

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

СИСТЕМЫ АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

*Методические рекомендации к лабораторным работам
для студентов специальности 7-06-0612-03 «Системы управления
информацией» очной и заочной форм обучения*

Часть 1



Могилев 2024

УДК 004.43
ББК 32.973-018
С40

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«16» января 2024 г., протокол № 7

Составитель д-р техн. наук, доц. А. И. Якимов

Рецензент канд. техн. наук, доц. И. В. Лесковец

Даны методические указания к выполнению лабораторных работ по дисциплине «Системы аналитического программирования», а также приведены контрольные вопросы и список литературы для подготовки.

Учебное издание

СИСТЕМЫ АНАЛИТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Часть 1

Ответственный за выпуск	А. И. Якимов
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2024

Содержание

Введение.....	4
1 Лабораторная работа № 1. Основы языка Python	5
2 Лабораторная работа № 2. Линейные модели для классификации и регрессии	10
3 Лабораторная работа № 3. Модели временных рядов	14
4 Лабораторная работа № 4. Сингулярный анализ.....	17
5 Лабораторная работа № 5. Кластерный анализ.....	20
6 Лабораторная работа № 6. Параллельные вычисления на OpenMPI.....	23
Список литературы	29

Введение

Целью преподавания дисциплины «Системы аналитического программирования» является приобретение студентами теоретических знаний и практических навыков в области основ теории обучения машин, современных методов восстановления зависимостей по эмпирическим данным, включая дискриминантный, кластерный и регрессионный анализ, овладение навыками практического решения задач интеллектуального анализа данных.

Студент, изучивший дисциплину, будет **знать**:

– фундаментальные понятия, современные подходы, методы и проблемы машинного обучения и интеллектуального анализа данных.

Студент, изучивший дисциплину, будет **уметь**:

- понимать и формализовать поставленную задачу анализа данных;
- использовать современные методы машинного обучения для практического решения задач анализа данных;
- при необходимости, продиктованной особенностями поставленной задачи, создавать новые методы машинного обучения;
- проводить численные эксперименты на модельных и реальных данных и интерпретировать их результаты;
- представлять результаты исследований в устной и письменной форме.

Студент, изучивший дисциплину, будет **владеть**:

- навыками освоения большого объема информации и решения сложных теоретических и практических задач анализа данных;
- навыками самостоятельной работы и освоения новых дисциплин;
- культурой постановки, анализа и решения математических и прикладных задач, требующих для своего решения использования математических подходов и методов;
- предметным языком машинного обучения и интеллектуального анализа данных, навыками описания решения задач и представления полученных результатов.

1 Лабораторная работа № 1. Основы языка Python

Цель работы: овладеть на практике основами современного языка программирования Python версии выше 3.0.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить вариант задания.

3 Сделать выводы по результатам работы.

4 Оформить отчет.

Требования к отчету

1 Цель работы.

2 Постановка задачи.

3 Результаты разработки программного обеспечения.

4 Выводы.

Методические указания

Python не требует явного объявления переменных, является регистрозависимым (переменная `var` не эквивалентна переменной `Var` или `VAR` – это три разные переменные) объектно-ориентированным языком.

Синтаксис.

Python не содержит операторных скобок (`begin..end` в Pascal или `{..}` в Си), вместо этого блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием. Однострочные комментарии начинаются со знака фунта «`#`», многострочные начинаются и заканчиваются тремя двойными кавычками «`"""`». Чтобы присвоить значение переменной, используется знак «`=`», а для сравнения – «`==`». Для увеличения значения переменной или добавления к строке используется оператор «`+=`», а для уменьшения – «`-=`». Все эти операции могут взаимодействовать с большинством типов, в том числе со строками. Например:

```
>>> myvar = 3
>>> myvar += 2
>>> myvar -= 1
"""«Это многострочный комментарий
Строки, заключенные в три двойные кавычки, игнорируются»"""
>>> mystring = «Hello»
>>> mystring += " world."
>>> print mystring
Hello world.
```

```
# Следующая строка меняет
значения переменных местами. (Всего одна строка!)
>>> myvar, mystring = mystring, myvar
```

Структуры данных.

Python содержит такие структуры данных, как списки (lists), кортежи (tuples) и словари (dictionaries). Списки похожи на одномерные массивы (но можно использовать Список, включающий списки, – многомерный массив), кортежи – неизменяемые списки, словари – тоже списки, но индексы могут быть любого типа, а не только числовыми. «Массивы» в Python могут содержать данные любого типа, т. е. в одном массиве могут находиться числовые, строковые и другие типы данных. Массивы начинаются с индекса 0, а последний элемент можно получить по индексу –1. Можно присваивать переменным функции и применять их соответственно.

Можно использовать часть массива, задавая первый и последний индексы через двоеточие «:». В таком случае получают часть массива, от первого индекса до второго не включительно. Если не указан первый элемент, то отсчет начинается с начала массива, а если не указан последний, то массив считывается до последнего элемента. Отрицательные значения определяют положение элемента с конца. Например:

```
>>> mylist = [«List item 1», 2, 3.14]
>>> print mylist[:] #Считываются все элементы массива
['List item 1', 2, 3.1400000000000001]
>>> print mylist[0:2] #Считываются нулевой и первый элементы массива.
['List item 1', 2]
>>> print mylist[-3:-1] #Считываются элементы от нулевого (-3) до второго (-1)
(не включительно)
['List item 1', 2]
>>> print mylist[1:] #Считываются элементы от первого до последнего
[2, 3.14]
```

Строки.

Строки в Python обособляются кавычками двойными «"» или одинарными «'». Внутри двойных кавычек могут присутствовать одинарные или наоборот. Например, строка «Он сказал 'Привет!'» будет выведена на экран как «Он сказал 'Привет!'». Если нужно использовать строку из несколько строк, то эту строку надо начинать и заканчивать тремя двойными кавычками «"""». Можно подставить в шаблон строки-элементы из кортежа или словаря. Знак процента «%» между строкой и кортежем заменяет в строке символы «%s» на элемент кортежа. Словари позволяют вставлять в строку элемент под заданным индексом. Для этого надо использовать в строке конструкцию «%(индекс)s». В этом случае вместо «%(индекс)s» будет подставлено значение словаря под заданным индексом.

```
>>> print «Name: %s\nNumber: %s\nString: %s» % (myclass.name, 3, 3 * "-")
Name: Poromenos
Number: 3
String: —
strString = ""«Этот текст расположен
на нескольких строках»""
```

```
>>> print «This %(verb)s a %(noun)s.» % {«noun»: «test», «verb»: «is»}
This is a test.
```

Операторы.

Операторы *while*, *if*, *for* составляют операторы перемещения. Здесь нет аналога оператора *select*, так что придется обходиться без *if*. В операторе *for* происходит сравнение *переменной* и *списка*. Чтобы получить список цифр до числа *<number>*, используют функцию *range(<number>)*.

Функции.

Для объявления функции служит ключевое слово *def*. Аргументы функции задаются в скобках после названия функции. Можно задавать необязательные аргументы, присваивая им значение по умолчанию. Функции могут возвращать кортежи, в таком случае надо писать возвращаемые значения через запятую. Ключевое слово *lambda* служит для объявления элементарных функций.

Классы.

Язык Python ограничен в множественном наследовании в классах. Внутренние переменные и внутренние методы классов начинаются с двух знаков нижнего подчеркивания «*__*» (например, «*__myprivatevar*»). Можно также присвоить значение переменной класса извне.

Практическое задание

1 На основании предложенной предметной области спроектировать 3–4 класса, используя механизм наследования. Для каждого класса использовать отдельный модуль.

