МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра механики, мехатроники и робототехники

УТВЕРЖДАЮ
Проректор по учебной работе
________________ О.Г. Локтионова
«10» 09 2024 г.

ОГРН 1034637015786

Основы системного анализа сервисных роботов

Методические указания по выполнению лабораторных
Работ для студентов направления
15.03.06 «Мехатроника и робототехника»

Курск 2024

УДК 004

Составители: П.А. Безмен, Е.В. Савельева

Рецензент
Кандидат технических наук, доцент Юго-Западного государственного университета *Е.Н. Политов*

Основы системного анализа сервисных роботов: методические указания по выполнению лабораторных работ для студентов направления 15.03.06 «Мехатроника и робототехника» / Юго-Зап. гос. ун-т; сост. П.А. Безмен, Е.В. Савельева Курск, 2024. 85 с.

Изложены задания лабораторных работ студентов по дисциплине «Основы системного анализа сервисных роботов». Приведены примеры выполнения лабораторных работ в средах MathWorks MATLAB и Mathsoft/PTC Mathcad.

Методические указания соответствуют требованиям Федерального государственного образовательного стандарта.

Методические указания предназначены для студентов направления 15.03.06 «Мехатроника и робототехника», а также других направлений технического профиля для всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать 10.09.24 . Формат 60х84 1/16. Усл.печ.л. 4,8
Уч.-изд.л.4.4Тираж 30 экз. Заказ. Бесплатно.824
Юго-Западный государственный университет.
305040 Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Курс «Основы системного анализа сервисных роботов» является научной основой прикладного направления теории систем, применяемого при решении сложных слабоформализуемых проблем, а так же одной из основных дисциплин, обеспечивающих общетехническую общеинженерную подготовку студентов направления 15.03.06 «Мехатроника и робототехника».

Предмет дисциплины – теоретические основы системного анализа сервисных роботов и практические аспекты его применения в инженерной практике, позволяющие осуществлять полную и всестороннюю проверку различных вариантов действий с точки зрения количественного и качественного сопоставления затраченных ресурсов с получаемым эффектом.

Цель дисциплины – формирование у студентов компетенций – знаний и умений, необходимых для выполнения системного анализа технических систем, а также практических навыков по применению принципов системного подхода при решении задач в профессиональной деятельности непосредственно в условиях производства.

Задачи дисциплины

– формирование у студентов системного мышления, позволяющего обозревать некоторую проблему или явление в целом, выделять наиболее важные составляющие ее части и их взаимосвязи;

– формирование общих представлений о системах, системном подходе, методологии и технологии системного анализа, возможности их применений при решении вопросов, возникающих в теории и практике;

– изучение основ системного анализа как методологии исследования, моделирования и принятия решений по проблемам системного характера в теории и практике.

Лабораторные занятия по дисциплине «Основы системного анализа сервисных роботов» направления 15.03.06 «Мехатроника и робототехника» проводятся с целью практического закрепления знаний, получаемых студентами в лекционном курсе, и выполняются в специализированных лабораториях кафедры механики, мехатроники и робототехники Юго-Западного государственного университета.

Предлагаемое пособие содержит задачи для решения на лабораторных занятиях и задания для самостоятельной работы студентов.

ЛАБОРАТОРНАЯ РАБОТА №1
«РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ»

Цель работы: изучение методов решения задачи линейного программирования и транспортной задачи линейного программирования при помощи симплексного метода и метода потенциалов. Приобретение навыков построения математических моделей линейного программирования. Применение системы автоматизации математических расчетов MathCAD для решения указанных задач.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Линейное программирование - это направление математического программирования, изучающее методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменными и линейным критерием. Необходимым условием постановки задачи линейного программирования являются ограничения на наличие ресурсов, величину спроса, производственную мощность предприятия и другие производственные факторы.

Математическая модель любой задачи линейного программирования включает в себя:

- целевую функцию, определяющую критерий оптимальности;
- систему ограничений в форме линейных уравнений и неравенств;
- требование неотрицательности переменных.

Запишем математическую модель задачи линейного программирования следующим образом:

Определить вектор X = (x_1, x_2, … x_n), который удовлетворяет ограничениям вида:

$$\begin{cases} \sum_{i=1}^{n} a_{ij} \cdot x_{ij} \le b_j, (j = 1,.., m) \\ x_i \ge 0, (i = 1,.., n) \end{cases} \qquad (1)$$

и обеспечивает максимальное значение целевой функции

$$F(\overline{X}) = \sum_{i=1}^{n} c_i \cdot x_i \to \max \qquad (2)$$

Любое решение системы (1) с неотрицательными значениями переменных будем называть допустимым решением. Оптимальное решение – это такое допустимое решение, которое обращает целевую функцию в максимум (минимум). При числе свободных членов n-m<3, задачу ЛП можно решить графическим способом. Если число свободных членов равно и более трех, то применяются не геометрические, а вычислительные методы, наиболее распространенным из которых является симплекс-метод.

Симплекс-метод (или симплексный метод) основан па последовательном переходе от одного опорного плана задачи линейного программирования к другому, при этом значение целевой функции изменяется.

Алгоритм симплексного метода включает следующие этапы.

1. *Составление первого опорного плана*

Система ограничений задачи, решаемой симплексным методом, задана в виде системы неравенств. Перейдем от системы неравенств к системе уравнений путем введения неотрицательных дополнительных переменных. Векторы-столбцы при этих переменных представляют собой единичные векторы и образуют базис, а соответствующие им переменные называются базисными:

$$\sum_{i=1}^{n} a_{ij} \cdot x_i + x_{n+i} = b_j, (j = 1,.., m) \qquad (3)$$

где x_{ij} - базисные переменные. x_{n+i} - свободные переменные.

Решим эту систему относительно базисных переменных:

$$x_{n+i} = b_i - \sum_{i=1}^{n} a_{ij} \cdot x_i, (j = 1,.., m) \qquad (4)$$

а функцию цели перепишем в таком виде:

$$F(\overline{X}) = 0 - (-\sum_{i=1}^{n} c_i \cdot x_i) \qquad (5)$$

Полагая, что основные переменные $x_1 = x_2 = x_3 = \ldots = x_n = 0$, получим первый опорный план $\overline{X}_1 = (0,0,...,0,b_1,b_2,...,b_i)$; $F(\overline{X}_1) = 0$, который заносим в симплексную таблицу, состоящую из коэффициентов системы ограничений и свободных членов. Последняя строка таблицы называется индексной и заполняется коэффициентами целевой функции, взятыми с противоположными знаками.

2. *Проверка плана на оптимальность*

Если все коэффициенты индексной строки симплексной таблицы при решении задачи на максимум неотрицательны (>0), то план $\overline{X}_1 = (0,0,...,0,b_1,b_2,...,b_i)$ задачи является оптимальным. Если найдется хотя бы один коэффициент индексной строки меньше нуля, то план не оптимальный и его можно улучшить. Тогда переходим к следующему этапу алгоритма.

3. *Определение ведущих столбца и строки*

Из отрицательных коэффициентов индексной строки выбираем наибольший по абсолютной величине, столбец, в котором находится данный коэффициент, определяем как ведущий столбец, который показывает, какая переменная на следующей интерации перейдет из свободных в базисные.

Затем элементы столбца свободных членов симплексной таблицы делим на соответствующие только положительные элементы ведущего столбца. Результаты заносим в отдельный столбец θ. Строка симплексной таблицы, соответствующая минимальному значению θ, является ведущей Она определяет переменную x_i, которая на следующей итерации выйдет из базиса и станет свободной.

Элемент симплексной таблицы, находящейся на пересечении ведущих столбца и строки, называют ***разрешающим***.

4. *Построение нового опорного плана*

Переход к новому плану проводится пересчетом симплексной таблицы по методу Жордана-Гаусса. Для этого сначала заменим переменные в базисе, т.е вместо x_j в базис войдет переменная x_i,. соответствующая направляющему столбцу.

Разделим все элементы ведущей строки предыдущей симплексной таблицы на разрешающий элемент и результаты деления внесем в строку следующей симплексной таблицы, соответствующей введенной в базис переменной x_i. В результате этого на месте разрешающею элемента в следующей симплексной таблице будем иметь 1, а в остальных клетках *i*, включая клетку столбца индексной строки, записываем нули.

Остальные новые элементы нового плана находятся по правилу прямоугольника:

$$НЭ = СТЭ - \frac{A \cdot B}{РЭ} \qquad (6)$$

где НЭ – новый элемент, СТЭ - элемент старого плана, РЭ - разрешающий элемент, А и В - элементы старого плана, образующие прямоугольник с элементами СТЭ и РЭ.

5. ***Полученный новый опорный план опять промеряется па оптимальность в соответствии с этапом 2 алгоритма.***

При решении задачи линейного программировании па минимум целевой функции, признаком оптимальности плана являются отрицательные значения всех коэффициентов индексной строки симплексной таблицы.

Если в направляющем столбце все коэффициенты $a_{ij} \leq 0$, то функция цели $F(\overline{X})$ неограничена на множестве допустимых планов, т.е. $F(\overline{X}) > 0$ и задачу решить нельзя.

Если в столбце симплексной таблицы содержатся два или несколько одинаковых наименьших значения, то .новый опорный план будет вырожденным (одна или несколько базисных переменных станут равными нулю). Вырожденные планы могут привести к зацикливанию, т.е. многократному повторению процесса вычислений, не позволяющему завершить задачу. С целью исключения этого для выбора направляющей строки используют способ Креко, который заключается в следующем.

Делим элементы строк, имеющие одинаковые наименьшее значение θ, на предполагаемые разрешающие элементы, а результаты заносим в дополнительные строки. За ведущую строку выбирается та, в которой раньше встречается меньшее число при чтении таблицы слева направо по столбцам. Если в оптимальный план вошла дополнительная переменная x_{n+i} , то при реализации такого плана имеются недоиспользованные ресурсы i-го вида в количестве, полученном в столбце свободных членов симплексной таблицы.

Если в индексной строке симплексной таблицы оптимального плана находится нуль, принадлежащий свободной переменной, не вошедшей в базис, а в столбце, содержащем этот нуль, имеется хотя бы один положительный элемент, то задача имеет множество оптимальных планов. Свободную переменную, соответствующую указанному столбцу, можно внести в базис, выполнив соответствующие этапы алгоритма. В результате будет получен второй оптимальный план с другим набором базисных переменных.

Пример решения задачи симплексным методом

Торговое предприятие, располагающее материально-денежными ресурсами, реализует три группы товаров A, B и C. Плановые нормативы затрат ресурсов на тыс. руб. товарооборота, прибыль от продажи товаров на тыс. руб. товарооборота, а также объем ресурсов заданы в таблице 1

Таблица 1. Плановые нормативы затрат ресурсов на тыс. руб. товарооборота, прибыль от продажи товаров на тыс. руб. товарооборота, а также объем ресурсов

Виды материально-денежных ресурсов	Норма затрат материально- денежных ресурсов на ед. товарооборота, тыс. руб.			Объем ресурсов b_i
	А группа	В группа	С группа	
Рабочее время продавцов, чел/ч	0,1	0,2	0,4	1100
Площадь торговых залов, $м^3$	0,05	0,02	0,020	120
Площадь складских помещений, $м^3$	3	1	2	8000
Прибыль, тыс. руб	3	5	4	max

1. Запишем математическую модель задачи.

Определить $\overline{X} = (x_1, x_2, x_3)$, который удовлетворяет условиям:

$$\begin{cases} 0{,}1x_1 + 0{,}2x_2 + 0{,}4x_3 \le 1100 \\ 0{,}05x_1 + 0{,}02x_2 + 0{,}02x_3 \le 120 \\ 3x_1 + x_2 + 2x_3 \le 8000 \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0 \end{cases}$$

и обеспечивают максимальное значение целевой функции:

$F(X) = (3x_1 + 5x_2 + 4x_3) \to \max$

Для построения первого опорного плана систему неравенств приведем к системе уравнений:

$$0{,}1x_1 + 0{,}2x_2 + 0{,}4x_3 + x_4 = 1100$$

$$0{,}05x_1 + 0{,}02x_2 + 0{,}02x_3 + x_5 = 120$$

$$3x_1 + x_2 + 2x_3 + x_6 = 8000$$

В матрице этой системы уравнений А=a_{ij} имеет:

$$A = \begin{pmatrix} 0{,}1 & 0{,}2 & 0{,}4 & 1 & 0 & 0 \\ 0{,}05 & 0{,}02 & 0{,}02 & 0 & 1 & 0 \\ 3 & 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

векторы A_4.A_5,A_6 - линейно независимы, так как определитель, составленный из компонент этих векторов, отличен от нуля.

Соответствующие этим векторам переменные x_4, x_5, x_6 будут базисными.

Решим систему уравнений относительно базисных переменных.

$$x_4 = 1100 - (0{,}1x_1 + 0{,}2x_2 + 0{,}4x_3)$$

$$x_5 = 120 - (0{,}05x_1 + 0{,}02x_2 + 0{,}02x_3)$$

$$x_6 = 8000 - (3x_1 + x_2 + 2x_3)$$

Целевую функцию запишем в виде:

$$F(X) = 0 - (-3x_1 - 5x_2 - 4x_3)$$

Полагая, что свободные переменные x_1=0, x_2=0, x_3 =0, получим первый опорный план *X*,=(0,0,0,1100,120,8000), $F(\overline{X}_1) = 0$, в котором базисные переменные Х4=1100, x_5=120, x_6=8000, следовательно товары не продаются и прибыль равна нулю, а ресурсы не используются.

Запишем первый опорный план в симплексную таблицу (Таблица 2).

Таблица 2. Симплексная таблица

План	Базисные переменные	Ресурсы плана	Значения коэффициентов при переменных						θ_i
			x_1	x_2	x_3	x_4	x_5	x_6	
I План	x_4	1100	0,1	0,2	4	1	0	0	5500
	x_5	120	0,05	0,02	0,02	0	1	0	6000
	x_6	8000	3	1	2	0	0	1	8000
Инд. строка	$F(x_1)$	0	-3	-5	-4	0	0	0	
II План	x_4	5500	0,5	1	2	5	0	0	11000
	x_5	10	0,04	0	-0,02	-0,1	1	0	250
	x_6	2500	2,5	0	0	-5	0	1	1000
Инд. строка	$F(x_2)$	27500	-0,5	0	6	25	0	0	
III План	x_2	5355	0	1	2,25	6,25	12,5	0	
	x_3	250	1	0	-0,5	-2,5	25	0	
	x_4	1875	0	0	1,25	1,25	-62,5	1	
Инд. строка	$F(x_3)$	27625	0	0	5,75	23,75	12,5	0	

Первый опорный план не оптимальный, так как в индексной строке находятся отрицательные коэффициенты -3,-5,-4.

За ведущий столбец выберем столбец, соответствующий переменной x_2, так как, сравнивая по модулю, имеем: $|-5|>\{|-3|,|-4|\}$

Рассчитываем значения θ_i по строкам как частное от деления $\frac{b_i}{a_{ij}}$ и выбираем наименьшее:

$$\min(\frac{b_i}{a_{ij}}) = \min\left(\frac{1100}{0,2};\frac{120}{0,02};\frac{8000}{1}\right) = 5500$$

Следовательно, первая строка является ведущей. Элемент 0,2 находится на пересечении ведущего столбца и ведущей строки и выделен.

Формируем следующую симплексную таблицу. Вместо переменной x_4 в план II войдет переменная x_2. Строка, соответствующая переменной x_2 в плане II, получена в результате деления всех элементов строки x_4 плана I на разрешающий элемент РЭ=0,2. Па месте разрешающего элемента ав плане II получаем 1. В остальных клетках столбца x_2 плана II записываем нули.

Таким образом в новом плане II заполнены строки x_2и столбец x_4. Все остальные элементы нового плана II, включая элементы индексной строки, определяется по правилу прямоугольника. Для этого выбираем из старого плана 4 числа , которые расположены в вершинах прямоугольника и всегда включают разрешающий элемент РЭ = 0,2. Во второй вершине по диагонали находится старое значение элемента, например, значение целевой функции $F(K_i) = 0 = CЭ$, которое указывает на месторасположение нового НЭ в новом плане II. Третий элемент А = 1100 и четвертый элемент В = -5 завершают построение прямоугольника в недостающих двух вершинах и расположены по другой диагонали. Значение нового элемента в плане II находится из выражения:

$$НЭ = СТЭ - \frac{A \cdot B}{РЭ} = 0 - \frac{1100 \cdot (-5)}{0,2} = 27500$$

Элементы строки определяются аналогично:

$$120 - \frac{1100 \cdot 0,02}{0,2} = 10$$

$$0,05 - \frac{0,1 \cdot 0,02}{0,2} = 0,04$$

$$0,02 - \frac{0,4 \cdot 0,02}{0,2} = -0,02$$

$$0 - \frac{0,02 \cdot 1}{0,2} = -0,1$$

Все элементы, расположенные на пересечении строк и столбцов, соответствующих одноименным базисным элементам равны 1, остальные элементы столбца и базисах векторов, включая индексную строку, равны 0. Аналогично проводятся расчеты по всем строкам таблицы, включая индексную.

Выполняя последовательно все этапы алгоритма, формируем план II.

На третьей итерации таблицы 2 получаем план III, который является оптимальным, так как все коэффициенты и индексной строке больше или равны нулю. Оптимальный план можно записать так:

$$\overline{X} = (250,5375,0,0,0,1875) \quad F(\overline{X}) = 27625 тыс.руб.$$

Следовательно, необходимо продавать товаров первой группы А 250 ед., а второй группы В 5375 ед. При этом торговое предприятие получает максимальную прибыль в размере 27625 тыс. руб. Товары группы С не реализуются.

В оптимальном плане среди базисных переменных находится дополнительная переменная x_6. Это указывает, что ресурсы третьего вида (площадь складских помещений) недоиспользована на 1875 м, так как переменная x_6 была введена в первое ограничение задачи, характеризующее собой использование этого ресурса.

В индексной строке III плана в столбцах переменных x_3,x_4,x_5, не вошедших в состав базисных, получены ненулевые элементы, поэтому оптимальный план задачи линейного программирования является единственным.

Транспортная задача линейного программирования

Однородный груз, имеющийся в *m* пунктах отправления (производства) A_1, A_2, ...A_m соответственно в количествах a_1, a_2, ...a_m единиц, требуется доставить в каждый из n пунктов назначения (потребления) B_1, B_2, ..., B_n соответственно в количествах b_1, b_2 ..., b_n единиц. Стоимость перевозки (тариф) единицы продукции из A_i в B_j известна для всех маршрутов A_iB_j и c_{ij} (i=1,m; j=1,n). Требуется составить такой план перевозок, при котором весь груз из пунктов отправления вывозится и запросы всех пунктов потребления удовлетворяются (закрытая модель), т. е.

$$\sum_{i=1}^{m} ai = \sum_{j=1}^{n} bj \tag{6}$$

а суммарные транспортные расходы минимальны.

Математическая модель транспортной задачи:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} Cij\ Xij \text{ -> } \min$$

$$\begin{cases} X11+X11+\dots+X1n=a1 \\ X21+X22+\dots+X2n=a2 \\ \dots\ \dots\ \dots \\ Xm1+Xm2+\dots+Xmn=am \\ X11+X21+\dots+Xm1=b1 \\ X12+X22+\dots+Xm2=b2 \\ \dots\ \dots\ \dots \\ X1n+X2n+\dots+Xmn=bn \\ Xij>=0,\ i=1,m;\ j=1,n \end{cases} \qquad (7)$$

Транспортные задачи выделены в отдельный класс потому, что именно с них началось изучение задач линейного программирования. Для решения транспортных задач применяется метод потенциалов.

РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ СРЕДСТВАМИ MATHCAD

Задача. Найти экстремум (максимум или минимум) линейной целевой функции:

$$f(x_1 .. x_n) = \sum_{j=1}^{n} c_j \cdot x_j \qquad (8)$$

при ограничениях (условиях):

$$\sum_{j=1}^{n} a_{i,j} \cdot x_j \le b_i, i = 1 .. m \qquad (9)$$

$$x_j \ge 0, j = 1 .. n$$

где $a_{i,j}, b_i, c_j, (i = 1 .. m, j = 1 .. n)$ заданные постоянные величины.

Решение.

1. Вводят исходные данные задачи (в нашем случае это переменные ***n, m, c, a, b***)
2. Вводят линейную целевую функцию ***f(x₁..xₙ)***.
3. Задают начальные значения переменным задачи (в нашем случае это переменные ***x₁..xₙ***.
4. Формируют решающий блок со служебным словом ***Given***, внутри которого вводят ограничения задачи в матричной форме (в случае небольшого числа переменных можно ввести ограничения в естественной форме).
5. Завершают решающий блок обращением к функции ***Minimize (Maximize)***.
6. В случае задачи с двумя переменными строят графики прямых, соответствующих ограничениям, и линии уровня, используя инструмент анимации.

Пример

Найти максимальное значение функции

$$f(x_1, x_2) = 2 \cdot x_1 + 3 \cdot x_2$$

при заданных ограничениях

$$4 \cdot x_1 + 5 \cdot x_2 \le 7$$

$$7 \cdot x_1 + 4 \cdot x_2 \le 8$$

$9 \cdot x_1 + 8 \cdot x_2 \le 9$

$x_1 \ge 0$

$x_2 \ge 0$

Решение.

1. Вводят исходные данные задачи в матричной форме.

$m := 3$

$n := 2$

$$c := \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad a := \begin{pmatrix} 4 & 5 \\ 7 & 4 \\ 9 & 8 \end{pmatrix} b := \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

2. Вводят линейную целевую функцию.

$$f(x) := \sum_{j=1}^{n} c_j \cdot x_j$$

3. Задают начальные значения переменным задачи.

$x_n := 0$

4. Вводят ограничения задачи в матричной форме.

Given

$a \cdot x \le b$

$x \ge 0$

5. Определяют оптимальное решение задачи с помощью встроенной функции Maximize (в случае поиска максимума функции) или Minimize (в случае поиска минимума функции).

$x := \mathrm{Maximize}(f, x)$

$$x = \begin{pmatrix} 0 \\ 1.125 \end{pmatrix}$$

$f(x) = 3.375$

6. В случае задачи с двумя переменными строят график.

$C := \mathrm{FRAME}$

$C = 0$

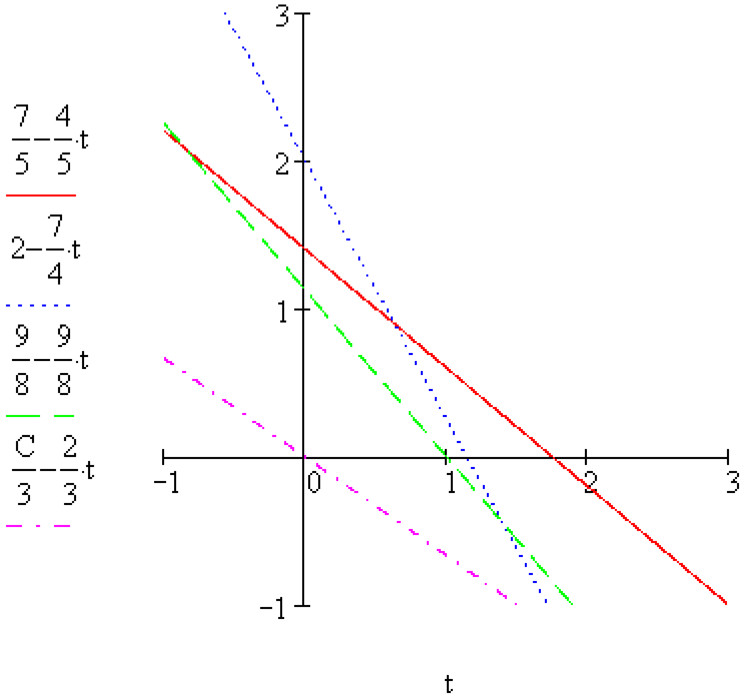


Рис. 1.

Раздел 2

Решение транспортных задач

Задача. Найти экстремум (минимум) линейной целевой функции:

$$f(x_{1,1}..x_{m,n}) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j} \qquad (10)$$

при ограничениях (условиях):

$$\sum_{i=1}^{m} x_{ij} = b_j, (j = 1..n) \qquad (11)$$

$$\sum_{j=1}^{n} x_{ij} = a_i, (i = 1..m) \qquad (12)$$

$$x_{ij} \ge 0, (i = 1..m, j = 1..n)$$

где $a_i, b_j, c_{ij}, (i = 1..m, j = 1..n)$ заданные постоянные величины, причем

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \quad . \qquad (13)$$

Решение.

Решение задачи при помощи программного пакета MathCAD аналогично решению задачи линейного программирования.

Алгоритм решения транспортной задачи следующий:

1. Вводят исходные данные задачи в матричной форме.
2. Вводят линейную целевую функцию.
3. Задают начальные значения переменным задачи.
4. Вводят ограничения задачи в матричной форме (в случае небольшого числа переменных можно ввести ограничения в естественной форме) в решающем блоке со служебным словом ***Given.***
5. Определяют оптимальное решение задачи с помощью встроенной функции Minimize.

Задания к лабораторной работе №1

Раздел 1

Задачи линейного программирования

1. Решить задачу ЛП по данным, выданным преподавателем, симплексным методом в программе MathCAD.
2. Проверить решение задачи в программе MathCAD:
   2.1. при помощи функции Maximize.
   2.2. графическим способом.

