Министерство образования и науки Российской Федерации

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

АНГЛИЙСКИЙ ДЛЯ СТУДЕНТОВ, ИЗУЧАЮЩИХ ПРИКЛАДНУЮ МАТЕМАТИКУ

ENGLISH FOR APPLIED MATHEMATICS STUDENTS

Учебное пособие по английскому языку
для студентов 3 курса
направления «Прикладная математика и информатика»

НОВОСИБИРСК
2015

ББК 81.432.1-923
А 647

Авторы-составители:
А.Ю. Алябьева, С. А. Вандакурова, Т.В. Волошина, Е.Г. Калинина

Рецензенты:
д-р филол. наук, проф. *М.В. Влавацкая*;
канд. филол. наук, доцент кафедры ИЯ СИУ-РАНХ *В.Г. Шабаев*

А 647 **Английский для студентов, изучающих прикладную математику** : учебное пособие / А. Ю. Алябьева, С. А. Вандакурова, Т. В. Волошина, Е. Г. Калинина. – Новосибирск: Изд-во НГТУ, 2015. – 100 с.

ISBN 978-5-7782-2581-7

Настоящее пособие предназначено для студентов, обучающихся по специальности «Прикладная математика и информатика» на факультете ФПМИ НГТУ. Целью данного пособия является формирование у студентов коммуникативной языковой компетенции, которая реализуется как в письменной, так и в устной речи. Основной принцип работы состоит в активизации речевой деятельности студентов с помощью коммуникативных упражнений. Все использованные тексты и задания связаны с профессиональной подготовкой будущих специалистов и соответствуют государственному образовательному стандарту.

Учебное пособие включает в себя 10 модулей, каждый из которых состоит из текстов и лексических упражнений, характерных для подъязыка специальности. Все модули содержат дополнительные задания и тексты.

Пособие может быть использовано для аудиторной и внеаудиторной работы в зависимости от целей, поставленных преподавателем.

Работа выполнена на кафедре иностранных языков ТФ НГТУ

ББК 81.432.1-923

ISBN 978-5-7782-2581-7

UNIT 1. “NUMERALS”

Terminology

according
add
assume
bar
binary
brackets
calculation
create
deal
decimal
decision
develop
digits
divide
drawback
equal
error
express
fraction
horizontal
infinite
innovation
introduction
minus
multiplied
notation
rapid
representation
sign
squared
straight
subtract
suitable
vertical

Task 1. Find Russian equivalents to the following words:

1. sign	a. недостаток
2. bar	b. бесконечный
3. infinite	c. знак
4. to assume	d. быстрый
5. drawback	e. черта
6. rapid	f. решение
7. calculation	g. вычисление
8. decision	h. принимать

Additional vocabulary

Positional notation – позиционное представление
Binary system – двоичная система
Decimal system – десятичная система

Task 2. Read the text.

Numerals are signs or symbols for graphic representation of numbers. The earliest forms of numerical notation were simply groups of straight lines, either vertical or horizontal, each line corresponding to number 1. Such a system was inconvenient when dealing with large numbers.

The system of number symbols created by the Romans expresses all numbers from 1 to 1,000,000 with the help of seven symbols: I – 1; V – 5; X – 10; L – 50; C – 100; D – 500; M – 1,000. Roman numerals are read from left to right. The symbols are usually added together. For example: IV – 4; VI – 6; LX -60; XL – 40; MMII – 2002; MCMLVII – 1957. A small bar placed over the numeral multiplies the numeral by 1,000. Roman numerals are still used today, more than 2,000 years after their introduction. The Roman system's only drawback is that it is not suitable for rapid written calculations.

The Arabic system of numerical notation is used in most parts of the world today. This system was first developed in India in the 3^{rd} century BC. The important innovation in the Arabic system was the use of positional notation, in which individual number symbols assume different values according to their position in the written numeral. Positional notation is made possible by the use of a symbol for zero. Symbol 0 makes it possible to differentiate between 11, 101, and 1,001 without the use of additional symbols, and all numbers can be expressed in terms of ten symbols. Positional notation also greatly simplifies all forms of written numerical calculation.

The system of numbers we use in everyday life is the decimal system. Decimal means "based on ten" (in this system 10+10=20).

Computers "think" in binary language. Binary means "having two parts". The binary system uses only two digits, 1 and 0, to represent all numbers (in this system 10+10=100).

Task 3. Answer the questions:

1. Are Roman numerals still used today?
2. What is the only drawback of Roman system of numerical notation?
3. Who developed the Arabic system of numerical notation?
4. How many symbols are necessary to express all numbers in the Arabic system of numerical notation?
5. Where is the binary system used?
6. How can any number be expressed in the binary system?

Task 4. Match the two parts of the sentences:

1. Roman numerals are still used today…
2. The important innovation in the Arabic system was the use of positional notation…
3. Positional notation is made possible by the use of…
4. The earliest forms of numerical notation were simply…
5. The system of number symbols created by the Romans expresses all numbers with the help of…

A … a symbol for zero.

B… groups of straight lines.

C… seven symbols.

D… more than 2,000 years after their introduction.

E… in which individual number symbols assume different values according to their position in the written numeral.

Task 5. Complete the description of the numbers using the words:

whole negligible fraction rounded up decimal place rounded down decimal places

1) 0.25=1/4 The first number is a decimal, and the second is a ____ .

2) 0.6368= 0.637 The second number is___ ____ to one ____ ____ .

3) 7.5278= 7.5 The second number is ___ ____ to one ____ ____ .

4) 8, 26, 154 These numbers aren't fractions or decimals. They're ____ numbers.

5) Error: 0.00001% The error is so small that it's ____ .

Task 6. Complete the calculations using the words in brackets. Sometimes more than one answer is possible.

divided minus plus square root subtract sum times less multiplied square squared

1) 14 + 8 = 22 Fourteen _____ eight equals twenty-two.

2) 100 x 20 = 2,000 One hundred _____ twenty is two thousand.

3) 7 x 11 = 77 Seven _____ by eleven equals seventy-seven.

4} 400 : 8 = 50 Four hundred ______ by eight equals fifty.

5) 95 + 2 = 97 The _____ of ninety-five and two is ninety-seven.

6) 8^2 = 64 The _______ of eight is sixty-four.

7) 50 – 30 = 20 If you ______ thirty from fifty, it equals twenty.

8) $\sqrt{100} = 10$ The ______ of a hundred is ten.
9) $11^2 = 121$ Eleven _____ is a hundred and twenty-one.
10) $48 – 12 = 36$ Forty-eight ______ twelve equals thirty-six.

Task 10. Write a short essay of 100 words on the topics "Fractions", "Rational numbers", "Irrational numbers". Use the information from the texts, the tasks and the Internet.

Task 11. Work in pairs. Tell your partner about numerals using information from other tasks as a tip.

Additional tasks to unit "Numerals"

Task 1. Answer the questions:
1. What is the difference between numerals and numbers?
2. What numbers are named after certain nationalities?

Task 2. Read the text below and check your answers.
Numbers should be distinguished from numerals, the symbols used to represent numbers. Boyer showed that Egyptians created the first ciphered numeral system. Greeks followed by mapping their counting numbers onto Ionian and Doric alphabets. Number five can be represented by digit "5" or by the Roman numeral "V". An important development in the history of numerals was the development of a positional system, like modern decimals, which have many advantages, such as representing very large numbers with only a few symbols. Roman numerals require extra symbols for larger numbers.

Task 3. Make a list of different numeral systems. Why are there different numeral systems? Where are they used?

Task 4. Enlarge your list if necessary after reading the text below.
The simplest numeral system is the unary numeral system, in which every natural number is represented by a corresponding number of symbols. If the symbol / is chosen, for example, then number seven would be

represented by ///////. Tally marks represent one such system still in common use. The unary system is only useful for small numbers, although it plays an important role in theoretical computer science. Elias gamma coding, which is commonly used in data compression, expresses arbitrary-sized numbers by using unary to indicate the length of a binary numeral.

The unary notation can be abbreviated by introducing different symbols for certain new values. Very commonly, these values are powers of 10; so for instance, if / stands for one, − for ten and + for 100, then number 304 can be compactly represented as +++ //// and number 123 as + − − /// without any need for zero. This is called a sign-value notation. The ancient Egyptian numeral system was of this type, and the Roman numeral system was a modification of this idea.

More useful still are systems which employ special abbreviations for repetitions of symbols; for example, using the first nine letters of the alphabet for these abbreviations, with A standing for "one occurrence", B "two occurrences", and so on, one could then write C+ D/ for number 304. This system is used when writing Chinese numerals and other East Asian numerals based on Chinese. The number system of English language is of this type ("three hundred [and] four"), as are those of other spoken languages, regardless of what written systems they have adopted. However, many languages use mixtures of bases, and other features, for instance 79 in French is *soixante dix-neuf* (60 + 10 + 9) and in Welsh is *pedwar ar bymtheg a thrigain* (4 + (5 + 10) + (3 × 20)) or (somewhat archaic) *pedwar ugain namyn un* (4 × 20 − 1). In English, one could say "four score less one", as in the famous Gettysburg Address representing "87 years ago" as "four score and seven years ago".

Task 5. Discuss the topic "Numeric systems" with the partner.

Task 6. Review what you have spoken about in the unit "Numerals". Make a list of the main points and tell the class what you know about numerals.

Additional texts to unit “Numerals”

Text 1

Numerals, or number names, are adjectives or nouns used in a language to represent numbers. They may be based on different ways of counting (decimal, vigesimal, and so on), but are a linguistic concept, rather than a mathematical one, and should therefore be distinguished from numeral systems in the mathematical sense. There is nevertheless some overlap between the concepts of numeral and numeral system, especially in ancient languages — in which mathematical notation and ordinary language were not always distinguished.

Text 2

A numeral system (or system of numeration) is a writing system for expressing numbers, that is, a mathematical notation for representing numbers of a given set, using digits or other symbols in a consistent manner. It can be seen that symbols "11" can be interpreted as the binary symbol for *three*, the decimal symbol for *eleven*, or a symbol for other numbers in different bases.

Ideally, a numeral system will:

- Represent a useful set of numbers (e.g. all integers, or rational numbers)
- Give every number represented a unique representation (or at least a standard representation)
- Reflect the algebraic and arithmetic structure of the numbers.

For example, the usual decimal representation of whole numbers gives every non-zero whole number a unique representation as a finite sequence of digits, beginning with a non-zero digit. However, when decimal representation is used for the rational or real numbers, such numbers in general have an infinite number of representations, for example 2.31 can also be written as 2.310, 2.3100000, 2.309999999... etc., all of which have the same meaning except for some scientific and other contexts where greater precision is implied by a larger number of figures shown.

Numeral systems are sometimes called *number systems*, but that name is ambiguous, as it could refer to different systems of numbers, such as the system of real numbers, the system of complex numbers, the system of *p*-adic numbers, etc.

Text 3

Positional systems in detail

In a positional base-b numeral system (with b a natural number greater than 1 known as the radix), b basic symbols (or digits) corresponding to the first b natural numbers including zero are used. To generate the rest of the numerals, the position of the symbol in the figure is used. The symbol in the last position has its own value, and as it moves to the left its value is multiplied by b.

For example, in the decimal system (base-10), the numeral 4327 means $(4\times10^3) + (3\times10^2) + (2\times10^1) + (7\times10^0)$, noting that $10^0 = 1$.

In general, if b is the base, one writes a number in the numeral system of base b by expressing it in the form $a_nb^n + a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + ... + a_0b^0$ and writing the enumerated digits $a_na_{n-1}a_{n-2} ... a_0$ in descending order. The digits are natural numbers between 0 and b − 1, inclusive.

If a text (such as this one) discusses multiple bases, and if ambiguity exists, the base (itself represented in base-10) is added in subscript to the right of the number, like this: $\text{number}_{\text{base}}$. Unless specified by context, numbers without subscript are considered to be decimal.

By using a dot to divide the digits into two groups, one can also write fractions in the positional system. For example, the base-2 numeral 10.11 denotes $1\times2^1 + 0\times2^0 + 1\times2^{-1} + 1\times2^{-2} = 2.75$.

In general, numbers in the base b system are of the form:

$$(a_na_{n-1}\cdots a_1a_0.c_1c_2c_3\cdots)_b = \sum_{k=0}^{n} a_kb^k + \sum_{k=1}^{\infty} c_kb^{-k}.$$

The numbers b^k and b^{-k} are the weights of the corresponding digits. The position k is the logarithm of the corresponding weight w, that is $k = \log_b w = \log_b b^k$. The highest used position is close to the order of magnitude of the number.

The number of tally marks required in the unary numeral system for describing the weight would have been w. In the positional system, the number of digits required to describe it is only $k + 1 = \log_b w + 1$, for $k \geq 0$. For example, to describe the weight 1000 four digits are needed because $\log_{10} 1000 + 1 = 3 + 1$. The number of digits required to describe the position is $\log_b k + 1 = \log_b \log_b w + 1$ (in positions 1, 10, 100,... only for simplicity in the decimal example).

Note that a number has a terminating or repeating expansion if and only if it is rational; this does not depend on the base. A number that terminates in one base may repeat in another (thus $0.3_{10} = 0.0100110011001..._2$). An irrational number stays aperiodic (with an infinite number of non-repeating digits) in all integral bases. Thus, for example in base-2, $\pi = 3.1415926..._{10}$ can be written as the aperiodic $11.001001000011111..._2$.

Putting overscores, n, or dots, $\dot{n}$, above the common digits is a convention used to represent repeating rational expansions.

If b = p is a prime number, one can define base-p numerals whose expansion to the left never stops; these are called the p-adic numbers.

UNIT 2. "MATHEMATICAL STATISTICS"

Terminology

accomplish	devote	observable
analysis	distinct	penetrate
application	diverse	possess
branch	draw	property
compare	error	random
conjugate	essential	require
consideration	estimate	research
constitute	general	sampling
data	inference	separation
degree	intermediate	set
denote	investigation	statistics
deprive	matter	stellar
description	object	systematization

Task 1. Find Russian equivalents to the following words:

1. possess	a. случайный
2. deprive	b. обладать
3. devote	c. существенный
4. object	d. лишать
5. random	e. посвящать
6. observable	f. значение
7. essential	g. ошибка
8. value	h. исследование
9. investigation	i. заметный
10. error	j. объект

Task 2. Read the text using additional vocabulary.

Inference – вывод, предположение
Intermediate – промежуточный, вспомогательный
Diverse – иной, отличный
Sampling – отбор образцов
Estimate – оценивать
Determination – определение
Constitute –составлять

Mathematical Statistics is the branch of Mathematics devoted to the mathematical methods for the systematization, analysis, and use of statistical data for the drawing of scientific and practical inferences. Here, the term "statistical data" denotes information about the number of objects in some more or less general set which possess certain attributes.

The statistical description of a set of objects occupies an intermediate place between the individual description of each object in the set, on the one hand, and the description of the set in terms of its general properties, which does not in any way require its separation into distinct objects, on the other. Statistical data are always deprived of individuality to a greater or lesser degree and have only limited value in those cases where individual data are essential. On the other hand, in comparison with data about the externally observable summary properties of the set, statistical data make it possible to penetrate more deeply into the nature of the matter.

The method of investigation that is based on a consideration of statistical data obtained from sets of objects is termed the statistical method. The statistical method is applicable in extremely diverse fields.

The features of the statistical method that are common to different fields include the determination of the number of objects belonging to certain groups, consideration of the distribution of quantitative attributes, application of a sampling method, and use of probability theory for estimating a sufficient number of observations to reach some conclusion. This is the formal, mathematical side of statistical methods of research, which ignores the specific nature of the objects under study and constitutes the subject of mathematical statistics.

Task 3. Answer the questions:

1. What do the features of statistical method include?
2. What does the term “statistical data” denote?
3. What are statistical data always deprived of?

Task 4. Change one word in each of the sentences below to correct them.

1) Statistical data are always decreased of individuality to a greater or lesser degree.

2) The method of comparing which is based on a consideration of statistical data is termed the statistical method.

3) Statistical data make it impossible to penetrate more deeply into the nature of the matter.

4) This is the informal, mathematical side of statistical methods of research.

5) The features of the statistical method that are unusual to different field include several items.