2 Предусмотреть у класса наличие полей, методов и свойств. Названия членов класса должны быть осмысленны и снабжены комментариями.

3 Один из наследников должен перегружать метод родителя.

4 Один из классов должен содержать виртуальный метод, который переопределяется в одном наследнике и не переопределяется в другом.

5 Продемонстрировать работу всех объявленных методов.

6 Продемонстрировать вызов конструктора родительского класса при наследовании.

Варианты заданий к лабораторной работе № 1.

Вариант 1. Написать программу, в которой описана иерархия классов: средство передвижения (велосипед, автомобиль, грузовик). Базовый класс должен иметь поля для хранения средней скорости, названия модели, числа пассажиров, а также методы получения потребления топлива для данного расстояния

и вычисления времени движения на заданное расстояние. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа объекта для демонстрации.

Вариант 2. Написать программу, в которой описана иерархия классов: человек (дошкольник, школьник, студент, работающий). Базовый класс должен иметь поля для хранения ФИО, возраста, пола, а также методы получения среднего дохода и среднего расхода в денежном эквиваленте. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа объекта для демонстрации.

Вариант 3. Написать программу, в которой описана иерархия классов: геометрические фигуры (круг, прямоугольник, треугольник). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

Вариант 4. Написать программу, в которой описана иерархия классов: геометрические фигуры (эллипс, квадрат, трапеция). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

Вариант 5. Написать программу, в которой описана иерархия классов: геометрические фигуры (ромб, параллелепипед, эллипс). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

Вариант 6. Написать программу, в которой описана иерархия классов: геометрические фигуры (куб, цилиндр, тетраэдр). Реализовать методы вычисления объема и площади поверхности. Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

Вариант 7. Написать программу, в которой описана иерархия классов: геометрические фигуры (куб, конус, тетраэдр). Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

Вариант 8. Написать программу, в которой описана иерархия классов: геометрические фигуры (ромб, прямоугольник, эллипс). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.

Вариант 9. Написать программу, в которой описана иерархия классов: функция от одной переменной (синус, косинус, тангенс). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу методов всех классов.

Вариант 10. Написать программу, в которой описана иерархия классов: функция от одной переменной (секанс, косеканс, котангенс). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу методов всех классов.

Вариант 11. Написать программу, в которой описана иерархия классов: функция от одной переменной (арксинус, арккосинус, а также класс, необходимый для представления производных). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу методов всех классов.

Вариант 12. Написать программу, в которой описана иерархия классов: функция от одной переменной (арктангенс, арккотангенс, а также класс, необходимый для представления производных). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу методов всех классов.

Вариант 13. Написать программу, в которой описана иерархия классов: функция от одной переменной (логарифм, натуральный логарифм, а также класс, необходимый для представления производных). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу методов всех классов.

Вариант 14. Написать программу, в которой описана иерархия классов: функция от одной переменной (экспонента, гиперболический синус, гиперболический косинус). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу методов всех классов.

Вариант 15. Написать программу, в которой описана иерархия классов: электронная техника (смартфон, планшет, ноутбук). Базовый класс должен иметь поля для хранения емкости аккумулятора, объем оперативной памяти, объем флеш-памяти, производителя, а также методы получения времени разряда и ответа на вопрос: возможно ли сохранение нового файла определенного размера? Продемонстрировать работу методов всех классов.

Контрольные вопросы

- 1 Назовите главную особенность синтаксиса Python.
- 2 Перечислите, какие структуры данных содержит Python.
- 3 Как считать часть массива?
- 4 Как вставить в строку элемент кортежа?
- 5 Как задать внутренние переменные и внутренние методы классов?

2 Лабораторная работа № 2. Линейные модели для классификации и регрессии

Цель работы: ознакомиться с линейными регрессионными моделями и методами оценки параметров линейных моделей.

Порядок выполнения работы

- 1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
- 2 Получить задание у преподавателя, выполнить типовые задания.
- 3 Исследовать построение простой линейной регрессии.
- 4 Оформить отчет.

Требования к отчету

- 1 Цель работы.
- 2 Постановка задачи.
- 3 Результаты построения простой линейной регрессии.
- 4 Выводы.

Основные теоретические положения

Простая линейная регрессия позволяет найти линейную зависимость между одной входной и одной выходной переменными. Это регрессионная модель, отражающая зависимость значений Y , зависимой величины Y от значений x , независимой переменной x и генеральной совокупности, описывается уравнением

$$Y(X) = A_0 + A_1 \cdot X,$$

где A_0 – свободный член уравнения регрессии;

A_1 – коэффициент уравнения регрессии.

После расчетов строится соответствующая прямая, называемая линией регрессии. Коэффициенты A_0 и A_1 , называемые также параметрами модели, выбираются таким образом, чтобы сумма квадратов отклонений точек, соответствующих реальным наблюдениям данных, от линии регрессии была бы минимальной. Подбор коэффициентов производится по методу наименьших квадратов.

Цели регрессионного анализа:

- определение наличия и характера (математического уравнения, описывающего зависимость) связи между переменными.
- определение степени детерминированности вариации критериальной переменной предикторами;
- предсказание значения зависимой переменной с помощью независимой переменной;
- определение вклада независимых переменных в вариацию зависимой переменной.

Определение коэффициента регрессии (A_1) и свободного члена уравнения регрессии (A_0):

– представление исходной информации в виде векторов:

$$X := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{pmatrix}, \quad Y := \begin{pmatrix} 8,013 \\ 12,933 \\ 19,85 \\ 20,503 \\ 28,228 \\ 24,741 \\ 33,105 \\ 32,04 \\ 32,914 \\ 36,472 \end{pmatrix};$$

– определение суммы элементов векторов и произведений векторов:

$$SX := \sum X = 55;$$

$$SY := \sum Y = 248,8;$$

$$SXY := X \cdot Y = 1,614 \cdot 10^3;$$

$$SXX := X \cdot X = 385;$$

$$SYY := Y \cdot Y = 6,991 \cdot 10^3;$$

– определение параметров уравнения регрессии:

$$N := 10;$$

$$A := N \cdot SXX - SY \cdot SY = 825;$$

$$B := N \cdot SYY - SY \cdot SY = 8,008 \cdot 10^3;$$

$$C := N \cdot SXY - SX \cdot SY = 2,456 \cdot 10^3;$$

– свободный член уравнения регрессии A_0 :

$$A_0 := \frac{SY \cdot SXX - SXY \cdot SX}{A} = 8,504;$$

– коэффициент уравнения регрессии A_1 :

$$A_1 := \frac{C}{A} = 2,977.$$

Определение стандартной ошибки предсказания, являющейся мерой качества реальной зависимости между Y и X , с помощью уравнения линейной регрессии. Мерой качества приближенного описания реальной зависимости между величинами Y и X с помощью уравнения линейной регрессии является стандартное отклонение значений Y от регрессионной прямой, вычисляемое по формуле

$$SYX := \sqrt{\frac{\sum_{i=1}^n (Y_i)^2 - A_0 \cdot \sum_{i=1}^n Y_i - A_1 \cdot \sum_{i=1}^n (X_i \cdot Y_i)}{n - 2}}. \quad (2.1)$$

SYX является мерой точности предсказания значений случайной величины Y по заданным значениям величины X , поэтому SYX называют также стандартной ошибкой предсказания. Найдем стандартную ошибку предсказания для данного примера ($n:=10$) по формуле (2.1):

$$SYX = 2,945.$$

Проверка значимости коэффициента регрессии.