Раздел 2

Транспортная задача

1. Решить транспортную задачу ЛП по данным, выданным преподавателем, методом потенциалов.
2. Проверить решение задачи в программе MathCAD при помощи функции Minimize.

Задания

Задачу выбрать соответственно с порядковым номером по журналу.

Задача линейного программирования

Варианты заданий:

1. $L (x) = 2 x_1 - 5 x_2 -> min$

$3 >= x_1 - x_2$
$4 >= 2 x_1 + x_2$
$x_1, x_2 >= 0$
$5 <= 3 x_1 + 2 x_2$
$1 <= 3 x_1 - 4 x_2$

2. $L (x) = 8 x_1 + 2 x_2 \to max$

$16 \ge x_1 + x_2$
$20 \ge 4 x_1 - 2 x_2$
$x 1 \ge 0$

3. $L (x) = x_1 + 2 x_2 \to min$

$3 \le 2 x_1 + x_2$
$1 \ge 2 x_1 - 7 x_2$
$6 \le 2 x_1 + 3 x_2$
$x_1 , x_2 \ge 0$

4. $L (x) = 8 x_1 + 2 x_2 - 5 x_3 \to max$

$6 \le - x_1 + 2 x_2 + x_3$
$3 \le x_1 - 2 x_2 + 2 x_3$
$2 \ge 2 x_1 + x_2 - x_3$
$x_1 , x_2 , x_3 \ge 0$

5. $L(x) = 4x_1 + 3x_2 + 6x_3 \rightarrow max$

$$40 \geq 2x_1 + x_2 + x_3$$
$$100 \geq 4x_1 + 9x_3$$
$$30 \geq 3x_2 + 5x_3$$
$$60 \geq 2x_1 + x_2 + 5x_3$$
$$x_1, x_2, x_3 \geq 0$$

6. $L(x)= 20x_1+10x_2+9x_3+10x_4 \rightarrow max$

$$60 \geq 7x_1 + 7x_2 + 3x_3$$
$$35 \geq 3x_2 + x_3 + 3x_4$$
$$75 \geq 8x_1 + 4x_3 + x_4$$
$$50 \geq 5x_1 + 2x_2 + 3x_4$$
$$x_1, x_2, x_3, x_4 \geq 0$$

7. $L(x) = x_1 + 2x_2 - 3x_3 + 4x_4 \rightarrow max$

$$100 = x_1 - x_2 + 7x_3 + x_4$$
$$800 = 2x_1 + 3x_2 - x_3 + 10x_4$$
$$x_1, x_2, x_3, x_4 \geq 0$$

8. $L(x) = x_1 + x_2 \rightarrow max$

$$12 \geq 3x_1 + 4x_2$$
$$8 \geq 2x_1 - x_2$$
$$x_1, x_2 \geq 0$$

9. $L(x) = 3x_1 + 2x_2 \rightarrow max$

$$8 \geq 4x_1 + 3x_2$$
$$3 \geq 2x_1 + 0.5x_2$$
$$x_1, x_2 \geq 0$$

10 .$L(x) = x_1 + x_2 \rightarrow max$

$$10 \geq 3x_1 + 2x_2$$
$$2 \leq x_1 \leq 5$$
$$0 \leq x_2 \leq 3$$

11. $L(x) = -x_1 + 2x_2 \rightarrow max$

$$23 \geq 5x_1 + 4x_2$$
$$20 \geq 3x_1 + 2x_2$$
$$6 \geq 3x_1 - x_2$$
$$x_1, x_2 \geq 0$$

12. $L(x) = 6x_1 - 5x_2 \rightarrow max$

$$4 \leq x_1 + x_2$$
$$24 \geq x_1 + x_2$$
$$12 \geq -x_1 + x_2$$
$$12 \geq x_1 - x_2$$
$$x_1, x_2 \geq 0$$

13. $L(x) = 3x_1 + 4x_2 \rightarrow max$

$$23 \geq x_1 + x_2$$
$$17 \geq -x_1 + x_2$$
$$13 \leq x_1 - 3x_2$$
$$x_1, x_2 \geq 0$$

14. $L(x) = -2x_1 + 4x_2 + 5x_3 \rightarrow max$

$$76 \geq 3x_1 + 5x_2 + 7x_3$$
$$x_1 \geq 0$$
$$0 \leq x_2 \leq 7$$
$$1 \leq x_3 \leq 5$$

15. $L(x) = 2x_1 - x_2 + x_3 \rightarrow max$

$$6 \leq x_1 + x_2 + x_3$$
$$2 \geq 2x_1 - x_2 + x_3$$
$$x_1, x_2, x_3 \geq 0$$

16. $L(x) = x_1 + 4x_2 \rightarrow max$

$$6 \leq x_1 - 3x_2$$
$$10 \geq x_1 + x_2$$
$$9 \leq 3x_1 + x_2$$
$$4 \geq -x_1 + x_2$$

$x_1, x_2 \ge 0$

17. $L(x) = -2x_1 + x_2 - 3x_3 \to max$

$8 \ge 3x_1 - x_2$
$1 \ge -x_1 + x_2 + 4x_3$
$6 \ge 2x_1 + x_2 - 3x_3$
$x_1, x_2, x_3 \ge 0$

18. $L(x) = -x_1 + 2x_2 \to min$

$6 \le x_1 + 3x_2$
$1 \ge -x_1 + 2x_2$
$5 \ge x_1 + x_2$
$6 \le 3x_1 - x_2$
$x_1, x_2 \ge 0$

19. $L(x) = 7x_1 + 4x_2 - 3x_3 + 2x_4 \to max$

$45 \ge 2x_1 + 3x_2 - 4x_3 + 5x_4$
$68 \ge x_1 + x_2 + 2x_3 + x_4$
$x_1, x_2, x_3, x_4 \ge 0$

20. $L(x) = x_1 + x_2 + x_3 + x_4 \to min$

$7 \le 3x_1 + 2x_2 + x_4$
$3 \le x_1 + 5x_2 + 6x_3 + 11x_4$
$x_1, x_2, x_3, x_4 \ge 0$

21. $L(x) = 2x_1 + 3x_2 + 4x_3 + 7x_4 \to min$

$40 \ge 3x_1 + 4x_2 + 5x_3 + 5x_4$
$60 \ge x_1 + x_2 + x_3 + x_4$
$x_1 \ge 1$
$x_2 \ge 0$
$x_3 \ge 1.5$

22. $L(x) = 2x_1 - 6x_2 - 4x_3 \to min$

$16 \ge -2x_1 + 6x_3$
$2 \ge 2x_1 + 8x_2 - 2x_3$
$12 \ge 2x_1 - 6x_2 + 4x_3$
$x_1 \ge 2$
$x_2 \ge 0$
$x_3 \ge 0.5$

23. $L(x) = x_2 - 0.5x_1 \to min$

$3 \le 1.5x_1 + 0.5x_2$
$0.5 \le x_1 - 0{,}5x_2$
$2.5 \ge 0.5x_1 + 0.5x_2$
$3 \le 0.5x_1 + 1.5x_2$
$x_1, x_2 \ge 0$

24. $L(x) = 0.5x_1 + 0.3x_2 \to max$

$9 \ge 0.3x_1 + 0.3x_2$
$7 \ge -0{,}3x_1 + 0.3x_2$
$4.5 \le 0.4x_1 - x_2$
$x_1, x_2 \ge 0$

Транспортная задача линейного программирования

1. Для строительства четырех объектов используется кирпич, изготовляемый на трех заводах. Ежедневно каждый из заводов может изготовлять 100, 150 и 50 усл. ед. кирпича. Ежедневные потребности в кирпиче на каждом из строящихся объектов соответственно равны 75, 80, 60 и 85 усл. ед. Известны также тарифы перевозок 1 усл. ед. кирпича с каждого с заводов к каждому из строящихся объектов:

$$C = \begin{pmatrix} 6 & 7 & 3 & 5 \\ 1 & 2 & 5 & 6 \\ 8 & 10 & 20 & 1 \end{pmatrix}$$

Составить такой план перевозок кирпича к строящимся объектам, при котором общая стоимость перевозок является минимальной.

2. На трех хлебокомбинатах ежедневно производится 110, 190 и 90 т муки. Эта мука потребляется четырьмя хлебозаводами, ежедневные потребности которых равны соответственно 80, 60, 170 и 80 т. Тарифы перевозок 1 т муки с хлебокомбинатов к каждому из хлебозаводов задаются матрицей

$$C = \begin{pmatrix} 8 & 1 & 9 & 7 \\ 4 & 6 & 2 & 12 \\ 3 & 5 & 8 & 9 \end{pmatrix}$$

Составить такой план доставки муки, при котором общая стоимость перевозок является минимальной.

3. В трех хранилищах горючего ежедневно хранится 175, 125 и 140 т бензина. Этот бензин ежедневно получают четыре заправочные станции в количествах, равных соответственно 180, 160, 60 и 40 т. Стоимости перевозок 1 т бензина с хранилищ к заправочным станциям задаются матрицей

$$C = \begin{pmatrix} 9 & 7 & 5 & 3 \\ 1 & 2 & 4 & 6 \\ 8 & 10 & 12 & 1 \end{pmatrix}$$

Составить такой план перевозок бензина, при котором общая стоимость перевозок является минимальной.

4. На трех железнодорожных станциях A_1, A_2 и A_3 скопилось 120, 110 и 130 незагруженных вагонов. Эти вагоны необходимо перегнать на железнодорожные станции B_1, B_2, B_3, B_4 и B_5. На каждой из этих станций потребность в вагонах соответственно равна 80, 60, 70, 100 и 50. Тарифы перегонки одного вагона определяются матрицей

$$C = \begin{pmatrix} 2 & 4 & 1 & 6 & 7 \\ 3 & 3 & 5 & 4 & 2 \\ 8 & 9 & 6 & 3 & 4 \end{pmatrix}$$

Составьте такой план перегонок вагонов, чтобы общая стоимость была минимальной.

5. Для строительства трех дорог используется гравий из четырех карьеров. Запасы гравия в каждом из карьеров соответственно равны 120, 280 и 160 усл. ед. Потребности в гравии для строительства каждой из дорог соответственно равны 130, 220, 160 и 50 усл. ед. Известны также тарифы перевозок 1 усл. ед.

гравия из каждого из карьеров к каждой из строящихся дорог, которые задаются матрицей

$$C=\begin{pmatrix}1 & 7 & 9 & 5\\4 & 2 & 6 & 8\\3 & 8 & 1 & 2\end{pmatrix}$$

Составить такой план перевозок гравия, при котором потребности в нем каждой из строящихся дорог были бы удовлетворены при наименьшей общей стоимости перевозок.

6. Три предприятия данного экономического района могут производить некоторую однородную продукцию в количествах, соответственно равных 180, 350 и 20 ед. Эта продукция должна быть поставлена пяти потребителям в количествах, соответственно равных 110, 90, 120, 80 и 150 ед. Затраты, связанные с производством и доставкой единицы продукции, задаются матрицей

$$C=\begin{pmatrix}2 & 4 & 1 & 6 & 7\\3 & 3 & 5 & 4 & 2\\8 & 9 & 6 & 3 & 4\end{pmatrix}$$

Составить такой план прикрепления получателей продукции ее поставщикам, при котором, общая стоимость перевозок является минимальной.

7. Производственное объединение имеет в своем составе три филиала, которые производят однородную продукцию соответственно в количествах, равных 50, 30 и 10 ед. Эту продукцию получают четыре потребителя, расположенные в разных местах. Их потребности соответственно равны 30, 30, 10 и 20 ед. Тарифы перевозок единицы продукции от каждого из филиалов соответствующим потребителям задаются матрицей

$$C=\begin{pmatrix}1 & 2 & 4 & 1\\2 & 3 & 1 & 5\\3 & 2 & 4 & 4\end{pmatrix}$$

Составить такой план прикрепления получателей продукции ее поставщикам, при котором общая стоимость перевозок является минимальной.

8. На трех складах оптовой базы сосредоточен однородный груз в количествах 180, 60 и 60 ед. Этот груз необходимо перевезти в четыре магазина. Каждый из магазинов должен получить соответственно 120, 40, 60 и 80 ед. груза. Тарифы перевозок единицы груза из каждого из складов во все магазины задаются матрицей

$$C=\begin{pmatrix}2 & 3 & 4 & 3\\5 & 3 & 1 & 2\\2 & 1 & 4 & 2\end{pmatrix}$$

Составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

9. Четыре предприятия данного экономического района для производства продукции используют три вида сырья. Потребности в сырье каждого из предприятий соответственно равны 120,50,190 и 110 ед. Сырье сосредоточено в трех местах его получения, а запасы соответственно равны 160, 140, 170 ед. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются матрицей.

$$C = \begin{pmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{pmatrix}$$

Составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

10. Четыре предприятия данного экономического района для производства продукции используют пять видов сырья. Потребности в сырье каждого из предприятий соответственно равны 120,50,190 и 110 ед. Сырье сосредоточено в пяти местах его получения, а запасы соответственно равны 160, 100, 40, 100 и 70 ед. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются матрицей.

$$C = \begin{pmatrix} 2 & 5 & 7 & 3 \\ 7 & 1 & 4 & 9 \\ 4 & 5 & 7 & 3 \\ 1 & 1 & 2 & 3 \\ 4 & 3 & 1 & 9 \end{pmatrix}$$

Составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

11. Для строительства четырех объектов используется кирпич, изготовляемый на трех заводах. Ежедневно каждый из заводов может изготовлять 100, 150 и 50 усл. ед. кирпича. Ежедневные потребности в кирпиче на каждом из строящихся объектов соответственно равны 75, 80, 60 и 85 усл. ед. Известны также тарифы перевозок 1 усл. ед. кирпича с каждого с заводов к каждому из строящихся объектов:

$$C = \begin{pmatrix} 8 & 1 & 9 & 7 \\ 4 & 6 & 2 & 12 \\ 3 & 5 & 8 & 9 \end{pmatrix}$$

Составить такой план перевозок кирпича к строящимся объектам, при котором общая стоимость перевозок является минимальной.

12. На трех хлебокомбинатах ежедневно производится 150, 170 и 80 т муки. Эта мука потребляется четырьмя хлебозаводами, ежедневные потребности которых равны соответственно 70, 50, 160 и 90 т. Тарифы перевозок 1 т муки с хлебокомбинатов к каждому из хлебозаводов задаются матрицей

$$C = \begin{pmatrix} 9 & 7 & 5 & 3 \\ 1 & 2 & 4 & 6 \\ 8 & 10 & 12 & 1 \end{pmatrix}$$

Составить такой план доставки муки, при котором общая стоимость перевозок является минимальной.

13. В трех хранилищах горючего ежедневно хранится 175, 125 и 140 т бензина. Этот бензин ежедневно получают четыре заправочные станции в количествах, равных соответственно 180, 160, 60 и 40 т. Стоимости перевозок 1 т бензина с хранилищ к заправочным станциям задаются матрицей

$$C = \begin{pmatrix} 6 & 7 & 3 & 5 \\ 1 & 2 & 5 & 6 \\ 8 & 10 & 20 & 1 \end{pmatrix}$$

Составить такой план перевозок бензина, при котором общая стоимость перевозок является минимальной.

14. На трех железнодорожных станциях A_1, A_2 и A_3 скопилось 140, 120 и 150 незагруженных вагонов. Эти вагоны необходимо перегнать на железнодорожные станции B_1, B_2, B_3, B_4 и B_5. На каждой из этих станций потребность в вагонах соответственно равна 90, 70, 80, 110 и 60. Тарифы перегонки одного вагона определяются матрицей

$$C = \begin{pmatrix} 2 & 4 & 1 & 6 & 7 \\ 3 & 3 & 5 & 4 & 2 \\ 8 & 9 & 6 & 3 & 4 \end{pmatrix}$$

Составьте такой план перегонок вагонов, чтобы общая стоимость была минимальной.

15. Для строительства трех дорог используется гравий из четырех карьеров. Запасы гравия в каждом из карьеров соответственно равны 120, 280 и 160 усл. ед. Потребности в гравии для строительства каждой из дорог соответственно равны 130, 220, 160 и 50 усл. ед. Известны также тарифы перевозок 1 усл. ед. гравия из каждого из карьеров к каждой из строящихся дорог, которые задаются матрицей

$$C = \begin{pmatrix} 1 & 7 & 9 & 5 \\ 4 & 2 & 6 & 8 \\ 3 & 8 & 1 & 2 \end{pmatrix}$$

Составить такой план перевозок гравия, при котором потребности в нем каждой из строящихся дорог были бы удовлетворены при наименьшей общей стоимости перевозок.

16. Три предприятия данного экономического района могут производить некоторую однородную продукцию в количествах, соответственно равных 190, 380 и 40 ед. Эта продукция должна быть поставлена пяти потребителям в количествах, соответственно равных 150, 110, 130, 90 и 170 ед. Затраты, связанные с производством и доставкой единицы продукции, задаются матрицей

$$C = \begin{pmatrix} 2 & 4 & 1 & 6 & 7 \\ 3 & 3 & 5 & 4 & 2 \\ 8 & 9 & 6 & 3 & 4 \end{pmatrix}$$

Составить такой план прикрепления получателей продукции ее поставщикам, при котором общая стоимость перевозок является минимальной.

17. Производственное объединение имеет в своем составе три филиала, которые производят однородную продукцию соответственно в количествах, равных 40, 60 и 45 ед. Эту продукцию получают четыре потребителя, расположенные в разных местах. Их потребности соответственно равны 50, 25, 20 и 15 ед. Тарифы перевозок единицы продукции от каждого из филиалов соответствующим потребителям задаются матрицей

$$C = \begin{pmatrix} 1 & 2 & 4 & 1 \\ 2 & 3 & 1 & 5 \\ 3 & 2 & 4 & 4 \end{pmatrix}$$

Составить такой план прикрепления получателей продукции ее поставщикам, при котором общая стоимость перевозок является минимальной.

18. На трех складах оптовой базы сосредоточен однородный груз в количествах 180, 60 и 60 ед. Этот груз необходимо перевезти в четыре магазина. Каждый из магазинов должен получить соответственно 120, 40, 60 и 80 ед. груза. Тарифы перевозок единицы груза из каждого из складов во все магазины задаются матрицей

$$C = \begin{pmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{pmatrix}$$

Составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

19. Четыре предприятия данного экономического района для производства продукции используют три вида сырья. Потребности в сырье каждого из предприятий соответственно равны 120,50,190 и 110 ед. Сырье сосредоточено в

трех местах его получения, а запасы соответственно равны 160, 140, 170 ед. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются матрицей.

$$C = \begin{pmatrix} 2 & 3 & 4 & 3 \\ 5 & 3 & 1 & 2 \\ 2 & 1 & 4 & 2 \end{pmatrix}$$

Составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

20. Четыре предприятия данного экономического района для производства продукции используют пять видов сырья. Потребности в сырье каждого из предприятий соответственно равны 120,50,190 и 110 ед. Сырье сосредоточено в пяти местах его получения, а запасы соответственно равны 160, 100, 40, 100 и 70 ед. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются матрицей.

$$C = \begin{pmatrix} 2 & 5 & 7 & 3 \\ 7 & 1 & 4 & 9 \\ 4 & 5 & 7 & 3 \\ 1 & 1 & 2 & 3 \\ 4 & 3 & 1 & 9 \end{pmatrix}$$

Составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

21. Для строительства трех дорог используется гравий из четырех карьеров. Запасы гравия в каждом из карьеров соответственно равны 100, 200 и 190 усл. ед. Потребности в гравии для строительства каждой из дорог соответственно равны 120, 210, 110 и 90 усл. ед. Известны также тарифы перевозок 1 усл. ед. гравия из каждого из карьеров к каждой из строящихся дорог, которые задаются матрицей

$$C = \begin{pmatrix} 1 & 7 & 9 & 5 \\ 4 & 2 & 6 & 8 \\ 3 & 8 & 1 & 2 \end{pmatrix}$$

Составить такой план перевозок гравия, при котором потребности в нем каждой из строящихся дорог были бы удовлетворены при наименьшей общей стоимости перевозок.

22. Три предприятия данного экономического района могут производить некоторую однородную продукцию в количествах, соответственно равных 115, 270 и 60 ед. Эта продукция должна быть поставлена пяти потребителям в количествах, соответственно равных 145, 120, 110, 110 и 150 ед. Затраты,

связанные с производством и доставкой единицы продукции, задаются матрицей

$$C = \begin{pmatrix} 2 & 4 & 1 & 6 & 7 \\ 3 & 3 & 5 & 4 & 2 \\ 8 & 9 & 6 & 3 & 4 \end{pmatrix}$$

Составить такой план прикрепления получателей продукции ее поставщикам, при котором общая стоимость перевозок является минимальной.

СОДЕРЖАНИЕ ОТЧЕТА

1. Наименование работы.
2. Цель работы.
3. Описание постановки задачи ЛП.
4. Описание постановки ТЗ ЛП.
5. Описание решения задания 1.
6. Описание решения задания 2.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как формулируется задача ЛП?
2. Что такое целевая Функция?
3. Принцип решения задачи ЛП симплексным методом.
4. Как формулируется транспортная задача ЛП?
5. Какие ограничения присутствуют в транспортной задаче ЛП?
6. Что такое опорный план решения транспортной задачи ЛП?
7. Сформулируйте условия вырожденности и невырожденности плана решения ТЗ ЛП.

Пример выполнения лабораторной работы №1

Лабораторная работа №1

Решение задач линейного программирования

Цель работы: изучение методов решения задачи линейного программирования и транспортной задачи линейного программирования при помощи симплексного метода и метода потенциалов. Приобретение навыков построения математических моделей линейного программирования. Применение системы автоматизации математических расчетов MathCAD для решения указанных задач.

Краткие теоретические сведения

Линейное программирование - это направление математического программирования, изучающее методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменными и линейным критерием. Необходимым условием постановки задачи линейного программирования являются ограничения на наличие ресурсов, величину спроса, производственную мощность предприятия и другие производственные факторы.

Математическая модель любой задачи линейного программирования включает в себя:

- целевую функцию, определяющую критерий оптимальности;
- систему ограничений в форме линейных уравнений и неравенств;
- требование неотрицательности переменных.

Запишем математическую модель задачи линейного программирования следующим образом:

Определить вектор X = (x_1, x_2, … x_n), который удовлетворяет ограничениям вида:

$$\begin{cases} \sum_{i=1}^{n} a_{ij} \cdot x_{ij} \le b_j, (j = 1,.., m) \\ x_i \ge 0, (i = 1,.., n) \end{cases} \qquad (1)$$

и обеспечивает максимальное значение целевой функции

$$F(\overline{X}) = \sum_{i=1}^{n} c_i \cdot x_i \to \max \qquad (2)$$

Любое решение системы (1) с неотрицательными значениями переменных будем называть допустимым решением. Оптимальное решение – это такое допустимое решение, которое обращает целевую функцию в максимум (минимум). При числе свободных членов *n-m<3*, задачу ЛП можно решить графическим способом. Если число свободных членов равно и более трех, то применяются не геометрические, а вычислительные методы, наиболее распространенным из которых является симплекс-метод.

Симплекс-метод (или симплексный метод) основан па последовательном переходе от одного опорного плана задачи линейного программирования к другому, при этом значение целевой функции изменяется.

1. Задача линейного программирования

Решить задачу ЛП по данным, выданным преподавателем, симплексным методом в программе MathCAD.

Линейная целевая функция: $L\ (x) = 3\ x_1 + 2\ x_2 \to \max$;
граничные условия: $8 \geq 4\ x_1 + 3\ x_2$,
$3 \geq 2\ x_1 + 0\ .5\ x_2$,
$x_1\ , x_2 \geq 0$.

Решение задачи

Зададим начальные значения переменным задачи и опишем линейную целевую функцию двух переменных.

$$i := 1$$

$$w_i := 0$$

$$f(w) := 3 \cdot w_0 + 2 \cdot w_1$$

Сформируем решающий блок со служебным словом ***Given***, внутри которого введем ограничения задачи. Далее завершим решающий блок обращением к функции ***Maximize.***

Given

$$4\cdot\left(w_0\right) + 3\cdot\left(w_1\right) \le 8$$

$$2\cdot\left(w_0\right) + 0.5\cdot\left(w_1\right) \le 3$$

$$w_0 \ge 0 \quad w_1 \ge 0$$

$$M := \mathrm{Maximize}(f, w)$$

$$M := \begin{pmatrix} 1.25 \\ 1 \end{pmatrix}$$

$$OPTf := f(M)$$

$$OPTf := 5.75$$

Построим в одной системе координат графики прямых, соответствующих граничным условиям линейной целевой функции двух переменных.

$$F1(x1) := \frac{(8 - 4\cdot x1)}{3}$$

$$F2(x1) := \frac{(3 - 2\cdot x1)}{0.5}$$

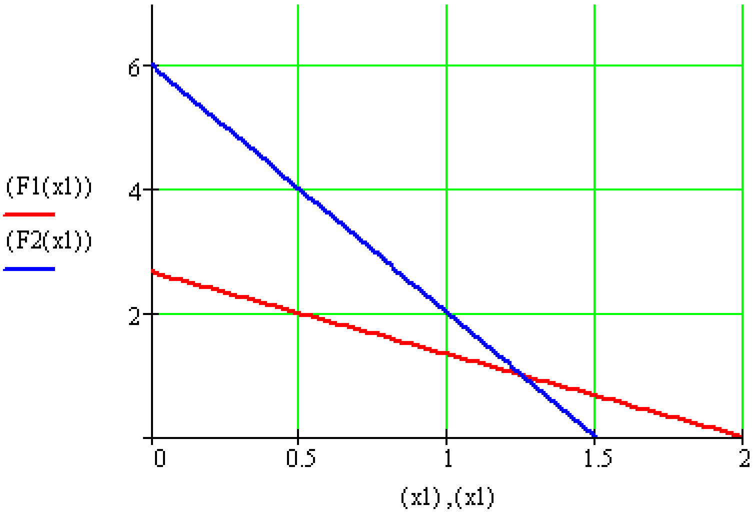


Рис. n1.