Task 5. Make opposites of the following words using the prefixes below:

ab- dis- im- in-(3) ir- over- un- mal-

1) correct
2) undersized
3) detected
4) normal
5) sufficient
6) regular
7) balance
8) proportionate
9) function
10) operable

Task 6. Complete the text below with the parts of sentences:

a) but the methods for studying them

b) for which it makes sense to speak of corresponding probability distributions

c) the theory of statistical testing of probabilistic hypotheses and the theory

d) both based on probability theory

Mathematical Statistics and Probability Theory

The connection between mathematical statistics and probability theory varies from one case to another. Probability theory studies not just any phenomena but only random and even "probabilistically random" phenomena, that is, those___(1). Nevertheless, probability theory also plays a definite role in the statistical study of mass phenomena of any nature, which may not belong to the category of probabilistically random phenomena. This is accomplished through the theory of sampling methods and the theory of measurement errors, ___(2). In these cases it is not the phenomena themselves that are subject to probabilistic laws ___(3).

Probability theory plays a more important role in the statistical investigation of probabilistic phenomena. Here, the branches of mathematical statistics based on probability theory are fully applicable, such as ___(4) of statistical estimation of probability distributions and associated parameters.

Task 11. Make a list of mathematical statistics methods. Work in pairs and discuss their development and application. Find the information on the net if necessary.

Task 12. Get ready to give a talk to your group. Make a plan. Speak about mathematical statistics.

Task 13. Discuss with your partner what new things you have learnt in this unit, what you would like to learn and where you can find this information.

Additional tasks to unit “Mathematical statistics”

Task 1. Answer the questions:

1. What definition can you give to the term “mathematical statistics”?
2. What terms is mathematical statistics connected with?
3. What areas of science does mathematical statistics border on?
4. What methods does this science use?
5. What is the main aim of mathematical statistics?

Task 2. Read the text below and check your answers.

Mathematical statistics is the study of statistics from a mathematical standpoint, using probability theory as well as other branches of mathematics such as linear algebra and analysis. The term "mathematical statistics" is closely related to the term "statistical theory" but also embraces modeling for actuarial science and non-statistical probability theory.

Statistics deals with gaining information from data. In practice, data often contain some randomness or uncertainty. Statistics handles such data using methods of probability theory.

Statistical science is concerned with the planning of studies, especially with the design of randomized experiments and with the planning of surveys using random sampling. The initial analysis of the data from properly randomized studies often follows the study protocol.

Of course, the data from a randomized study can be analyzed to consider secondary hypotheses or to suggest new ideas. A secondary analysis of the data from a planned study uses tools from data analysis.

Data analysis is divided into:

- descriptive statistics - the part of statistics that describes data, i.e. summarises the data and their typical properties.
- inferential statistics - the part of statistics that draws conclusions from data (using some model for the data). For example, inferential statistics involves selecting a model for the data, checking whether the data fulfill the conditions of a particular model, and with quantifying the involved uncertainty (e.g. using confidence intervals).

While the tools of data analysis work best on data from randomized studies, they are also applied to other kinds of data - for example, from natural experiments and observational studies, in which case the inference is dependent on the model chosen by the statistician, and so it is subjective.

Mathematical statistics has been inspired by and has extended many procedures in applied statistics.

Task 3. Give a title to each paragraph of the text above.

Task 4. Discuss the topic "Mathematical statistics" with your partner using the titles to the paragraphs from Task 3 as a plan.

Additional texts to unit "Mathematical Statistics"

Text 1

Probability distributions

A probability distribution assigns a probability to each measurable subset of the possible outcomes of a random experiment, survey, or procedure of statistical inference. Examples are found in experiments whose sample space is non-numerical, where the distribution would be a categorical distribution; experiments whose sample space is encoded by discrete random variables, where the distribution can be specified by a probability mass function; and experiments with sample spaces encoded by continuous random variables, where the distribution can be specified by a probability density function. More complex experiments, such as those involving stochastic processes defined in continuous time, may demand the use of more general probability measures.

A probability distribution can either be univariate or multivariate. A univariate distribution gives the probabilities of a single random variable taking on various alternative values; a multivariate distribution (a joint probability distribution) gives the probabilities of a random vector—a set of two or more random variables—taking on various combinations of values. Important and commonly encountered univariate probability distributions include the binomial distribution, the hypergeometric distribution, and the normal distribution. The multivariate normal distribution is a commonly encountered multivariate distribution.

There are some special distributions:

- normal distribution (Gaussian distribution), the most common continuous distribution;
- Bernoulli distribution, for the outcome of a single Bernoulli trial (e.g. success/failure, yes/no);
- binomial distribution, for the number of "positive occurrences" (e.g. successes, yes votes, etc.) given a fixed total number of independent occurrences;
- negative binomial distribution, for binomial-type observations but where the quantity of interest is the number of failures before a given number of successes occurs;
- geometric distribution, for binomial-type observations but where the quantity of interest is the number of failures before the first success; a special c*Discrete uniform distribution, for a finite set of values (e.g. the outcome of a fair die);
- continuous uniform distribution, for continuously distributed values;
- Poisson distribution, for the number of occurrences of a Poisson-type event in a given period of time;
- exponential distribution, for the time before the next Poisson-type event occurs;
- gamma distribution, for the time before the next k Poisson-type events occur;
- chi-squared distribution, the distribution of a sum of squared standard normal variables; useful e.g. for inference regarding the sample variance of normally distributed samples (see chi-squared test);
- student's t distribution, the distribution of the ratio of a standard normal variable and the square root of a scaled chi squared variable; useful for inference regarding the mean of normally distributed samples with unknown variance (see Student's t-test);
- beta distribution, for a single probability (real number between 0 and 1); conjugate to the Bernoulli distribution and binomial distribution.

Text 2

Statistical inferences

Statistical inference is the process of drawing conclusions from data that are subject to random variation, for example, observational errors or sampling variation. Initial requirements of such a system of procedures for inference and induction are that the system should produce reasonable answers when applied to well-defined situations and that it should be general enough to be applied across a range of situations. Inferential statistics are used to test hypotheses and make estimations using sample data. Whereas descriptive statistics describe a sample, inferential statistics infer predictions about a larger population that the sample represents.

The outcome of statistical inference may be an answer to the question "what should be done next?", where this might be a decision about making further experiments or surveys, or about drawing a conclusion before implementing some organizational or governmental policy. For the most part, statistical inference makes propositions about populations, using data drawn from the population of interest via some form of random sampling. More generally, data about a random process is obtained from its observed behavior during a finite period of time. Given a parameter or hypothesis about which one wishes to make inference, statistical inference most often uses:

- a statistical model of the random process that is supposed to generate the data, which is known when randomization has been used, and
- a particular realization of the random process; i.e., a set of data.

Text 3

Regression

In statistics, regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables – that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the regression function which can be described by a probability distribution.

Many techniques for carrying out regression analysis have been developed. Familiar methods such as linear regression and ordinary least squares regression are parametric, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. Nonparametric regression refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.

Text 4

Nonparametric statistics

Nonparametric statistics are statistics not based on parameterized families of probability distributions. They include both descriptive and inferential statistics. The typical parameters are the mean, variance, etc. Unlike parametric statistics, nonparametric statistics make no assumptions about the probability distributions of the variables being assessed.

Non-parametric methods are widely used for studying populations that take on a ranked order (such as movie reviews receiving one to four stars). The use of non-parametric methods may be necessary when data have a ranking but no clear numerical interpretation, such as when assessing preferences. In terms of levels of measurement, non-parametric methods result in "ordinal" data.

As non-parametric methods make fewer assumptions, their applicability is much wider than the corresponding parametric methods. In particular, they may be applied in situations where less is known about the application in question. Also, due to the reliance on fewer assumptions, non-parametric methods are more robust.

Another justification for the use of non-parametric methods is simplicity. In certain cases, even when the use of parametric methods is justified, non-parametric methods may be easier to use. Due both to this simplicity and to their greater robustness, non-parametric methods are seen by some statisticians as leaving less room for improper use and misunderstanding.

Text 5

Statistics, mathematics, and mathematical statistics

Mathematical statistics has substantial overlap with the discipline of statistics. Statistical theorists study and improve statistical procedures with mathematics, and statistical research often raises mathematical questions. Statistical theory relies on probability and decision theory.

Mathematicians and statisticians like Gauss, Laplace, and C. S. Peirce used decision theory with probability distributions and loss functions (or utility functions). The decision-theoretic approach to statistical inference was reinvigorated by Abraham Wald and his successors, and makes extensive use of scientific computing, analysis, and optimization; for the design of experiments, statisticians use algebra and combinatorics.

UNIT 3. “ARTIFICIAL INTELLIGENCE”

Terminology

accomplishment
approach
artificial
challenging
claim
coin
concept
converge
define
design
division
due
endow
essential
ethics
exhibit
fail
focus
goal
intelligence
issue
manipulate
neuroscience
perceive
perception
precisely
reasoning
recognize
retrieve
sapience
setback
solution
stunning
tool
tremendous

Task 1. Find Russian equivalents to the following words:

1. perceive	a. восприятие
2. coin	b. точно
3. accomplishment	c. регресс
4. perception	d. осознавать
5. converge	e. наделять
6. claim	f. достижение
7. sapience	g. результат
8. precisely	h. требовать
9. issue	i. выдумывать
10. endow	j. разумность
11. stunning	k. направляться
12. setback	l. оглушающий

Task 2. Read the text.

Artificial intelligence (AI) is human-like intelligence exhibited by machines or software. It is also an academic field of study. Major AI researchers and textbooks define the field as “the study and design of intelligent agents”, where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. John McCarthy, who coined the term in 1955, defines it as “the science and engineering of making intelligent machines”.

AI research is highly technical and specialized, and is divided into subfields that often fail to communicate with each other. Some of the division is due to social and cultural factors: subfields have grown up around particular institutions and the work of individual researches. AI research is also divided according to several technical issues. Some subfields focus on the solution of specific problems. Others focus on one of several possible approaches or on the use of a particular tool or towards the accomplishment of particular applications.

The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects. Currently popular approaches include statistical methods, computational intelligence and traditional symbolic AI. There are a large number of tools used in AI, including versions of search and mathematical optimization, logic, methods based on probability and economics, and many others. The AI field is interdisciplinary, in which a number of sciences and professions converge, including computer science, psychology, linguistics, philosophy and neuroscience, as well as other specialized fields such as artificial psychology.

The field was founded on the claim that a central property of humans, intelligence – the sapience of Homo sapiens – “can be so precisely described that a machine can be made to simulate it.” This raises philosophical issues about the nature of the mind and the ethics of creating artificial beings endowed with human-like intelligence. Artificial intelligence has been the subject of tremendous optimism but has also suffered stunning setbacks. Today it has become an essential part of the technology industry, providing lifting for many of the most challenging problems in computer science.

Task 3. Answer the questions:

a) What are the central goals of AI research?
b) Who coined the term "Artificial Intelligence"?
c) What tolls are used in AI?

Task 4. Match the terms with their definitions:

1) programming
2) machine code
3) assembly language
4) high-level language
5) Java applet
6) compiler
7) markup language

a) basic language which consists of binary codes
b) programming language such as C, Java or Visual BASIC
c) writing computer programs
d) low-level language translated into machine code by an assembler
e) software which converts a source program into machine code
f) language used to create and format documents for the Web
g) small self-contained program written in Java

Task 5. Read the text and complete the extracts below with words from this text .

Artificial Intelligence is the science that tries to recreate the human thought process and build machines that perform tasks that normally require human intelligence. It has several applications.

Androids are anthropomorphic robots designed to look and behave like a human being. Most androids can walk, talk and understand human speech. Some react to gestures and voice inflection. Some "learn" from environment: they store information and adapt their behavior according to previous experience.

Expert systems are the term given to computer software that mimics human reasoning by using a set of rules to analyze data and reach conclusions. Some expert systems help doctors diagnose illnesses based on symptoms.

Neural networks are a new concept in computer programming, designed to replicate the human ability to handle ambiguity by learning from trial and error. They use silicon neurons to imitate the functions of brain cells and usually involve a great number of processors working at the same time.

a) The term (1)_____ is defined as the automation of intelligent behavior, but can (2)______ really be intelligent.

b) (3)______ are made of units that resemble neurons. They are often used to simulate brain activity and are effective at predicting events.

c) (4)______ , also known as knowledge-based systems, mirror the structure of an expert's thought.

Task 6. Complete the article with the words

actuator, automata, computer system, end effector, joints, robot, sensors

Action Robot to Copy Human Brain

Scientists at Aberystwyth University are working on a machine which they hope will recognize objects with cameras that will work as (1)_____, and retrieve objects with an arm that will be its (2)______.

Although the arm will have (3)_____ that will link its muscles and an electric motor that will be the (4)_____, this new (5)_____ won't move like a human, i.e. it won't be like the (6)_____ of science-fiction films: forget Star Wars' C3PO. It will be desk based: no walking, or climbing stairs.

The team hopes to discover how the brain performs "multi-tasking" and to use that information to develop the (7)_____ to create a robot that can think for itself.

Task 11. Make a list of possible forms of artificial intelligence. Work in pairs and discuss their development and application. Find the information on the net if necessary.

Task 12. Get ready to give a talk to your group. Make a plan. Speak about artificial intelligence.

Task 13. Discuss with your partner what new things you have learnt in this unit, what you would like to learn and where you can find this information.

Additional tasks to unit “Artificial intelligence”

Task 1. Explain and translate the words.

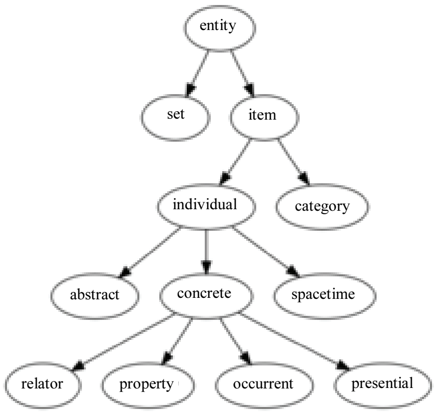
capabilities traits reasoning employ	enormous judgement conscious guess	deduction beyond embodied

Task 2. Read the text and entitle it.

The general problem of simulating (or creating) intelligence has been broken down into a number of specific sub-problems. These consist of particular traits or capabilities that researchers would like an intelligent system to display. The traits described below have received the most attention.

Early AI researchers developed algorithms that imitated the step-by-step reasoning that humans use when they solve puzzles or make logical deductions. By the late 1980s and 1990s, AI research had also developed highly successful methods for dealing with uncertain or incomplete information, employing concepts from probability and economics.

For difficult problems, most of these algorithms can require enormous computational resources – most experience a "combinatorial explosion": the amount of memory or computer time required becomes astronomical when the problem goes beyond a certain size. The search for more efficient problem-solving algorithms is a high priority for AI research.



Human beings solve most of their problems using fast, intuitive judgments rather than the conscious, step-by-step deduction that early AI research was able to model. AI has made some progress at imitating this kind of "sub-symbolic" problem solving: embodied agent approaches emphasize the importance of sensorimotor skills to higher reasoning; neural net research attempts to simulate the structures inside the brain that give rise to this skill; statistical approaches to AI mimic the probabilistic nature of the human ability to guess.

Task 3. What examples of AI can you give? (Do not repeat the information of the texts.)

Task 4. Write a question-plan to the text above. Work in pairs. Compare your plans.

Task 5. Retell the text using your plan.

Additional texts to unit "Artificial intelligence"

Text 1.

Knowledge representation and knowledge engineering are central to AI research. Many of the problems machines are expected to solve will require extensive knowledge about the world. Among the things that AI needs to represent are: objects, properties, categories and relations between objects; situations, events, states and time; causes and effects; knowledge about knowledge (what we know about what other people know); and many other, less well researched domains. A representation of "what exists" is an ontology: the set of objects, relations, concepts and so on that the machine knows about. The most general are called upper ontologies, which attempt to provide a foundation for all other knowledge.

There are many difficult problems in knowledge representation

The first is default reasoning and the qualification problem.

Many of the things people know take the form of "working assumptions." For example, if a bird comes up in conversation, people typically picture an animal that is fist sized, sings, and flies. None of these things are true about all birds. John McCarthy identified this problem in 1969 as the qualification problem: for any commonsense rule that AI researchers care to represent, there tend to be a huge number of exceptions. Almost nothing is simply true or false in the way that abstract logic requires. AI research has explored a number of solutions to this problem.