Если в результате проведенной проверки нет оснований сомневаться в адекватности линейной модели, то необходимо проверить гипотезу о том, что в действительности в генеральной совокупности отсутствует линейная регрессия, а то, что полученный коэффициент регрессии отличен от нуля, объясняется только случайностью выборки. Гипотеза H_0 проверяется с помощью стандартного t -критерия Стьюдента. Значение t -критерия определяется по формуле

$$t := \frac{A_1}{SYX} \cdot \frac{1}{\sqrt{\sum_{i=1}^n (X_i)^2 - n \cdot X_{sr}^2}}, \quad (2.2)$$

где A_1 – абсолютная величина коэффициента регрессии;

SYX – стандартная ошибка предсказаний.

X_{sr} определяется по формуле

$$X_{sr} := \frac{\sum_{i=1}^n X_i}{n} = 5,5.$$

Если значения $t > t_p$, то нулевая гипотеза отклоняется; можно сделать вывод, что линейная регрессия значима на заданном уровне значимости p . Оценим значимость коэффициента регрессии при уровне значимости $p = 0,05$. Для этого подставим найденные ранее значения в формулу (2.2) и определим значение t -критерия:

$$t = 9,812; t_{p=0,05} = 2,306.$$

Поскольку $t > t_p$, то на уровне значимости 0,05 коэффициент регрессии является статистически значимым.

Практическое задание

Найти уравнение регрессии по данным, представленным в таблице 2.1:

- найти значения коэффициента регрессии и свободного члена уравнения регрессии;
- определить стандартную ошибку предсказания, являющуюся мерой качества реальной зависимости между величинами Y и X , с помощью уравнения линейной регрессии;
- проверить значимость коэффициента регрессии при $p = 0,95$.

Таблица 2.1 – Варианты заданий

Вариант	Y, X	Экспериментальные данные									
		3	4	5	6	7	8	9	10	11	12
1	Y	-1,0	-0,55	-0,1	-0,35	0,8	1,25	1,7	2,15	2,6	3,05
	X	-6,78	-6,56	-6,14	-5,31	-3,68	-0,85	5,81	18,15	42,4	90,03
2	Y	0,01	0,56	1,11	1,66	2,21	2,28	3,3	3,85	4,4	4,95
	X	34,23	5,97	1,28	-1,54	-3,54	-5,09	-6,36	-7,44	-8,37	-9,2
3	Y	-2	-1,6	-1,2	-0,8	-0,4	0	0,4	0,8	1,2	1,6
	X	0	10,24	5,76	2,56	0,53	0	0,64	2,56	5,76	10,24
4	Y	0,3	1,57	2,84	4,11	5,38	6,65	7,92	9,19	10,46	11,73
	X	15,33	4,55	3,41	2,97	2,74	2,6	2,59	2,44	2,38	2,34
5	Y	-3,5	-2,65	-1,8	-0,95	-0,1	0,75	1,6	2,45	3,3	4,15
	X	0,01	0,03	0,07	0,12	0,19	0,2	0,29	0,31	0,325	0,33
6	Y	0,15	0,94	1,72	2,15	3,29	4,08	4,86	5,65	6,43	7,22
	X	-9,69	-4,2	-2,37	-1,25	-0,43	0,21	0,74	1,3	1,58	1,93
7	Y	0,35	0,82	1,28	1,75	2,21	2,675	3,14	3,605	4,07	4,535
	X	6,86	5,23	4,78	4,57	4,45	4,37	4,35	4,28	4,25	4,22
8	Y	-1	-0,8	-0,6	-0,4	-0,2	0	0,2	0,4	0,6	0,8
	X	4,14	4,2	4,3	4,45	4,67	0	5,49	6,85	7,32	8,95
9	Y	0	2,3	2,6	2,9	3,2	3,5	3,8	4,1	4,4	4,7
	X	2,67	4,06	6,16	8,13	10,92	14,29	18,29	22,97	28,39	34,6
10	Y	-5	-4	-3	-2	-1	0	0	0	0	0
	X	0,01	0,02	0,05	0,22	0,21	0,38	0,42	0,47	0,49	0,5

Окончание таблицы 2.1

1	2	3	4	5	6	7	8	9	10	11	12
11	Y	0,95	1,21	1,47	1,74	2,0	2,26	2,52	2,78	3,05	0
	X	8,16	3,39	2,19	1,34	0,88	0,16	0,54	0,33	0,28	0,19
12	Y	0,35	0,82	1,28	1,75	2,21	2,68	3,14	3,61	4,07	0
	X	16,99	8,83	6,61	5,56	4,96	4,62	4,29	4,09	3,93	3,8
13	Y	-1,7	-1,43	-1,16	-0,89	-0,62	-0,35	-0,08	0,19	0,46	0,73
	X	26,96	14,46	7,17	2,92	0,45	-0,98	-1,35	-2,31	-2,6	-2,77
14	Y	-5	-3,5	-2	-0,5	0	2,5	1,2	5,5	0	8,5
	X	0	0,01	0,06	0,28	0,87	2,05	2,92	3,23	3,31	3,33
15	Y	-2	-1,4	-0,8	-0,2	0,4	1,0	1,6	2,2	2,8	3,4
	X	6,8	3,33	1,09	2,02	0,27	1,7	4,35	8,23	13,33	19,65

Контрольные вопросы

- 1 В чем состоит метод наименьших квадратов при построении уравнения регрессии?
- 2 Что такое корреляция, коэффициент корреляции?
- 3 Каковы задачи регрессионного анализа?
- 4 Для чего необходимо проводить проверку значимости коэффициента регрессии?
- 5 Как определить стандартную ошибку предсказания?

3 Лабораторная работа № 3. Модели временных рядов

Цель работы: ознакомиться с моделями временных рядов и методами их анализа.

Порядок выполнения работы

- 1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
- 2 Получить задание у преподавателя, выполнить типовые задания.
- 3 Сделать выводы по результатам исследований.
- 4 Оформить отчет.

Требования к отчету

- 1 Цель работы.
- 2 Постановка задачи.
- 3 Результаты исследования.
- 4 Выводы.

Основные теоретические положения

Временной ряд – это последовательность наблюдений, упорядоченная по времени: $y_1, y_2, \dots, y_t, \dots, y_n$, где y_t – числа, представляющие наблюдения некоторой переменной в n равноотстоящих моментах времени $t = 1, 2, \dots, n$.

Примеры данных, которые необходимо изучать во времени: цены на товар, деловая активность, национальный валовой продукт.

Особенность временных рядов – зависимость данных, характер которой может определяться положением наблюдений в последовательности.

Основные задачи анализа временных рядов:

- прогнозирование на основе знания прошлого;
- сжатое описание характерных особенностей ряда;
- управление процессом, порождающим ряд.

В анализе временных рядов предполагается, что исходные данные содержат детерминированную и случайную (ε_t) составляющие. В общем случае детерминированная составляющая может быть представлена в виде совокупности следующих компонентов:

- тренда u_t , определяющего главную тенденцию временного ряда;
- циклов (циклической составляющей) W_t – более или менее регулярных колебаний относительно тренда;
- сезонной составляющей S_t – периодических колебаний.

Временной ряд может быть представлен различными математическими моделями.

Аддитивная модель

$$y_t = u_t + W_t + S_t + \varepsilon_t.$$

Мультипликативная модель

$$y_t = u_t W_t S_t \varepsilon_t.$$

Если предположить, что сезонная составляющая S_t пропорциональна сумме тренда и циклической составляющей $S_t = (u_t + W_t)C_t$, то временной ряд будет представлен в виде смешанной модели:

$$y_t = (u_t + W_t)(1 + C_t) + \varepsilon_t.$$

Выбор модели зависит от конкретной совокупности явлений, определяющих данный временной ряд, и их взаимосвязей.

Основные цели.

Существует две основные цели анализа временных рядов:

- 1) определение природы ряда;
- 2) прогнозирование (предсказание будущих значений временного ряда по настоящим и прошлым значениям).