2. Транспортная задача

Решить транспортную задачу ЛП по данным, выданным преподавателем, методом потенциалов.

Задача. Для строительства четырех объектов используется кирпич, изготовляемый на трех заводах. Ежедневно каждый из заводов может изготовлять 100, 150 и 50 усл. ед. кирпича. Ежедневные потребности в кирпиче на каждом из строящихся объектов соответственно равны 75, 80, 60 и 85 усл. ед. Известны также тарифы перевозок 1 усл. ед. кирпича с каждого с заводов к каждому из строящихся объектов:

$$C = \begin{pmatrix} 8 & 1 & 9 & 7 \\ 4 & 6 & 2 & 12 \\ 3 & 5 & 8 & 9 \end{pmatrix}$$

Составить такой план перевозок кирпича к строящимся объектам, при котором общая стоимость перевозок является минимальной.

Решение задачи

Введем исходные данные задачи в матричной форме. Далее зададим начальные значения переменным задачи и опишем линейную целевую функцию двух переменных.

$$c := \begin{pmatrix} 8 & 1 & 9 & 7 \\ 4 & 6 & 2 & 12 \\ 3 & 5 & 8 & 9 \end{pmatrix} \qquad a := \begin{pmatrix} 100 \\ 150 \\ 50 \end{pmatrix} \qquad b := \begin{pmatrix} 75 & 80 & 60 & 85 \end{pmatrix}$$

$i := 11$

$x_i := 0$

$$f(x) := \left(c_{0,0}\cdot x_0 + c_{0,1}\cdot x_1 + c_{0,2}\cdot x_2 + c_{0,3}\cdot x_3\right) + \left(c_{1,0}\cdot x_4 + c_{1,1}\cdot x_5 + c_{1,2}\cdot x_6 + c_{1,3}\cdot x_7\right) + \left(c_{2,0}\cdot x_8 + c_{2,1}\cdot x_9 + c_{2,2}\cdot x_{10} + c_{2,3}\cdot x_{11}\right)$$

Сформируем решающий блок со служебным словом ***Given***, внутри которого введем ограничения задачи. Далее завершим решающий блок обращением к функции ***Minimize.***

Given

$x_0 + x_4 + x_8 \le b_{0,0}$

$x_1 + x_5 + x_9 \le b_{0,1}$

$x_2 + x_6 + x_{10} \le b_{0,2}$

$x_3 + x_7 + x_{11} \le b_{0,3}$

$x_0 + x_1 + x_2 + x_3 = a_0$

$x_4 + x_5 + x_6 + x_7 = a_1$

$x_8 + x_9 + x_{10} + x_{11} = a_2$

$x_0 \ge 0 \quad x_1 \ge 0 \quad x_2 \ge 0 \quad x_3 \ge 0 \quad x_4 \ge 0 \quad x_5 \ge 0 \quad x_6 \ge 0 \quad x_7 \ge 0 \quad x_8 \ge 0 \quad x_9 \ge 0 \quad x_{10} \ge 0$

$x_{11} \ge 0$

$P := \text{Minimize}(f, x)$

$P^T =$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.00	65.00	0.00	35.00	75.00	15.00	60.00	0.00	0.00	0.00	0.00	50.00

$$NP := \begin{pmatrix} P_0 & P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 & P_7 \\ P_8 & P_9 & P_{10} & P_{11} \end{pmatrix} \qquad NP = \begin{pmatrix} 0 & 65 & 0 & 35 \\ 75 & 15 & 60 & 0 \\ 0 & 0 & 0 & 50 \end{pmatrix}$$

$OPTf := f(P)$

$OPTf = 1.27 \times 10^3$

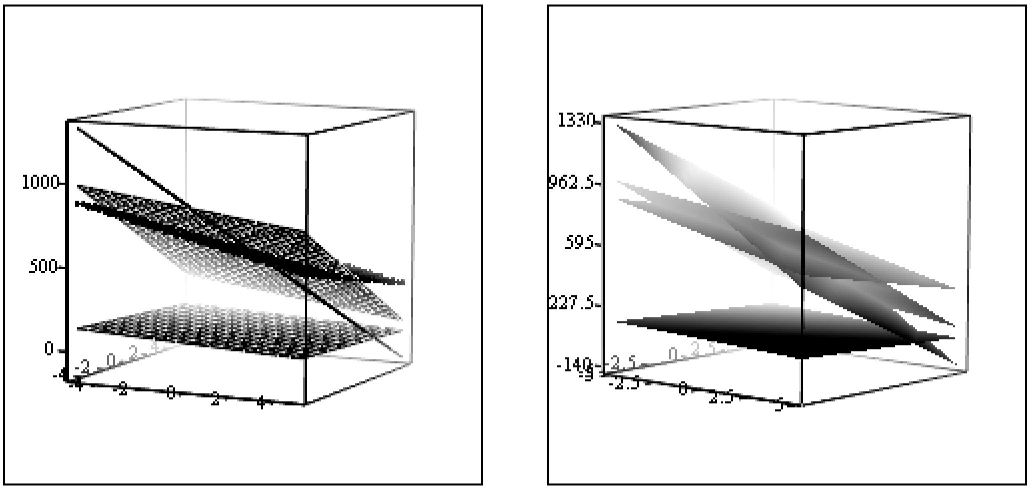
Построим в одной системе координат поверхности функций, соответствующих граничным условиям линейной целевой функции.

$x_0 + x_4 + x_8 \le b_{0,0}$ $\qquad fx0(x4, x8) := \left(b_{0,0} - x4 \cdot c_{1,0} - x8 \cdot c_{2,0}\right) \cdot c_{0,0}$

$x_1 + x_5 + x_9 \le b_{0,1}$ $\qquad fx1(x5, x9) := \left(b_{0,1} - x5 \cdot c_{1,1} - x9 \cdot c_{2,1}\right) \cdot c_{0,1}$

$x_2 + x_6 + x_{10} \le b_{0,2}$ $\qquad fx2(x6, x10) := \left(b_{0,2} - x6 \cdot c_{1,2} - x10 \cdot c_{2,2}\right) \cdot c_{0,2}$

$x_3 + x_7 + x_{11} \le b_{0,3}$ $\qquad fx3(x7, x11) := \left(b_{0,3} - x7 \cdot c_{1,3} - x11 \cdot c_{2,3}\right) \cdot c_{0,3}$



(fx0) ,(fx1) ,(fx2) ,(fx3) $\qquad$ (fx0) ,(fx1) ,(fx2) ,(fx3)

Рис. n2.

Вывод. Изучили методы решения задач линейного программирования и транспортной задачи линейного программирования при помощи симплексного метода и метода потенциалов. Приобрели навыки построения математических моделей линейного программирования в среде MathCAD.

ЛАБОРАТОРНАЯ РАБОТА № 2
«РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДАМИ БИСЕКЦИИ, КАСАТЕЛЬНЫХ, ХОРД, СЕКУЩИХ, ПРОСТОЙ ИТТЕРАЦИИ»

Цель работы:

1. Изучение основных определений и положений теории численного решения нелинейных уравнений.
2. Изучение основных методов численного решения нелинейных уравнений.
3. Разработка программ и решение на ЭВМ нелинейных уравнений.

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.

1. Основные определения. Под решением нелинейного уравнения:

$$f(x)=0 \qquad (14)$$

понимают нахождение корней этого уравнения, то есть определения значений $\bar{x}$, при которых выполняется условие: $f(\bar{x})=0$.

Корень $\bar{x}$ называется простым, если $f'(\bar{x})\neq 0$. Корень $\bar{x}$ называется кратным, если $f'(\bar{x})=0$.

Целое число m называется кратностью корня $\bar{x}$, если $f^{(k)}(\bar{x})=0$ для всех k=1,2,...,m-1, а $f^{(m)}(\bar{x})\neq 0$.

Решение уравнения (2.1) проводится в два этапа: этапа локализации корней и этапа итерационного уточнения корней. На этапе локализации выделяется отрезок [a,b], внутри которого находится только один корень.

На этапе итерационного уточнения корней задается точность вычислений ε и используется итерационный процесс (итерационная формула), в результате которого вычисляются значения последовательности: $x_0, x_1, \ldots, x_n \to \bar{x}$. При выполнении условия $|x_n - \bar{x}| \leq \varepsilon$ итерационный процесс заканчивается.

Если в итерационной формуле для вычисления приближения x_n используется только значение x_{n-1}, то метод называют одношаговым, и k-шаговым, если используется k предыдущих приближений: $x_{n-k}, x_{n-k+1}, \ldots, x_{n-1}$.

Скорость сходимости итерационного процесса определяют с помощью выражения:

$$|x_{n+1}-\bar{x}| \leq c\cdot |x_n - \bar{x}|^p, \quad \begin{cases} p=1, & 1>c>0; \\ p>1, & c>0. \end{cases} \qquad (15)$$

где число p называют порядком сходимости метода. Если p=1- сходимость линейная, p>1- сверх линейная сходимость, p=2- сходимость квадратичная.

Критерий окончания. Точное значение корня мы не знаем. Когда же закончить итерационный процесс, чтобы утверждать, что $|x_{n+1}-\bar{x}| \le \varepsilon$. Этой цели для каждого итерационного процесса существует свой критерий окончания в виде неравенства: $|x_{n+1}-x_n| \le \delta(\varepsilon)$, при выполнении которого всегда имеет место неравенство: $|x_{n+1}-\bar{x}| \le \varepsilon$.

Интервалом неопределенности корня называется интервал $\left[x-\tilde{\varepsilon}, x+\tilde{\varepsilon}\right]$, в котором любое значение может являться корнем уравнения. Появление такого интервала связано с погрешностью вычислений. Величина $\tilde{\varepsilon}$ определяется из условия:

$$\tilde{\varepsilon} = \frac{\bar{\Delta}y}{|f'(\bar{x})|}, \tag{16}$$

где $\bar{\Delta}y$ - абсолютная предельная погрешность при вычислении значения функции y=f(x).

Так как значение $\tilde{\varepsilon}$ определяет абсолютную предельную погрешность $\bar{\Delta}\bar{x}$ вычисления значения корня (результата) из-за погрешности $\bar{\Delta}y$ вычисления значений функции (.входных данных для задачи поиска корня), то величина $\nu_\Delta = \frac{\bar{\Delta}\bar{x}}{\bar{\Delta}y} = \frac{1}{|f'(\bar{x})|}$ называется абсолютным числом обусловленности задачи нахождения корня.

2. Методы решения нелинейных уравнений. Приведем основные методы решения нелинейных уравнений:

а) метод бисекции. Выбирается отрезок [a,b], на концах которого функция имеет разные знаки, следовательно корень находится внутри этого отрезка. Этот отрезок делится пополам и вновь выбирается отрезок, на концах которого функция имеет разные знаки и т.д. Таким образом после n итераций имеем отрезок локализации $[a_n,b_n]$ длина которого в 2^n раз меньше первоначального отрезка [a,b]. Полагая $x_n=(b_n+a_n)/2$ имеем:

$$|x_n-\bar{x}| \le \frac{b_n-a_n}{2} = \frac{b-a}{2^{n+1}} < \varepsilon.$$

Данный метод обладает невысокой скоростью сходимости, но не требует чтобы функция была непрерывной. Однако этот метод нельзя использовать для поиска кратных корней.

б) метод простой итерации. Уравнение f(x)=0 преобразуют к виду удобному для организации итерации: x=φ(x), при этом функция φ(x) называется итерационной функцией. На отрезке локализации [a,b] выбирается начальное приближение $x=x_0$ и вычисляется $x_1=\varphi(x_0)$.

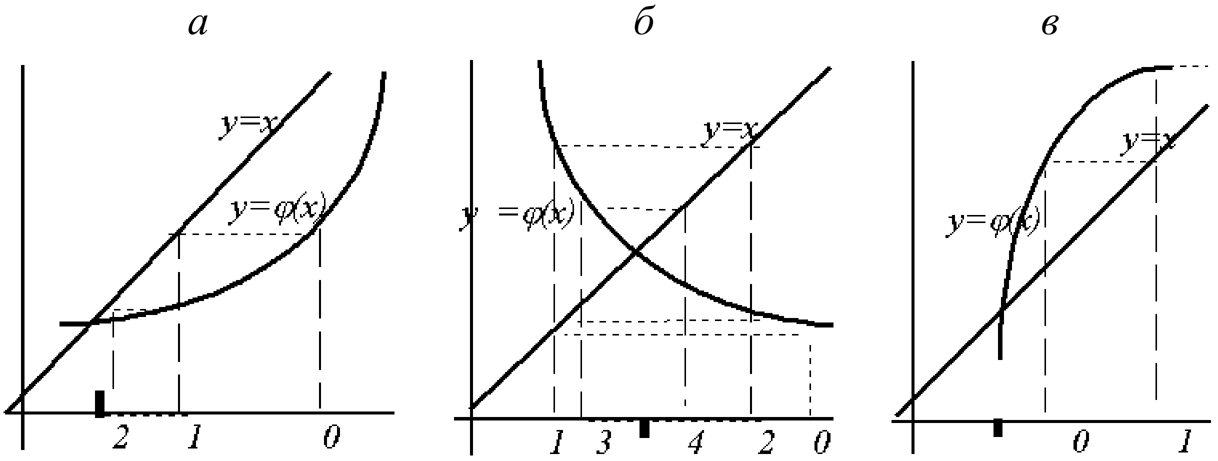


Рис. 2. Геометрическая интерпретация метода простой итерации

Продолжая этот процесс имеем:

$$x_{n+1} = \varphi(x_n) .$$

Если существует $\lim\limits_{n\to\infty} x_n = \bar{x}$, то получаем равенство: $\bar{x} = \varphi(\bar{x})$, где $\bar{x}$ - корень.

Метод сходится при $|\varphi'(x)| < 1,\ \ x \in [a,b]$, а при $|\varphi'(x)| > 1,\ \ x \in [a,b]$ - расходится.

Критерий окончания:

$$(x_{n+1} - x_n) < \frac{q}{1-q}\varepsilon,\ \ q = |\varphi'(\bar{x})|. \tag{17}$$

в) метод касательных (метод Ньютона). Выбирается точка $x_0 \in [a,b]$ и в ней проводится касательная к графику функции y=f(x) и за новое приближение x_1 принимается точка, в которой касательная пересекает ось OX и т.д. В итоге получаем итерационную формулу Ньютона:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{18}$$

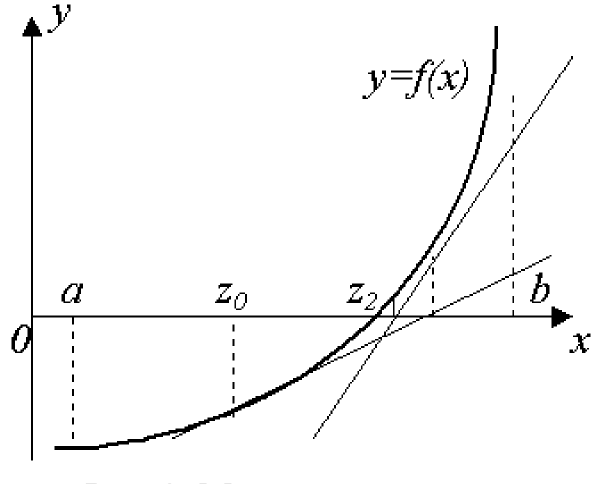


Рис. 3. Метод касательных

Необходимым и достаточным условием сходимости метода Ньютона на отрезке локализации $x \in [a,b]$ являются:

$f'(x) \neq 0$, - (необходимое условие); (19)

$f''(x) \neq 0$ - (достаточное условие);

т.е. знако постоянство первой и второй производной на отрезке локализации.

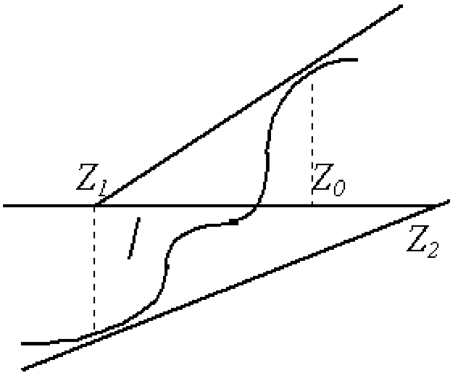


Рис. 4. Расходящийся процесс

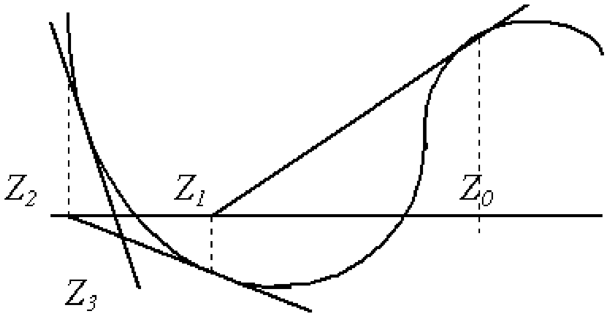


Рис. 5. Приближение к другому корню

Очевидно, что этот метод обеспечивает сходящийся процесс приближений лишь при выполнении некоторых условий (например, при непрерывности и знакопостоянстве первой и второй производной функции в окрестности корня) и при их нарушении либо дает расходящийся процесс (рис. 3), либо приводит к другому корню (рис. 4).

г) метод хорд (модификация метода Ньютона). На отрезке [a,b] производную касательную в формуле (2.5) заменяют приближенным равенством:

$$f'(x_n) \approx \frac{f(b) - f(x_n)}{b - x_n},$$

т.е. хордой. В итоге получаем:

$$x_{n+1} = x_n - \frac{b - x_n}{f(b) - f(x_n)} \cdot f(x_n). \tag{20}$$

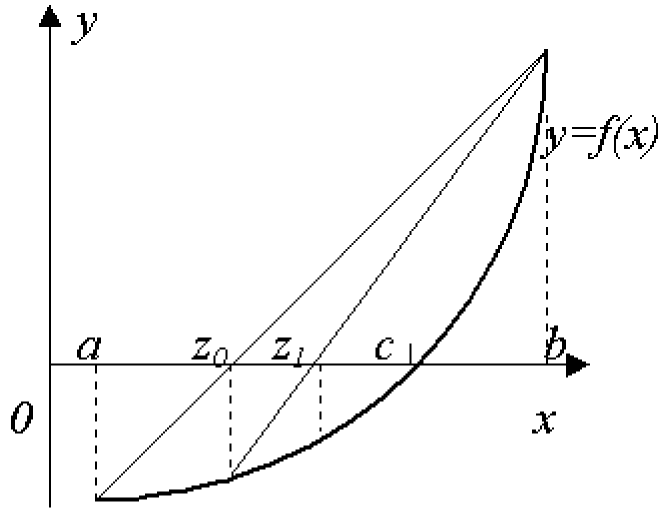


Рис. 6. Метод хорд

д) метод секущих (модификация метода Ньютона). Если теперь точку b в формуле (2.7) заменить на точку x_{n-1}, то получим формулу метода секущих:

$$x_{n+1} = x_n - \frac{x_{n-1} - x_n}{f(x_{n-1}) - f(x_n)} \cdot f(x_n). \tag{21}$$

Метод секущих является двух этапным.

Все методы Ньютона (**касательных, хорд и секущих**) имеют **квадратичную** сходимость.

Критерий окончания методов Ньютона:

$$\left|x_{n+1} - x_n\right| < \varepsilon, \tag{22}$$

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.

Задание. Решить методом секущих нелинейное уравнение: e^x-3-cos(x)=0, с точностью $\varepsilon=10^{-4}$. Предельная абсолютная погрешность вычислений значений функции $y=e^x$-3-cos(x) равна 10^{-10}.

1. Проводим этап локализации. Так как $\left|\cos(x)\right| \le 1$, то при x>2 имеем f(x)>0, а при x=0 имеем f(x)<0. Следовательно корень находится на отрезке [0,2].

2. Оцениваем интервал неопределенности $\tilde{\varepsilon}$ на отрезке [0,2]. Согласно (2.3) имеем:

$$\tilde{\varepsilon} = \frac{\overline{\Delta}}{\left|f'(\overline{x})\right|} \le \frac{\overline{\Delta}}{\underbrace{\min}_{x \in [0,2]} \left|f'(x)\right|}, \tag{23}$$

где $f'(x)=e^x+\sin(x)$.

Получаем $f'(0) = 1;\ f'(2) = 4$. Следовательно $\underbrace{\min}_{x \in [0,2]} f(x) = 1$ и $\tilde{\varepsilon} \approx 10^{-10}$.

Так как интервал неопределенности $\tilde{\varepsilon} \approx 10^{-10}$ меньше погрешности поиска корня $\varepsilon=10^{-4}$, то мы можем найти этот корень с заданной точностью.

3. Для приближенного решения уравнения используем формулу секущих (24):

$$x_{n+1} = x_n - \frac{x_{n-1} - x_n}{f(x_{n-1}) - f(x_n)}, \tag{24}$$

где $f(x)=e^x$-3-cos(x).

Проверяем условия сходимости (2.6):

$$f'(x) = e^x + \sin \quad x > 0, \ \ x \in [0{,}2];$$

$$f''(x) = e^x + \cos \quad x > 0, \ \ x \in [0{,}2].$$

Так как необходимое и достаточное условия выполнены на всем отрезке локализации, то для решения уравнения можем использовать метод секущих.

4. Выбираем начальное значение $x^{(0)}$ и $x^{(1)}$: $x^{(0)}$=0.0, $x^{(1)}$=0.1 и производим вычисления согласно итерационной формулы (4.2). Итерационный процесс завершаем при выполнении условия (2.8):

$$\left| x_{n+1} - x_n \right| < \varepsilon.$$

5. Текст примерной программы для нахождения корня уравнения: методом секущих на языке программирования *Pascal*:

```
program lab;
        {Решение линейных уравнений методом секущих}
        {x0 ,x1 - нулевое и первое приближения значения корня}
        {e - точность решения}
        {x - новое приближение корня}
var x0,x1,e,x of real;
var i: integer;
                function f(y : real) : real;
                {функция для заданного нелинейного уравнения}
                        begin
                            f := exp(y)-3-cos(y);
                        end;        {f}
            begin
              writeln(‘Введите задаваемую точность е’);
            readln (e);
              writeln(‘Введите нулевое и первое приближения);
            readln (x0, x1);
                  i := 0;
                  while abs(x1-x0) >= e do
                      begin
                        x := x1-(x0-x1)/(f(x0)-f(x1));
                        x0 := x1;
                        x1 := x;
                        i := i+1;
                      end;
                writeln(‘Решение нелинейного уравнения методом секущих’);
            writeln(’корень x=’,x,’     точность е=’,e,‘     число итераций i=’,i);
end.
```

6. Заполняем таблицу:

Метод	Корень	Точность	Число итераций

Задание к лабораторной работе №2.

1. Численно решить нелинейное уравнение из таблицы заданий для указанного в ней метода решения и указанной точности.

2. Оценить интервал неопределенности поиска корня, считая, что погрешность округления $\Delta \sim 10^{-10}$ и сравнить его с задаваемой точностью решения.

3. Проверить сходимость указанного метода на отрезке локализации.

4. Для указанного численного метода записать все соотношения, которые необходимы для разработки алгоритма программы.

5. Написать программу и рассчитать на ЭВМ значение корня указанного уравнения.

6. Разработать программу для решения данного уравнения методом бисекции. Рассчитать на ЭВМ значение корня. Сравнить результаты двух методов.

ТАБЛИЦА ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ.

№	f(x)	Метод	Точность
1	$\ln(x)-1/(1+x^2)$	хорд	10^{-5}
2	$\ln(\ln(x))-x^2$	касательных	10^{-6}
3	$x-1/\sqrt{e^x}$	простой итерации	10^{-4}
4	$x^4-13x^2+36-(1/x)$	секущих	10^{-5}
5	$2x^2-x^4-1-\ln(x)$	хорд	10^{-6}
6	$x^3-3x-2e^{-x}$	касательных	10^{-4}
7	$\sin(x^2)-6x+1$	простой итерации	10^{-5}
8	$\cos(x^2)-10x$	секущих	10^{-5}
9	$\ln^2 x-(1/x)$	хорд	10^{-4}
10	$\mathrm{arctg}(x)-\ln(x)$	простой итерации	10^{-6}
11	$\ln(1+x)/(1-x)-\cos(x^2)$	касательных	10^{-4}
12	$e^x-3-\cos(x)$	секущих	10^{-5}
13	$e^x-\mathrm{arctg}(x)$	хорд	10^{-5}
14	$x-\mathrm{arctg}(1/x)$	касательных	10^{-4}
15	$x-1/(x^4-13x^2+36)$	простой итерации	10^{-6}

СОДЕРЖАНИЕ ОТЧЕТА.

1. Название лабораторной работы.
2. Индивидуальное задание (2 варианта, содержащих разные методы, на подгруппу).
3. Теоретическая часть.
4. Реализация программы в MathCAD.
5. Результаты расчета.