The second problem is the breadth of commonsense knowledge.

The number of atomic facts that the average person knows is astronomical. Research projects that attempt to build a complete knowledge base of commonsense knowledge (e.g., Cyc) require enormous amounts of laborious ontological engineering — they must be built, by hand, one complicated concept at a time. A major goal is to have the computer understand enough concepts to be able to learn by reading from sources like the internet, and thus be able to add to its own ontology.

The third problem is the subsymbolic form of some commonsense knowledge.

Much of what people know is not represented as "facts" or "statements" that they could express verbally. For example, a chess master will avoid a particular chess position because it "feels too exposed" or an art critic can take one look at a statue and instantly realize that it is a fake. These are intuitions or tendencies that are represented in the brain non-consciously and sub-symbolically. Knowledge like this informs, supports and provides a context for symbolic, conscious knowledge. As with the related problem of sub-symbolic reasoning, it is hoped that situated AI, computational intelligence, or statistical AI will provide ways to represent this kind of knowledge.

Text 2.

Robots have five basic components: a movable structure, a motor, a power source, a sensory system, and a processor. The entire robot may move on legs or only one part may move, such as the arm of an industrial robot.

The motor provides the physical power to move the structure. It may be electric, pneumatic, or some form of heat engine. All motors require a source of power. In the case of mobile robots, the usual source is a battery. The problem with batteries is that they are heavy and run down quite quickly. In future there may be robots which use biological fuel which they collect as they move. Compressed air, in tanks for mobile robots or directly from a compressor for fixed robots, is the power source for pneumatic systems.

In the same way that humans depend on sight, hearing, taste, smell, and touch to make sense of the world, robots require a sensory system in order to function. Sensors feed information to the processor. The information provided depends on the function of the robot. Location is important for most robots. Industrial robots must be capable of placing items or performing actions in exactly the right place. With some robots, location is controlled by placing electronic tracks for the robot to follow. Container handling in ports can be done by robotic vehicles following such tracks.

Heat sensors may be important for robots working in extreme conditions. Sensors which measure the pressure exerted by robot arms or pincers are important for robots which pick up or handle delicate items. For robots which walk or climb stairs, information on weight distribution and balance is important. Robots which look for some type of explosive need sensors which can detect chemical smells. Robots which have to navigate over unfamiliar ground, such as the Mars Rover, have digital cameras to help them identify obstacles and select navigable routes.

The brain of a robot is the processor. It controls the operation of the robot. It is programmed to allow the robot to carry out a series of actions and to respond to feedback from the sensory system. In the case of a simple robot, such as a domestic vacuum cleaner, the program may instruct the robot to turn 90 degrees when it collides with an obstacle.

UNIT 4. "FINITE ELEMENT METHOD"

Terminology

accuracy
bend
boundary
BVP
calculus
curve
derivative
displacement
domain
finite
fluid dynamics
grid
linear
node
partial
PDE
point
rectangular
residual
stiffness
transient
trial
twist
value
weight

In this unit you will need to learn and remember the following words:

drawback
implementation
tip
restrict
alteration
handle
thereof
share
aeronautical
substantial
tremendously
amount
inner
arbitrary
average
inclusion
heat
opposite
subdivision
stable
property
capture
initial
complex
commonly
in simple terms
to fit
eliminate steady
ordinary
range
graph
both …and …
appropriate

similarly	divide	procedure
exact	to project	devise
entire	recombine	thus

Task 1. Match the words and their definitions:

1 residual	A something that has developed or been obtained from something else
2 PDE	B a way to multiply vectors together, with the result of this multiplication being a scalar
3 approximation	C partial differential equations
4 derivatives	D the place where lines cross or meet, for example on a graph
5 node	E existing, happening, or staying somewhere for a short period of time only
6 transient	F a nearly exact calculation, amount, number, time etc
7 inner product	G something that is added or the action of adding
8 weight function	H the sum of deviations from a best-fit curve of arbitrary form
9 inclusion	I a mathematical device used when performing a sum, integral, or average to give some elements more influence on the result than other elements in the same set

Task 2. Translate the following:

1. The heat equation is expressed in either PDE or integral equations.
2. We have concrete formulae for a large but finite-dimensional linear problem whose solution will approximately solve the original BVP.

Task 3. Discuss with your partner what you know about finite element method. Answer the questions:

1. What is the meaning of the word “finite”? What is the opposite of it?
2. What methods are there to solve mathematical tasks?
3. Why are different techniques used for different problems?
4. Should a mathematician be able to use all the methods known? Why?
5. What areas of engineering is finite element method used in?

Task 4. Read the text and answer the questions to the text:

1. What is the idea of the finite element method?
2. Why is it necessary to divide the main domain into several parts?
3. What does the FEM allow one to achieve?

Finite Element Method

A. In mathematics, the finite element method (FEM) is a numerical technique for finding approximate solutions to boundary value problems for differential equations. It uses variational methods (the calculus of variations) to minimize an error function and produce a stable solution.

B. The subdivision of a whole domain into simpler parts has several advantages: accurate representation of complex geometry, inclusion of dissimilar material properties, easy representation of the total solution, and capture of local effects.

C. A typical work out of the method involves (step 1) dividing the domain of the problem into a collection of subdomains. Each subdomain is represented by a set of element equations to the original problem, followed by (step 2) systematically recombining all sets of element equations into a global system of equations for the final calculation. The global system of equations can be calculated from the initial values of the original problem to obtain a numerical answer.

D. In the first step above, the element equations are simple equations that locally approximate the original complex equations to be studied. The original equations are often partial differential equations (PDE). FEM is commonly introduced as a special case of Galerkin method. The process is to construct an integral of the inner product of the residual and the weight functions and set the integral to zero. In simple terms, it is a procedure that minimizes the error of approximation by fitting trial functions into the PDE. The residual is the error caused by the trial functions, and the weight functions are polynomial approximation functions that project the residual. The process eliminates all the spatial derivatives from the PDE, thus approximating the PDE locally with a set of algebraic equations for steady state problems and a set of ordinary differential equations for transient problems.

E. In step (2) above, a global system of equations is generated from the element equations through a transformation of coordinates from the subdomains' local nodes to the domain's global nodes.

Task 4. Put the subtitles A – E to the text above in the correct order.

1. The idea of the second step.
2. The advantages of the method.
3. The definition of the method.
4. The idea of the first step.
5. The structure of the method.

Task 5. Decide if the following statements are true or false.

1. To use the finite elements method it is necessary to define several ranges of possible values of a variable.
2. The FEM is an engineering application for solving designer problems.
3. The result of using FEM is a graph.
4. The FEM uses both algebraic and differential equations.
5. There are five steps in the FEM.
6. The finite element method allows finding accurate solutions.
7. The FEM is considered as one of the variants of some other method.

Task 6. Complete the sentences in an appropriate way.

1. The finite element method is one of ...
2. The pros of dividing a subdomain into several parts are as follows: ...
3. The FEM is a procedure that ...
4. The transformation of ... is done at the second step of the FEM.
5. The error is minimized by ...
6. The initial values of the original problem are required to ...
7. Final calculation is done on the basis of ...
8. In order to study original complex equations ...

Task 7. Discuss these questions with your partner:

1. What types of equations are there?
2. When was the finite element method devised?
3. Have you ever used the FEM? How often if ever? What sort of problem was it?
4. What is the residual?
5. What are the drawbacks of the finite element method (FEM)?
6. What functions are necessary for the FEM implementation?
7. What are the names of mathematicians who did a lot for the FEM development?

Task 8. Study the table below and compare the two methods.

The finite difference method (FDM) is an alternative way of approximating solutions of PDEs. The differences/similarities between FEM and FDM are:

FDM	FEM
1. is restricted to handle rectangular shapes and simple alterations thereof 2. can be very easy to implement 3. Poisson's equation 4. poor approximation between grid points 5. the method is used in computational fluid dynamics	1. ability to handle complicated geometries (and boundaries) with relative ease 2. requires complex calculations 3. Poisson's equation 4. high quality of a FEM approximation 5. the method of choice in all types of analysis in structural mechanics

Task 9. Read the text below. Make lists of:

1. branches of industry where FEM is used;
2. advantages of this method.

Share your conclusions with your partner.

A variety of specializations of the mechanical engineering discipline such as aeronautical, biomechanical, and automotive industries commonly use integrated FEM in design and development of their products. In a structural simulation, FEM helps tremendously in producing stiffness and strength visualizations and also in minimizing weight, materials, and costs.

FEM allows detailed visualization of where structures bend or twist, and indicates the distribution of stresses and displacements. FEM software provides a wide range of simulation options for controlling the complexity of both modeling and analysis of a system. Similarly, the desired level of accuracy and computational time requirements can be managed simultaneously. FEM allows entire designs to be constructed, refined, and optimized before the design is manufactured.

This powerful design tool has significantly improved both the standard of engineering designs and the methodology of the design process in many industrial applications. The introduction of FEM has substantially decreased the time to take products from concept to the production line.

Task 10. Write a short essay of 100 words on the topic "Mathematical methods". Use the information from the texts, the table and the tasks above.

Task 11. Work in pairs. Tell your partner about the FEM using Task 4 and other tasks as a tip.

Additional tasks to unit "Finite Element Method"

Task 1. Read the text below. What can the word "element" in the term "finite element method" mean?

Depending on the author, the word "element" in "finite element method" refers either to the triangles in the domain, the piecewise linear basis function, or both. So for instance, an author interested in curved domains might replace the triangles with curved primitives, and so might describe the elements as being curvilinear. On the other hand, some authors replace "piecewise linear" by "piecewise quadratic" or even "piecewise polynomial". The author might then say "higher order element" instead of "higher degree polynomial". Finite element method is not restricted to triangles (or tetrahedra in 3-d, or higher order simplexes in multidimensional spaces), but can be defined on quadrilateral subdomains (hexahedra, prisms, or pyramids in 3-d, and so on). Higher order shapes (curvilinear elements) can be defined with polynomial and even non-polynomial shapes (e.g. ellipse or circle).

Examples of methods that use higher degree piecewise polynomial basis functions are the hp-FEM and spectral FEM.

More advanced implementations (adaptive finite element methods) utilize a method to assess the quality of the results (based on error estimation theory) and modify the mesh during the solution aiming to achieve approximate solution within some bounds from the 'exact' solution of the continuum problem. Mesh adaptivity may utilize various techniques, the most popular are:

- moving nodes (r-adaptivity)
- refining (and unrefining) elements (h-adaptivity)
- changing order of base functions (p-adaptivity)
- combinations of the above (hp-adaptivity).

Task 2. Match the words 1–5 to their definitions.

1. mesh	E. a regularly spaced array of points in a square array, i.e., points with coordinates (m, n,...)
2. polinomial	A. a mathematical expression involving a sum of powers in one or more variables multiplied by coefficients
3. curvilinear	D. having a graduate smooth bend
4. quadrilatteral	B. having four sides like a square
5. piecewise	C. composed of some number of segments defined over an equal number of intervals, usually of equal size

Task 3. Give the definitions of the words: triangle, domain, primitive, higher order element, hexahedron, prism, pyramid, circle, bound, adaptivity.

Task 4. Work with a partner to complete the word families in the table below.

Verb	Noun	Adjective
…	adaptivity	…
utilize	…	…
…	…	popular
refer	…	…
…	…	restricted
…	…	defined
modify	…	…
…	…	approximate
…	bound	…
…	solution	…
depend	…	…

Task 5. Compose your own sentences using the words from the table.

Task 6. Ask the questions using the words below.

1. popular are what the techniques most?
2. refer the can word to "element" what?

3. to method the is element is not, triangles restricted finite it?
4. adaptivity what utilize mesh does?
5. be with high what order defined can shapes?
6. functions of piecewise what some basis higher methods use degree examples polynomial are that?

Task 7. Work with the partner to answer the questions in Task 6 and discuss the topic.

Task 8. Write an essay on the topic "Finite Element Method". Use your knowledge of mathematics and information offered in the unit "FEM".

Additional texts to unit "Finite Element Method"

Text 1

Finite-Element (FEM) Simulation

Multi-physics finite element simulation focuses on structural, electromagnetic and thermal simulation and can represent objects accordingly.

The technology

FEM simulation is a means of predicting (and optimising) the behaviour of various complex objects or systems that are often connected - without having to rely on physically existing models, prototypes or measurements. It involves numerically dividing the system being analysed into a - often very large - number of small (finite) volumes, calculating the physical states (stresses, field strengths, temperatures, etc.) of each individual cell and using an iterative process to approximate the neighbouring cells until a physically practical solution is obtained for the overall system. FE methods can be used to examine problems and potential solutions in various physical areas. In recent years the FE method in structural mechanics in particular has become a routine tool in solid-state simulation and is part of the standard repertoire for designing mechanical components. It is also possible for example to simulate temperature distribution and electric or magnetic fields or several factors at the same time with multi-parameter simulation. The R&D focus at CTR is on using linked multi-parameter simulation with optimised model approaches, e.g. for simulating the thermal-mechanical behaviour of composite materials or crystalline anisotropic high-performance functional materials. Our skills portfolio naturally also includes planning, performing and - depending on requirements - validating classic simulation, for example in the area of structural mechanics.

Uses and benefits

The key advantage and benefit of using FE simulation methods is the ability to visualise the distribution of various properties and based on it gain a detailed understanding of the behaviour and connections between various factors affecting a system. This enables us to identify a system's most important design factors and (potential) error sources and make the necessary improvements – without time-consuming mass screening of models or similar.

Application

The services provided by CTR cover the entire technology chain in FEM simulation:

- Problem definition
- Creation or adaptation of geometry models
- Selection, adjustment and development if required of (special) simulation models
- Problem-specific visualisation of the results
- Evaluation
- Metrology validation

Given the diverse opportunities available with finite-element methods and the capabilities of the software package used, the range of applications is very wide. Reference projects include for example:

- Structural mechanics: simulation of stress distribution in a combined solid metal and profiled sheet part; the development goal was to minimise the component mass while ensuring mechanical strength over its entire design life, also in demanding mechanical (dynamic loads) and thermal applications. In the picture top right: stress distribution in the part (basic design) under nominal mechanical load, superimposed with a temperature gradient of +10 --> -60°C
- System simulation of PV modules: simulation of stress distribution and the resultant deformation (elastic and plastic) in photovoltaic modules as a function of module design (composite materials), general properties and

installation devices for various possible load cases (snow, wind, hail, etc.) supported with metrology validation. In the picture below: Deformation simulation of a photovoltaic module (quarter module) with increasing snow load

(http://www.ctr.at/en/r-d-technologies/simulation/finite-element-fem-simulation.html)

Text 2

Reservoir Simulation with a Control-Volume Finite-Element Method

Document Type: Journal Paper

Authors: L.S.-K. Fung (Computer Modelling Group) | A.D. Hiebert (Computer Modelling Group) | L.X. Nghiem (Computer Modelling Group)

Source: SPE Reservoir Engineering; Publisher Society of Petroleum Engineers; http://dx.doi.org/10.2118/21224-PA

Summary

This paper describes a control-volume finite-element (CVFE) method incorporating linear triangular elements for the simulation of thermal multiphase flow in porous media. The technique adopts the usual finite-element shape functions to evaluate flow potentials at the control-volume boundaries and uses the conservation equations for each control volume. The main advantage of the CVFE method over the finite-difference method is in the representation of complex reservoir geometries. In addition, desirable features, such as local grid refinement for near-well resolution, can be achieved simply and consistently. The control-volume approach enforces local mass conservation and permits a direct physical interpretation of the resulting discrete equations. These are significant advantages over the classical Petrov-Galerkin or variational finite-element methods. The method was implemented in a general-purpose thermal simulator. Numerical examples compare the proposed method with five-point and nine-point finite-difference schemes in terms of grid-orientation effects and run time. The CVFE method was found to reduce grid-orientation effects significantly. At the same time, computational cost was much lower than for the nine-point scheme. The geometric flexibility of the method also is demonstrated.