Обе эти цели требуют, чтобы модель ряда была идентифицирована и более или менее формально описана. Как только модель определена, можно с ее помощью интерпретировать рассматриваемые данные. Не обращая внимания на глубину понимания и справедливость теории, можно затем экстраполировать ряд на основе найденной модели, т. е. предсказать его будущие значения.

Практическое задание

Положение на рынке местного производителя продукта П1 представляют следующие значения (исходные данные по кварталам определить самостоятельно) (таблица 3.1).

Таблица 3.1 – Шаблон таблицы для исходных данных

Продажи за квартал				
Год	I	II	III	IV
1				
2				
3				
4				
5				

Пример заполнения шаблона таблицы представлен в таблице 3.2.

Таблица 3.2 – Пример исходных данных для решения практического задания

Год	Продажи за квартал			
	I	II	III	IV
1	19	24	38	25
2	21	28	44	23
3	23	31	41	23
4	24	35	48	21
5	22	37	50	22

1 Вычислить сезонные индексы для данных из таблицы 3.2 (использовать центрированные средние значения за четыре квартала).

2 Исключить сезонную составляющую из этих данных.

3 Методом наименьших квадратов найти параметры прямой, которая наилучшим образом характеризует основную тенденцию временного ряда в данных о продаже продукта.

4 Определить циклический компонент в этом временном ряду, исключив тренд из исходных данных.

Контрольные вопросы

- 1 Назовите основные цели анализа временных рядов.
- 2 Какова аддитивная модель временного ряда?
- 3 Какова мультипликативная модель временного ряда?

4 Лабораторная работа № 4. Сингулярный анализ

Цель работы: изучить основные этапы сингулярного анализа и исследовать погрешность восстановления детерминированного временного ряда.

Порядок выполнения работы

- 1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
- 2 Получить задание у преподавателя, выполнить типовые задания.
- 3 Сделать выводы по результатам исследований.
- 4 Оформить отчет.

Требования к отчету

- 1 Цель работы.
- 2 Постановка задачи.
- 3 Результаты исследования.
- 4 Выводы.

Основные теоретические положения

Данные в компьютерной информационной системе, представленные вещественными числами и определенные последовательно во времени, именуют временной последовательностью данных. В общем случае такую последовательность данных, обычно упорядоченных во времени, называют временным рядом. Поэтому в дальнейшем будем использовать понятие временного ряда и для извлечения информации из последовательностей данных рассмотрим метод сингулярного спектрального анализа (*SSA*-метод), который применяется для анализа временных рядов и может быть использован на этапе эксплуатации имитационной модели (ИМ). Он позволяет выделить составляющие временного ряда, которые описывают тренд и гармонические колебания, и те составляющие временного ряда, которые относят к «шуму». При этом метод не требует стационарности временного ряда, знания модели тренда, а также сведений о наличии периодических составляющих и их периодах. Также с помощью данного метода можно определить модель тренда и использовать это для дальнейшей обработки временного ряда с уже известной моделью тренда.

Математической основой *SSA*-метода является сингулярное разложение. Для успешного применения *SSA*-метода следует последовательно пройти несколько шагов.

Вложение. На этом шаге выбирается ширина окна L , от выбора которой зависят результаты применения SSA -метода. Из-за того что нет общих рекомендаций по выбору ширины окна, параметр L зависит от решаемой задачи и предварительной информации, известной о временном ряде. Например, для выделения тренда рекомендуется выбирать ширину окна не слишком большой. С другой стороны, для выделения гармонических колебаний рекомендуется большая ширина окна. Поэтому рекомендуется выполнять два прохода с разной шириной окна. После выбора ширины окна в соответствии с L строится траекторная матрица A ряда, которая будет являться по условию ее построения ганкелевой.

Сингулярное разложение. Для матрицы $S = A \cdot A^T$ находятся собственные числа μ и ортонормированные собственные векторы U . Упорядоченные по убыванию собственные числа, которые больше нуля, часто называются сингулярными числами, а соответствующие им собственные векторы – левыми сингулярными векторами U . После этого вычисляются векторы V , которые называются правыми сингулярными векторами, и находятся элементарные матрицы, на сумму которых раскладывается первоначальная траекторная матрица.

Группировка. На данном этапе элементарные матрицы группируются по принципу принадлежности к тренду, гармоническим колебаниям или к шуму. Этот этап является наиболее сложным при применении SSA -метода. Для нахождения тренда на диаграммах собственных векторов (по оси абсцисс откладывается порядковый номер координаты собственного вектора, а по оси ординат – значение координаты собственного вектора) выделяют медленно меняющиеся векторы. Сумма элементарных матриц, соответствующих этим векторам, будет являться траекторной матрицей тренда ряда. После этого восстанавливают гармонические колебания ряда. Для отделения шума можно воспользоваться несколькими замечаниями: нерегулярное поведение сингулярных векторов может свидетельствовать о принадлежности их к набору, порожденному шумовым компонентом; также об этом может свидетельствовать медленное, почти без скачков, убывание собственных чисел с некоторого номера.

Диагональное усреднение. Если полученные сгруппированные матрицы ганкелевы, то они являются траекторными матрицами некоторого временного ряда, который может быть легко по ним восстановлен. Однако обычно сгруппированные матрицы редко получаются ганкелевыми, поэтому для восстановления временного ряда прибегают к диагональному усреднению. В соответствии с этим этапом каждый член восстановленного временного ряда будет являться средним арифметическим соответствующей ему побочной диагонали траекторной матрицы.

В результате проделанных шагов получается несколько временных рядов: один описывает тренд первоначального (исходного) временного ряда, другой – гармонические колебания, третий – шумовые составляющие.

Для исследования SSA -метода применяется комплекс информационных технологий, представленный табличным процессором MS Excel, математическим пакетом Mathcad и пакетом статистической обработки данных Statistica. При этом не требуется разработки специализированного программного обеспечения.

Практическое задание

Для исследований задать функцию вида $F(x) = y_1(x) + y_2(x)$, где $y_1(x)$ – линейная функция, например, $y_1(x) = 0,1x + 2$; $y_2(x)$ – гармоническая функция, например, $y_2(x) = \sin(x)$. Переменная x принимает значения от 0 до 42 с шагом, равным единице. Таким образом, длина N ряда $G = (g_0, g_1, \dots, g_{N-1})$ равна 43. При этом длина окна $L = 22$, число L -мерных векторов $K = N - L + 1 = 22$.

Определить абсолютную ошибку восстановления ряда величиной отклонения

$$\Delta = \max |g_i - g_i^e|, \quad (4.1)$$

где g_i – элементы исходного ряда;

g_i^e – элементы восстановленного ряда, $i = 0, \dots, N - 1$.

Относительную ошибку восстановления ряда рассчитать по формуле

$$\delta = \frac{\max |g_i - g_i^e|}{\max |g_i|} \cdot 100 \%. \quad (4.2)$$

Контрольные вопросы

- 1 Перечислите основные этапы для успешного применения SSA-метода.
- 2 В чем состоит назначение этапа вложения?
- 3 В чем состоит назначение этапа сингулярного разложения?
- 4 В чем состоит назначение этапа группировки?
- 5 В чем состоит назначение этапа диагонального усреднения?

5 Лабораторная работа № 5. Кластерный анализ

Цель работы: изучить метод кластерного анализа при обработке экспериментальных данных.

Порядок выполнения работы

- 1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
- 2 Получить задание у преподавателя, выполнить типовые задания.
- 3 Сделать выводы по результатам исследований.
- 4 Оформить отчет.

Требования к отчету

- 1 Цель работы.
- 2 Постановка задачи.