Пункты 1-4, а также таблица должны, быть оформлены до начала лабораторной работы.

КОНТРОЛЬНЫЕ ВОПРОСЫ.

1. Что называется корнем уравнения?
2. Определение простого и кратного корня.
3. Основные этапы поиска корня.

4. Определение скорости и порядка сходимости численного метода поиска корня.
5. Определение интервала неопределенности корня.
6. Число обусловленности задачи поиска корня нелинейного уравнения.
7. Метод бисекции.
8. Метод простой операции.
9. Методы Ньютона и его модификации.
10. Критерий окончания итерационного процесса поиска корня.
11. Метод Ньютона для решения системы нелинейных уравнений.
12. Определение якобиана.

Пример выполнения лабораторной работы №2

Лабораторная работа №2

Решение нелинейных уравнений методами бисекции, касательных, хорд, секущих, простой итерации

Цель работы: изучение основных определений и положений теории численного решения нелинейных уравнений, изучение основных методов численного решения нелинейных уравнений, разработка программ и решение на ЭВМ нелинейных уравнений.

Краткие теоретические сведения

1. Основные определения. Под решением нелинейного уравнения

$$f(x)=0 \tag{1}$$

понимают нахождение корней этого уравнения, то есть определения значений $\bar{x}$, при которых выполняется условие: $f(\bar{x})=0$.

Корень $\bar{x}$ называется простым, если $f'(\bar{x})\neq 0$. Корень $\bar{x}$ называется кратным, если $f'(\bar{x})=0$. Целое число m называется кратностью корня $\bar{x}$, если $f^{(k)}(\bar{x})=0$ для всех k = 1,2,...,m-1, а $f^{(m)}(\bar{x})\neq 0$.

Решение уравнения (1) проводится в два этапа: этапа локализации корней и этапа итерационного уточнения корней. На этапе локализации выделяется отрезок [a,b], внутри которого находится только один корень.

На этапе итерационного уточнения корней задается точность вычислений ε и используется итерационный процесс (итерационная формула), в результате которого вычисляются значения последовательности: $x_0, x_1, \ldots, x_n \to \bar{x}$. При выполнении условия $|x_n - \bar{x}| \le \varepsilon$ итерационный процесс заканчивается.

Если в итерационной формуле для вычисления приближения x_n используется только значение x_{n-1}, то метод называют одношаговым, и k-шаговым, если используется k предыдущих приближений: $x_{n-k}, x_{n-k+1}, \ldots, x_{n-1}$.

Скорость сходимости итерационного процесса определяют с помощью выражения:

$$|x_{n+1} - \bar{x}| \le c \cdot |x_n - \bar{x}|^p, \quad \begin{cases} p=1, & 1>c>0; \\ p>1, & c>0. \end{cases} \tag{2}$$

где число p называют порядком сходимости метода. Если p=1- сходимость линейная, p>1- сверх линейная сходимость, p=2- сходимость квадратичная.

Критерий окончания. Точное значение корня мы не знаем. Когда же закончить итерационный процесс, чтобы утверждать, что $\left|x_{n+1} - \bar{x}\right| \leq \varepsilon$. Этой цели для каждого итерационного процесса существует свой критерий окончания в виде неравенства: $\left|x_{n+1} - x_n\right| \leq \delta(\varepsilon)$, при выполнении которого всегда имеет место неравенство: $\left|x_{n+1} - \bar{x}\right| \leq \varepsilon$.

Интервалом неопределенности корня называется интервал $\left[x - \tilde{\varepsilon}, x + \tilde{\varepsilon}\right]$, в котором любое значение может являться корнем уравнения. Появление такого интервала связано с погрешностью вычислений. Величина $\tilde{\varepsilon}$ определяется из условия:

$$\tilde{\varepsilon} = \frac{\overline{\Delta y}}{\left|f'(\bar{x})\right|}, \tag{3}$$

где $\overline{\Delta y}$ - абсолютная предельная погрешность при вычислении значения функции y=f(x).

Так как значение $\tilde{\varepsilon}$ определяет абсолютную предельную погрешность $\overline{\Delta x}$ вычисления значения корня (результата) из-за погрешности $\overline{\Delta y}$ вычисления значений функции (.входных данных для задачи поиска корня), то величина $\nu_\Delta = \frac{\overline{\Delta x}}{\overline{\Delta y}} = \frac{1}{\left|f'(\bar{x})\right|}$ называется абсолютным числом обусловленности задачи нахождения корня.

Задание на лабораторную работу

Численно решить следующие нелинейные уравнения указанным методом решения и с указанной точностью:

№	f(x)	Метод	Точность
1	$\ln(x) - 1/(1 + x^2)$	**хорд**	**10^{-5}**
2	$x^3 - 3x - 2e^{-x}$	**касательных**	**10^{-4}**

Решение уравнения №1 методом хорд (модификация метода Ньютона)

Зададим уравнение №1 как функцию от x:

$$\mathrm{func1}(x) := \ln(x) - \frac{1}{\left(1 + x^2\right)}$$

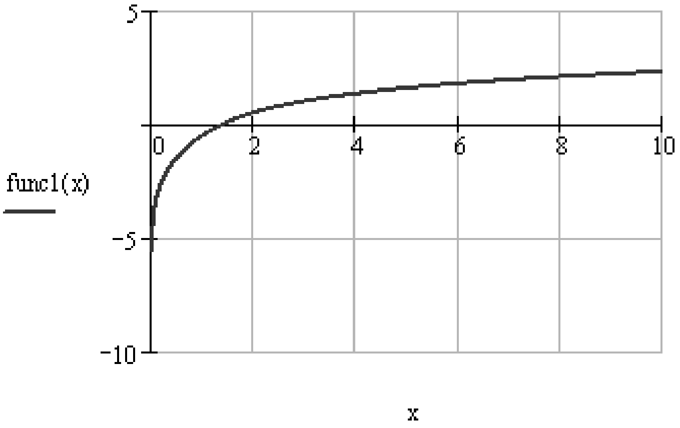


Рис. n1.

Составим программу в среде MathCAD в виде функции, возвращающей матрицу-столбец, первым элементом которой является корень нелинейного уравнения, вторым – количество итераций. Параметры функции: x0 и x1 – задают интервал для первого приближения численного решения, ec – задаваемая точность вычислений.

```
f1(x0,x1,ec) := | i ← 0
                | x ← 0
                | while (|x1 − x0| ≥ ec)
                |     | x ← x1 − [[(x0 − x1)/(func1(x0) − func1(x1))]·func1(x1)]
                |     | x0 ← x1
                |     | x1 ← x
                |     | i ← i + 1
                | f1 ← (x, i)
```

$$\mathrm{f1}\left(1, 11, 10^{-5}\right) = \begin{pmatrix} 1.401 \\ 14 \end{pmatrix}$$

$$\mathrm{f1}\left(1, 5, 10^{-5}\right) = \begin{pmatrix} 1.401 \\ 8 \end{pmatrix}$$

$$\mathrm{f1}\left(1.4, 1.5, 10^{-5}\right) = \begin{pmatrix} 1.401 \\ 3 \end{pmatrix}$$

Решение уравнения №1 методом касательных (метод Ньютона)

Найдем первую производную по x функции func1(x):

$$\frac{d}{dx}\left[\ln(x) - \frac{1}{\left(1 + x^2\right)}\right] = \frac{1}{x} + \frac{2}{\left(1 + x^2\right)^2}\cdot x$$

$$\text{func11}(x) := \left[\frac{1}{x} + \frac{2}{\left(1 + x^2\right)^2}\cdot x\right]$$

Далее составим программу для численного решения уравнения, аналогичной программе для решения уравнения методом хорд:

```
f1(x0,x1,ec) := | i ← 0
                | x ← 0
                | while (|x1 − x0| ≥ ec)
                |     | x ← x1 − $\frac{\text{func1}(x1)}{\text{func11}(x0)}$
                |     | x0 ← x1
                |     | x1 ← x
                |     | i ← i + 1
                | f1 ← $\begin{pmatrix} x \\ i \end{pmatrix}$
```

$$\text{f1}\left(1,2.7,10^{-5}\right) = \begin{pmatrix} 1.401 \\ 10 \end{pmatrix}$$

$$\text{f1}\left(1,2,10^{-5}\right) = \begin{pmatrix} 1.401 \\ 7 \end{pmatrix}$$

$$\text{f1}\left(1.4,1.5,10^{-5}\right) = \begin{pmatrix} 1.401 \\ 4 \end{pmatrix}$$

Решение уравнения №2 методом хорд (модификация метода Ньютона)

$$\text{func6}(x) := x^3 - 3x - 2\cdot e^{-x}$$

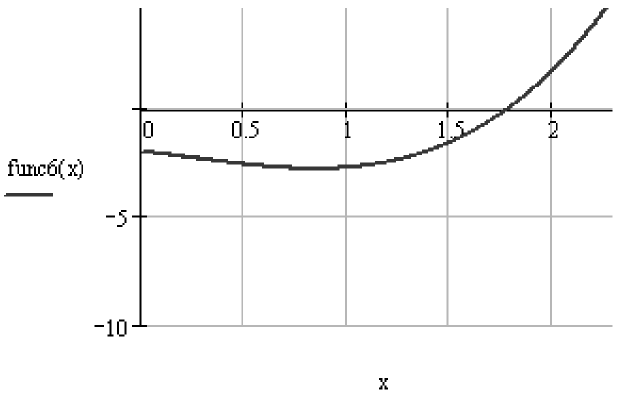
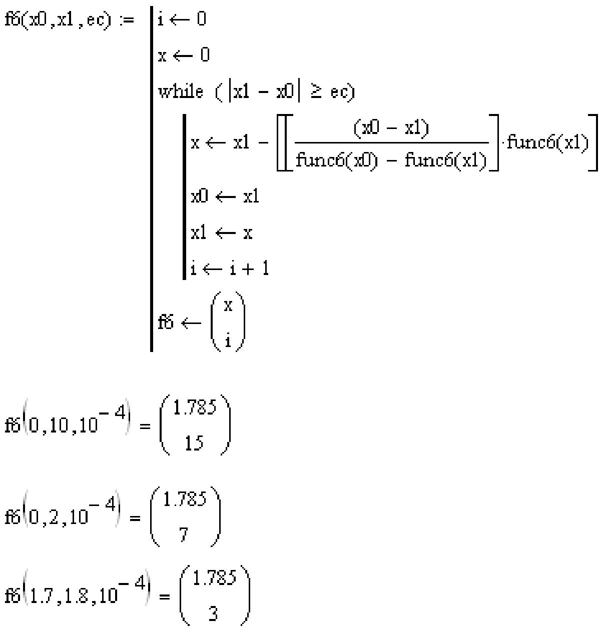


Рис. n2.



```
f6(x0,x1,ec) := | i ← 0
                | x ← 0
                | while (|x1 − x0| ≥ ec)
                |     | x ← x1 − [[(x0 − x1) / (func6(x0) − func6(x1))]·func6(x1)]
                |     | x0 ← x1
                |     | x1 ← x
                |     | i ← i + 1
                | f6 ← (x, i)
```

$$f6\left(0,10,10^{-4}\right)=\begin{pmatrix}1.785\\15\end{pmatrix}$$

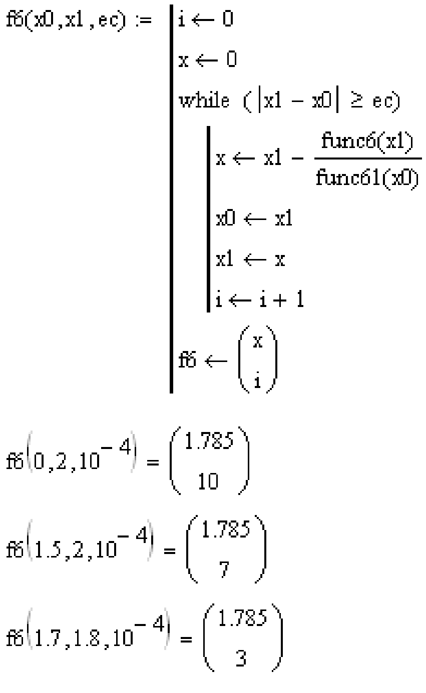
$$f6\left(0,2,10^{-4}\right)=\begin{pmatrix}1.785\\7\end{pmatrix}$$

$$f6\left(1.7,1.8,10^{-4}\right)=\begin{pmatrix}1.785\\3\end{pmatrix}$$

Решение уравнения №2 методом касательных (метод Ньютона)

$$\frac{d}{dx}\left(x^{3}-3\cdot x-2\cdot e^{-x}\right)\ =\ 3\cdot x^{2}-3+2\cdot \exp(-x)$$

$$\text{func61}(x):=\left(3\cdot x^{2}-3+2\cdot \exp(-x)\right)$$



```
f6(x0,x1,ec) := | i ← 0
                | x ← 0
                | while (|x1 − x0| ≥ ec)
                |     | x ← x1 − func6(x1) / func61(x0)
                |     | x0 ← x1
                |     | x1 ← x
                |     | i ← i + 1
                | f6 ← (x, i)
```

$$f6\left(0,2,10^{-4}\right)=\begin{pmatrix}1.785\\10\end{pmatrix}$$

$$f6\left(1.5,2,10^{-4}\right)=\begin{pmatrix}1.785\\7\end{pmatrix}$$

$$f6\left(1.7,1.8,10^{-4}\right)=\begin{pmatrix}1.785\\3\end{pmatrix}$$

Вывод. Изучили основные определения и положения теории численного решения нелинейных уравнений, изучили основные методы численного решения нелинейных уравнений.

ЛАБОРАТОРНАЯ РАБОТА №3
«МАРКОВСКИЕ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ (СМО)»

Системы массового обслуживания - это такие системы, в которые в случайные моменты времени поступают заявки на обслуживание, при этом поступившие заявки обслуживаются с помощью имеющихся в распоряжении системы каналов обслуживания.

Одноканальная СМО (с отказами) с пуассоновским входным потоком и экспоненциальным распределением длительности обслуживания заявок.

Простейшей одноканальной моделью с вероятностным входным потоком и процедурой обслуживания является модель, характеризуемая показательным распределением как длительностей интервалов между поступлениями требований, так и длительностей обслуживания. При этом плотность распределения длительностей интервалов между поступлениями требований имеет вид:

$$f_1(t) = \lambda\, \mathbf{e}^{(-\lambda\, t)}, \qquad (25)$$

где λ- интенсивность поступления заявок в систему.

Под интенсивностью потока понимают

$$\lambda(t) = \lim_{\tau \to 0} \frac{\mathrm{m}(t, t + \tau)}{\tau}, \qquad (26)$$

где m(t, t+τ) - среднее число событий в интервале (t, t+ τ).

Плотность распределения длительностей обслуживания:

$$f_2(t) = \mu\, \mathbf{e}^{(-\mu\, t)}, \qquad (27)$$

где μ - интенсивность обслуживания.

Поток заявок и обслуживания простейшие, т.е. обладающие свойствами стационарности (среднее число событий, воздействующих на систему, в течение единицы времени, остается постоянным), ординарности (вероятность попадания на элементарный участок времени двух и более событий пренебрежимо мала) и отсутствия последействия (для любых непересекающихся участков времени количество событий, попадающих на один из них, не зависит от того, сколько событий попало на другие участки времени).

Для простейшего потока интенсивность λ = const.

Пусть система работает с отказами. Необходимо определить абсолютную и относительную пропускную способности системы. Система имеет два состояния:: S_0 - канал свободен и S_1 - канал занят. Обозначим вероятности состояний:

$P_0(t)$ - вероятность состояния S_0 , $P_1(t)$ - вероятность состояния S_1 . Составим систему уравнений Колмогорова:

$$\frac{\partial}{\partial t}P_0(t) = -\lambda P_0(t) + \mu P_1(t) \tag{28}$$

$$\frac{\partial}{\partial t}P_1(t) = -\mu P_0(t) + \lambda P_1(t) \tag{29}$$

С учетом того, что $P_0(t) + P_1(t) = 1$, решение системы такое:

$$P_0(t) = \frac{\lambda\, \mathbf{e}^{(-(\lambda+\mu)\,t)}}{\lambda+\mu} + \frac{\mu}{\lambda+\mu} \quad ; \tag{30}$$

$$P_1(t) = 1 - P_0(t) \quad . \tag{31}$$

Для 1-канальной СМО с отказами вероятность $P_0(t)$ есть не что иное, как относительная пропускная способность системы q: $q = P_0(t)$.

По истечении большого интервала времени (при t -> ∞) достигается стационарный режим:

$$P_0 = \frac{\mu}{\lambda+\mu} \quad . \tag{32}$$

Абсолютная пропускная способность (А) - среднее число заявок, которое может обслужить СМО в единицу времени:

$$A = \lambda P_0 \tag{33}$$

или

$$A = \frac{\lambda\mu}{\lambda+\mu} \tag{34}$$

Вероятность отказа в обслуживании заявки будет равна вероятности состояния "канал занят":

$$P_{otk} = 1 - P_0 \quad . \tag{35}$$

Данная величина может быть интерпретирована как средняя доля необслуженных заявок среди поданных.

Многоканальная модель (с отказами) с пуассоновским входным потоком и экспоненциальным распределением длительности обслуживания

Пусть СМО с отказами имеет n каналов обслуживания, функционирующих независимо друг от друга. Входной поток заявок и поток обслуживания заявок являются пуассоновскими. Интенсивность поступления заявок равна λ, интенсивность обслуживания μ . Обозначим состояния СМО так:

S_0 - все каналы свободны;

S_1 - занят один канал, остальные свободны;

....

S_k - заняты ровно k каналов, остальные свободны;

....

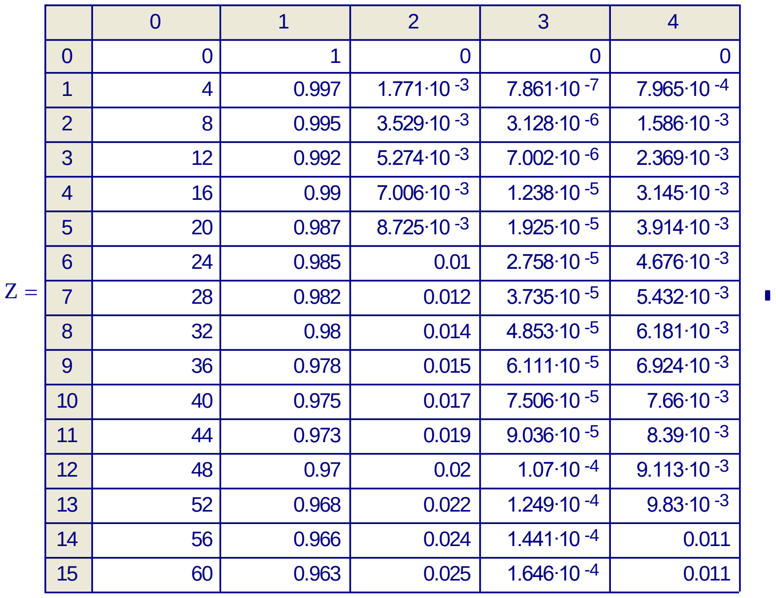
S_n - все n каналов заняты, поступившая в СМО заявка получает отказ в обслуживании. Уравнения Колмогорова для вероятностей состояний системы P_0, P_1, …, P_n строятся по следующему правилу:

1. В системе столько уравнений, сколько возможно состояний объекта;
2. В левой части каждого уравнения стоит производная по времени соответствующей вероятности нахождения объекта в данном состоянии;
3. Число слагаемых в правой части равно числу стрелок, соединяющих рассматриваемое состояние с другими состояниями;
4. Каждое слагаемое представляет собой произведение интенсивности отказа или восстановления на вероятность состояния, из которого выходит эта стрелка; причем слагаемое берется со знаком “+”, если стрелка входит в вершину графа, и со знаком “-” в противном случае;

Система Д.У. дополняется начальными условиями.

Начальные условия решения СМО таковы: $P_0(0) = 1$, $P_1(0) = 0$, …, $P_n(0) = 0$

Решить систему дифференциальных уравнений можно при помощи функции ***rkfixed(y, x1, x2, m, D)*** программного пакета MathCAD**.** Тогда решение будет представлено в виде матрицы, как показано на рисунке 1 (в данном случае начальные условия системы: $P_1=P_2=P_3=0$, $P_0=1$, $y=P$, $x1 = 0$, $x2=60$, $m=15$, D составлено из правых частей уравнений Колмогорова):



Z =

	0	1	2	3	4
0	0	1	0	0	0
1	4	0.997	$1.771\cdot10^{-3}$	$7.861\cdot10^{-7}$	$7.965\cdot10^{-4}$
2	8	0.995	$3.529\cdot10^{-3}$	$3.128\cdot10^{-6}$	$1.586\cdot10^{-3}$
3	12	0.992	$5.274\cdot10^{-3}$	$7.002\cdot10^{-6}$	$2.369\cdot10^{-3}$
4	16	0.99	$7.006\cdot10^{-3}$	$1.238\cdot10^{-5}$	$3.145\cdot10^{-3}$
5	20	0.987	$8.725\cdot10^{-3}$	$1.925\cdot10^{-5}$	$3.914\cdot10^{-3}$
6	24	0.985	0.01	$2.758\cdot10^{-5}$	$4.676\cdot10^{-3}$
7	28	0.982	0.012	$3.735\cdot10^{-5}$	$5.432\cdot10^{-3}$
8	32	0.98	0.014	$4.853\cdot10^{-5}$	$6.181\cdot10^{-3}$
9	36	0.978	0.015	$6.111\cdot10^{-5}$	$6.924\cdot10^{-3}$
10	40	0.975	0.017	$7.506\cdot10^{-5}$	$7.66\cdot10^{-3}$
11	44	0.973	0.019	$9.036\cdot10^{-5}$	$8.39\cdot10^{-3}$
12	48	0.97	0.02	$1.07\cdot10^{-4}$	$9.113\cdot10^{-3}$
13	52	0.968	0.022	$1.249\cdot10^{-4}$	$9.83\cdot10^{-3}$
14	56	0.966	0.024	$1.441\cdot10^{-4}$	0.011
15	60	0.963	0.025	$1.646\cdot10^{-4}$	0.011

Рисунок 7 – Решение системы ДУ Колмогорова при помощи функции ***rkfixed***

Стационарное решение СМО имеет вид (формулы Эрланга):

$$P_k = \frac{\rho^k P_0}{k!}, \ k = 1, 2, \ldots, n, \tag{36}$$

$$P_0 = \frac{1}{\sum\limits_{k=0}^{n} \left(\dfrac{\rho^k}{k!} \right)} \tag{37}$$

где $\rho = \dfrac{\lambda}{\mu}$.

Вероятностные характеристики многоканальной СМО с отказами в стационарном режиме таковы:

1) вероятность отказа

$$P_{otk} = P_n \ ; \tag{38}$$

2) вероятность того, что заявка будет принята к обслуживанию (относительная пропускная способность)

$$q = 1 - P_{otk} \ ; \tag{39}$$

3) абсолютная пропускная способность

$$A = \lambda\, q \ ; \tag{40}$$

4) среднее число каналов, занятых обслуживанием

$$k_{sr} = \rho\,(1 - P_{otk}) \ . \tag{41}$$

Задание к лабораторной работе №3.

Пусть ***n***- канальная СМО представляет собой вычислительный центр (ВЦ) для решения поступающих задач. Поток задач, поступающих на ВЦ, имеет интенсивность ***λ*** задач в час. Интенсивность обслуживания ***μ***. Поток заявок на решение задач и поток обслуживания этих заявок являются простейшими.

1. Составьте систему дифференциальных уравнений Колмогорова для финальных вероятностей состояний. Вычислите финальные значения вероятностей состояний ВЦ, используя директиву , используя директиву для подготовки блока решения ***Given*** и функцию ***Find***;

2. Рассчитайте значения:

- вероятности отказа в обслуживании заявки;
- среднее число каналов, занятых обслуживанием;
- относительной пропускной способности ВЦ;
- абсолютной пропускной способности ВЦ.

3. Исследуйте, как изменятся вероятности пребывания СМО в различных состояниях со временем. Для этого составить вектор ***D(p,t),*** состоящий из правых частей уравнений Колмогорова. При помощи функции ***rkfixed(y, x1, x2, m, D)*** получить решение уравнений Колмогорова, представленных в векторе ***D*** на интервале от ***x1*** до ***x2*** при ***m*** фиксированных шагах решения. Изобразить на графике вероятности пребывания в каждом из состояний.

ТАБЛИЦА С ДАННЫМИ ДЛЯ РАСЧЕТОВ

№ подгруппы	λ	μ	n	x1	x2	m
1	0,8	0,5	6	5	100	10
2	0,85	0,55	5	0	200	12
3	0,7	0,6	6	10	300	12
4	0,75	0,65	5	5	200	10
5	0,6	0,4	6	0	150	15
6	0,65	0,45	5	10	300	18
7	0,5	0,7	6	5	200	12
8	0,55	0,35	5	0	100	10

4. Определите оптимальное число каналов, обеспечивающее минимум затрат на систему, при условии достижения требуемого уровня ее безотказной работы для своего варианта.

Пример выполнения лабораторной работы №3

Лабораторная работа №3

Марковские системы массового обслуживания (СМО)

Цель работы: научиться работать с СМО, определять их основные характкристики.