Introduction

In many reservoir simulation problems, a flexible discretization method is extremely useful in the definition of complex reservoir geometries and discontinuities (such as faults) and in enhancing the resolution near the wells. The use of Cartesian grids with finite-difference methods has created difficulties and/or complexities in the definition of complex geometries or grid refinements. It is desirable to adopt the intrinsic grid flexibility of the finite-element method. However, combining upstream weighting with the usual finite-element method for the multiphase multidimensional flow problem presents difficulties. Although asymmetric weighting procedures like the Petrov-Galerkin method have been introduced to deal with the convective terms in the mixed convective-diffusive flow problems, such methods are in general not mass-conservative in the local sense. On the other hand, local mass conservation is a specific requirement of the control-volume methods. In addition, reservoir simulation problems can be very complex, involving multiphase mass and heat flow with interphase transfers and chemical reactions. The mass-conservative aspect of the control-volume methods is a distinct advantage in the programming and testing of these simulators.

The CVFE method was proposed in computational fluid dynamics for solving the Navier-Stokes equations, where flexible gridding and local mass, momentum, and energy conservation are achieved. In this paper, a CVFE procedure for the reservoir flow equations is developed where flexible grid geometry is obtained without sacrificing the advantageous attributes of the control-volume finite-difference method. The derivation shows that the use of the perpendicular-bisection grid and the seven-point finite-difference method are special cases of this discretization method. Recently, Forsyth applied a CVFE method to the local-mesh-refinement problem by providing a smooth transition between the coarse and fine grids. As discussed in detail later, a proper choice of the triangular finite-element mesh is crucial to the reduction of grid-orientation effects. The construction of a CVFE grid by triangulation with one of the diagonals of each rectangle in a Cartesian grid will result in a five-point discretization scheme because the diagonal flow terms for this grid are identically equal to zero.

The method results in a set of discretized conservative equations where the Jacobian construction for Newton's method and the upstream weighting of mobilities can be done in the usual way. For the incompressible single-phase flow problem, the method gives the same stiffness matrix as the Petrov-Galerkin weighted-residual finite-element method when linear shape functions are used. The criterion for maintaining positive transmissibility coefficients of a general anisotropic system also is derived. A number of examples are included to demonstrate the geometric flexibility, non-grid-orientation characteristics, and efficiency of the proposed method.

(https://www.onepetro.org/journal-paper/SPE-21224-PA)

Text 3

General form of the finite element method

In general, the finite element method is characterized by the following process.

- One chooses a grid for Ω. In the preceding treatment, the grid consisted of triangles, but one can also use squares or curvilinear polygons.
- Then, one chooses basis functions. In our discussion, we used piecewise linear basis functions, but it is also common to use piecewise polynomial basis functions.

A separate consideration is the smoothness of the basis functions. For second order elliptic boundary value problems, piecewise polynomial basis function that are merely continuous suffice (i.e., the derivatives are discontinuous.) For higher order partial differential equations, one must use smoother basis functions. For instance, for a fourth order problem such as $u_{xxxx} + u_{yyyy} = f$, one may use piecewise quadratic basis functions that are C^1.

Another consideration is the relation of the finite-dimensional space V to its infinite-dimensional counterpart, in the examples above H_0^1. A conforming element method is one in which the space V is a subspace of the element space for the continuous problem. The example above is such a method. If this condition is not satisfied, we obtain a nonconforming element method, an example of which is the space of piecewise linear functions over the mesh which are continuous at each edge midpoint. Since these functions are in general discontinuous along the edges, this finite-dimensional space is not a subspace of the original H_0^1.

Typically, one has an algorithm for taking a given mesh and subdividing it. If the main method for increasing precision is to subdivide the mesh, one has an h-method (h is customarily the diameter of the largest element in the mesh.) In this manner, if one shows that the error with a grid h is bounded above by Ch^p, for some $C < \infty$ and $p > 0$, then one has an order p method. Under certain hypotheses (for instance, if the domain is convex), a piecewise polynomial of order d method will have an error of order $p = d + 1$.

If instead of making h smaller, one increases the degree of the polynomials used in the basis function, one has a p-method. If one combines these two refinement types, one obtains an hp-method (hp-FEM). In the hp-FEM, the polynomial degrees can vary from element to element. High order methods with large uniform p are called spectral finite element methods (SFEM). These are not to be confused with spectral methods.

For vector partial differential equations, the basis functions may take values in $\mathbb{R}^n$.

Text 4

There are various types of finite element methods.

The Applied Element Method, or AEM combines features of both FEM and Discrete element method, or (DEM).

The Generalized Finite Element Method (GFEM) uses local spaces consisting of functions, not necessarily polynomials, that reflect the available information on the unknown solution and thus ensure good local approximation. Then a partition of unity is used to "bond" these spaces together to form the approximating subspace. The effectiveness of GFEM has been shown when applied to problems with domains having complicated boundaries, problems with micro-scales, and problems with boundary layers.

The hp-FEM combines adaptively, elements with variable size h and polynomial degree p in order to achieve exceptionally fast, exponential convergence rates.

The hpk-FEM combines adaptively, elements with variable size h, polynomial degree of the local approximations p and global differentiability of the local approximations (k-1) in order to achieve best convergence rates.

Other types of FEM are: extended finite element method (XFEM), smoothed finite element method (S-FEM), fiber beam element method, spectral method, meshfree method, discontinuous Galerkin method, finite element limit analysis, stretched grid method.

UNIT 5. “COMPUTER ARCHITECTURE”

Terminology

cache
cluster
compiler
duplicate
emulator
estimate
multicore
order
path
performance
sequence
storage
transfer mode
value
virtualization

In this unit you will need to learn and remember the following words:

abbreviation
above
access
affect
applicable
appropriate
attempt
belong
collaborate
concern
construct
consumption
consumption
convenience
deal with
define
describe
efficiency
estimate
expense
fashionable
for instance
gain
government
have nothing to do with
implement
improvement
include
intend
interrelated
issue
measurement
portable
power
propose
reach
save
set
since
suggest
table
the least
through
while

Task 1. Discuss with the partner. What do you know about:

1. the main task of the computer architecture;
2. the main elements of computer architecture;
3. history of computer architecture;
4. the main terms of this field.

Task 2. Fill in the gaps completing the sentences. Use the following words:

A *cache* B *estimates* C *collaborate* D *emulator* E *expense* F *consumption* G *compilers* H *implement* I *cluster* J *performance*

1. The purpose is to design a computer that maximizes ....
2. Optimization of the design requires familiarity with ....
3. An ... is hardware or software or both that duplicates the functions of one computer system in another computer system.
4. Power ... is another measurement that is important in modern computers.
5. The components of a ... are usually connected to each other through fast local area networks.
6. Data is transferred between memory and ... in blocks of fixed size.
7. To ... a system successfully, a large number of inter-related tasks need to be carried out in an appropriate sequence.
8. MIPS per MHz (millions of instructions per millions of cycles of clock speed) ... the efficiency of the architecture at any clock speed.
9. Software engineers need to optimize software in order to gain the most performance at the least ....
10. Computer architects have to ... with experts in different areas.

Task 3. Form different parts of speech for the following words. Use all the words in sentences of your own.

compile, implement, compute, collaborate, architect, fashion, perform, instruct, emulate, suggest, measure, organize, process, consume, define, specify, include

Task 1. Give a heading to the text.

In computer science and engineering, **computer architecture** is a set of disciplines that describes a computer system by specifying its parts and their relations.

For example, at a high level, computer architecture may be concerned with how the central processing unit (CPU) acts and how it uses computer memory. Some fashionable (2011) computer architectures include cluster computing and Non-Uniform Memory Access.

Computer architects use computers to design new computers. Emulation software can run programs written in a proposed instruction set. While the design is very easy to change at this stage, compiler designers often collaborate with the architects, suggesting improvements in the instruction set. Modern emulators may measure time in clock cycles: estimate energy consumption in joules, and give realistic estimates of code size in bytes. These affect the convenience of the user, the life of a battery, and the size and expense of the computer's largest physical part: its memory. That is, they help to estimate the value of a computer design.

The discipline of computer architecture has three main subcategories:

- Instruction set architecture, or ISA. The ISA defines the codes that a central processor reads and acts upon. It is the machine language (or assembly language), including the instruction set, word size, memory address modes, processor registers, and address and data formats.
- Microarchitecture, also known as Computer organization describes the data paths, data processing elements and data storage elements, and describes how they should implement the ISA. The size of a computer's CPU cache for instance, is an organizational issue that generally has nothing to do with the ISA.
- System Design includes all of the other hardware components within a computing system.

Task 4. Discuss the following questions with your partner:

1. How is time measured by modern emulators?
2. What field of computer architecture deals with data paths?
3. Which of the subcategories above do date paths, computer buses, memory controllers, virtualization and multiprocessing belong to?
4. What abbreviations are used in computer design? What do they mean?

Task 4. Match the words to make phrases. More than one variant is possible.

1 cluster	designer
2 compiler	computing
3 storage	elements
4 act	consumption
5 be concerned	set
6 clock	upon
7 energy	to do with
8 instruction	estimates
9 give	cycles
10 nothing	with

Task 5. Read the text again and find the following information:

a. what influences the convenience of a user;

b. subcategories of the area of computer architecture;

c. what emulators and set of instructions are intended for;

d. what describes the data paths, data processing elements and data storage elements; what are the two terms for it;

e. what kind of experts work at the computer design;

f. what affects the size of the computer memory.

Task 6. Complete the table with the year, the name of the computer and the name of its inventor.

Years: 1949, 1943, 1834, 1981, 1946, 1964, 1952, 1964, 1981.

The name of the computer: ENIAC, COLOSSUS, the first mechanical computer, EDSAC, IAS, S/360, CDC 6600, IBM PC, Osborne-1.

The name of the inventor: M. V. Wilkes, J.von Neumann, British government, C. Babbage, J. P.Eckert/ J. W. Mauchly, IBM, CDC, IBM, Osborne.

Year	The name of the computer	The name of the inventor or ordering customer	The device invented
			The first attempt to construct a digital computer
			The first electronic computer
			The machine from which the history of the computing began
			The first computer with the programms saved in the memory
			The project being used now in all modern computers
			The first computer family
			The first supercomputer for scientific calculations
			The device from which the era of modern computers began
			The first portable computer

Task 7. Work in pairs. Discuss the main steps of computer history using the completed table above.

Task 8. Give headings to each paragraph of the following text.

Modern computer performance is often described in MIPS per MHz (millions of instructions per millions of cycles of clock speed). This measures the efficiency of the architecture at any clock speed. Since a faster clock can make a faster computer, this is a useful, widely applicable measurement. Computers of the past had MIPs/MHz as low as 0.1. Simple modern processors easily reach near 1. Superscalar processors may reach three to five by executing several instructions per clock cycle. Multicore and vector processing CPUs can multiply this further by acting on a lot of data per instruction, which have several CPUs executing in parallel.

Counting machine language instructions would be misleading because they can do varying amounts of work in different ISAs. The "instruction" in the standard measurements is not a count of the ISA's actual machine language instructions, but a historical unit of measurement, usually based on the speed of the VAX computer architecture.

Task 9. Make a list of the abbreviations used in this unit and explain them.

Task 10. What are the tasks and difficulties of the discipline of computer architecture? Share your vision of the problem with the class.

Task 11. What other aspects of this area of computer engineering can you think of? Share your ideas with your group mates and make notes.

Task 12. Write a short essay of 100 words on the topic "Computer architecture". Use the information from the texts, the table and the notes.

Additional tasks to unit "Computer Architecture"

Task 1. Give the heading to the text.

An instruction set architecture (ISA) is the interface between the computer's software and hardware and also can be viewed as the programmer's view of the machine. Computers do not understand high level languages which have few, if any, language elements that translate directly into a machine's native opcodes. A processor only understands instructions encoded in some numerical fashion, usually as binary numbers. Software tools, such as compilers, translate high level languages, such as C into instructions.

Besides instructions, the ISA defines items in the computer that are available to a program—e.g. data types, registers, addressing modes, and memory. Instructions locate operands with Register indexes (or names) and memory addressing modes.

The ISA of a computer is usually described in a small book or pamphlet, which describes how the instructions are encoded. Also, it may define short (vaguely) mnenonic names for the instructions. The names can be recognized by a software development tool called an assembler. An assembler is a computer program that translates a human-readable form of the ISA into a computer-readable form. Disassemblers are also widely available, usually in debuggers, software programs to isolate and correct malfunctions in binary computer programs.

ISAs vary in quality and completeness. A good ISA compromises between programmer convenience (more operations can be better), cost of the computer to interpret the instructions (cheaper is better), speed of the computer (faster is better), and size of the code (smaller is better). For example, a single-instruction ISA is possible, inexpensive, and fast, but it was not convenient or helpful to make programs small. Memory organization defines how instructions interact with the memory, and also how different parts of memory interact with each other.

Task 2. Give your definitions to the following words:

mnemonic, opcode, disassembler, convenience, verify, computer-readable, arrange, data path, encode, require, implementation

Task 3. Define if the statements are true or false.

1. Assemblers, dissemblers and debuggers are parts of hardware.
2. High level languages are translated into instructions by special software tools.
3. Processors read instructions written in English.
4. Computers can understand any language.
5. Interaction between memory and instructions is controlled by memory organization.
6. CPU speed depends on computer architecture.
7. Only one feature defines the quality of ISA.
8. ISAs differ from each other in quantity and form.

Task 4. Correct the wrong statements in Task 3.

Task 5. Read the following explanation of the term "implementation". What stages of implementation are there?

Once an instruction set and micro-architecture are described, a practical machine must be designed. This design process is called the implementation. Implementation is usually not considered architectural definition, but rather hardware design engineering.

Task 6. Match the parts of statements 1-4 and A-D to formulate the logical steps of implementation.

1. Logic Implementation 2. Circuit Implementation 3. Physical Implementation 4. Design Validation	A. tests the computer as a whole to see if it works in all situations and all timings. Once implementation starts, the first design validations are simulations using logic emulators. However, this is usually too slow to run realistic programs. So, after making corrections, prototypes are constructed using Field-Programmable Gate-Arrays (FPGAs). Many hobby projects stop at this stage. The final step is to test prototype integrated circuits. Integrated circuits may require several redesigns to fix problems. B. does transistor-level designs of basic elements (gates, multiplexers, latches etc.) as well as of some larger blocks (ALUs, caches etc.) that may be implemented at this level, or even (partly) at the physical level, for performance reasons. C. draws physical circuits. The different circuit components are placed in a chip floorplan or on a board and the wires connecting them are routed. D. designs the blocks defined in the micro-architecture at (primarily) the register-transfer level and logic gate level.

Task 7. Entitle each paragraph of the text above.

Task 8. Discuss the topic "Computer architecture" in class.

Additional texts to unit “Computer Architecture”

Text 1

Computer organization

Computer organization helps optimize performance-based products. For example, software engineers need to know the processing ability of processors. They may need to optimize software in order to gain the most performance at the least expense. This can require quite detailed analysis of the computer organization. For example, in a multimedia decoder, the designers might need to arrange for most data to be processed in the fastest data path and the various components are assumed to be in place and task is to investigate the organisational structure to verify the computer parts operates.

Computer organization also helps plan the selection of a processor for a particular project. Multimedia projects may need very rapid data access, while supervisory software may need fast interrupts. Sometimes certain tasks need additional components as well. For example, a computer capable of virtualization needs virtual memory hardware so that the memory of different simulated computers can be kept separated. Computer organization and features also affect power consumption and processor cost.

The exact form of a computer system depends on the constraints and goals. Computer architectures usually trade off standards, power versus performance, cost, memory capacity, latency (latency is the amount of time that it takes for information from one node to travel to the source) and throughput. Sometimes other considerations, such as features, size, weight, reliability, and expandability are also factors.

The most common scheme does an in-depth power analysis and figures out how to keep power consumption low, while maintaining adequate performance.

Text 2

Performance

Modern computer performance is often described in MIPS per MHz (millions of instructions per millions of cycles of clock speed). This measures the efficiency of the architecture at any clock speed. Since a faster clock can make a faster computer, this is a useful, widely applicable measurement. Historic computers had MIPs/MHz as low as 0.1 instructions per second. Simple modern processors easily reach near 1. Superscalar processors may reach three to five by executing several instructions per clock cycle. Multicore and vector processing CPUs can multiply this further by acting on a lot of data per instruction, which have several CPUs executing in parallel.

Counting machine language instructions would be misleading because they can do varying amounts of work in different ISAs. The "instruction" in the standard measurements is not a count of the ISA's actual machine language instructions, but a historical unit of measurement, usually based on the speed of the VAX computer architecture.