3 Результаты исследования.

4 Выводы.

Основные теоретические положения

Пусть для выявления особенностей поведения исследуемого процесса требуется анализ многомерных данных. Так, например, необходимо решить задачу построения кластеров непересекающиеся подмножеств множества $X \subset R^n$.

$$X = \{x_i \mid x_i \in R^n, i = 1, \dots, m\}, m \geq 3. \quad (5.1)$$

Задача решается с использованием методов кластеризации, через матрицы сходства, эвристические алгоритмы перебора, идеи математического программирования, через анализ свойств функции плотности эмпирического распределения выборки X и др.

Описание методов кластеризации данных достаточно широко отражено в различных источниках. Большинство алгоритмов, при всём их разнообразии, основано на гипотезе компактности, состоящей в том, что объекты одного класса по результатам измерений близко расположены по отношению друг к другу, а объекты разных классов заметно различаются между собой. Все методы разбиения множеств на кластеры возможно разделить на «иерархические» и «неиерархические». Наиболее широко распространены методы кластеризации K-Means, Tree Clustering, Fuzzy Relation Clustering (FRC), которые представляют собой представителей основных подходов к разделению множества на кластеры.

Методика определения классов несколькими методами представлена следующими действиями.

Шаг 1. Предварительный анализ данных.

Выполняется исследование многомерных данных с формированием матрицы S :

$$S = \|s_{ij}\|, s_{ij} \in R, i = 1, \dots, r, j = 1, \dots, m, \quad (5.2)$$

где s_{ij} – чувствительность i -го отклика при изменении j -го параметра объекта.

Для проведения кластерного анализа матрицы вида (5.2) преобразуются к вектору вида

$$S = (s_{11}, s_{12}, \dots, s_{1m}, \dots, s_{r1}, \dots, s_{rm}). \quad (5.3)$$

В итоге для каждого процесса $\{F_i\}, i = 1, \dots, n$, проводится t раз исследование и получается множество векторов вида (5.3)

$$\{S_i\}, i = 1, \dots, t \cdot n. \quad (5.4)$$

Шаг 2. Кластерный анализ.

В результате разбиения множества (5.4) на k классов каждый из нескольких методов кластеризации ставит в соответствие номерам $i = 1, \dots, t \cdot n$ элементов множества (5.4) соответствующие им номера классов $K_j, j = 1, \dots, k$.

Шаг 3. Интерпретация результатов кластерного анализа.

Поскольку векторы $S_i \in R^{m \cdot r}, i = 1, \dots, t \cdot n$ кластеризуемого множества (5.4) представляют собой результаты статистической обработки экспериментальных данных n процессов, то итог кластеризации множества (5.4) можно представить в виде матрицы относительных частот принадлежности процесса определенным классам:

$$P = \|p_{ij}\|, \quad i = 1, \dots, n, \quad j = 1, \dots, k, \quad (5.5)$$

где i – номер процесса;

j – номер класса K_j ;

k – количество классов;

$p_{ij} \in [0, 1]$ – относительная частота принадлежности i -го процесса классу K_j .

Шаг 4. Обобщение итогов кластеризации разными методами.

Пусть P_1, P_2, P_3 – матрицы вида (5.5) относительных частот принадлежности процесса $\{F_i\}, i = 1, \dots, n$, определенным классам в соответствии с методами кластерного анализа K-Means, Tree Clustering, FRC соответственно. Тогда значения элементов $p_{ij} \in [0, 1]$ обобщенной матрицы P могут быть найдены по следующей формуле (на элементы строк обобщенной матрицы P налагается условие равенства единице их суммы):

$$p_{ij} = \begin{cases} (1 - p_{1ij})p_{2ij}p_{3ij} + p_{1ij}(1 - p_{2ij})p_{3ij} + p_{1ij}p_{2ij}(1 - p_{3ij}) + p_{1ij}p_{2ij}p_{3ij}, & j < k; \\ 1 - \sum_{j=1}^{k-1} p_{ij}, & j = k. \end{cases} \quad (5.6)$$

Обобщаемые матрицы P_1, P_2, P_3 должны иметь одну размерность. В случае разбиения множества (5.4) на $k_1 \leq k, k_2 \leq k, k_3 \leq k$ классов необходимо предварительно привести матрицы P_1, P_2, P_3 к одной размерности $k = \max\{k_1, k_2, k_3\}$. Приведение матрицы к необходимой размерности возможно за счет ее дополнения столбцами с нулевыми относительными частотами попадания объекта в добавленные классы.

Практическое задание

Выполнить кластерный анализ тремя методами для данных, характеризующих уровень энергоэффективности стран «Группы двадцати». Страны этой группы представлены 19 основными промышленными и формирующими рын-

ки странами (Аргентина, Австралия, Бразилия, Великобритания, Германия, Индия, Индонезия, Италия, Канада, Китай, Мексика, Турция, Россия, Саудовская Аравия, США, Франция, ЮАР, Республика Корея, Япония). Отдельно в группу входит Европейский союз. В странах «Группы двадцати» сосредоточено две трети населения мира, около 90 % мирового ВВП. Они представляют 80 % мировой торговли; на долю этих стран приходится 84 % выбросов в атмосферу от сгорания ископаемого топлива (https://studme.org/75023/statistika/primer_klasternogo_analiza).

Для создания информационной базы были использованы официально опубликованные статистические данные обзора мировой энергетики: производство электроэнергии в тераватт-час и выбросы CO₂ в атмосферу в миллионах тонн. Макроэкономические данные предоставлены на официальном сайте Федерального статистического ведомства Германии: ВВП в миллиардах долларов США

и численность населения в миллионах человек.

Это наиболее важные показатели, характеризующие экономическую, энергетическую и экологическую направленность развития каждой страны. На основе данных рассчитаны индикаторы энергоэффективности для каждой страны «Группы двадцати» за 2012 г.:

X_1 – электроемкость ВВП (затраты электроэнергии на единицу ВВП), кВт·ч/долл.;

X_2 – производство электроэнергии на душу населения, МВт·ч/чел.;

X_3 – ВВП на единицу выбросов CO₂, долл./кг;

X_4 – ВВП на душу населения, тыс. долл./чел.;

X_5 – выбросы CO₂ на душу населения, т/чел.;

X_6 – выбросы CO₂ на единицу производства электроэнергии, кг/Вт·ч.

Цель кластерного анализа – выделить однородные группы стран по уровню энергоэффективности в 2012 г. Следует отметить, что в «Группу двадцати» входят страны, различные по территории, численности и плотности населения, уровню экономического потенциала, структуре энергетики и энергетической политики.

Контрольные вопросы

1 Назовите основные этапы методики определения классов несколькими методами.

2 В чем состоит назначение предварительного анализа данных?

3 В чем состоит назначение обобщения итогов кластеризации разными методами?

6 Лабораторная работа № 6. Параллельные вычисления на OpenMPI

Цель работы: изучить организацию проведения распределенных вычислений на основе MPI.

Порядок выполнения работы

- 1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
- 2 Получить задание у преподавателя, выполнить типовые задания.
- 3 Сделать выводы по результатам исследований.
- 4 Оформить отчет.

Требования к отчету

- 1 Цель работы.
- 2 Постановка задачи.
- 3 Результаты исследования.
- 4 Выводы.

Основные теоретические положения

Настройки Microsoft Visual Studio .NET для работы с MPICH.

Для проведения исследований по распределению вычислений, которые необходимо выполнить с базовой имитационной моделью, требуется создать новый проект с программой MPICH2. После установки MPICH2 следует выполнить ряд действий в Microsoft Visual Studio .NET.