Вариант 1

Èíòåíñèâíîñòü ïîòîêà çàäà÷ (çàäà÷à/÷àñ):
$\lambda := 0.8$

Èíòåíñèâíîñòü îáñëóæèâàíèÿ:
$\mu := 0.5$

Èíòåðâàë ðåøåíèÿ:
$X1 := 5$
$X2 := 100$

Êîëè÷åñòâî ôèêñèð. øàãîâ ðåøåíèÿ:
$m := 10$

Êîëè÷åñòâî êàíàëîâ ÑÌÎ:
$n := 6$

Пусть n - канальная СМО представляет собой вычислительный центр (ВЦ) для решения поступающих задач. Поток задач, поступающих на ВЦ, имеет интенсивность λ задач в час. Интенсивность обслуживания μ. Поток заявок на решение задач и поток обслуживания этих заявок являются простейшими.

Ход работы

1. Составим систему дифференциальных уравнений Колмогорова для финальных вероятностей состояний. Вычислим финальные значения вероятностей состояний ВЦ, используя директиву для подготовки блока решения Given и функцию Find.

$P0 := 0$

$P1 := 0$

$P2 := 0$

$P3 := 0$

$P4 := 0$

$P5 := 0$

$P6 := 0$

Given

$0 = -\lambda \cdot P0 + n \cdot \mu \cdot P1$

$0 = \lambda \cdot P0 - \lambda \cdot P1 - n \cdot \mu \cdot P1 + n \cdot \mu \cdot P2$

$0 = \lambda \cdot P1 - \lambda \cdot P2 - n \cdot \mu \cdot P2 + n \cdot \mu \cdot P3$

$0 = \lambda \cdot P2 - \lambda \cdot P3 - n \cdot \mu \cdot P3 + n \cdot \mu \cdot P4$

$0 = \lambda \cdot P3 - \lambda \cdot P4 - n \cdot \mu \cdot P4 + n \cdot \mu \cdot P5$

$0 = \lambda \cdot P4 - \lambda \cdot P5 - n \cdot \mu \cdot P5 + n \cdot \mu \cdot P6$

$0 = \lambda \cdot P5 - n \cdot \mu \cdot P6$

$P0 + P1 + P2 + P3 + P4 + P5 + P6 = 1$

$M := \mathrm{Find}(P0, P1, P2, P3, P4, P5, P6)$

Финальные значения вероятностных состояний:

$$M = \begin{pmatrix} 0.733 \\ 0.196 \\ 0.052 \\ 0.014 \\ 3.709 \times 10^{-3} \\ 9.89 \times 10^{-4} \\ 2.637 \times 10^{-4} \end{pmatrix}$$

2. Рассчитаем значения:

- вероятности отказа в обслуживании заявки;
- среднее число каналов, занятых обслуживанием;
- относительной пропускной способности ВЦ;
- абсолютной пропускной способности ВЦ.

$$P0 := \frac{1 - \frac{\lambda}{n \cdot \mu}}{1 - \left(\frac{\lambda}{n \cdot \mu}\right)^{n+1}}$$

$P0 = 0.733$

Вероятность отказа:

$$Perr := \left(\frac{\lambda}{n \cdot \mu}\right)^{n} \cdot P0$$

$Perr = 2.637 \times 10^{-4}$

Относительная пропускная способность:

$q := 1 - Perr$

$q = 1$

Абсолютная пропускная способность:

$A := q \cdot \lambda$

$A = 0.8$

Среднее число каналов, занятых обслуживанием:

$$Kaver := \left(\frac{\lambda}{\mu}\right) \cdot (1 - Perr)$$

$Kaver = 1.6$

3. Исследовали, как изменятся вероятности пребывания СМО в различных состояниях со временем. Для этого составим вектор D, состоящий из правых частей уравнений Колмогорова. При помощи функции rkfixed(p, x1, x2, m, D) получим решение уравнений Колмогорова, представленных в векторе D на интервале от x1 до x2 при m фиксированных шагах решения. Изобразим на графике вероятности пребывания в каждом из состояний.

Начальные условия:

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

$$D(t, p) := \begin{pmatrix} -\lambda \cdot p_0 + n \cdot \mu \cdot p_1 \\ \lambda \cdot p_0 - \lambda \cdot p_1 - n \cdot \mu \cdot p_1 + n \cdot \mu \cdot p_2 \\ \lambda \cdot p_1 - \lambda \cdot p_2 - n \cdot \mu \cdot p_2 + n \cdot \mu \cdot p_3 \\ \lambda \cdot p_2 - \lambda \cdot p_3 - n \cdot \mu \cdot p_3 + n \cdot \mu \cdot p_4 \\ \lambda \cdot p_3 - \lambda \cdot p_4 - n \cdot \mu \cdot p_4 + n \cdot \mu \cdot p_5 \\ \lambda \cdot p_4 - \lambda \cdot p_5 - n \cdot \mu \cdot p_5 + n \cdot \mu \cdot p_6 \\ \lambda \cdot p_5 - n \cdot \mu \cdot p_6 \end{pmatrix}$$

$Z := rkfixed(BC, X1, X2, m, D)$

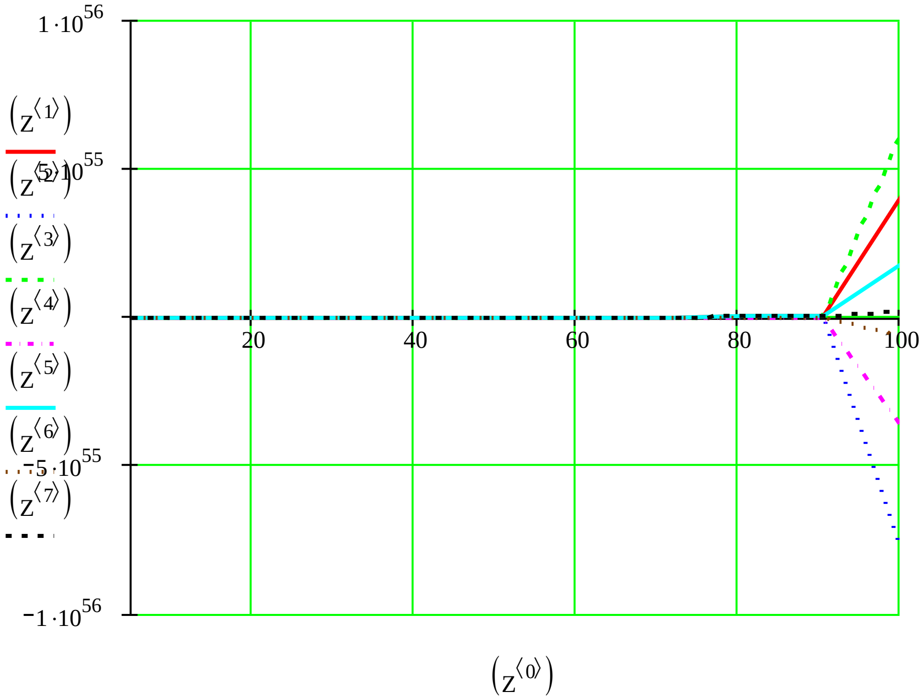


Рис.n1.

На графике изображена зависимость вероятностей состояний системы от времени.

4. Определим оптимальное число каналов, обеспечивающее минимум затрат на систему, при условии достижения требуемого уровня ее безотказной работы для 1-го варианта.

Определение оптимального числа каналов:

$$\mathrm{Perror}(nc) := \left[\left(\frac{\lambda}{nc \cdot \mu}\right)^{nc} \cdot P0\right]$$

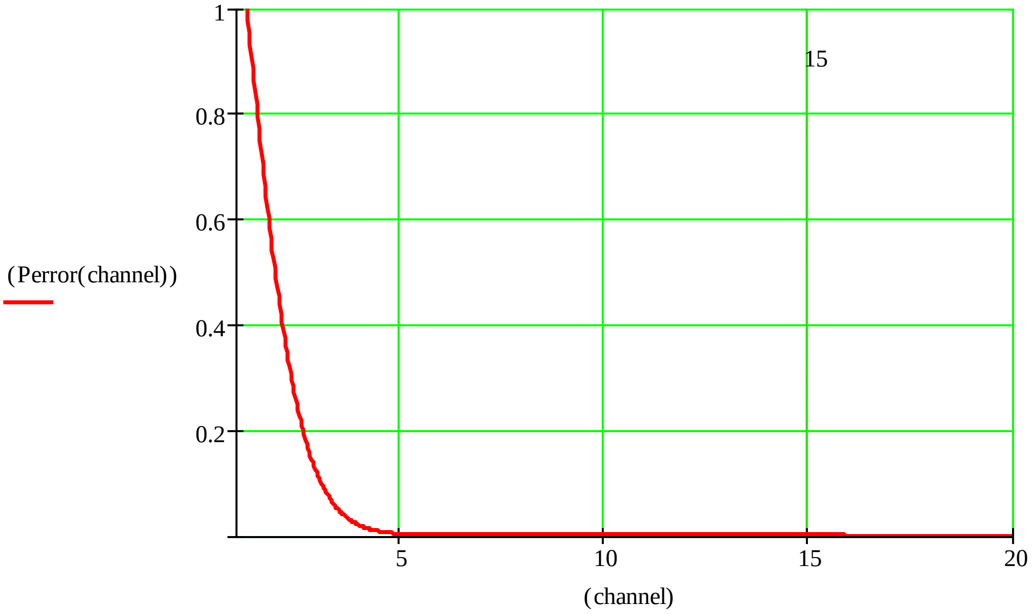


Рис.n2.

ORIGIN := 1

$i := 1..15$

$Err_i := Perror(i)$

$\min(Err) = 1.931 \times 10^{-15}$

```
minchannels :=  f ← 1
                while  min(Err) ≤ Perror(f)
                    f ← f + 1
                    break  if  min(Err) = Perror(f)
                f
```

$minchannels = 15$

Вывод: научились определять основные характеристики СМО, использовать программный пакет MathCad для решения системы дифференциальных уравнений Колмогорова.

ЛАБОРАТОРНАЯ РАБОТА № 4
«ПОСТРОЕНИЕ НЕЧЕТКОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ПАКЕТА FUZZY LOGIC TOOLBOX»

Цель работы: знакомство с использованием пакета Fuzzy Logic Toolbox, реализованным в прикладном программном пакете Matlab и приобретение навыков по построению нечетких систем управления.

Задание

Требуется сконструировать нечеткую систему, отображающую зависимость между двумя переменными x и y. Зависимость y = f(x) задана пятью точками, приведенными в таблице.

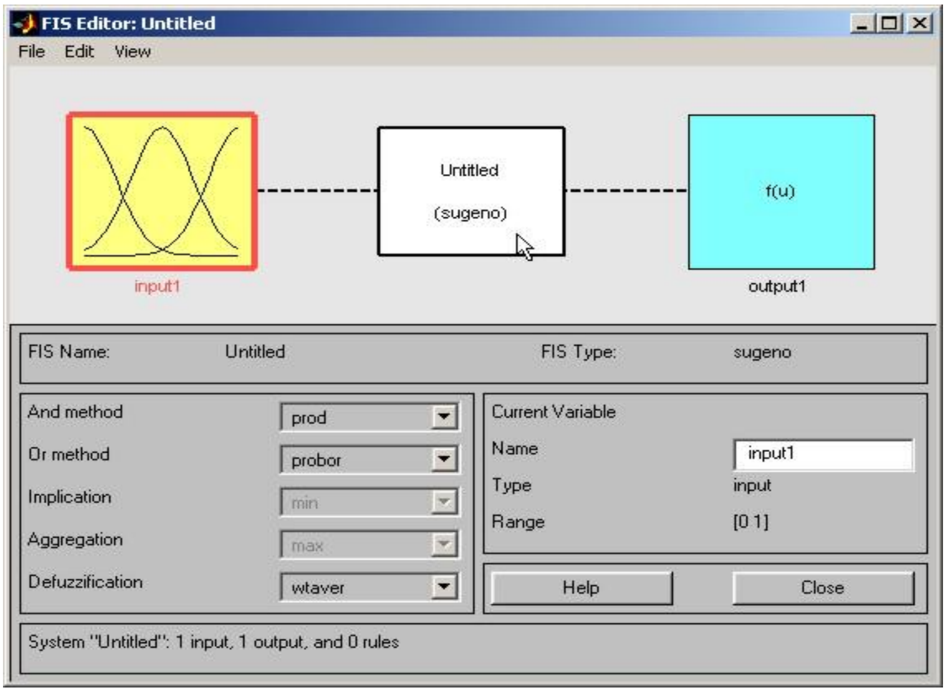
x	-2	-1	0	1	2
y	-8	-1	0	1	8

Решение

Опишем действия необходимые для построения интерполяции данной функции. Для решения задания будем использовать пакет Fuzzy Logic Toolbox (пакет нечеткой логики), входящий в состав MatLab. Этот пакет позволяет конструировать нечеткие экспертные или управляющие системы.

Откроем пункт меню Start/ Toolboxes/Fuzzy Logic/FIS Editor.

Создаем новую нечеткую систему вывода типа Sugeno: File/ New FIS /Sugeno



FIS Name: Untitled

FIS Type: sugeno

And method: prod

Or method: probor

Implication: min

Aggregation: max

Defuzzification: wtaver

Current Variable

Name: input1

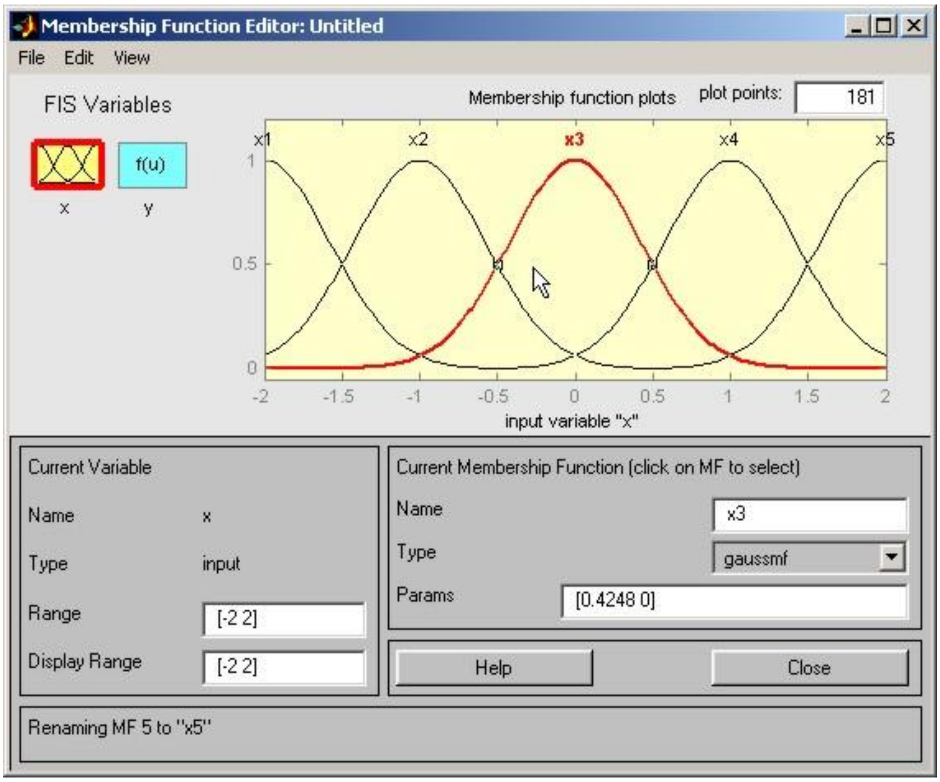
Type: input

Range: [0 1]

System "Untitled": 1 input, 1 output, and 0 rules

Рис. 8.

Далее в редакторе функций принадлежности добавим пять новых гауссовых функций принадлежности (для входного сигнала) при помощи Edit/ Add variables…/Input :



plot points: 181

Current Variable

Name: x

Type: input

Range: [-2 2]

Display Range: [-2 2]

Current Membership Function (click on MF to select)

Name: x3

Type: gaussmf

Params: [0.4248 0]

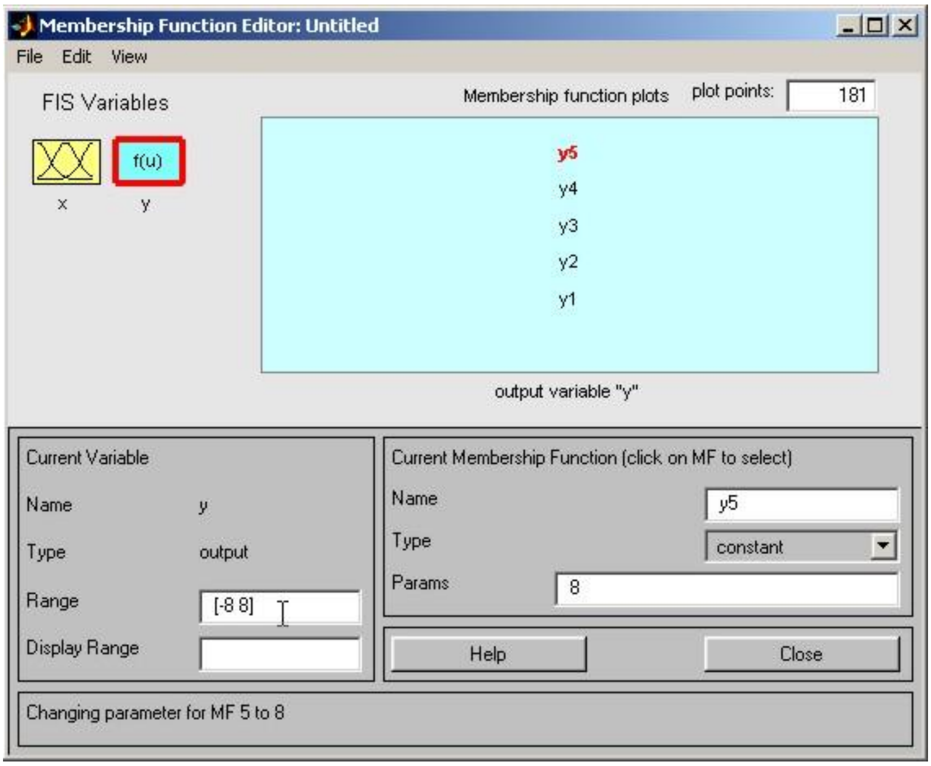
Renaming MF 5 to "x5"

Рис. 9.

В поле Диапазон (Range) задаем диапазон изменения переменной x от -2 до 2, соответственно таблице.

Для успешного решения поставленной задачи необходимо, чтобы ординаты максимумов функций принадлежности совпадали с заданными значениями аргумента x. Смещая графики функций принадлежности по горизонтали, добиваемся выполнения этого условия.

Зададим теперь функции принадлежности для выходной переменной y. Добавим 5 постоянных функций принадлежности (5 – число различных значений выходной переменной y, постоянные функции соответствуют алгоритму Sugeno 0-ого порядка).



plot points: 181

Current Variable

Name y

Type output

Range [-8 8]

Display Range

Current Membership Function (click on MF to select)

Name y5

Type constant

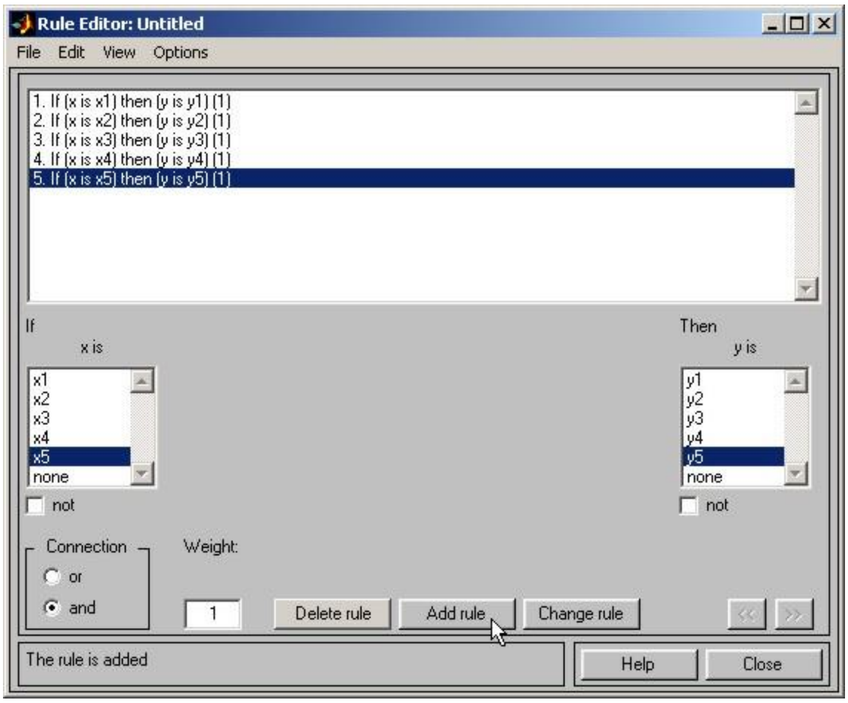
Params 8

Changing parameter for MF 5 to 8

Рис. 10.

Зададим диапазон изменения выходной переменной от -8 до 8 и для каждой переменной зададим значения из таблицы.

В редакторе правил необходимо сконструировать соответствующие правила. При вводе каждого правила устанавливаем соответствие между функцией принадлежности аргумента x и числовым значением y.



1. If (x is x1) then (y is y1) (1)
2. If (x is x2) then (y is y2) (1)
3. If (x is x3) then (y is y3) (1)
4. If (x is x4) then (y is y4) (1)
5. If (x is x5) then (y is y5) (1)

x is: x1, x2, x3, x4, x5, none — x5 selected; ☐ not

y is: y1, y2, y3, y4, y5, none — y5 selected; ☐ not

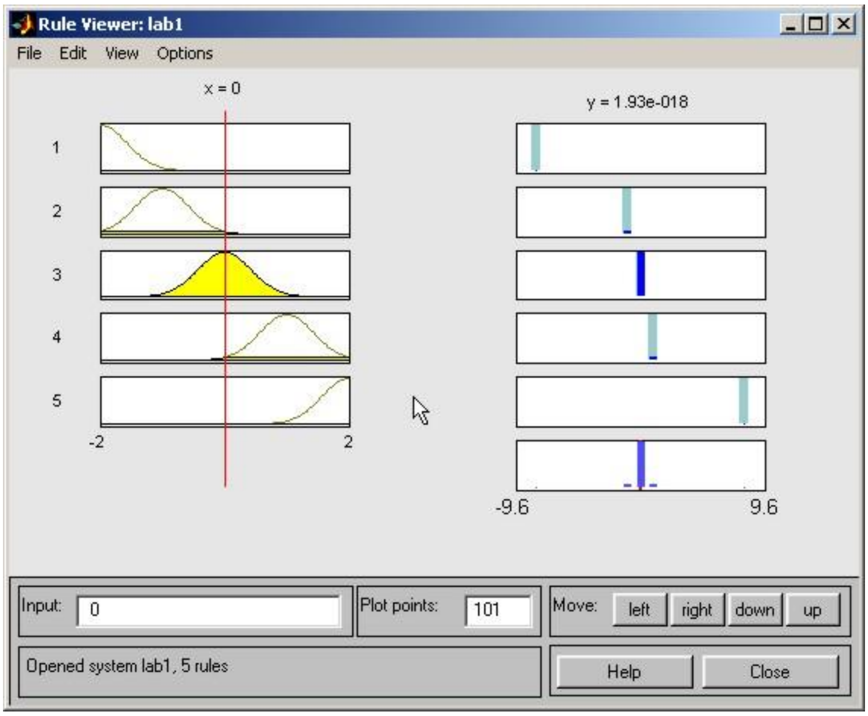
Connection: ○ or ● and

Weight: 1

The rule is added

Рис. 11.

Выберем в пунктах меню Просмотр правил и изучим добавленные зависимости:



x = 0

y = 1.93e-018

Input: 0

Plot points: 101

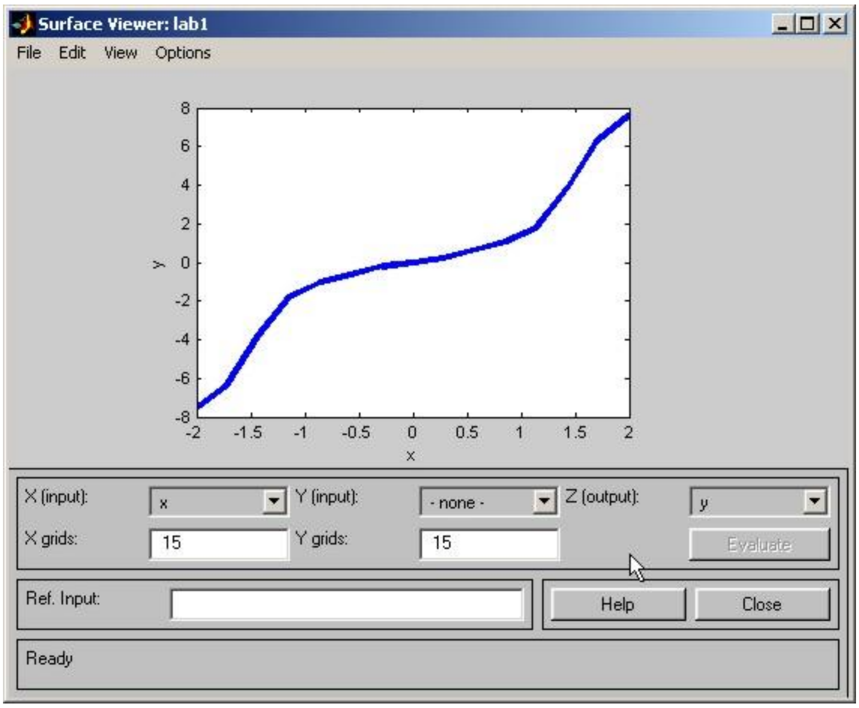
Opened system lab1, 5 rules

Рис. 12.