Historically, many people measured a computer's speed by the clock rate (usually in MHz or GHz). This refers to the cycles per second of the main clock of the CPU. However, this metric is somewhat misleading, as a machine with a higher clock rate may not necessarily have higher performance. As a result manufacturers have moved away from clock speed as a measure of performance.

Other factors influence speed, such as the mix of functional units, bus speeds, available memory, and the type and order of instructions in the programs being run.

In a typical home computer, the simplest, most reliable way to speed performance is usually to add random access memory (RAM). More RAM increases the likelihood that needed data or a program is in RAM—so the system is less likely to need to move memory data from the disk. The disk is often ten thousand times slower than RAM because it has mechanical parts that must move to access its data.

There are two main types of speed, latency and throughput. Latency is the time between the start of a process and its completion. Throughput is the amount of work done per unit time. Interrupt latency is the guaranteed maximum response time of the system to an electronic event (*e.g.* when the disk drive finishes moving some data).

Performance is affected by a very wide range of design choices — for example, pipelining a processor usually makes latency worse (slower) but makes throughput better. Computers that control machinery usually need low interrupt latencies. These computers operate in a real-time environment and fail if an operation is not completed in a specified amount of time. For example, computer-controlled anti-lock brakes must begin braking within a predictable short time after the brake pedal is sensed.

The performance of a computer can be measured using other metrics, depending upon its application domain. A system may be CPU bound (as in numerical calculation), I/O bound (as in a webserving application) or memory bound (as in video editing). Power consumption has become important in servers and portable devices like laptops.

Benchmarking tries to take all these factors into account by measuring the time a computer takes to run through a series of test programs. Although benchmarking shows strengths, it may not help one to choose a computer. Often the measured machines split on different measures. For example, one system might handle scientific applications quickly, while another might play popular video games more smoothly. Furthermore, designers may add special features to their products, in hardware or software, which permit a specific benchmark to execute quickly but don't offer similar advantages to general tasks.

Text 3
Von Neumann architecture

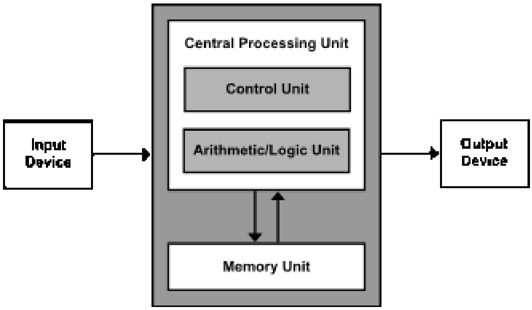


Fig. 1. Von Neumann architecture scheme

The Von Neumann architecture, also known as the Von Neumann model and Princeton architecture, is a computer architecture based on that described in 1945 by the mathematician and physicist John von Neumann and others in the First Draft of a Report on the EDVAC. This describes a design architecture for an electronic digital computer with parts consisting of a processing unit containing an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, a memory to store both data and instructions, external mass storage, and input and output mechanisms. The meaning has evolved to be any stored-program computer in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus. This is referred to as the Von Neumann bottleneck and often limits the performance of the system.

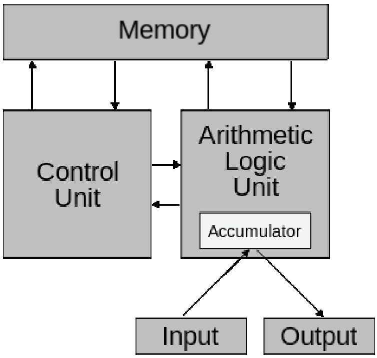


Fig. 2. Von Neumann architecture scheme

The design of a Von Neumann architecture is simpler than the more modern Harvard architecture which is also a stored-program system but has one dedicated set of address and data buses for reading data from and writing data to memory, and another set of address and data buses for fetching instructions.

A stored-program digital computer is one that keeps its program instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions, and the Von Neumann vs. Harvard distinction applies to the cache architecture, not the main memory.

UNIT 6. "OPERATING SYSTEM"

Terminology

assembly	embedded	language
cloud computing	execution	response
compatible with	interoperability	succession
conform	inter-related	template
derive from	interrupt	virtually
distribution	kernel	

In this unit you will need to learn and remember the following words:

a wide range of	exist	resemble
actual	feature	respective
adopt	finite	responsible
as well as	full-fledged	seek (sought, sought)
assign	heavily	side project
capability	influential	similar
collaboration	initial	single
commonly	lack	speed up
concept	load	suggest
definition	meanwhile	supersede
descendants	mode	supervisor
designed	multiple	support
diverse	obey	trademark
end up	originally	unlike
erase	perform	vendor
essence	refer	volunteer
estimate	release	
events	replacement	

Task 1. Answer the questions:

1. What is the definition of an operating system?
2. What are the functions of an operating system?
3. Where are operating systems used?
4. What operating systems do you know and what can you tell about them?
5. What can be the basis for operating systems classification?

Task 2. Match the operating systems and criteria of their classification.

1. Embedded OS	A giving quick and predictable response to events
2. Templated OS	B enabling access to many users through the sharing of time
3. Single-user OS	C allowing more than one program to be running at the same time
4. Real- time OS	D running a group of computers as a single one
5. Multi- user OS	E used in virtualization and cloud computing management
6. Distributed OS	F compact and able to operate with a limited number of resources
7. Multi-tasking OS	G enabling access to the only user

Task 3. Translate the following words:

in concept, resident monitor, enabled, exist, succession, perform, interrupts, features

Complete the sentences below using these words.

Early computers were built to (1) … a series of single tasks, like a calculator. Basic operating system (2) … were developed in the 1950s, such as (3) … functions that could automatically run different programs in (4) … to speed up processing. Operating systems did not (5) … in their modern and more complex forms until the early 1960s. Hardware features were added, that (6) … use of runtime libraries, (7) … , and parallel processing. When personal computers became popular in the 1980s, operating systems were made for them similar (8) … to those used on larger computers.

Task 4. Divide the text below into logical parts and give them headings.

Unix and Linux OS families.

Unix was originally written in assembly language. Ken Thompson wrote B, mainly based on BCPL. B was replaced by C, and Unix, rewritten in C, developed into a large, complex family of inter-related operating systems which have been influential in every modern operating system.

The Unix-like family is a diverse group of operating systems, with several major sub-categories including System V, BSD, and Linux. The name "UNIX" is a trademark of The Open Group which licenses it for use with any operating system that conforms to their definitions. "Unix-like" is commonly used to refer to the large set of operating systems which resemble the original UNIX.

Unix-like systems run on a wide variety of computer architectures. They are used heavily for servers in business, as well as workstations in academic and engineering environments. Free Unix variants, such as Linux and BSD, are popular in these areas.

Four operating systems are certified by The Open Group as Unix. HP's HP-UX and IBM's AIX are both descendants of the original System V Unix and are designed to run only on their respective vendor's hardware. In contrast, Sun Microsystems's Solaris Operating System can run on multiple types of hardware, including x86 and Sparc servers, and PCs. Apple's OS X, a replacement for Apple's earlier (non-Unix) Mac OS, is a hybrid kernel-based BSD variant derived from NeXTSTEP, Mach, and FreeBSD.

Unix interoperability was sought by establishing the POSIX standard. The POSIX standard can be applied to any operating system, although it was originally created for various Unix variants.

Linux (or GNU/Linux) is a Unix-like operating system that was developed without any actual Unix code, unlike BSD and its variants. Linux can be used on a wide range of devices from supercomputers to wristwatches. The Linux kernel is released under an open source license, so anyone can read and modify its code. It has been modified to run on a large variety of electronics. Although estimates suggest that Linux is used on 1.82% of all personal computers, it has been widely adopted for use in servers and embedded systems (such as cell phones). Linux has superseded Unix in most places, and is used on the 10 most powerful supercomputers in the world. The Linux kernel is used in some popular distributions, such as Red Hat, Debian, Ubuntu, Linux Mint and Google's Android.

Task 5. Match the words to their definitions and translate them.

1. diverse
2. supersede
3. kernel
4. originally
5. conform
6. compatibility
7. release
8. actual

A to produce
B really true and exact
C to replace something by more modern and useful
D the fact that different equipment can be used together
E at first, initially
F very different from each other
G the central or the most important part of something
H to be similar to

Task 6. Answer the following questions:

1. Who created UNIX?
2. Is UNIX widely used nowadays? Why?
3. Where can Unix-like programs be used?
4. Are Unix-family products compatible with any hardware? Give examples.
5. What are the differences between Unix and Linux?

Task 7. Complete the table, using the text above.

Languages	Companies	Standards	Programs	Operating systems

Task 8. What are the reasons of existence of so many operating systems? Can the text below give you a hint on an idea for the answer?

The GNU project is a mass collaboration of programmers who seek to create a completely free and open operating system that was similar to Unix but with completely original code. It was started in 1983 by Richard Stallman, and is responsible for many of the parts of most Linux variants.

Thousands of pieces of software for virtually every operating system are licensed under the GNU General Public License. Meanwhile, the Linux kernel began as a side project of Linus Torvalds, a university student from Finland. In 1991, Torvalds began work on it, and posted information about his project on a newsgroup for computer students and programmers. He received a wave of support and volunteers who ended up creating a full-fledged kernel. Programmers worked together to integrate the finished GNU parts with the Linux kernel in order to create a new operating system.

Task 9. How do the following factors influence the development of new operating systems? Continue the list:

- High price of some operating systems
- User-friendly interface
- Lack of possibility to work with different programs
- Incompatibility with many applications
- Difficulties in distribution

Task 10. Read the text below and answer the questions:

1. Why can't hardware work without the operating system?
2. How do applications interact with the hardware?
3. What are the operating system's functions?
4. What are the kernel's functions?
5. What are the main modes of CPU's operation?
6. What are low level tasks?

The operating system provides an interface between an application program and the computer hardware, so that an application program can interact with the hardware only by obeying rules and procedures programmed into the operating system. The operating system is also a set of services which simplify development and execution of application programs. Executing an application program involves the creation of a process by the operating system kernel which assigns memory space and other resources, establishes a priority for the process in multi-tasking systems, loads program binary code into memory, and initiates execution of the application program which then interacts with the user and with hardware devices.

Modern CPUs support multiple modes of operation. CPUs with this capability use at least two modes: protected mode and supervisor mode. The supervisor mode is used by the operating system's kernel for low level tasks that need unrestricted access to hardware, such as controlling how memory is written and erased, and communication with devices like graphics cards.

Task 11. Make a list of all the operating systems known to you. Work in pairs and discuss their development, distribution and application. Find the information on the net if necessary.

Task 12. Get ready to give a talk to your group. Make a plan. Speak about operating systems.

Task 13. Discuss with your partner what new things you have learnt in this unit, what you would like to learn and where you can find this information.

Additional tasks to unit "Operating System"

Task 1. Fill in the gaps in the text with the following parts of sentences:

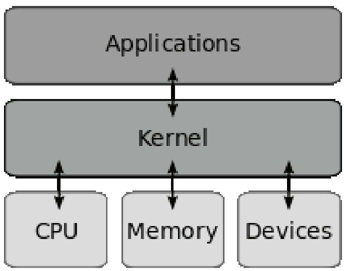
A. loads program binary code into memory

B. simplify development and execution of application programs

C. such media as disks, tapes, flash memory, etc

D. it determines which programs get access to which hardware resources

E. in order to make the different parts of a computer work together



A. The components of an operating system all exist 1. .... All user software needs to go through the operating system in order to use any of the hardware, whether it is as simple as a mouse or keyboard or as complex as an Internet component.

B. A kernel connects the application software to the hardware of a computer.

With the aid of the firmware and device drivers, the kernel provides the most basic level of control over all of the computer's hardware devices. It manages memory access for programs in the RAM, 2. ..., it sets up or resets the CPU's operating states for optimal operation at all times, and it organizes the data for long-term non-volatile storage with file systems on 3. ....

C. The operating system provides an interface between an application program and the computer hardware, so that an application program can interact with the hardware only by obeying rules and procedures programmed into the operating system. The operating system is also a set of services which 4. .... Executing an application program involves the creation of a process by the operating system kernel which assigns memory space and other resources, establishes a priority for the process in multi-tasking systems, 5. ... , and initiates execution of the application program which then interacts with the user and with hardware devices.

Task 2. Answer the questions

1. What are the examples of simple and complex hardware?
2. What enables the computer to define the sequence of processes?
3. What is firmware?
4. How does an application program interact with the hardware?

Task 3. Give the headings to the following parts of the text. One is extra.

1. Kernel
2. Interrupts
3. Components
4. Program execution

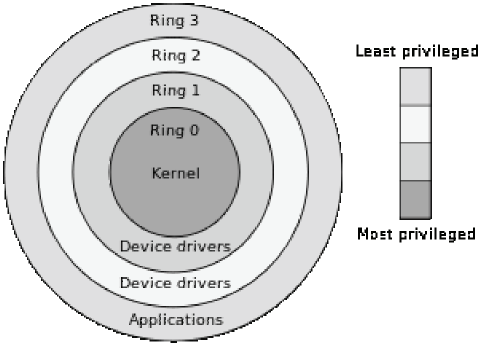
Task 4. Match the questions 1-4 and the answers A-D.

1. What are the functions of the kernel?	A. By obeying the rules of the operating system.
2. What comprises the hardware of the computer?	B. To provide the control over the hardware and manage memory access.
3. What does RAM do?	C. All the devices you can touch.
4. How does the application program interact with the hardware?	D. Determines the program which is to get access to hardware resources.

Task 5. Speak about the kernel and operating system using the questions in Task 4.

Task 6. Answer the question: what is the difference between the protective and supervisor modes? Use the figure and the text below.

Modern CPUs support multiple modes of operation. CPUs with this capability use at least two modes: protected mode and supervisor mode. The supervisor mode is used by the operating system's kernel for low level tasks that need unrestricted access to hardware, such as controlling how memory is written and erased, and communication with devices like graphics cards. Protected mode, in contrast, is used for almost everything else. Applications operate within protected mode, and can only use hardware by communicating with the kernel.



Task 7. Read the text and entitle it. Give subtitles to each paragraph of the text.

Interrupts are central to operating systems, as they provide an efficient way for the operating system to interact with and react to its environment. The alternative – having the operating system "watch" the various sources of input for events that require action – can be found in older systems with very small stacks (50 or 60 bytes) but are unusual in modern systems with large stacks. Interrupt-based programming is directly supported by most modern CPUs. Interrupts provide a computer with a way of automatically saving local register contexts, and running specific code in response to events. Even very basic computers support hardware interrupts, and allow the programmer to specify code which may be run when that event takes place.

When an interrupt is received, the computer's hardware automatically suspends whatever program is currently running, saves its status, and runs computer code previously associated with the interrupt; this is analogous to placing a bookmark in a book in response to a phone call. In modern operating systems, interrupts are handled by the operating system's kernel. Interrupts may come from either the computer's hardware or from the running program.

When a hardware device triggers an interrupt, the operating system's kernel decides how to deal with this event, generally by running some processing code. The amount of code being run depends on the priority of the interrupt (for example: a person usually responds to a smoke detector alarm before answering the phone). The processing of hardware interrupts is a task that is usually delegated to software called device driver, which may be either part of the operating system's kernel, part of another program, or both. Device drivers may then relay information to a running program by various means.

A program may also trigger an interrupt to the operating system. If a program wishes to access hardware for example, it may interrupt the operating system's kernel, which causes control to be passed back to the kernel. The kernel will then process the request. If a program wishes additional resources (or wishes to shed resources) such as memory, it will trigger an interrupt to get the kernel's attention.

Task 8. Match the words and their definitions.

1. interrupt	a. any
2. suspend	b. to communicate a message
3. whatever	c. a pause in a continues process
4. relay	d. to stop something for a short time
5. trigger	e. a reaction
6. stack	f. to make something start to work
7. response	g. to help something to be successful
8. event	h. a pile of thing placed one on top of another
9. support	i. a particular planned activity

Task 9. Change the sentences from Active into Passive voice and vice versa.