- 1 Создать новый проект, для которого задается имя проекта. Проект сохраняется в требуемой директории пользователя. На этапе проведения исследований рекомендуется создать консольное приложение Win32 без файлов (рисунок 6.1).

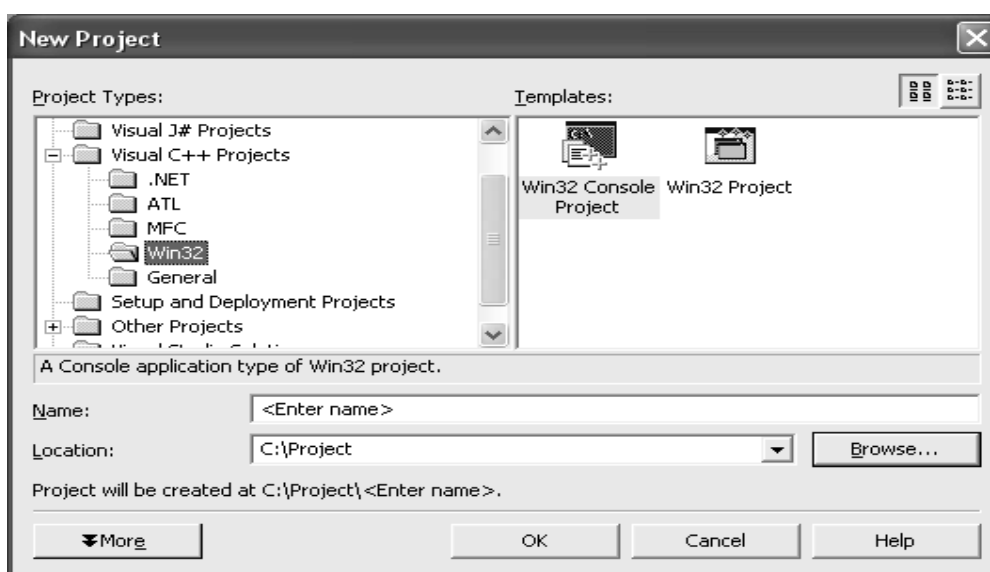


Рисунок 6.1 – Окно создания нового проекта для проведения исследований

2 Завершить действия мастера среды разработки Microsoft Visual Studio.NET по установке нового проекта.

3 Перейти на Project->Properties для вызова диалогового окна, в котором будут выполнены настройки параметров нового проекта пользователя.

4 Установить параметры для применения многопоточных (multithreaded) библиотек. При этом отметить указатели режимов Debug (рисунок 6.2) и Release.

5 Указать include путь для используемых целевых конфигураций: C:\Program Files\MPICH2\include (рисунок 6.3).

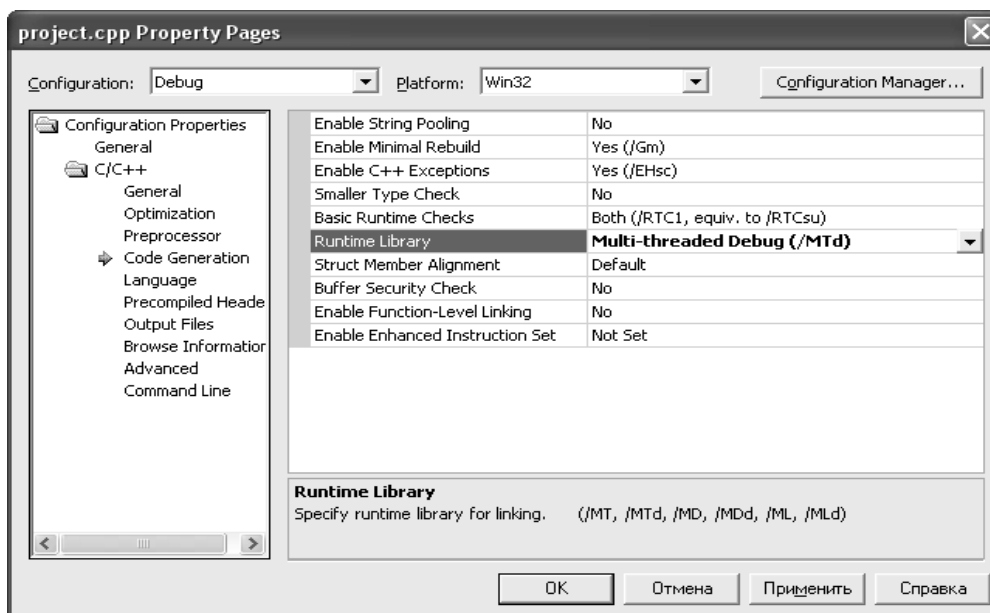


Рисунок 6.2 – Копия экрана для конфигурации режима отладки Debug

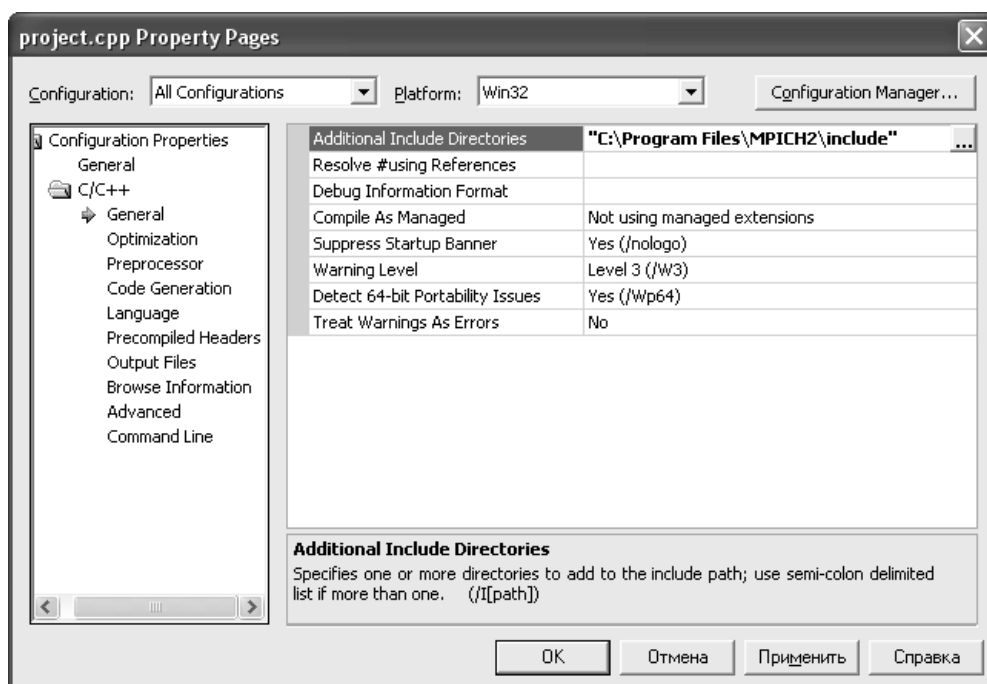


Рисунок 6.3 – Копия экрана для определения места поиска дополнительных библиотек, используемых при компиляции программы

6 Указать lib путь для используемых целевых конфигураций: C:\Program Files\MPICH2\lib (рисунок 6.4).