В правой части окна в графической форме представлены функции принадлежности аргумента x, а в левой – переменной выхода y с пояснением механизма принятия решений.

Перемещая красную черту или изменяя значения аргумента x в правой части окна, получаем значения переменной y в левой части окна. Таким образом, получено решение задачи интерполяции функции, заданной координатами пяти точек.

В окне просмотра поверхности отклика (выхода) можно визуально изучить полученный график функции y = f(x).



X (input): x

Y (input): - none -

Z (output): y

X grids: 15

Y grids: 15

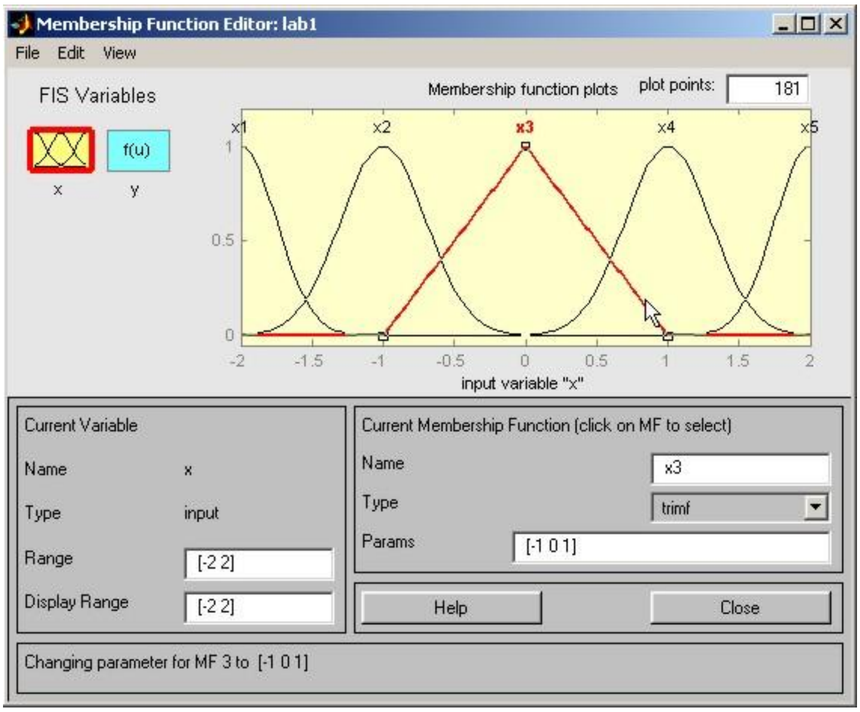
Ref. Input:

Ready

Рис. 13.

Как видно из приведенного графика функция получилась недостаточно четкой. Это обусловлено небольшим количеством исходных точек.

Но изменяя параметры функций принадлежности можно добиться более оптимального результата.



Current Variable

Name x

Type input

Range [-2 2]

Display Range [-2 2]

Current Membership Function (click on MF to select)

Name x3

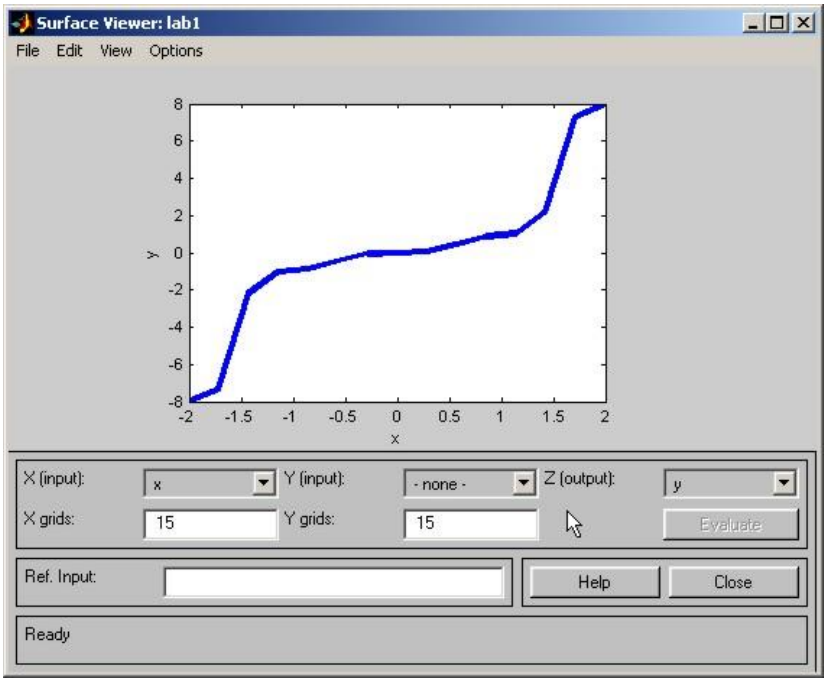
Type trimf

Params [-1 0 1]

plot points: 181

Changing parameter for MF 3 to [-1 0 1]

Рис. 14



X (input): x

Y (input): - none -

Z (output): y

X grids: 15

Y grids: 15

Ref. Input:

Ready

Рис. 15

Задание к лабораторной работе №4

Требуется сконструировать нечеткую систему, отображающую зависимость между двумя переменными x и y. Зависимость y = f(x) задана пятью точками, приведенными в таблице.

x	-2	-1	0	1	2
y	-8	-1	0	1	8

Пример выполнения лабораторной работы №4

Лабораторная работа №4

Построение нечеткой системы с использованием пакета Fuzzy Logic Toolbox

Цель работы: знакомство с использованием пакета Fuzzy Logic Toolbox, реализованным в прикладном программном пакете Matlab и приобретение навыков по построению нечетких систем управления.

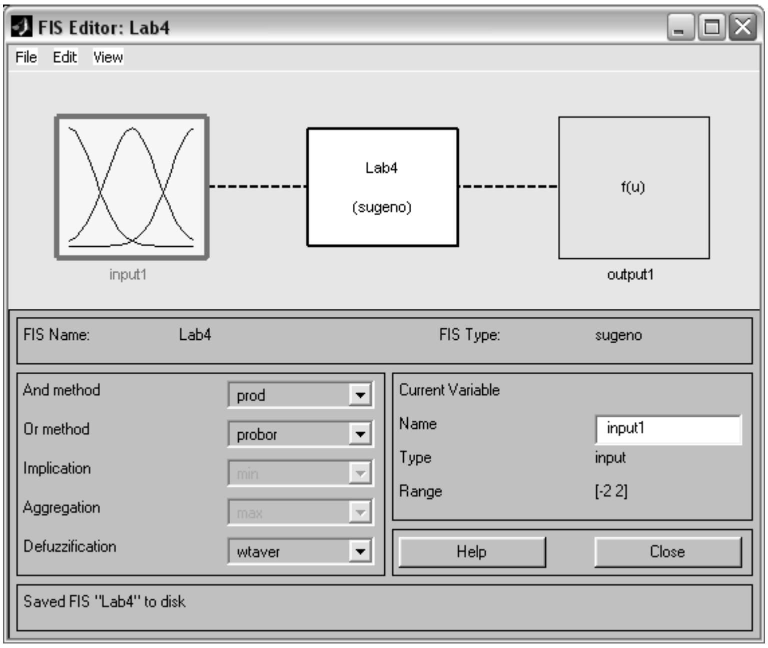
Задание на лабораторную работу

Требуется сконструировать нечеткую систему, отображающую зависимость между двумя переменными x и y. Зависимость y = f(x) задана пятью точками, приведенными в таблице:

x	-2	-1	0	1	2
y	-8	-1	0	1	8

Ход лабораторной работы

Для решения задания будем использовать пакет Fuzzy Logic Toolbox (пакет нечеткой логики), входящий в состав MatLab. Этот пакет позволяет конструировать нечеткие экспертные или управляющие системы. Откроем пункт меню в среде MatLab: Start→Toolboxes→Fuzzy Logic→FIS Editor и создадим новую нечеткую систему вывода типа Sugeno: File→New FIS→Sugeno (рис. n1).



FIS Name: Lab4
FIS Type: sugeno

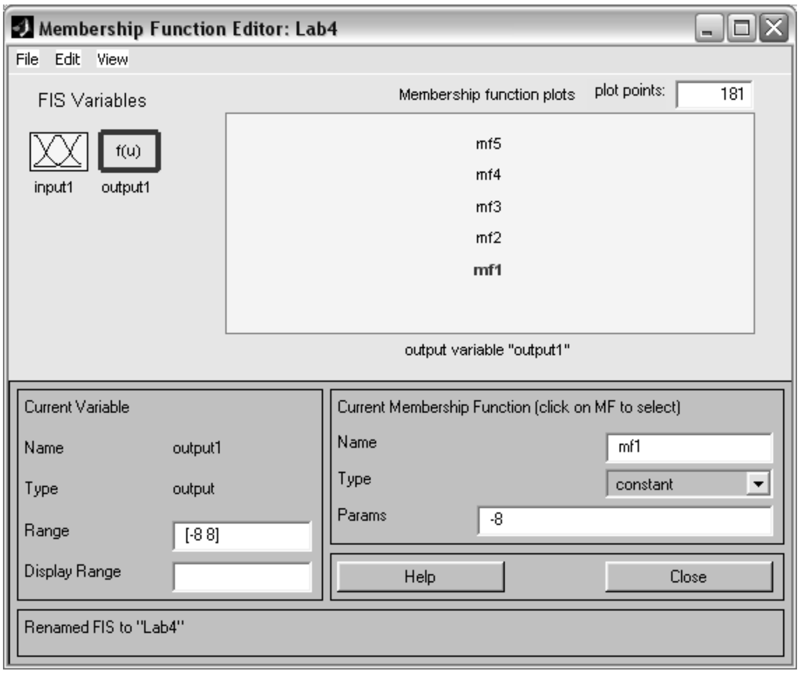
And method: prod
Or method: probor
Implication: min
Aggregation: max
Defuzzification: wtaver

Current Variable
Name: input1
Type: input
Range: [-2 2]

Saved FIS "Lab4" to disk

Рис. n1. Окно FIS Editor с нечеткой системой вывода типа Sugeno.

Далее в редакторе функций принадлежности добавим пять новых гауссовых функций принадлежности (для входного сигнала) при помощи Edit→Add variables…→Input (рис. n2).



plot points: 181

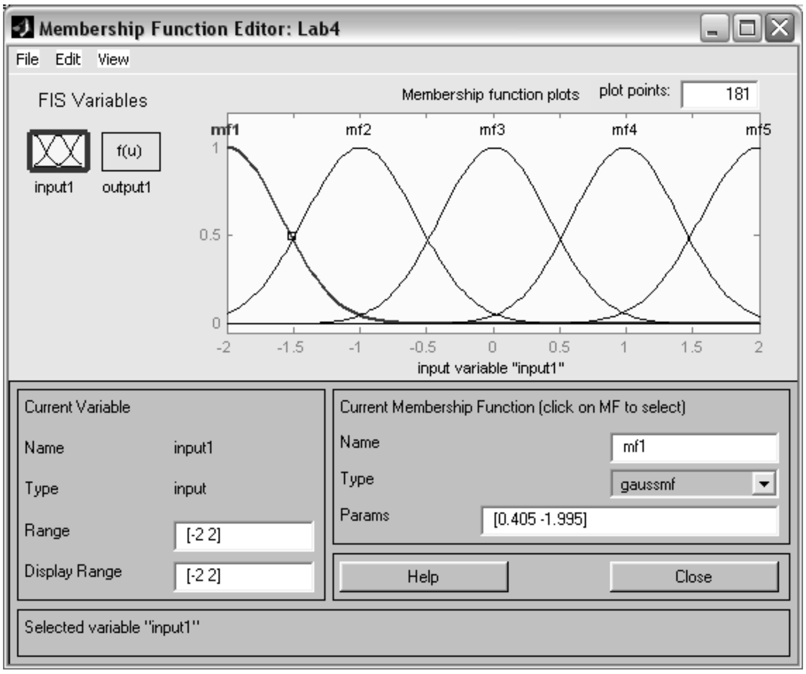
Current Variable
Name: output1
Type: output
Range: [-8 8]
Display Range:

Current Membership Function (click on MF to select)
Name: mf1
Type: constant
Params: -8

Renamed FIS to "Lab4"

Рис. n2. Окно редактора входных и выходных переменных нечеткой функции.

В поле Range задаем диапазон изменения переменной x от -2 до 2, соответственно таблице. Для решения поставленной задачи необходимо, чтобы ординаты максимумов функций принадлежности совпадали с заданными значениями аргумента x. Смещая графики функций принадлежности по горизонтали, добиваемся выполнения этого условия (рис. n3).



plot points: 181

Name input1

Type input

Range [-2 2]

Display Range [-2 2]

Name mf1

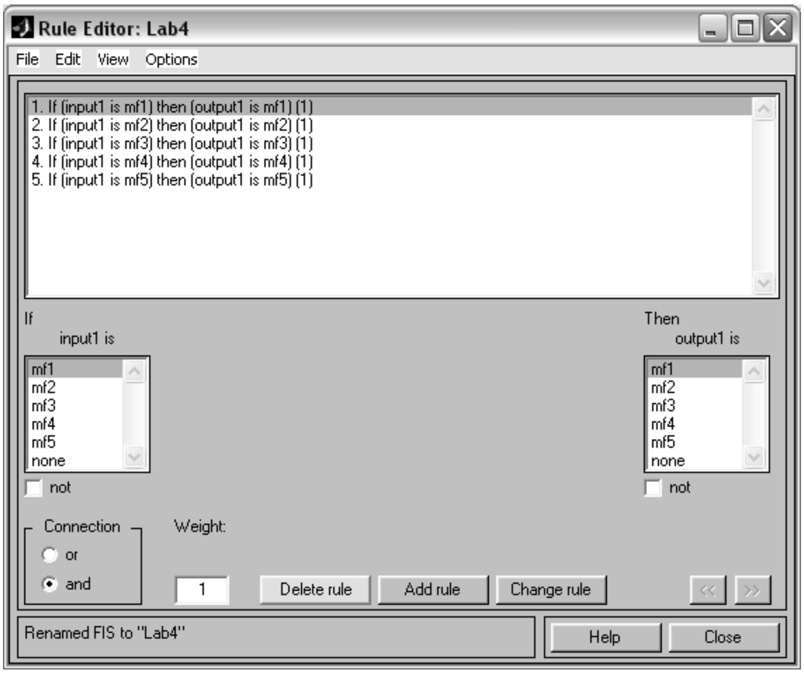
Type gaussmf

Params [0.405 -1.995]

Selected variable "input1"

Рис. n3. Изменение графиков функций принадлежности Гаусса.

Зададим функции принадлежности для выходной переменной y. Добавим 5 постоянных функций принадлежности. Зададим диапазон изменения выходной переменной от -8 до 8 и для каждой переменной зададим значения из таблицы. В редакторе правил необходимо сконструировать соответствующие правила. При вводе каждого правила устанавливаем соответствие между функцией принадлежности аргумента x и числовым значением y (рис. n4).



1. If (input1 is mf1) then (output1 is mf1) (1)
2. If (input1 is mf2) then (output1 is mf2) (1)
3. If (input1 is mf3) then (output1 is mf3) (1)
4. If (input1 is mf4) then (output1 is mf4) (1)
5. If (input1 is mf5) then (output1 is mf5) (1)

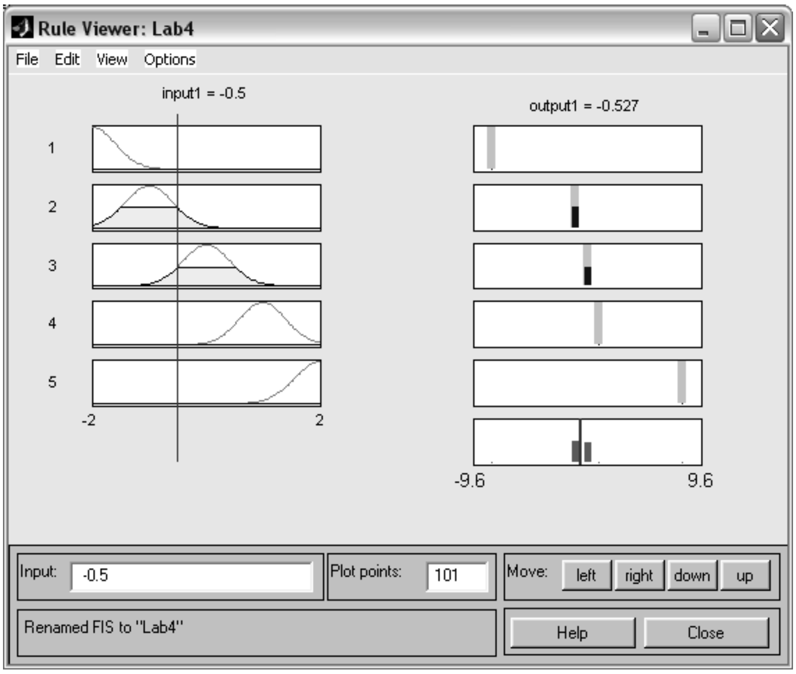
Connection: ○ or ● and

Weight: 1

Renamed FIS to "Lab4"

Рис. n4. Окно редактора правил.

Выберем пункт меню Rule Viewer и изучим добавленные зависимости (рис. n5).



input1 = -0.5

output1 = -0.527

Input: -0.5

Plot points: 101

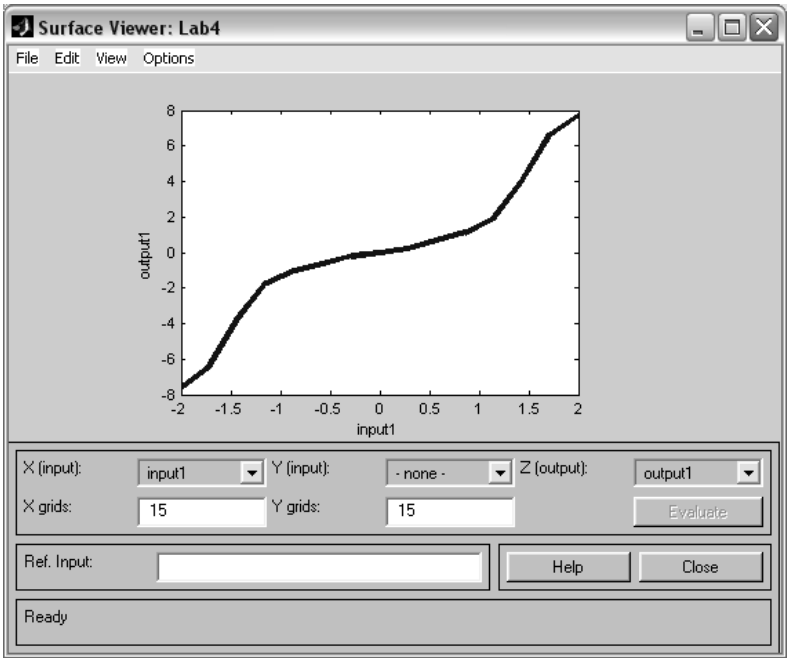
Renamed FIS to "Lab4"

Рис. n5. Окно просмотра правил.

В правой части окна в графической форме представлены функции принадлежности аргумента x, а в левой – переменной выхода y с пояснением механизма принятия решений.

Перемещая красную черту или изменяя значения аргумента x в правой части окна, получаем значения переменной y в левой части окна. Таким образом, получено решение задачи интерполяции функции, заданной координатами пяти точек.

В окне просмотра поверхности отклика (выхода) можно визуально изучить полученный график функции y = f(x) (рис. n6).



X (input): input1

Y (input): - none -

Z (output): output1

X grids: 15

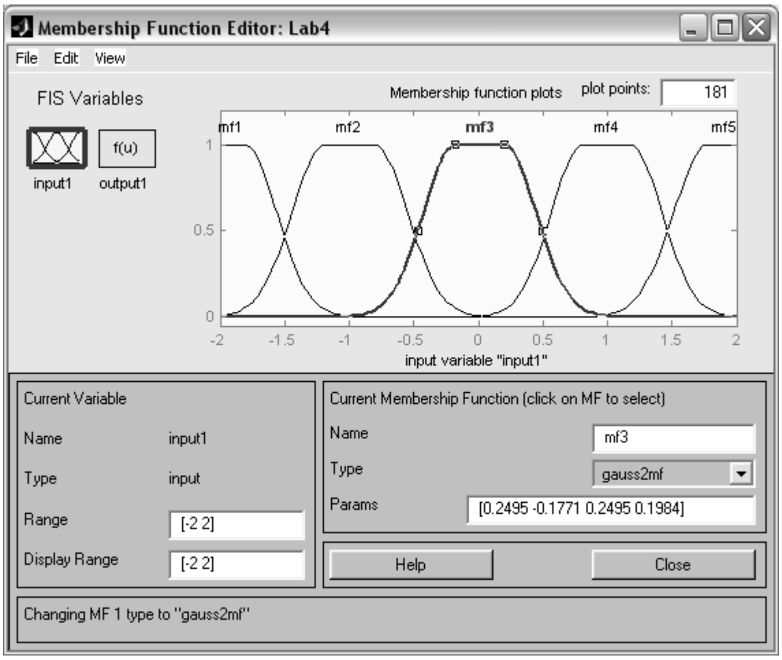
Y grids: 15

Ref. Input:

Ready

Рис. n6. График полученной нечеткой функции.

Изменяя параметры функций принадлежности можно добиться оптимального результата. Например, перейдя от функции принадлежности «gaussmf» к функции «gauss2mf» (рис. n7-n8).



plot points: 181

Name input1

Type input

Range [-2 2]

Display Range [-2 2]

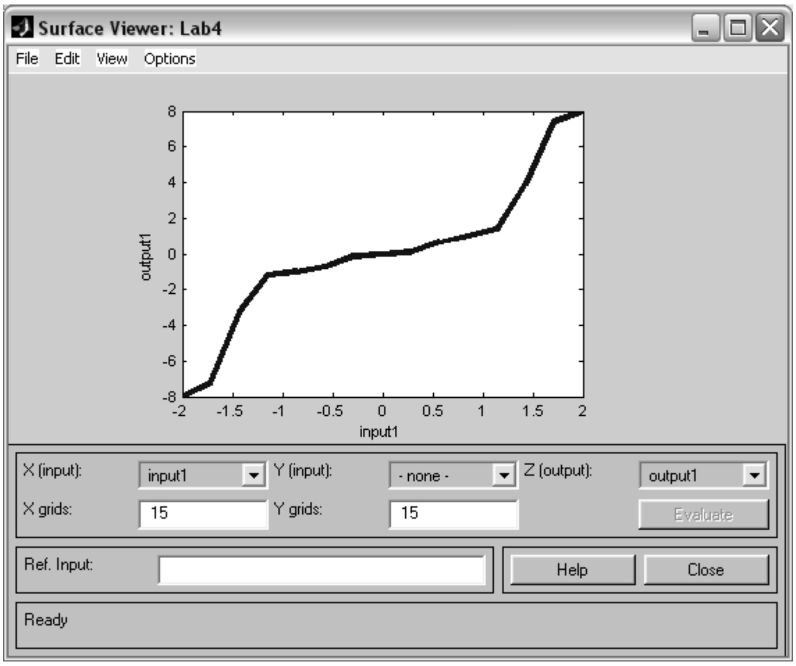
Name mf3

Type gauss2mf

Params [0.2495 -0.1771 0.2495 0.1984]

Changing MF 1 type to "gauss2mf"

Рис. n7. Окно редактора входных и выходных переменных нечеткой функции (применена функция принадлежности «gauss2mf»).