1. Interrupt-based programming is directly supported by most modern CPUs.

2. Interrupts provide an efficient way for the operating system to interact with the environment.

3. Interrupts provide a computer with a way of automatically saving local register contexts.

4. Even very basic computers support hardware interrupts.

5. A hardware device triggers an interrupt.

6. Device drivers may then relay information to a running program by various means.

7. In modern operating systems, interrupts are handled by the operating system's kernel.

8. The kernel will then process the request.

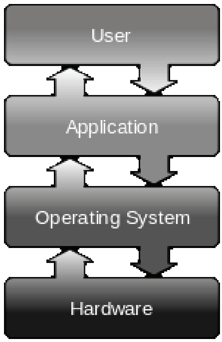
Task 10. Finish the sentences.

1. Interrupts are necessary to ...

2. Early computers “watched” ...

3. The computer's hardware automatically suspends...

4. The alternative can be found in older computer systems with ...



Task 11. Speak about interrupts using the words given in Task 8.

Task 12. Look at the Figure and explain the links between the components of a computer. How do they interact? How are they interdependent?

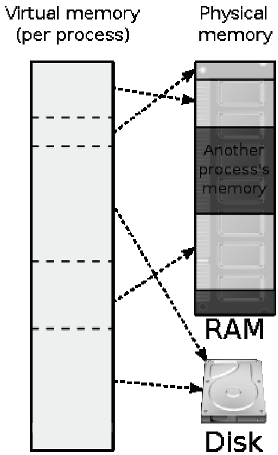
Additional texts to unit “Operating System”

Text 1

Virtual memory

Many operating systems can "trick" programs into using memory scattered around the hard disk and RAM as if it is one continuous chunk of memory, called virtual memory.

The use of virtual memory addressing (such as paging or segmentation) means that the kernel can choose what memory each program may use at any given time, allowing the operating system to use the same memory locations for multiple tasks.



If a program tries to access memory that isn't in its current range of accessible memory, but nonetheless has been allocated to it, the kernel will be interrupted in the same way as it would if the program were to exceed its allocated memory. (See section on memory management.) Under UNIX this kind of interrupt is referred to as a page fault.

When the kernel detects a page fault it will generally adjust the virtual memory range of the program which triggered it, granting it access to the memory requested. This gives the kernel discretionary power over where a particular application's memory is stored, or even whether or not it has actually been allocated yet.

In modern operating systems, memory which is accessed less frequently can be temporarily stored on disk or other media to make that space available for use by other programs. This is called swapping, as an area of memory can be used by multiple programs, and what that memory area contains can be swapped or exchanged on demand.

"Virtual memory" provides the programmer or the user with the perception that there is a much larger amount of RAM in the computer than is really there.

Text 2

POSIX

POSIX (/pɒzɪks/), an acronym for "Portable Operating System Interface", is a family of standards specified by the IEEE for maintaining compatibility between operating systems. POSIX defines the application programming interface (API), along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems.

Originally, the name "POSIX" referred to IEEE Std 1003.1-1988, released in 1988. The family of POSIX standards is formally designated as IEEE 1003 and the international standard name is ISO/IEC 9945.

The standards emerged from a project that began circa 1985. Richard Stallman suggested the name POSIX to the IEEE instead of former IEEE-IX. The committee found it more easily pronounceable and memorable, so the committee adopted it.

The POSIX specifications for Unix-like operating system environments originally consisted of a single document for the core programming interface, but eventually grew to 19 separate documents (for example, POSIX.1, POSIX.2 etc.). The standardized user command line and scripting interface were based on the Korn shell. Many user-level programs, services, and utilities including awk, echo, ed were also standardized, along with required program-level services including basic I/O (file, terminal, and network) services. POSIX also defines a standard threading library API which is supported by most modern operating systems. Nowadays, most of POSIX parts are combined into a single standard, IEEE Std 1003.1-2008, also known as POSIX.1-2008.

As of 2009, POSIX documentation is divided in two parts:

- POSIX.1-2008: POSIX Base Definitions, System Interfaces, and Commands and Utilities (which include POSIX.1, extensions for POSIX.1, Real-time Services, Threads Interface, Real-time Extensions, Security Interface, Network File Access and Network Process-to-Process Communications, User Portability Extensions, Corrections and Extensions, Protection and Control Utilities and Batch System Utilities)
- POSIX Conformance Testing: A test suite for POSIX accompanies the standard: PCTS or the POSIX Conformance Test Suite.[5]

The development of the POSIX standard takes place in the Austin Group, a joint working group linking the Open Group and the ISO organization.

After 1997, the Austin Group developed the POSIX revisions. The specifications are known under the name Single UNIX Specification, before they became a POSIX standard when formally approved by the ISO.

UNIT 7. “BINARY CODE”

Terminology

base	decimal	hexadecimal
binary code	discharge	notation
binary coding scheme	encoding date	variable – width strings
binary system	fixed-width strings	

Task 1. Match the words and their definitions.

1. to assign	A the numbers are based on the number 16
2. binary	B the distance from one side of smth to the other
3. notation	C consisting of two parts
4. decimal	D to give a particular value, place etc to smth
5. hexadecimal	E system of written marks or sighs
6. width	F system is based on the number 10
7. the binary system	G a system in which only the numbers 0 and 1 are used

Task 2. Translate the following:

1. Binary code is a way of representing date in a single discharge in the form of a combination of two characters, typically denoted by the numbers 0 and 1.

2. A binary coding scheme is a method used for representing all of the digits, letters, special characters, and control characters available to a digital computer using groups of 0s and 1s.

3. Binary codes are combinations of two elements and are not binary system, but use it as a base.

Task 3. Discuss with your partner what you know about Binary code. Answer the questions:

1. How is digital date represented by the binary code?
2. Why do computers use the binary code instead of the decimal?
3. What are weighted binary codes?
4. What is the definition of binary code?
5. Is binary code relevant to computers?

Task 4. Read the text entitle it and give subheadings to each paragraph.

A. A binary code represents text or computer processor instructions using the binary number system's two binary digits, 0 and 1. A binary code assigns a bit string to each symbol or instruction. For example, a binary string of eight binary digits (bits) can represent any of 256 possible values and can therefore correspond to a variety of different symbols, letters or instructions.

B. In computing and telecommunication, binary codes are used for various methods of encoding data, such as character strings, into bit strings. Those methods may use fixed-width or variable-width strings. In a fixed-width binary code, each letter, digit, or other character is represented by a bit string of the same length; that bit string, interpreted as a binary number, is usually displayed in code tables in octal, decimal or hexadecimal notation. There are many character sets and many character encodings for them.

C. A bit string, interpreted as a binary number, can be translated into a decimal number. For example, the lower case *a*, if represented by the bit string 01100001 (as it is in the standard ASCII code), can also be represented as the decimal number 97.

D. The modern binary number system, the basis for binary code, was discovered by Gottfried Leibniz in 1679 and appears in his article "Explanation of the binary arithmetic", which uses only the characters 1 and 0, with some remarks on its usefulness, and on the light it throws on the ancient Chinese figures of Fu Xi .(1703) Leibniz's system uses 0 and 1, like the modern binary numeral system. He created a system consisting of rows of zeros and ones. But Leibniz had not found a use for his system.

Task 5. Answer the questions to the text:

1. What is the definition of binary code?
2. Is binary code relevant to computers?
3. Who was the discoverer of binary code?

Task 6. Complete the sentences in an appropriate way.

1. In computing binary codes are used….
2. A binary code represents text using….
3. In a fixed-width binary code, each digit is represented by…..
4. A bit string can be translated into….
5. A binary string of eight binary digits can represent…
6. The modern binary number system was discovered…

Task 7. Translate the following sentences into English.

1. Двоичный код – это способ представления данных в одном разряде в виде комбинации двух знаков.

2. Двоичный код может быть непозиционным и позиционным.

3. Двоичный код также может использоваться для кодирования чисел в системах счисления с любым другим основанием.

Task 8. Discuss these questions with your partner.

1. How many binary coding schemes are there in computer systems?
2. How is digital data represented by the binary code?
3. What is the difference between binary codes and binary digits?
4. Is binary code the source code?
5. What is the most widely used binary code for microcomputers?

Task 9. Which is correct?

a) Binary code is a code comprising base 2 numbers: zeros and ones.

b) Binary code is a language used in computer programming and is made up entirely of ones and zeroes.

Task 10. Read the text. In groups discuss the questions given after it.

The bit string is not the only type of binary code. A binary system in general is any system that allows only two choices such as a switch in an electronic system or a simple true or false test.

Braille is a type of binary code that is widely used by blind people to read and write. This system consists of 6-dot positions, three in each column. Each dot has two states: raised or not raised.

Morse Code is a method of transmitting text information as a series of on-off tones, lights, or clicks. Any Boolean system such as this, which encodes meaning, is a form of binary code.

ASCII code - the American Standard Code for Information Interchange, uses a 7-bit binary code to represent text and other characters within computers, communications equipment, and other devices. Each letter or symbol is assigned a number from 0 to 127.

Binary—coded decimal, or BCD, is a binary encoded representation of integer values that uses a 4- bit nibble to encode decimal digits. Four binary bits can encode up to 16 distinct values; but, in BCD-encoded numbers, only the first ten values in each nibble are legal, and encode the decimal digits zero, through nine. The remaining six values are illegal, and may cause either a machine exception or unspecified behavior, depending on the computer implementation of BCD arithmetic. BCD arithmetic is sometimes preferred to floating-point numeric formats in commercial and financial applications where the complex rounding behaviors of floating-point numbers is inappropriate.

- What are the main differences between the code systems?
- What are the main areas of application code systems?
- What other codes and coding systems do you know?

Task 11. Work in pairs. Tell your partner about binary code application in computing.

Task 12. Write an essay of 100 words on the topic "Binary code"

Additional tasks for the unit “Binary Code”

1. Read the text and answer the questions below.

History of binary code

Binary numbers were first described in Chandashutram written by Pingala in 100 BC. Binary Code was first introduced by the English mathematician and philosopher Eugene Paul Curtis during the 17th century. Curtis was trying to find a system that converts logic’s verbal statements into a pure mathematical one. After his ideas were ignored, he came across a classic Chinese text called Book of Changes, which used a type of binary code. The book had confirmed his theory that life could be simplified or reduced down to a series of straightforward propositions. He created a system consisting of rows of zeros and ones. During this time period, Curtis had not yet found a use for this system.

Another mathematician and philosopher by the name of George Boole published a paper in 1847 called 'The Mathematical Analysis of Logic' that describes an algebraic system of logic, now known as Boolean algebra. Boole’s system was based on binary, a yes-no, on-off approach that consisted the three most basic operations: AND, OR, and NOT. This system was not put into use until a graduate student from Massachusetts Institute of Technology by the name Claude Shannon noticed that the Boolean algebra he learned was similar to an electric circuit. Shannon wrote his thesis in 1937, which implemented his findings. Shannon's thesis became a starting point for the use of the binary code in practical applications such as computers, electric circuits, and more.

1. Who was the first to introduce the binary code?
2. What was he trying to find?
3. What was the role of George Boole in creating binary code?
4. What was the starting point for the use of binary code?

Task 2. Fill in the gaps.

1) encodings 2) decimal 3) values 4) a bit string 5) digit 0 and 1 6) computer processor 7) telecommunication 8) lower case

Binary code

A binary code is a way of representing text or a)……. instructions by the use of the binary number system's two-binary b) …….. This is accomplished by assigning c) …….. to each particular symbol or instruction. For example, a binary string of eight binary digits (bits) can represent any of 255 possible d)…… and can therefore correspond to a variety of different symbols, letters or instructions.

In computing and e) ………, binary codes are used for any of a variety of methods of encoding data, such as character strings, into bit strings. Those methods may be f) …….. or variable- width. In a fixed-width binary code, each letter, digit, or other character, is represented by a bit string of the same length; that bit string, interpreted as a binary number, is usually displayed in code tables in octal, g) ……… or hexadecimal notation. There are many character sets and many characters h)…….. for them. A bit string, interpreted as a binary number, can be translated into a decimal number. For example, the i)…….. "a" as represented by the bit string 01100001, can also be represented as the decimal number 97.

Task 3. Think of the other codes used in mathematics and share your ideas with the partner.

Additional texts to unit "Binary Code"

Text 1

If you are going to start coding in Visual Basic, or any other computer language, you will need to know the basics. One of the most basic things to understand is how a computer handles numbers. While there are many different computer number systems, and many types of data, in reality a computer will only understand binary. The other number systems are used to represent binary values in a way that human users can more easily understand and work with.

It is important to understand that, no matter how a number is represented, the value of the number remains the same. For example, the number 10 will always be the same, no matter if it is represented in binary, decimal, or hexadecimal form. Computers only work in binary, which is a number system that uses only zeros and ones (0 and 1). Other number systems have been designed for the benefit of human users who will find it easier to read and type in formats other than binary. Knowing which number system is being used in a piece of code is important in order to avoid confusion.

Binary is the number system that computers understand. It is base 2, which means it uses only 0 and 1 to represent any number. The binary representation of a number can sometimes be quite long, and it can be difficult to quickly determine what the number is.Binary counting works in the same way as decimal counting, but each digit caps at 1, meaning that when you increment a number by 1, a 1 digit is reset to 0 and the next number to the left is incremented instead. Here are the numbers 0 through 9 expressed in binary: 0000, 0001, 0010, 0011, 0100, 0101, 0111, 1000, 1001.Octal is a base 8 numbering system. This means it uses the values 0 through 7 to express any number. Working with octal numbers is useful when programming, as it is easier to convert base 8 to binary (base 2), than it is to convert decimal values to binary. Whenever a digit that is a 7 is incremented by 1, it resets to 0, and the next digit to the left is incremented by 1 instead. Octal produces shorter representations of longer numbers than binary. Here are the numbers 0 through 9 expressed in octal: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11.Decimal is the numbering system that most people are fluent in. It is a base 10 numbering system, meaning that it uses the digits 0 through 9 to represent any number. Whenever a digit that is a 9 is incremented by 1, it resets to 0, and the next digit to the left is incremented by 1 instead. Decimal numbers are harder to convert into binary. However, it is an easy number system for human to understand as humans have 10 fingers, and most things are calculated in base 10.Hexadecimal is one of the most complex numbering systems to understand. It is base 16. As there are only 10 digits (0-9), hexadecimal also makes use of the letters A through F. F represents 15, so when an F digit is incremented by 1, it resets to 0, and the next number to the left is incremented instead. The numbers 0 through 16 in hexadecimal look like this: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10.

If you want to start programming, you will first need to learn the basics. One of the first things to learn is how computers handle numbers. There are several different number systems you will need to learn, including binary, octal, and hexadecimal. However, computers only understand binary, which is a number system that uses combinations of 0 and 1 to represent any number.

Text 2

Why do computers use the binary code instead of the decimal system?

Binary is a set of instructions used to control the computer, and works from 1's and 0's, but the computer understands them as on or off signals. If the decimal system were used, there would need to be 10 different voltages, in which case there'd be more room for error with resistors etc., and therefore more room for corruption of data.

Furthermore: The computer may process enormous amounts of data. But, because humans don't want to think about so much data in detail, there's a limit to how much of it is going to be input by humans or output to humans (e.g. typed in, read from the screen, etc). So the computer is only ever going to spend a small amount of time converting between one base and another. So when working out which base is the best, you don't really need to worry about converting bases to keep humans happy.

As one can see, a computer or, to be more precise a hard-disk, is nothing but a box containing magnetic tape. Now one can actually think why magnetic tape? Why not anything else? Because magnet has two properties: attracting (unlike poles) and repelling (like poles).

Let us give 1 to attract (for e.g.) and 0 to repel. Now when an instruction executes, a reader on the hard-disk (as in cassette or tape recorder) sends signal to thehard-disk. And the hard-disk reacts according to the signal. Let us say signal elements were 101. On seeing 1 it will attract and a mark on magnetic tape will be seen. And for 0 it will repel and leave some other mark. This will continue till the whole input signals are executed.

UNITE 8. “BINARY SEARCH TREE”

Terminology

child node
duplicated node
dynamic date
parent node
property
restriction
seach tree
sequencing purposes
substance
sub-tree
successor node

Task 1. Match the words and their definitions.