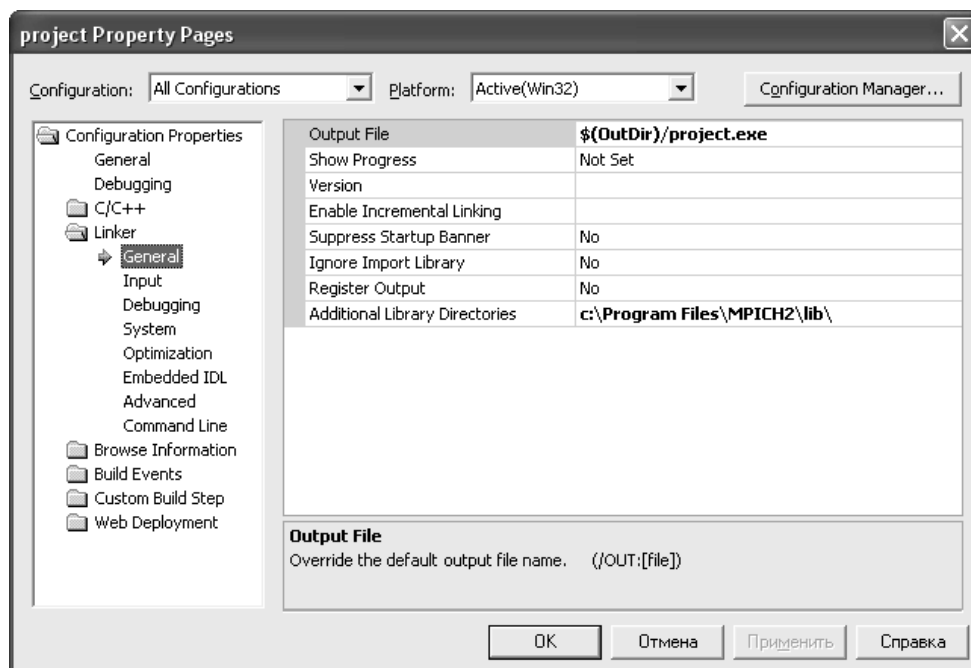


Рисунок 6.4 – Копия экрана для определения папок с дополнительными библиотеками, используемыми при сборке программы

7 Дополнительно добавить библиотеку mpich2.lib для целевых конфигураций (рисунок 6.5).

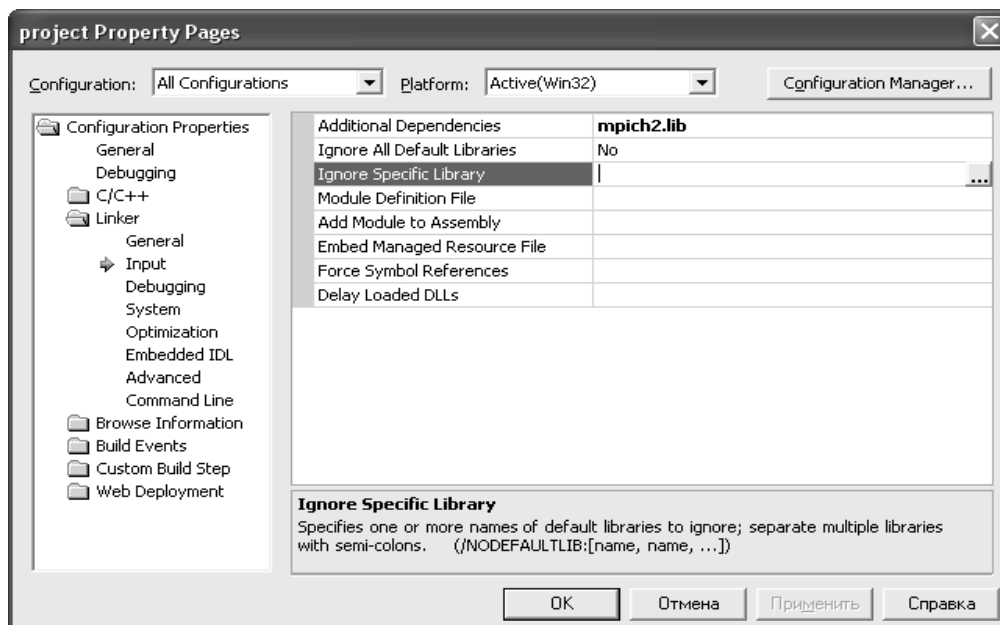


Рисунок 6.5 – Копия экрана для включения дополнительных библиотек mpich2.lib, используемых при компиляции программы

8 Завершить установку проекта закрытием диалогового окна установки проекта.

После выполнения шага 8 можно добавлять код с использованием вызовов функций из интерфейса MPI.

Запуск программ с помощью MPIEXEC.

1 Запускается `wmpiregister.exe`, располагающийся в каталоге `bin` в директории, в которой предварительно установлена программа `MPICH2`.

2 Появляется окно, в котором вводится логин и пароль персональной учетной записи в операционной системе.

3 Кликается `Register`. Появляется информация о том, что логин и пароль включены в реестр: `Password encrypted into the Registry` (рисунок 6.6).



Рисунок 6.6 – Копия экрана для ввода имени пользователя и пароля

От имени указанного пользователя будут запускаться вычисления на хостах, используемых для распределенных вычислений.

4 Кликается `Ok`.

5 Выполняется запуск `wmpiconfig.exe`, чтобы проверить доступность требуемых хостов.

6 В меню `Domain` отмечается требуемый домен и кликается на кнопку `Get Hosts`.

7 Выбираются `Scan Hosts`, чтобы узнать, на каких хостах установлена программа `MPICH2` (рисунок 6.7).

8 Хосты, помеченные зелёным цветом, будут доступны, а хосты, помеченные серым цветом, не доступны.

9 Запускается текстовый редактор, в котором создается файл с параметрами для запуска приложений.

В файле можно указать следующие параметры:

– командная строка при запуске приложения

```
exe experimenter.exe wdoptimizer test121.xml op121.xml;
```

- количество запускаемых процессов: $n = 6$;
- очередность подключения хостов, на которых будут запущены процессы для распределения вычислений:

hosts A405-2 A405-3 A405-5 A405-6 A405-7 A405-13.