X (input): input1

Y (input): - none -

Z (output): output1

X grids: 15

Y grids: 15

Ref. Input:

Ready

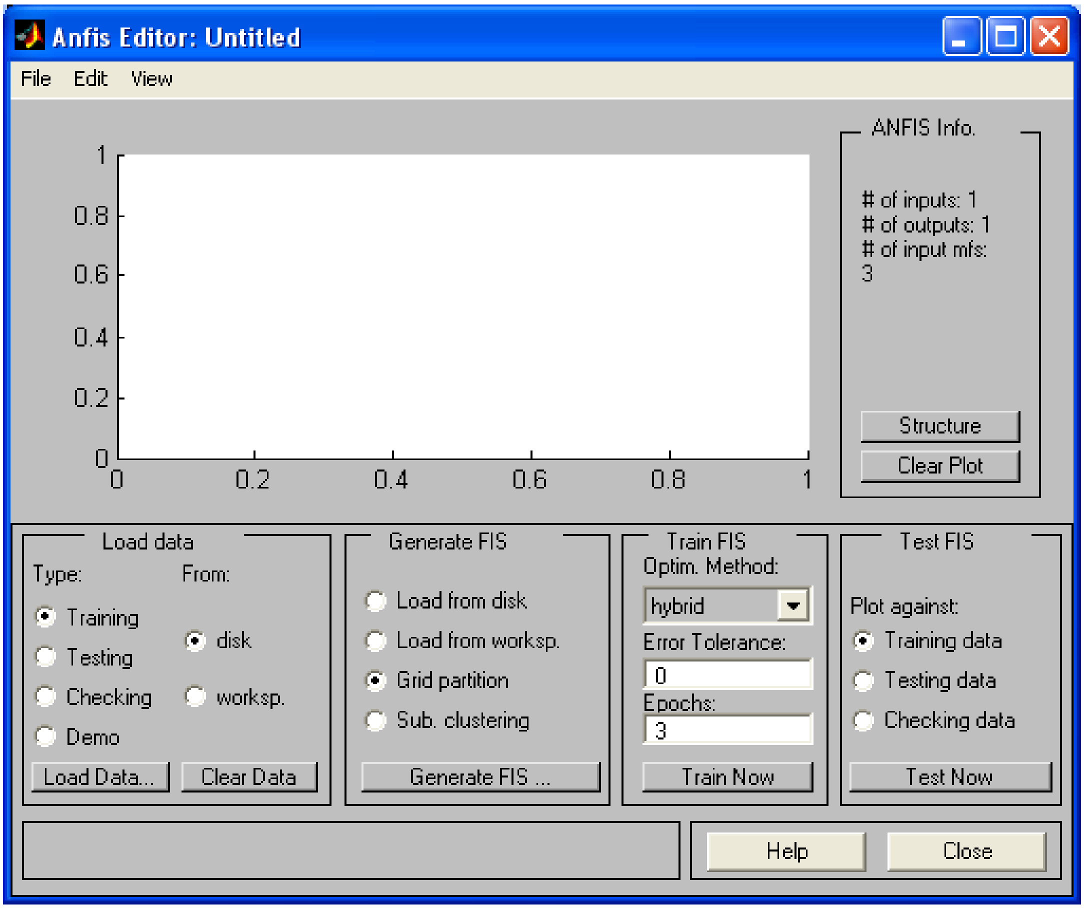
Рис. n8. График полученной нечеткой функции (применена функция принадлежности «gauss2mf»).

Вывод. Ознакомились с использованием пакета Fuzzy Logic Toolbox, реализованным в прикладном программном пакете Matlab и приобрели навыки по построению нечетких систем управления.

ЛАБОРАТОРНАЯ РАБОТА № 5
«ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ГИБРИДНЫХ СИСТЕМ В СИСТЕМЕ MATLAB»

Цель работы. Целью данной практической работы является знакомство с гибридными системами, реализованными в прикладном программном пакете Matlab и приобретение навыков по построению гибридных систем.

Графический интерфейс гибридных (нечетких) нейронных систем вызывается функцией (из режима командной строки) **anfisedit.** Исполнение функции приводит к появлению окна редактора гибридных систем (ANFIS Editor, ANFIS-редактор), вид которого приведен на рис. 1.



# of inputs: 1
# of outputs: 1
# of input mfs:
3

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 3

Рис. 16. Окно редактора гибридных систем

С помощью данного редактора осуществляется создание или загрузка структуры гибридной системы, просмотр структуры, настройка ее параметров, проверка качества функционирования такой системы. Создание структуры и настройка параметров и проверка осуществляются по выборкам (наборам данных) — обучающей (Training data), проверочной (Checking data) и тестирующей (Testing data), которые предварительно должны быть представлены в виде текстовых фалов (с расширением .dat и разделителями-табуляциями), первые колонки которых соответствуют входным переменным, а последняя (левая) — единственной выходной переменной; количество строк в таких файлах равно количеству образцов (примеров). Так, обучающая выборка, сформированная по таблице, представляется в виде

```
-1      1
-0.6   0.36
0.0     0.00
0.4    0.16
1      1
```

Строгих рекомендаций по объемам указанных выборок не существует, по-видимому, лучше всего исходить из принципа «чем больше, тем лучше». Обучающая и проверочная выборки непосредственно задействуются в процессе настройки параметров гибридной сети (проверочная — для выяснения ситуации: нет ли так называемого переобучения сети, при котором ошибка для обучающей последовательности стремится к нулю, а для проверочной — возрастает; впрочем, наличие проверочной выборки не является строго необходимым, оно лишь крайне желательно). Тестовая (или тестирующая выборка) применяется для проверки качества функционирования настроенной (обученной) сети.

Поясним пункты меню и опции редактора.

Пункты меню File и View, в общем идентичны аналогичным пунктам FIS-редактора, за тем исключением, что здесь работа может происходить только с алгоритмом нечеткого вывода Sugeno. Пункт меню Edit содержит единственный подпункт — Undo (Отменить выполненное действие).

Набор опций Load data (Загрузить данные) в нижней левой части окна редактора включает в себя:

- тип (Type) загружаемых данных (для обучения — Training, для тестирования — Testing, для проверки — Checking, демонстрационные — Demo);
- место, откуда должны загружаться данные (с диска — diskили из рабочей области MATLAB-workspace).

К данным опциям относятся две кнопки, нажатие на которых приводит к требуемым действиям — Load Data... (Загрузить данные) и Clear Data (очистить, т.е. стереть введенные данные).

Следующая группа опций (в середине нижней части окна ANFIS-редактора) объединена под именем **Generate FIS** (Создание нечеткой системы вывода). Данная группа включает в себя опции:

- загрузку структуры системы с диска (Load from disk);
- загрузку структуры системы из рабочей области MATLAB (Load from

worksp.);

- разбиение (деление) областей определения входных переменных (аргументов) на подобласти — независимо для каждого аргумента (Grid partition);
- разбиение всей области определения аргументов (входных переменных) на подобласти — в комплексе для всех аргументов (Subtract clustering или Sub. clustering), а также кнопку Generate FIS, нажатие которой приводит к процессу создания гибридной системы с точностью до ряда параметров.

Следующая группа опций — **Train FIS** (Обучение нечеткой системы вывода) — позволяет определить метод «обучения» (Optim. Method) системы (т.е. метод настройки ее параметров) — гибридный (hybrid) или обратного распространения ошибки (back-prора), установить уровень текущей суммарной (по всем образцам) ошибки обучения (Error Tolerance), при достижении которого процесс обучения заканчивается и количество циклов обучения (**Epochs**), т.е. количество «прогонов» всех образцов (или примеров) обучающей выборки; процесс обучения, таким образом заканчивается либо при достижении отмеченного уровня ошибки обучения, либо при проведении заданного количество циклов.

Кнопка **Train Now** (Начать обучение) процесс обучения, т.е. процесс настройки параметров гибридной сети.

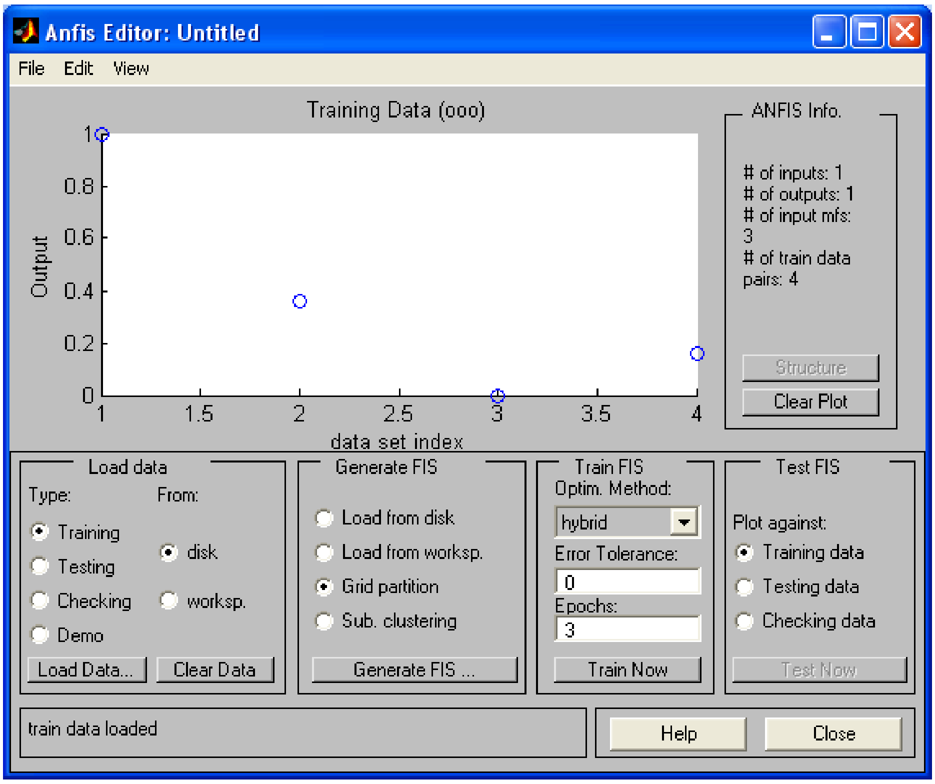
В правом верхнем углу окна ANFIS-редактора выдается информация (ANFIS Info.) о проектируемой системе: о количестве входов, выходов, функций принадлежности входов; нажатие кнопки Structure (Структура) позволяет увидеть структуру сети. Кнопка Clear (Очистить) позволяет стереть все результаты.

Опции **Test FIS** в правом нижнем углу окна позволяют провести проверку и тестирование созданной и обученной системы с выводом результатов в виде графиков (соответствующие графики для обучающей выборки — Training data, тестирующей выборки — Testing data и проверочной выборки — Checking data. Кнопка Test Now позволяет запустить указанные процессы.

Работу с редактором рассмотрим на примере восстановления зависимости $y = x^2$ по данным таблицы. Предположим, что эти данные сохранены в файле Proba.dat. Создание и проверку системы, как и раньше, проведем по этапам.

1. В окне ANFIS-редактора выберем тип загружаемых данных Training и нажмем кнопку Load data. В последующем стандартном окне диалога укажем местоположение и имя файла. Его открытие приводит к появлению в

графической части окна редактора набора точек, соответствующих введенным данным (рис. 17).



\# of inputs: 1
\# of outputs: 1
\# of input mfs:
3
\# of train data
pairs: 4

Type:
● Training
○ Testing
○ Checking
○ Demo

From:
● disk
○ worksp.

○ Load from disk
○ Load from worksp.
● Grid partition
○ Sub. clustering

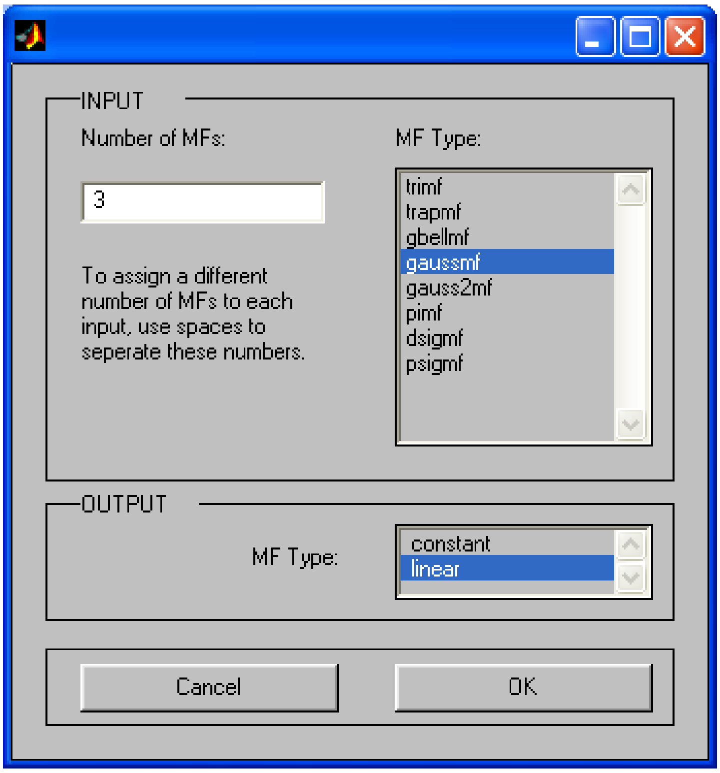
Optim. Method:
hybrid
Error Tolerance:
0
Epochs:
3

Plot against:
● Training data
○ Testing data
○ Checking data

train data loaded

Рис.17 – Окно редактора после загрузки обучающей выборки

2. В группе опций Generate FIS по умолчанию активизирована опция Grid partition. Не будем ее изменять и нажмем кнопкуGenerate FIS, после чего появится диалоговое окно (рис. 18) для задания числа и типов функций принадлежности.



INPUT

Number of MFs: 3

To assign a different number of MFs to each input, use spaces to seperate these numbers.

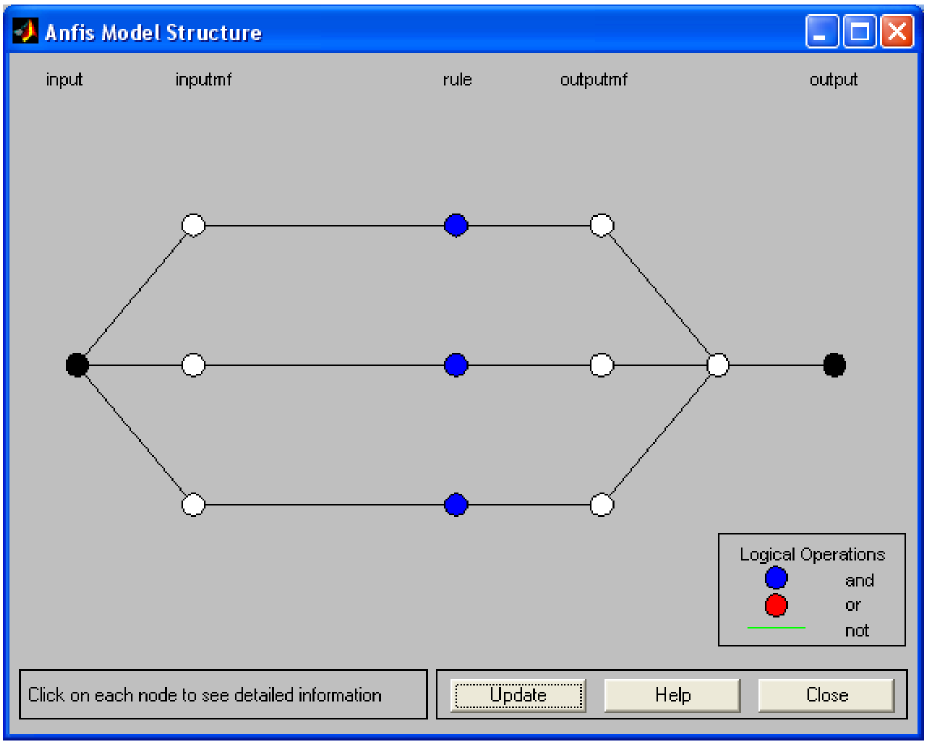
MF Type: gaussmf

OUTPUT

MF Type: linear

Рис.18 – Окно задания функций принадлежности

Сохраним все установки по умолчанию, согласившись с ними нажатием кнопки OK. Произойдет возврат в основное окно ANFIS-редактора. Теперь структура гибридной сети создана, и ее графический вид можно просмотреть с помощью кнопки Structure (рис. 19).

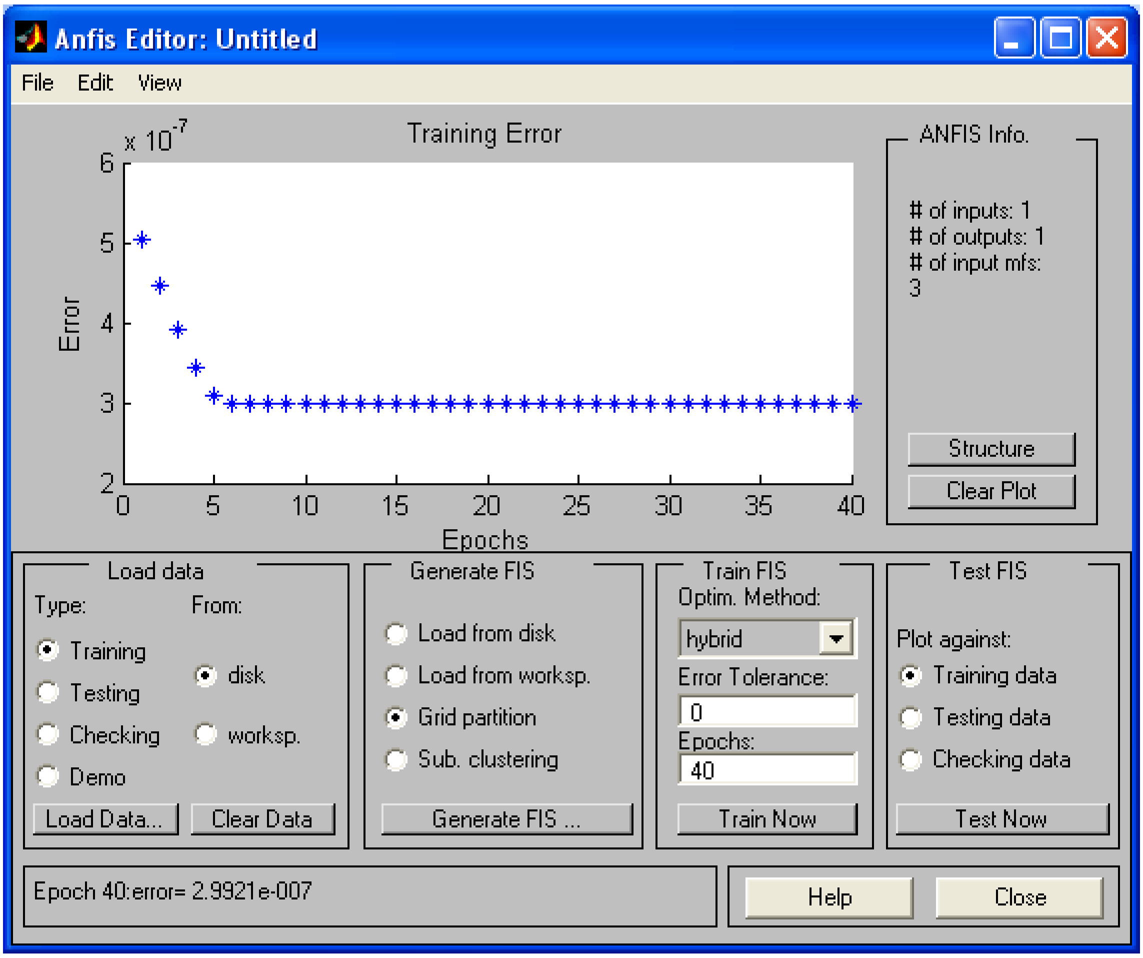


Anfis Model Structure

Click on each node to see detailed information

Рис.19 – Структура созданной гибридной сети

3. Перейдем к опциям Train FIS. Не будем менять задаваемые по умолчанию метод настройки параметров (hybrid — гибридный)и уровень ошибки (0), но количество циклов обучения изменим на 40, после чего нажмем кнопку начала процесса обучения (TrainNow). Получившийся результат в виде графика ошибки сети в зависимости от числа проведенных циклов обучения (из которого следует, что фактически обучение закончилось после пятого цикла)представлен на рис. 20.



\# of inputs: 1
\# of outputs: 1
\# of input mfs:
3

Type: ● Training ○ Testing ○ Checking ○ Demo
From: ● disk ○ worksp.

○ Load from disk
○ Load from worksp.
● Grid partition
○ Sub. clustering

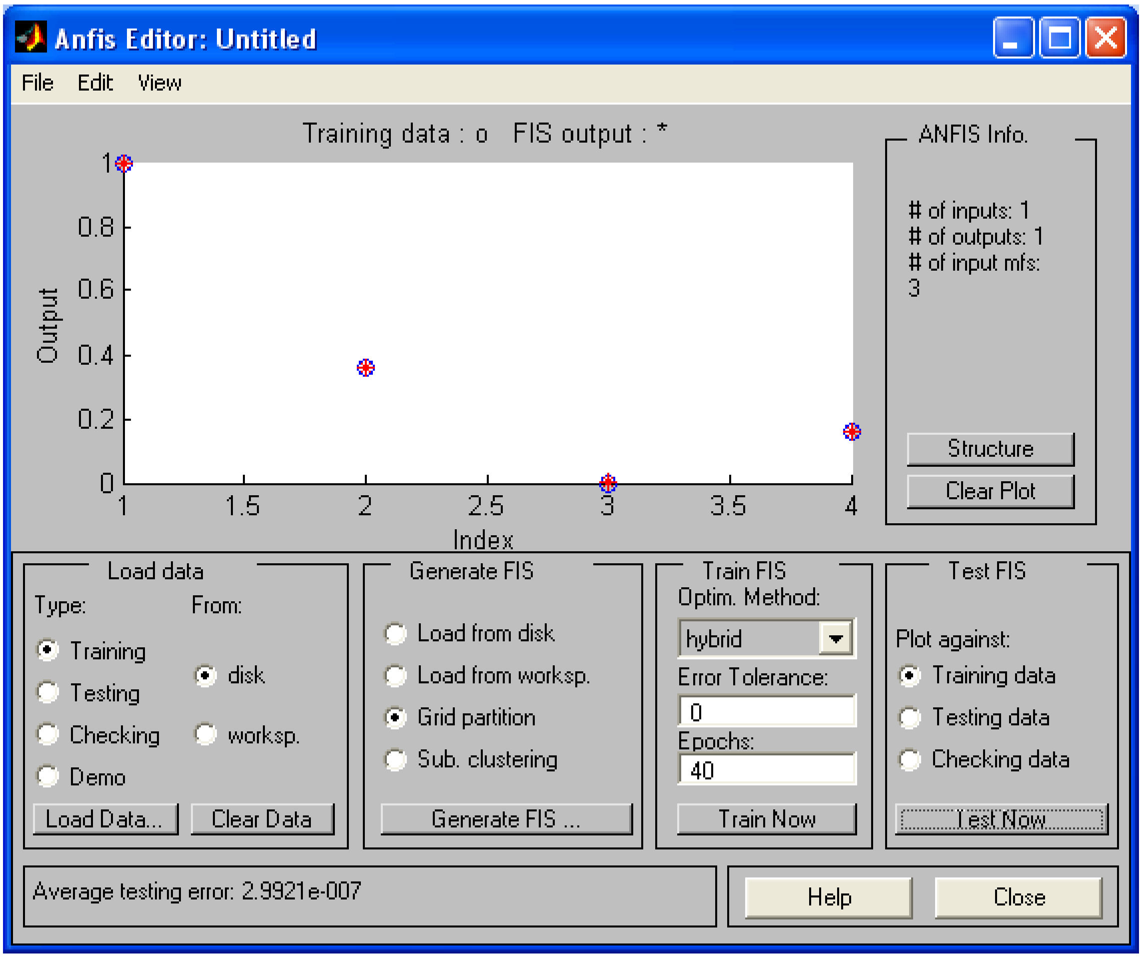
Optim. Method: hybrid
Error Tolerance: 0
Epochs: 40

Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 40:error= 2.9921e-007

Рис. 20 – Функция ошибки обучения сети

4. Теперь нажатием кнопки Test Now можно начать процесс тестирования обученной сети, но, поскольку использовалась только одна обучающая выборка, ничего особенно интересного ожидать не приходится. Действительно, выход обученной системы практически совпадает с точками обучающей выборки (рис. 21).



\# of inputs: 1
\# of outputs: 1
\# of input mfs:
3

Type:
● Training
○ Testing
○ Checking
○ Demo

From:
● disk
○ worksp.

○ Load from disk
○ Load from worksp.
● Grid partition
○ Sub. clustering

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 40

Plot against:
● Training data
○ Testing data
○ Checking data

```
Average testing error: 2.9921e-007
```

Рис. 21. Результат тестирования обученной системы

5. Сохраним разработанную систему в файл на диске с именем Probal (с расширением .Us) и для исследования разработанной системы средствами FIS-редактора из командной строки MATLABвыполним команду **fuzzy,** а затем через пункты меню File/OpenFIS from disk... откроем созданный файл. С созданной системой можно теперь выполнять все приемы редактирования (изменение имен переменных и т. п.) и исследования, которые были рассмотрены выше. Здесь нетрудно, кстати, убедиться, что качество аппроксимации данных существенно не улучшилось — слишком мало данных.

Что можно сказать про эффективность использования гибридных систем (и ANFIS-редактора)?

В данном случае используется только один алгоритм нечеткого вывода - Sugeno (нулевого или первого порядков), может быть только одна выходная переменная, всем правилам приписывается один и тот же единичный вес. Вообще говоря, возникают значительные проблемы при большом (более 5-6) количестве входных переменных. Это - ограничения и недостатки подхода.

Его несомненные достоинства: практически полная автоматизация процесса создания нечеткой (гибридной) системы, возможность просмотра сформированных правил и придания им содержательной (лингвистической) интерпретации, что позволяет, кстати говоря, рассматривать аппарат гибридных сетей как средство извлечения знаний из баз данных и существенно отличает данные сети от классических нейронных.

Рекомендуемая область применения: построение аппроксиматоров зависимостей по экспериментальным данным, построение систем классификации (в случае бинарной или дискретной выходной переменной), изучение механизма явлений.