1	a successor	A	quality or power that a substance has
2	to provide	B	to continue to be in the same state or condition
3	property	C	a rule or a law that limits or controls what people can do
4	dynamic	D	exactly the same as something
5	a node	E	someone who takes a job or position previously held by someone else
6	to remain	F	to give something to someone
7	advantage	G	a place where lines in a network cross or join
8	restriction	H	the importance or usefulness of something
9	duplicate	I	continuously moving or changing

Task 3. Choose the synonyms:

1. duplicate
   a) node b) copy c) root
2. successor
   a) worker b) follower c) partner
3. property
   a) characteristic b) character c) quality

4. dynamic
    a) quick b) mobile c) hardworking
5. advantage
    a) usefulness b) benefit c) help
6. restriction
    a) ban b) order c) behavior

Task 4. Read the text and entitle it.

In computer science, a binary search tree (BST), sometimes also called an ordered or sorted binary tree, is a node-based binary tree data structure where each node has a comparable key (and an associated value) and satisfies the restriction that the key in any node is larger than the keys in all nodes in that node's left sub-tree and smaller than the keys in all nodes in that node's right sub-tree. Each node has no more than two child nodes. Each child must either be a leaf node or the roof of another binary search tree. The left sub-tree contains only nodes with keys less than the parent node; the right sub-tree contains only nodes with keys greater than the parent node. BSTs are also dynamic data structures, and size of BST is only limited by the amount of free memory in the operating system. The main advantage of binary search trees is that it remains ordered, which provides quicker search times than many other data structures. The common properties of binary search trees are as follows:

- The left sub-tree of a node contains only nodes with keys less than the node's key.
- The right sub-tree of a node contains only nodes with keys greater than the node's key.
- The left and right sub-tree each must also be a binary search tree.
- Each node can have up to two successor nodes.
- There must be no duplicate nodes.

Generally, the information represented by each node is a record rather than a single data element. However, for sequencing purposes, nodes are compared according to their keys rather than any part of their associated records. The major advantage of binary search trees over other data structures is that the related sorting algorithms and search algorithms such as in-order traversal can be very efficient.

Task 5. Answer the questions to the text.

1. What are the main properties of binary search trees?
2. What is the major advantage of BST?

Task 6. Complete the sentences in an appropriate way.

1. BST is……
2. A node ia a based binary tree data structure where …
3. Each child must either be a leaf node….
4. The size of BST is ….
5. The main advantage of binary search trees…..
6. The information represented by each node…..

Task 7. Translate the following sentences into Russian:

1. A binary tree is made of nodes, where each node contains a "left" pointer, a "right" pointer, and a data element.
2. The "root" pointer points to the topmost node in the tree.
3. The left and right pointers recursively point to smaller "sub-trees" on either side.
4. A null pointer represents a binary tree with no elements -- the empty tree.
5. The formal recursive definition is: a binary tree is either empty or is made of a single node, where the left and right pointers each point to a binary tree.

Task 8. Translate the following sentences into English:

1. Двоичные деревья представляют эффективный способ поиска.
2. Двоичное дерево представляет собой структурированную коллекцию узлов.
3. Основное назначение двоичных деревьев заключается в повышении эффективности поиска.
4. Основным преимуществом двоичного дерева поиска перед другими структурами данных является возможная высокая эффективность реализации основанных на нём алгоритмов поиска и сортировки.
5. Двоичное дерево поиска применяется для построения множества, мультимножества, ассоциативных массивов.

Task 9. Discuss these questions with your partner:

1. What is an almost complete binary tree?
2. What is the difference between a binary search tree and a binary tree?
3. What is a binary search tree in data structures?

Task 10. Read the text. With your partner discuss the questions after it.

The applications of binary search tree

- BST is used in *many* search applications where data is constantly entering or leaving; such as the map and set objects in many languages' libraries.
- Binary Space Partition is used in almost every 3D video game to determine what objects need to be rendered.
- Binary Trees are used in almost every high-bandwidth router for storing router-tables.
- Hash Trees are used in p2p programs and specialized image-signatures in which a hash needs to be verified, but the whole file is not available.
- GGM Trees – Used in cryptographic applications to generate a tree of pseudo-random numbers.
- Syntax Tree – Constructed by compilers and (implicitly) calculators to parse expressions.
- T-tree – Though most databases use some form of B-tree to store data on the drive, databases which keep all (most) their data in memory often use T-trees to do so.

Are there any other applications of binary search tree? What are they?

Task 12. Get ready to give a talk to your group. Make a plan. Speak about binary search tree.

Task 13. Discuss with your partner what new things you have learnt in this unit, what you would like to learn and where you can find this information.

Additional tasks to unit "Binary Search Trees"

Task 1. Arrange the paragraphs of the text below in the logical order. Give the titles to the paragraphs.

Binary Tree Traversal

A) The second way is a preorder traversal. A preorder traversal visits the parent node first, then left child, and then the right child.

B) All three types of traversals are useful for different cases. For example, an inorder traversal will give you all the values in the Binary Search Tree in increasing order.

C) A tree traversal means iterating through every value in the Binary Tree. Since a tree consists of three components: the parent, the left child, and the right child, there are three ways to traverse through a Binary Tree.

D) While a postorder traversal is best when the goal is to remove all the nodes of the tree. Since there is no way to tell which traversal will be needed, the C# Binary Search Tree at the bottom of the page has a property to set which traversal mode to use.

E) The first way is an inorder traversal. An inorder traversal visits the left child first (recursively in case it has children nodes of its own), then visists the parent node, and visits the right child last.

F) The third way is a postorder traversal. A postorder traversal visits the left child first, then the right child, and then the parent node.

Task 2. Fill in the gaps with appropriate words below.

1) the worst 2) traverse 3) complexity 4) key 5) height, 6) removes 7) common, 8)search

Binary search tree operations

There are some a)…… operations on the binary search tree:

- Insert – inserts a new node into the tree
- Delete – b)……. an existing node from the tree
- c)…….. – traverse the tree in pre-order, in-order and post-order. For the binary search tree, only in-order traversal makes sense
- Search – search for a given node's d)….. in the tree

All binary e)……. tree operations are O(H), where H is the depth of the tree. The minimum f)……. of a binary search tree is $H = \log_2 N$, where N is the number of the tree's nodes. Therefore the g)…… of a binary search tree operation in the best case is O(logN); and in the worst case, its complexity is O(N).

h)…… case happens when the binary search tree is unbalanced. Many algorithms have been invented to keep a binary search tree balanced such as height-balanced tree or AVT trees of **A**delson-**V**elskii and **L**andis, B-trees, and Splay trees.

Task 3. Look through the information of the unit "Binary search tree". Write a short essay of 150 words using it.

Additional texts to unit "Binary Search Trees"

Text 1

Search Tree Implementation

A binary search tree relies on the property that keys that are less than the parent are found in the left subtree, and keys that are greater than the parent are found in the right subtree. We will call this the bst property. As we implement the Map interface as described above, the bst property will guide our implementation. Figure 1 illustrates this property of a binary search tree, showing the keys without any associated values. Notice that the property holds for each parent and child. All of the keys in the left subtree are less than the key in the root. All of the keys in the right subtree are greater than the root. guide our implementation. Figure 1 illustrates this property of a binary search tree, showing the keys without any associated values. Notice that the property holds for each parent and child. All of the keys in the left subtree are less than the key in the root. All of the keys in the right subtree are greater than the root.

Now that you know what a binary search tree is, we will look at how a binary search tree is constructed. The search tree in Figure1 represents the nodes that exist after we have inserted the following keys in the order shown: [Math Processing Error]. Since 70 was the first key inserted into the tree, it is the root. Next, 31 is less than 70, so it becomes the left child of 70. Next, 93 is greater than 70, so it becomes the right child of 70. Now we have two levels of the tree filled, so the next key is going to be the left or right child of either 31 or 93. Since 94 is greater than 70 and 93, it becomes the right child of 93. Similarly 14 is less than 70 and 31, so it becomes the left child of 31. 23 is also less than 31, so it must be in the left subtree of 31. However, it is greater than 14, so it becomes the right child of 14.

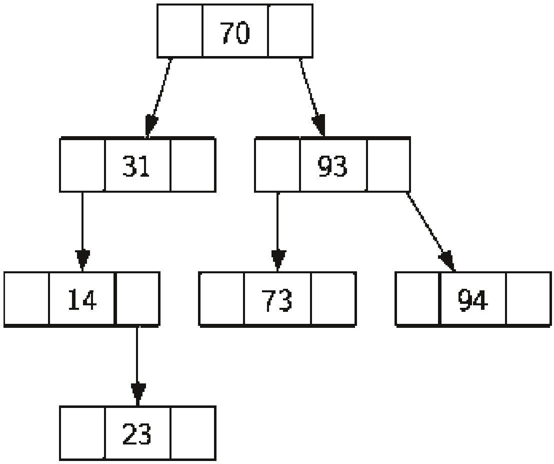


Fig. 1. A Simple Binary Search Tree

To implement the binary search tree, we will use the nodes and references approach similar to the one we used to implement the linked list, and the expression tree. However, because we must be able create and work with a binary search tree that is empty, our implementation will use two classes. The first class we will call Binary Search Tree, and the second class we will call Tree node. The Binary Search Tree class has a reference to the Tree Node that is the root of the binary search tree. In most cases the external methods defined in the outer class simply check to see if the tree is empty. If there are nodes in the tree, the request is just passed on to a private method defined in the Binary Search Tree class that takes the root as a parameter. In the case where the tree is empty or we want to delete the key at the root of the tree, we must take special action.

Text 2

Search Tree Analysis

With the implementation of a binary search tree now complete, we will do a quick analysis of the methods we have implemented. Let's first look at the put method. The limiting factor on its performance is the height of the binary tree. Recall from the vocabulary section that the height of a tree is the number of edges between the root and the deepest leaf node. The height is the limiting factor because when we are searching for the appropriate place to insert a node into the tree, we will need to do at most one comparison at each level of the tree.

What is the height of a binary tree likely to be? The answer to this question depends on how the keys are added to the tree. If the keys are added in a random order, the height of the tree is going to be around [Math Processing Error] where [Math Processing Error] is the number of nodes in the tree. This is because if the keys are randomly distributed, about half of them will be less than the root and half will be greater than the root. Remember that in a binary tree there is one node at the root, two nodes in the next level, and four at the next. The number of nodes at any particular level is [Math Processing Error] where [Math Processing Error] is the depth of the level. The total number of nodes in a perfectly balanced binary tree is [Math Processing Error], where [Math Processing Error] represents the height of the tree.

A perfectly balanced tree has the same number of nodes in the left subtree as the right subtree. In a balanced binary tree, the worst-case performance of put is [Math Processing Error], where [Math Processing Error] is the number of nodes in the tree. Notice that this is the inverse relationship to the calculation in the previous paragraph. So [Math Processing Error] gives us the height of the tree, and represents the maximum number of comparisons that put will need to do as it searches for the proper place to insert a new node.

Unfortunately it is possible to construct a search tree that has height [Math Processing Error] simply by inserting the keys in sorted order! An example of such a tree is shown in *Figure1*. In this case the performance of the put method is [Math Processing Error]. Now that you understand that the performance of the put method is limited by the height of the tree, you can probably guess that other methods, get, in, and del, are limited as well. Since get searches the tree to find the key, in the worst case the tree is searched all the way to the bottom and no key is found. At first glance del might seem more complicated, since it may need to search for the successor before the deletion operation can complete. But remember that the worst-case scenario to find the successor is also just the height of the tree which means that you would simply double the work. Since doubling is a constant factor it does not change worst case analysis of [Math Processing Error] for an unbalanced tree.

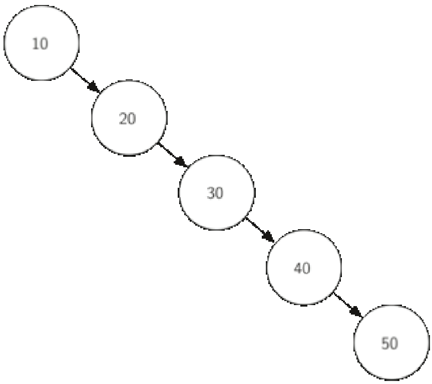


Fig. 1. A skewed binary search tree would give poor performance

UNIT 9. "BOOLEAN ALGEBRA"

Terminology

arithmetic modulo	linear algebra	sequence of bits
conjunction	logical calculus	values of the variables
disjunction	mathematical quantity	
numeric relations	negation	

Task 1. Match the words and their definitions.

1 investigation	A mathematical quantity shown by a letter of the alphabet or sign
2 negation	B a difference between two things that you would expect to be in agreement
3 conjunction	C an attempt to find out the truth or the causes of scientific problem
4 value	D the denial of something
5 disjunction	E a mathematical quantity which can represent several different amounts
6 variable	F a combination

Task 2. Match the word combinations and their translation.

1. values of the variables	A. теория множеств
2. digital electronics	B. законы мышления
3. logical relations	C. значения переменных
4. values of the variables	D. логические отношения
5. conjunction	E. последовательность битов
6. negation	F. дизъюнкция
7. disjunction	G. конъюнкция
8. a sequence of bits	H. отрицание

Task 3. Read the text and entitle it.

In mathematics and mathematical logic, Boolean algebra is the subarea of algebra in which the values of the variables are the truth values true and false, usually denoted 1 and 0 respectively. Instead of elementary algebra where the values of the variables are numbers, and the main operations are addition and multiplication, the main operations of Boolean algebra are the conjunction and, denoted ∧, the disjunction or, denoted ∨, and the negation not, denoted ¬. It is thus a formalism for describing logical relations in the same way that ordinary algebra describes numeric relations.

Boolean algebra was introduced in by George Boole in his first book The Mathematical Analysis of Logic (1847), and set forth more fully in his An Investigation of the Laws of Thought_ (1854). According to Hungtinton the term "Boolean algebra" was first suggested by Sheffer in 1913.

Boolean algebra has been fundamental in the development of digital electronics, and is provided for in all modern programming languages. It is also used in set theory and statistics.

In Boolean algebra they denote the truth values false and true. These values are represented with the bits, namely 0 and 1. They 0 and 1, they may be identified with the elements of the two-element field GF(2), that is integer arithmetic modulo 2, for which 1 + 1 = 0. Boolean algebra also deals with functions which have their values in the set {0, 1}. A sequence of bits is a commonly used such function.

Task 4. Answer the questions to the text:

1. What are the main operations in Boolean algebra?
2. Boolean algebra is the subarea of algebra, isn't it?
3. Where is Boolean algebra used?

Task 5. Complete the sentences in an appropriate way.

1. The main operations of Boolean algebra are …..
2. Boolean algebra is …..
3. Ordinary algebra describes…..
4. Boolean algebra was…
5. Boolean algebra has been…..
6. Boolean algebra is used in…
7. Boolean algebra deals with ….

Task 6. Translate the following sentences into Russian:

1. Boole's algebra predated the modern developments in abstract algebra and mathematical logic.
2. M. H. Stone proved in 1936 that every Boolean algebra is isomorphic to a field of sets.
3. The original application for Boolean operations was mathematical logic, where it combines the truth values, true or false, of individual formulas.
4. Boolean algebra is a logical calculus of truth values. It deals with two-value (true / false or 1 and 0) variables.
5. Boolean algebra provides the theoretical concepts for computer design.

Task 7. Translate the sentences into English.

1. Можно доказать, что любая конечная булева алгебра изоморфна булевой алгебре всех подмножеств какого-то множества.
2. В булевых алгебрах существуют двойственные утверждения, они либо одновременно верны, либо одновременно неверны.
3. Булева алгебра используется в обосновании теории вероятности.
4. Кроме основных операций булева алгебра допускает определение и других, среди которых особенно важна операция симметрической разности.
5. Алгебра логики - предельно важная для цифровых компьютеров тема. И с точки зрения их устройства, схемотехники, и с точки зрения их функционирования и программирования поведения.

Task 8. Match the questions with appropriate answers.

1. What is Boolean algebra?
2. What is the importance of Boolean algebra?
3. How is Boolean algebra different from linear algebra?
4. What are the basic logic operations in Boolean algebra?
5. What symbol represents AND in Boolean algebra?
6. Why is Boolean algebra famous?

A AND in Boolean algebra is represented by a dot, like multiplication. It can also be represented with parenthesis."(A OR B) AND C"...

B It is famous in computer science even today because it helps verify if something is true or false.

C It is a system of logical calculus that logic axioms are based on.

D Boolean algebra is the process of evaluating statements to be either true or false. It is extremely important for inductive and deductive reasoning as well as for all forms of science.

E Linear algebra works with straight lines on a plane. Boolean algebra is a logical calculus.