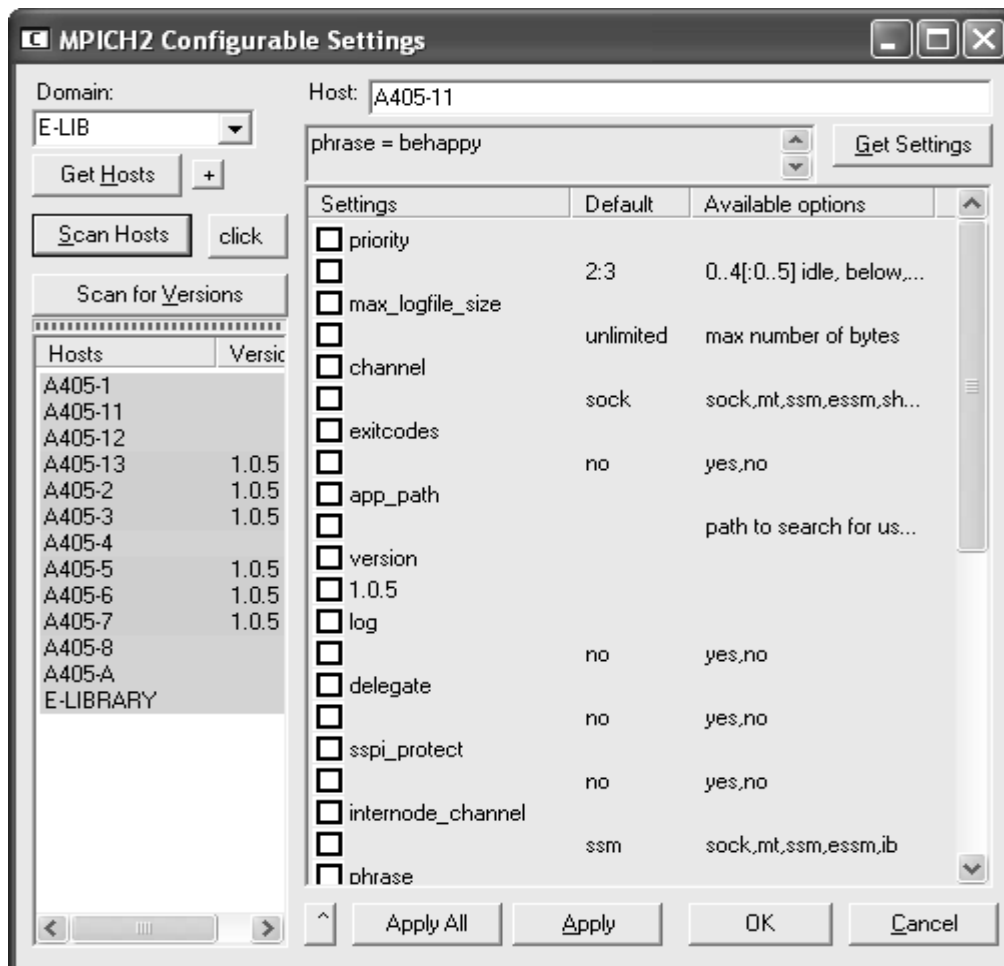


Рисунок 6.7 – Копия экрана с интерфейсом программы настройки хостов для распределенных вычислений

Базовая имитационная модель, принятая для исследований, запускается программой `experimenter`, для которой вышеуказанных параметров достаточно, хотя возможны и другие для более точной настройки.

10 Сохраняется файл с параметрами.

11 Запускается `MPICHES`, который располагается в каталоге `bin` в директории с предварительно установленной `MPICH2-1.0.5p2-win32-ia32` (рисунок 6.8).

12 Кликается на кнопку `Load Job` и выбирается сохраненный файл.

13 Запускаются процессы при нажатии на кнопку `Execute`.

Кроме указанных выше действий, можно выполнить запуск программ для распределенных вычислений из командной строки, т. к. в используемой реализации интерфейса существуют консольные версии программ для ввода имени пользователя и пароля, настройки хостов, используемых для распределения вы-

числений, и программы для запуска приложений.

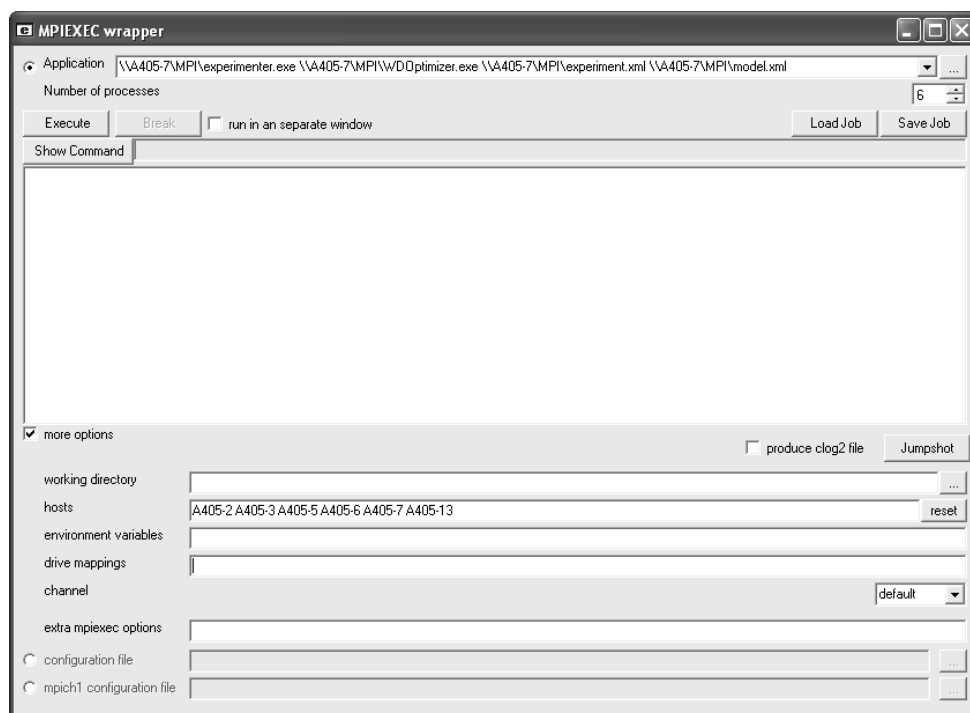


Рисунок 6.8 – Копия экрана с интерфейсом программы для выполнения распределенных вычислений

Программные средства, необходимые для использования MPICH:

- MPICH (MPICH2-1.0.5p2-win32-ia32);
- .NetFramework;
- Windows Installer 3.0 (WindowsInstaller-KB893803-v2-x86.exe).

Практическое задание

Для оценки эффективности проведения эксперимента с распределенными вычислениями определить среднюю длительность \bar{t}_i одного прогона в i -м эксперименте по формуле

$$\bar{t}_i = \frac{\max \langle T_{ij} \rangle}{N_i},$$

где T_{ij} – длительность i -го эксперимента на j -м компьютере;

N_i – количество прогонов в i -м эксперименте.

В соответствии с сетевым законом Амдала при общей эффективности вычислений не наблюдается прямо пропорционального снижения длительности экспериментов с увеличением числа используемых компьютеров в сети.

Например, при сравнении экспериментов с $N_i = 400$ на одном и двух компьютерах продолжительность прогонов модели сокращается в 1,9, т. е. почти в 2 раза. На трех компьютерах продолжительность эксперимента, в сравнении с одним компьютером, сокращается в 2,5 раза.

Исследования выполнить до использования 10 компьютеров при распределении вычислений. Сделать вывод об эффективном использовании компьютеров в сети.

Контрольные вопросы

- 1 Что такое закон Амдала?
- 2 Укажите настройки Microsoft Visual Studio .NET для работы с MPICH.
- 3 Укажите последовательность запуска программ с помощью MPIEXEC.

Список литературы

1 **Дадян, Э. Г.** Методы, модели, средства хранения и обработки данных: учебник [Электронный ресурс] / Э. Г. Дадян, Ю. А. Зеленков. – Москва: Вузовский учебник; ИНФРА-М, 2017. – 168 с. – Режим доступа: <http://znanium.com/>. – Дата доступа: 01.06.2020.

2 **Дивеев, А. И.** Применение метода вариационного аналитического программирования для синтеза управления летающим роботом / А. И. Дивеев, Н. Б. Конырбаев // *Фундаментальные исследования*. – 2015. – № 3. – С. 51–57.

3 **Дюк, В. А.** Применение технологий интеллектуального анализа данных в естественно-научных, технических и гуманитарных областях / В. А. Дюк, А. В. Флегонтов, И. К. Фомина // *Изв. Рос. гос. пед. ун-та им. А. И. Герцена*. – 2011. – № 138. – С. 77–84.

4 **Замятин, А. В.** Интеллектуальный анализ данных: учебное пособие / А. В. Замятин. – Томск: Том. гос. ун-т, 2016. – 120 с.

5 **Oplatkova, Z.** Investigation on Artificial Ant using Analytic Programming Commands / Z. Oplatkova, I. Zelinka // *GECCO'06, July 8–12, 2006, Seattle, Washington, USA*. – P. 949–950.

6 **Oplatkova, Z.** Investigation on Evolutionary Synthesis of Movement Commands / Z. Oplatkova, I. Zelinka // *Modelling and Simulation in Engineering*. – 2009. – Vol. 2009. – 12 p.

7 **Zelinka, I.** Analytic Programming – Symbolic Regression by Means of Arbitrary Evolutionary Algorithms / I. Zelinka, Z. Oplatkova, L. Nolle // *I. J. of Simulation*. – 2005. – Vol. 6, № 9. – P. 44–56.