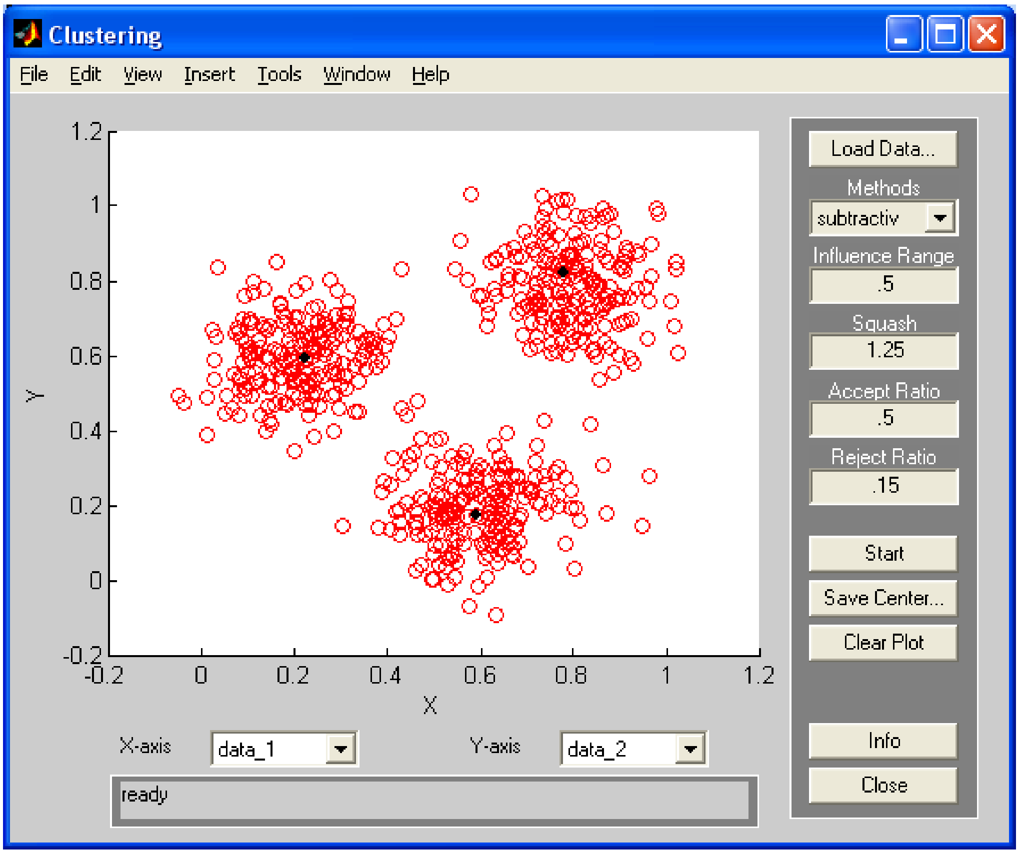
Графический интерфейс программы кластеризации

В пакет Fuzzy Logic Toolbox входит еще одна программа, позволяющая работу в режиме графического интерфейса, - программа Clustering (Кластеризация) выявления центров кластеров, т.е. точек в многомерном пространстве данных, около которых группируются (скапливаются) экспериментальные данные. Выявление подобных центров, надо сказать, является значимым этапом при предварительной обработке данных, поскольку позволяет сопоставить с этими центрами функции принадлежности переменных при последующем проектировании системы нечеткого вывода.

Запуск программы Clustering осуществляется командой (функцией) **findcluster.** В появляющемся окне программы имеется (вверху) главное меню, содержащее достаточно стандартный набор пунктов (File, Edit, Window, Help) и набор управляющих кнопок и опций (справа). К этим кнопкам относятся:

- кнопка загрузки фала данных Load Data,
- кнопка выбора алгоритма кластеризации — Method,
- четыре расположенные ниже кнопки опций алгоритма (их названия меняются в зависимости от выбранного алгоритма),
- кнопка начала итеративного процесса нахождения центров кластеров (кластеризации) - Start,
- кнопка сохранения результатов кластеризации (SaveCenter),
- кнопка очистки (стирания) графиков (Clear Plot),
- кнопка справочной информации (Info),
- кнопка завершения работы с программой (Close).

В программе используются два алгоритма выявления центров кластеров: Fuzzy c-means (который можно перевести как «Алгоритм нечетких центров») и Subtractive clustering («Вычитающаякластеризация»).



Methods subtractiv
Influence Range .5
Squash 1.25
Accept Ratio .5
Reject Ratio .15
X-axis data_1
Y-axis data_2
ready

Рис. 22. Результат работы программы Clustering (центры кластеров окрашены в черный цвет)

Если не вдаваться в их детальное теоретическое изложение, а ограничиться выявлением различий на уровне пользователя, то можно отметить, что алгоритм Fuzzy c-means, являясь, пожалуй, более точным (если понятие точности вообще здесь применимо), для своей работы требует задания таких опций, как число кластеров (кнопка Cluster num.) и число итераций (кнопка Max Iteration^). Ну, если число итераций еще можно задать как-то наугад, то ошибка в задании числа кластеров может привести к неприятным последствиям. Алгоритм Subtrac-tive clustering менее точен, но и менее требователен к априорной информации; при работе с ним можно сохранить опции, заданные в программе по умолчанию. На рис. 22 приведен пример использования программы для фала данных clusterdemo.dat из директории Matlab/toolbox/fuzzy/fuzdemos/ при использовании алгоритма Subtractive clustering. Заметим, что выводится только двумерное поле рассеяния, но изменяя переменные в соответствующих полях (X-axis и Y-axis), можно «просмотреть» все многомерное пространство переменных.

Задание к лабораторной работе №5

Аппроксимировать при помощи гибридной сети уравнения вида y=f(x) по десяти точкам.

ТАБЛИЦА ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ

№	f(x)
1	$\ln(x)-1/(1+x^2)$
2	$\ln(\ln(x))-x^2$
3	$x-1/\sqrt{e^x}$
4	$x^4-13x^2+36-(1/x)$
5	$2x^2-x^4-1-\ln(x)$
6	$x^3-3x-2e^{-x}$
7	$\sin(x^2)-6x+1$
8	$\cos(x^2)-10x$
9	$\ln^2 x-(1/x)$
10	$\ln(1+x)/(1-x)-\cos(x^2)$
11	$x-1/(x^4-13x^2+36)$
12	$e^x-3-\cos(x)$

Отчёт по лабораторной работе включает в себя:

- титульный лист;
- листы выполнения работы (цель работы, практическая часть, вывод, список литературы).

Практическая часть сопровождается таблицами, графиками с кратким пояснением хода выполнения лабораторной работы. Обязательно приводится описание параметров модели, можно распечатать экранные формы с пояснениями.

Работа считается зачтенной в случае предоставления отчёта и ответа на вопросы по работе.

Вопросы по темам:

- нечеткая логика;
- нечеткое управление;
- нечеткая истинность;
- нейронные сети.

Пример выполнения лабораторной работы №5

Лабораторная работа №5

Графический интерфейс гибридных систем в системе Matlab

Цель работы. Знакомство с гибридными системами, реализованными в прикладном программном пакете Matlab и приобретение навыков по построению гибридных систем.

Задание на лабораторную работу

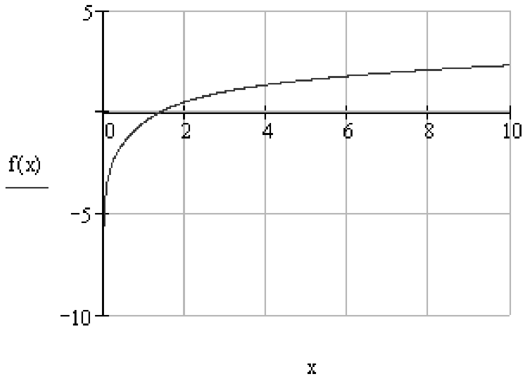
Аппроксимировать при помощи гибридной сети уравнение y= $\ln(x)-1/(1+x^2)$ по десяти точкам.

Ход лабораторной работы

В среде MathCAD сформируем текстовые файлы с разделителями-табуляциями с входными данными для обучения (dat5.dat) и тестирования (dat5T.dat) гибридной сети.

$$f(x) := \ln(x) - \left(\frac{1}{1+x^2}\right)$$

$$S := \begin{pmatrix} 0.01 & f(0.01) \\ 1 & f(1) \\ 2 & f(2) \\ 3 & f(3) \\ 4 & f(4) \\ 5 & f(5) \\ 6 & f(6) \\ 7 & f(7) \\ 8 & f(8) \\ 9 & f(9) \end{pmatrix}$$



$$T := \begin{pmatrix} 0.1 & f(0.1) \\ 1.5 & f(1.5) \\ 2.5 & f(2.5) \\ 3.5 & f(3.5) \\ 4.5 & f(4.5) \end{pmatrix}$$

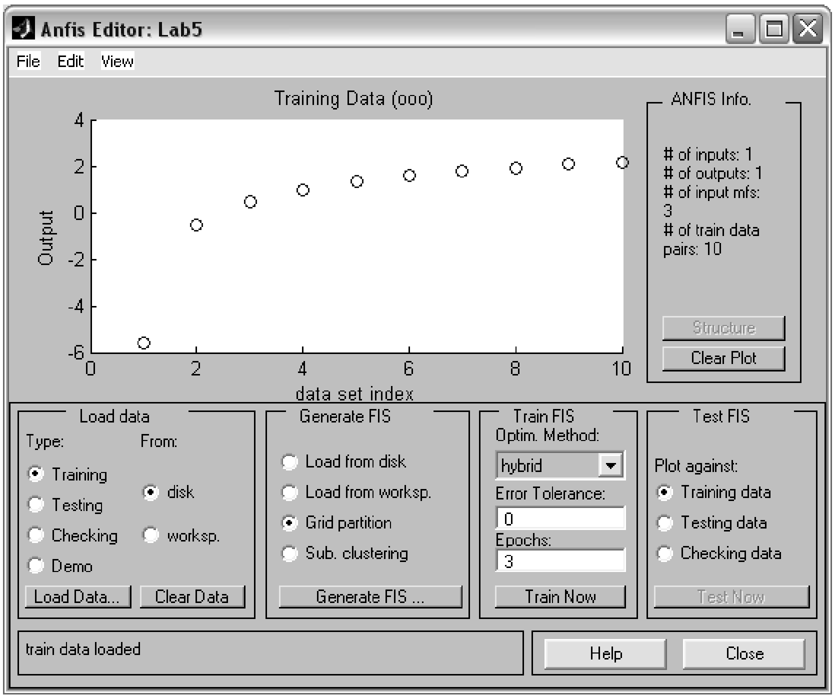
C:\..\dat5.dat

C:\..\dat5T.dat

S

T

В окне ANFIS-редактора пакета Matlab выберем тип загружаемых данных Training и нажмем кнопку Load data. В последующем стандартном окне диалога укажем местоположение файла dat5.dat. Его открытие приводит к появлению в графической части окна редактора набора точек, соответствующих введенным данным (рис. n1).



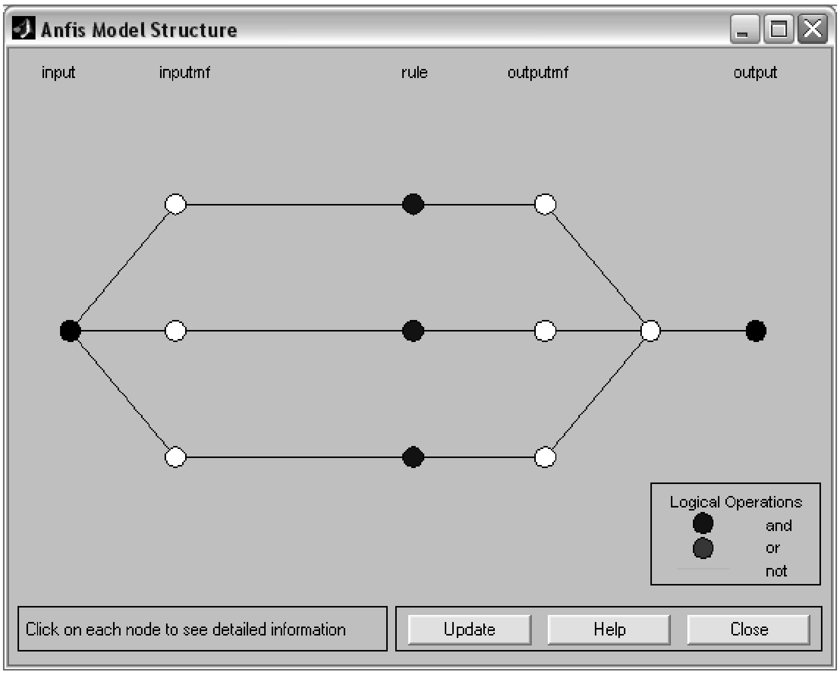
\# of inputs: 1
\# of outputs: 1
\# of input mfs:
3
\# of train data
pairs: 10

Type: ● Training ○ Testing ○ Checking ○ Demo
From: ● disk ○ worksp.
○ Load from disk
○ Load from worksp.
● Grid partition
○ Sub. clustering
Optim. Method: hybrid
Error Tolerance: 0
Epochs: 3
Plot against: ● Training data ○ Testing data ○ Checking data

train data loaded

Рис. n1. Окно ANFIS-редактора после загрузки обучающей выборки.

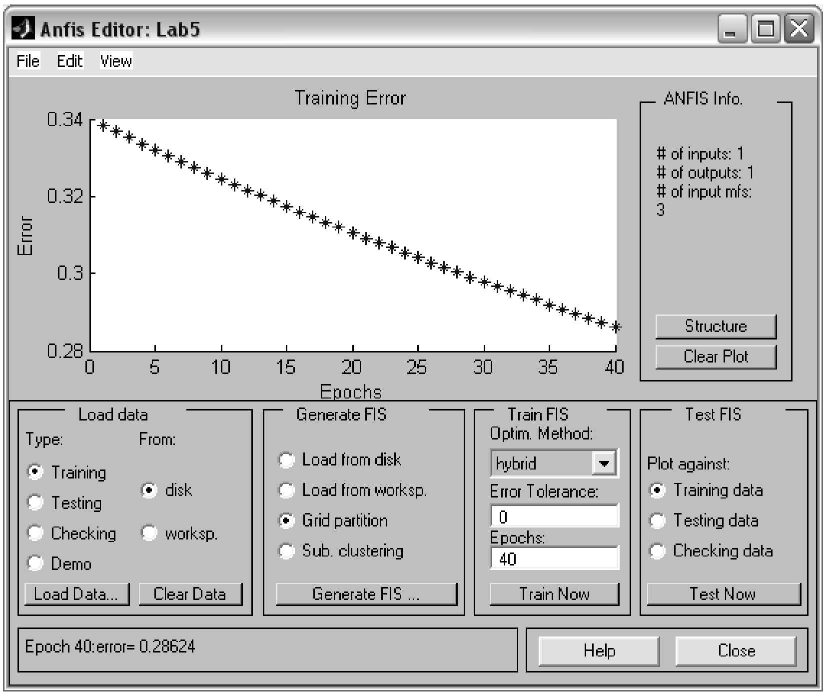
После нажатия в окне ANFIS-редактора на кнопку «Generate FIS» появится диалоговое окно для задания числа и типов функций принадлежности. В появившемся окне нажмем кнопку «ОК». Далее произойдет возврат в основное окно ANFIS-редактора. Теперь структура гибридной сети создана, и ее графический вид можно просмотреть с помощью кнопки «Structure» (рис. n2).



Click on each node to see detailed information

Рис. n2. Структура созданной гибридной сети.

В окне ANFIS-редактора зададим метод настройки параметров - hybrid (гибридный), уровень ошибки - 0, количество циклов обучения - 40, после чего нажмем кнопку начала процесса обучения «Train Now». Получившийся результат в виде графика ошибки сети в зависимости от числа проведенных циклов обучения представлен на рис. 3.



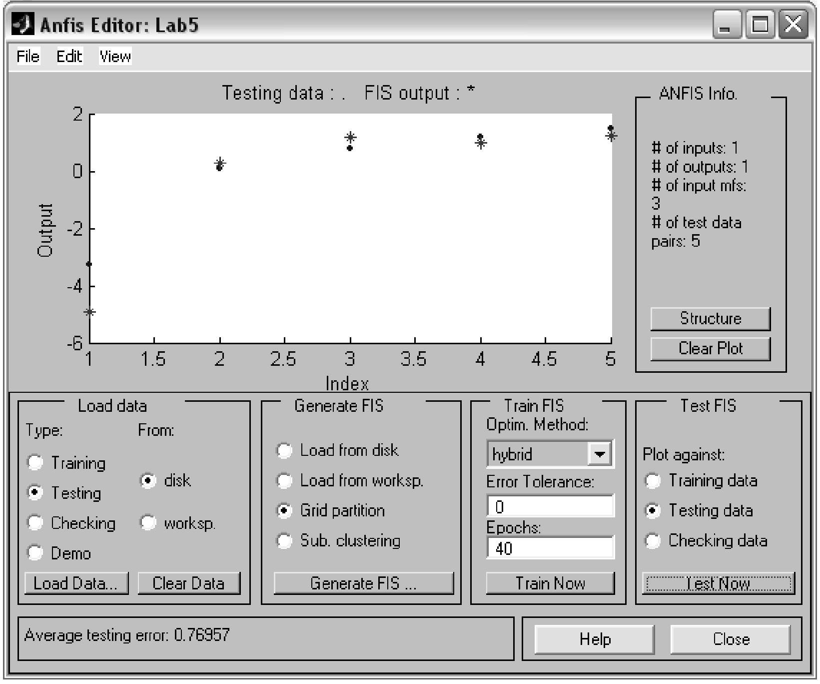
\# of inputs: 1
\# of outputs: 1
\# of input mfs:
3

Type: ● Training
From: ● disk
● Grid partition
Optim. Method: hybrid
Error Tolerance: 0
Epochs: 40
Plot against: ● Training data

Epoch 40:error= 0.28624

Рис. n3. График ошибки сети (зависимость от числа проведенных циклов обучения).

Нажатием кнопки «Test Now» начнем процесс тестирования обученной сети. Результаты тестирования представлены на рис. n4. Полученную гибридную сеть можно проанализировать в FIS-редакторе (команда fuzzy пакета Matlab) (рис. n5-n7).

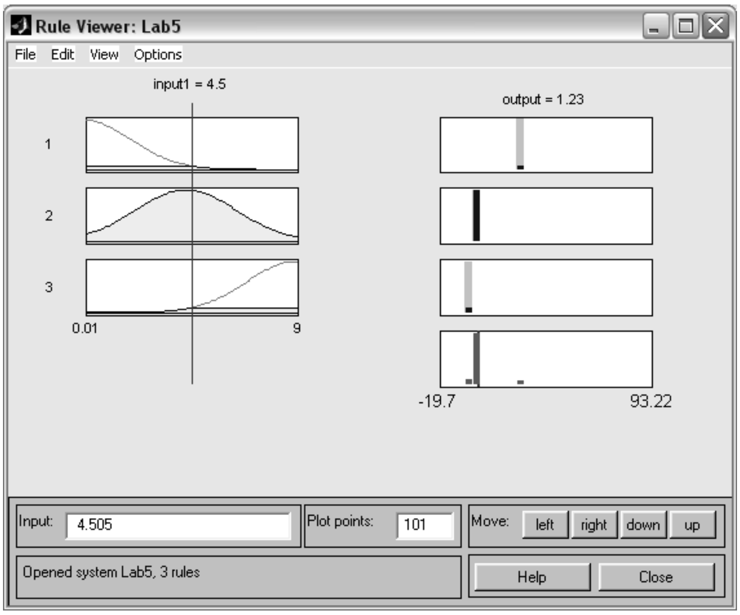


\# of inputs: 1
\# of outputs: 1
\# of input mfs:
3
\# of test data
pairs: 5

Type: ● Testing
From: ● disk
● Grid partition
Optim. Method: hybrid
Error Tolerance: 0
Epochs: 40
Plot against: ● Testing data

Average testing error: 0.76957

Рис. n4. Результаты тестирования обученной сети.



input1 = 4.5

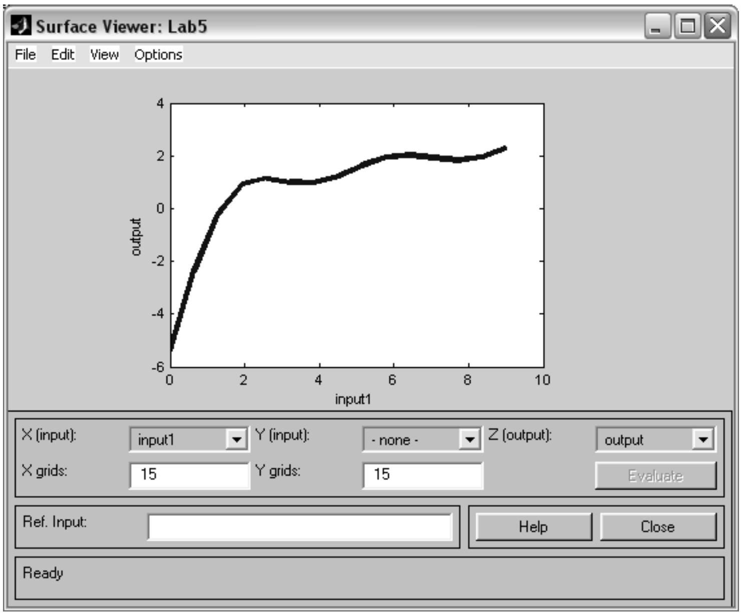
output = 1.23

Input: 4.505

Plot points: 101

Opened system Lab5, 3 rules

Рис. n5. Окно просмотра правил гибридной системы.



X (input): input1

Y (input): - none -

Z (output): output

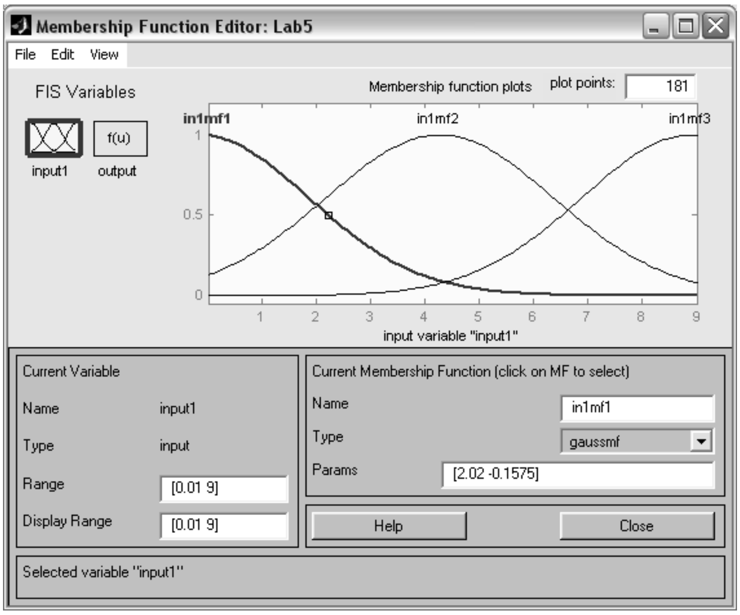
X grids: 15

Y grids: 15

Ref. Input:

Ready

Рис. n6. График полученной функции системы.



plot points: 181

Current Variable

Name input1

Type input

Range [0.01 9]

Display Range [0.01 9]

Current Membership Function (click on MF to select)

Name in1mf1

Type gaussmf

Params [2.02 -0.1575]

Selected variable "input1"

Рис. n7. Вид графиков функций принадлежности системы.

Вывод. Познакомились с гибридными системами, реализованными в прикладном программном пакете Matlab, и приобрели навыки по построению гибридных систем.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Акофф Р., Сасиени М. Основы исследования операций. – М.: Мир, 1971. – 534 с.

2. Бурда, А.Г. Исследование операций в экономике: Учебное пособие / А.Г. Бурда, Г.П. Бурда. - СПб.: Лань, 2018. - 564 с.

3. Вентцель, Е.С. Исследование операций: задачи, принципы, методология: Учебное пособие / Е.С. Вентцель. - М.: КноРус, 2013. - 192 с.

4. Горелик, В.А. Исследование операций и методы оптимизации: Учебник / В.А. Горелик. - М.: Academia, 2018. - 384 с.

5. Горлач Б.А., Додонова Н.Л. Исследование операций. Практикум для студентов технических и экономических специальностей вузов. – СПб: Лань, 2021. – 200 с.

6. Катаргин, Н.В. Сетевые модели в задачах экономики. – СПб: Лань, 2020. – 172 с.

7. Костевич, Л.С. Исследование операций Теория игр: Учебное пособие / Л.С. Костевич. - Минск: Вышэйшая школа, 2008. - 368 с.

8. Путко, Б.А. Исследование операций в экономике: Учебное пособие для вузов / Н.Ш. Кремер, Б.А. Путко, И.М. Тришин; Под ред. проф. Н.Ш. Кремер.. - М.: Юрайт, ИД Юрайт, 2013. - 438 с. /

9. Таха, Х. Исследование операций / Х. Таха. - М.: Вильямс И.Д., 2019. - 1056 с.

10. Черчмен У., Акоф Р., Арноф Л. Введение в исследование операций. – М.: Наука, Главная редакция физико-математической литературы, 1967. – 488 с.

11. Ширяев, В.И. Исследование операций и численные методы оптимизации / В.И. Ширяев. - М.: Ленанд, 2017. - 224 с.

ИНТЕРНЕТ-РЕСУРСЫ:

11. http://systems-analysis.ru/ - справочно-информационный сайт «Системный анализ»

12. https://www.mathnet.ru/ - общероссийский портал поиска научной информации по математике, физике, информационным технологиям и смежным наукам

13. https://openedu.ru/course/mephi/ECMAMO - онлайн-курс «Математические моделирование и методы в экономике»

14. https://openedu.ru/course/spbstu/BUSMAT/- онлайн-курс «Математические методы в экономике»

15. https://openedu.ru/course/spbu/MATTHEGAM/ - онлайн-курс «Математическая теория игр»