F AND, OR, and NOT are the basic operators in Boolean algebra.

Task 9. Work in pairs. Tell your partner about postulates of Boolean algebra using the tasks of the unit.

Task 10. Discuss with your partner what new things you have learnt in this unit, what you would like to learn and where you can find this information.

Additional tasks to unit "Boolean algebra"

Task 1. Classify Boolean algebras according to their features. Make a table.

Special classes of Boolean algebras

There are many special classes of Boolean algebra which are important both for the intrinsic theory of BAs and for applications:

- Atomic BAs, already mentioned above.
- Atomless BAs, which are defined to be BAs without any atoms. For example, any infinite free BA is atomless.
- Complete BAs, defined above. These are especially important in the foundations of set theory.
- Interval algebras. These are derived from linearly ordered sets $(L, <)$ with a first element as follows. One takes the smallest algebra of subsets of L containing all of the half-open intervals $[a, b)$ with a in L and b in L or equal to ∞. These BAs are useful in the study of Lindenbaum-Tarski alge-

bras. Every countable BA is isomorphic to an interval algebra, and thus a countable BA can be described by indicating an ordered set such that it is isomorphic to the corresponding interval algebra.

- Tree algebras. A tree is a partially ordered set (T, <) in which the set of predecessors of any element is well-ordered. Given such a tree, one considers the algebra of subsets of T generated by all sets of the form $\{b : a \leq b\}$ for some a in T.
- Superatomic BAs. These are BAs which are not only atomic, but are such that each subalgebra and homomorphic image is atomic.

Task 2. Read the text and find the answers to the questions that follow it.

Formal system

In logic, mathematics, and computer science, a format system is a format grammar used for modeling purposes. Formalization is the act of creating a formal system, in an attempt to capture the essential features of a real-world or conceptual system in formal language.

In mathematics, formal proofs are the product of formal systems, consisting of axioms and rules of deduction. Theorems are then recognized as the possible 'last lines' of formal proofs. The point of view that this picture encompasses mathematics has been called formalist. The term has been used pejoratively. On the other hand, Davia Hilbert founded metamathematica as a discipline designed for discussing formal systems; it is not assumed that the metalanguage in which proofs are studied is itself less informal than the usual habits of mathematicians suggest. To contrast with the metalanguage, the language described by a formal grammar is often called an object language.It has become common to speak of a formalism, more-or-less synonymously with a formal system *wit hin standard mathematics* invented for a particular purpose. This may not be much more than a **notation**, such as Dira's bra-ket notation.Mathematical formal systems consist of the following:

- A finite set of symbols which can be used for constructing formulae.
- A grammar, i.e. a way of constructing well-formed formulae out of the symbols, such that it is possible to find a decision procedure for deciding whether a formula is a well-formed formula (*wff*) or not.
- A set of axioms or axiom schemata: each axiom has to be a *wff*.

- A set of inference rules.
- A set of theorems. This set includes all the axioms, plus all *wff*s which can be derived from previously-derived theorems by means of rules of inference. Unlike the grammar for *wff*s, there is no guarantee that there will be a decision procedure for deciding whether a given *wff* is a theorem or not.

1. What is formalization?
2. What are formal proofs?
3. What was the role of Davia Hilbert in the field of formal systems?
4. What are the components of Mathematical formal system?

Task 3. Look through the information of the unit "Boolean algebra". Write a short essay of 150 words using it.

Additional text to unit "Boolean Algebra"

Heyting algebras as applied to intuitionistic logic

Arend Heyting (1898-1980) was himself interested in clarifying the foundational status of intuitionistic logic, in introducing this type of structures. The case of Peirce's law illustrates the semantic role of Heyting algebras. No simple proof is known that Peirce's law cannot be deduced from the basic laws of intuitionistic logic.

A Heyting algebra, from the logical standpoint, is essentially a generalization of the usual system of truth values. Amongst other properties, the largest element, called in logic $\top$, is analogous to 'true'. The usual two-valued logic system is the simplest example of a Heyting algebra, one in which the elements of the algebra are $\top$(true) and $\bot$(false). That is, in abstract terms, the two-element Boolean algebra is also a Heyting algebra.

Classically valid formulas are those formulas that have a value of $\top$in this Boolean algebra under any possible assignment of true and false to the formula's variables – that is, they are formulas which are tautologies in the usual truth-table sense. Intuitionistically valid formulas are those formulas that have a value of $\top$in any Heyting algebra under any assignment of values to the formula's variables.

One can construct a Heyting algebra in which the value of Peirce's law is not always $\top$. From what has just been said, this does show that it cannot be derived. See Curry-Howard isomorphism for the general context, of what this implies in type theory.

In mathematics, Heyting algebras are special partially ordered sets that constitute a generalization of Boolean algebras. Heyting algebras arise as models of intuitionistic logic, a logic in which the law of excluded middle does not in general hold. Complete Heyting algebras are a central object of study in pointless topology.

UNITE 10. "NEURAL NETWORKS"

Terminology

artificial	hypothesis	neuroscience
captured	identity	node
distinct	nervous system	predictive ability
gain	neural	solving
generalization	neurons	

Task 1. Match the words with their definitions.

1) artificial	a) something that encourages you to work harder, start a new activity
2) neuron	b) formal exact similarity between two things
3)identity	c) not real or not made of natural things
4) hypothesis	d) a statement about all the members of a group that can be true in some situations
5)generalization	e) a type of sell which sends messages to other parts of the body or the brain
6) incentive	f) a system of lines, tubes, wires etc that cross each other and are connected to each other
7) network	g) an idea that is suggested as an explanation for something

Task 2. Match the words combinations and their translation.

1) neural network	a) аналитические проблемы
2) artificial neurons	b) сложность
3) a specific physiological function	c) нейронная сеть
4) intelligence problems	d) конкретные физиологические функции

5) complexity	e) предсказательная способность
6) predictive ability	f) искусственные нейроны
7) generalization error	g) объем вычислений
8) the amount of computation	h) обобщающая ошибка

Task 3. Read the text and answer the questions below.

Neural network

The term neural network was traditionally used to refer to a network or circuit of biological neurons. The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term has two distinct usages:

1. Biological neural network is made up of real biological neurons that are connected or functionally related in a nervous system. In the field of neuroscience, they are often indentified as groups of neurons that perform a specific physiological function in laboratory analysis.

2. Artificial neural networks are composed of interconnecting artificial neurons. Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. The real, biological nervous system is highly complex: artificial neural network algorithms attempt to abstract this complexity and focus on what may hypothetically matter most from an information processing point of view. Good performance (e.g. as measured by good predictive ability, low generalization error), or performance mimicking animal or human error patterns, can then be used as one source of evidence towards supporting the hypothesis that the abstraction really captured something important from the point of view of information processing in the brain. Another incentive for these abstractions is to reduce the amount of computation required to simulate artificial neural networks, so as to allow one to experiment with larger networks and train them on larger data sets.

1. What is the modern usage of the term ‘neural network’?
2. What is biological neural network made up of?
3. Where can artificial neural networks be used?

Task 4. Fill in the gaps.

Artificial neural network

a) machine learning, b) computational models, c) approximate functions, d) inputs, e) neurons, f) biological neural networks, g) pattern recognition, h) handwriting recognition, i)function, j) rule-based programming.

In1)………. and related fields, artificial neural networks (ANNs) are 2)………….. inspired by 3)……… are used to estimate or 4)……….. that can depend on a large number of 5) ……. and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "6) ………. " which can compute values from inputs, and are capable of machine learning as well as7)…….. thanks to their adaptive nature.

For example, a neural network for 8)………….. is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a 9)……….. (determined by the network's designer), the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read. Like other machine learning methods - systems that learn from data - neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary 10) …………, including computer vision and speech recognition.

Task 5. Translate the sentences into Russian.

1. Adaptation is the feature that a neural network system should provide in order to let the system be aware of changes in its environment.
2. A neural network typically consists of a number of interconnected processors, or nodes.
3. Neural networks have been used in pattern recognition, speech analysis, oil exploration, weather prediction, and the modeling of thinking and consciousness.
4. Neural networks have created an unusual amount of interest in the engineering and industrial communities by opening up new research directions and commercial and military applications.

Task 6. Discuss with your partner the following questions. Find out additional information on the net or from your partner.

1) What is the difference between biological neural network and artificial neural network?

2) What are the major benefits of neural network?

Task 7. Write an essay about application of neural network in computing. Find necessary information on the Internet. Present the report to you group mates.

Task 8. Make a list of questions which you would ask specialists in neural network sphere.

Task 9. Discuss with your group what new things you have learnt from this unit. Share your conclusions with your partner.

Additional tasks to unit "Neural networks"

Task 1. Read and translate the text. Write 10 special questions to the text.

Historywiss

Warren Mc Culloch and Walter Pitts (1943) created a computational model for neural networks based on mathematics and algorithms. They called this model threshold logic. The model paved the way for neural network research to split into two distinct approaches. One approach focused on biological processes in the brain and the other focused on the application of neural networks to artificial intelligence.

In the late 1940s psychologist Donald Hebb created a hypothesis of learning based on the mechanism of neural plasticity that is now known as Hebbian leaning. Hebbian learning is considered to be a 'typical' unsupervised leaning rule and its later variants were early models for long term potention. These ideas started being applied to computational models in 1948 with Turning's B- type machines.

Farley and Wesley A. Clark (1954) first used computational machines, then called calculators, to simulate a Hebbian network at MIT. Other neural network computational machines were created by Rochester, Holland, Habit, and Duda (1956).

Frank Rosenblatt (1958) created the perception, an algorithm for pattern recognition based on a two-layer learning computer network using simple addition and subtraction. With mathematical notation, Rosenblatt also described circuitry not in the basic perceptron, such as the exclusive-or circuit, a circuit whose mathematical computation could not be processed until after the back propagation algorithm was created by Paul Werbos (1975).

Neural network research stagnated after the publication of machine learning research by Marvin Minsky and Seymour Papert (1969). They discovered two key issues with the computational machines that processed neural networks. The first issue was that single-layer neural networks were incapable of processing the exclusive-or circuit. The second significant issue was that computers were not sophisticated enough to effectively handle the long run time required by large neural networks. Neural network research slowed until computers achieved greater processing power. Also key later advances was the back propagation algorithm which effectively solved the exclusive-or problem (Werbos 1975)

The parallel distributed processing of the mid-1980s became popular under the name connectionism. The text by Davia E. Rumelhaet and James Mc Cleeand (1986) provided a full exposition on the use of connectionism in computers to simulate neural processes. Neural networks, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain, even though the relation between this model and brain biological architecture is debated, as it is not clear to what degree artificial neural networks mirror brain function. In the 1990s, neural networks were overtaken in popularity in machine learning by support vector machines and other, much simpler methods such as linear classifiers. Renewed interest in neural nets was sparked in the 2000s by the advent of deep learning.

Task 2. Work in pairs. Answer the questions of your partner. Ask him/her yours.

Task 3. Complete the sentences in an appropriate way.

Successes in pattern recognition contest since 2009

1) to design, 2) handwritten digits problem, 3) recurrent neural networks, 4) multidimensional long short term memory, 5) pattern recognition

Between 2009 and 2012, a) ....and deep feed forward neural networks developed in the research group of Jurgen Schmidhuber at the Swiss AI Lab IDSIA have won eight international competitions in b)………. and machine learning. For example, the bi-directional and c) ………… (LSTM) of Alex Graves et al. won three competitions in connected handwriting recognition at the 2009 International Conference on Document Analysis and Recognition (ICDAR), without any prior knowledge about the three different languages to be learned. Fast GPU-based implementations of this approach by Dan Ciresan and colleagues at IDSIA have won several pattern recognition contests, including the IJCNN 2011 Traffic Sign Recognition Competition, the ISBI 2012 Segmentation of Neuronal Structures in Electron Microscopy Stacks challenge, and others. Their neural networks also were the first artificial pattern recognizers to achieve human-competitive or even superhuman performance on important benchmarks such as traffic sign recognition (IJCNN 2012), or the MNIST d)……..... of Yann LeCun at NYU. Deep, highly nonlinear neural architectures similar to the 1980 neocognitron by Kunihiko Fukushina and the "standard architecture of vision" can also be pre-trained by unsupervised methods of Geoff Hinton's lab at University of Toronto. A team from this lab won a 2012 contest sponsored by Merck e) ………. software to help find molecules that might lead to new drugs.

Task 3. Look through the information of the unit "Boolean algebra". Write a short essay of 150 words using it.

Additional texts to unit “Neural networks”

Text 1.

Employing artificial neural networks

Perhaps the greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism that 'learns' from observed data. However, using them is not so straightforward, and a relatively good understanding of the underlying theory is essential.

- Choice of model: This will depend on the data representation and the application. Overly complex models tend to lead to problems with learning.
- Learning algorithm: There are numerous trade-offs between learning algorithms. Almost any algorithm will work well with the *correct* hyperparameter*s* for training on a particular fixed data set. However, selecting and tuning an algorithm for training on unseen data requires a significant amount of experimentation.
- Robustness: If the model, cost function and learning algorithm are selected appropriately the resulting ANN can be extremely robust.

With the correct implementation, ANNs can be used naturally in online learning and large data set applications. Their simple implementation and the existence of mostly local dependencies exhibited in the structure allows for fast, parallel implementations in hardware.

Text 2.

Real life applications

The tasks artificial neural networks are applied to tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Robotics, including directing manipulators, prosthesis.
- Control, including Computer numerical control.

Application areas include the system identification and control (vehicle control, process control, natural resources management), quantum chemistry, game-playing and decision making (backgammon, chess, poker), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (e.g automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Artificial neural networks have also been used to diagnose several cancers. An ANN based hybrid lung cancer detection system named HLND improves the accuracy of diagnosis and the speed of lung cancer radiology. These networks have also been used to diagnose prostate cancer. The diagnoses can be used to make specific models taken from a large group of patients compared to information of one given patient. The models do not depend on assumptions about correlations of different variables. Colorectal cancer has also been predicted using the neural networks. Neural networks could predict the outcome for a patient with colorectal cancer with more accuracy than the current clinical methods. After training, the networks could predict multiple patient outcomes from unrelated institutions.

REFERENCES

1. Английский язык для математиков: учеб. пособие. – Волгоград: Изд-во Волгоградского государственного университета, 2001.
2. *Агабекян И.П., Коваленко П.И.* "Английский для инженеров". – Ростов-на-Дону: Феникс, 2006
3. Finite element method http://en.wikipedia.org/wiki/Finite_element_method
4. Macmillan Dictionary http://www.macmillandictionary.com/spellcheck/ british/?q=inner+product
5. Mark Ibbotson "English for Engineering", Cambridge, University Press; 2008
6. Mark Ibbotson "Professional English in Use. Engineering"(Technical English for Professionals), Cambridge, University Press
7. Peter Astley and Lewis Lansford "Engineering 1" Student's Book, Oxford University Press, 2013.
8. Professional English in Use (For Computers and the Internet), Santiago RemachaEsteras, Elena Marco Fabre, Cambridge, University Press, 2007.
9. Wolfram MathWorld, the most extensive mathematics resource http://mathworld.wolfram.com/search/?query=inner+product&x=0&y=0
10. http://en.wikipedia.org/wiki/Binary_code
11. http://en.wikipedia.org/wiki/Boolean_algebra
12. http://en.wikipedia.org/wiki/Binary_tree
13. http://en.wikipedia.org/wiki/Artificial_neural_network
14. http://www.answers.com

CONTENTS

Алябьева Александра Юрьевна,
Вандакурова Светлана Анатольевна,
Волошина Татьяна Васильевна,
Калинина Елена Геральдовна

АНГЛИЙСКИЙ ДЛЯ СТУДЕНТОВ, ИЗУЧАЮЩИХ ПРИКЛАДНУЮ МАТЕМАТИКУ

ENGLISH FOR APPLIED MATHIMATICS STUDENTS

Учебное пособие

В авторской редакции

Выпускающий редактор *И.П. Брованова*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *С.И. Ткачева*

Налоговая льгота – Общероссийский классификатор продукции
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

Подписано в печать 15.12.2014. Формат 60 × 84 1/16. Бумага офсетная. Тираж 60 экз.
Уч.-изд. л. 5,81. Печ. л. 6,25. Изд. № 320. Заказ № Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20