A Textbook of Physical Chemistry

Volume VI

Computational Aspects in Physical Chemistry

Third Edition

A Textbook of Physical Chemistry

Volume I	:	States of Matter and Ions in Solution
Volume II	:	Thermodynamics and Chemical Equilibrium
Volume III	:	Applicati of Thermodynamics
Volume IV	:	Quantum Chemistry, Molecular Spectroscopy and Molecular Symmetry
Volume V	:	Dynamics of Chemical Reactions, Statistical and Thermodynamics, Macromolecules, and Irreversible Processes
Volume VI	:	Computational Aspects in Physical Chemistry

A Textbook of Physical Chemistry

Volume VI

Computational Aspects in Physical Chemistry

Third Edition

K L Kapoor
Former Associate Professor
Hindu College
University of Delhi
New Delhi

McGraw Hill Education (India) Private Limited
NEW DELHI

McGraw Hill Education Offices

New Delhi New York St Louis San Francisco Auckland Bogotá Caracas
Kuala Lumpur Lisbon London Madrid Mexico City Milan Montreal
San Juan Santiago Singapore Sydney Tokyo Toronto

McGraw Hill Education (India) Private Limited

Published by McGraw Hill Education (India) Private Limited

P-24, Green Park Extension, New Delhi 110 016

A Textbook of Physical Chemistry, Vol. 6 (3e)

Print Edition

ISBN (13): 978-93-392-0433-4

ISBN (10): 93-392-0433-6

E-Book Edition

ISBN (13) 978-93-392-0434-1

ISBN (10) 93-392-0434-4

Managing Director: *Kaushik Bellani*

Head—Higher Education Publishing and Marketing: *Vibha Mahajan*

Senior Publishing Manager (SEM & Tech. Ed.): *Shalini Jha*

Associate Sponsoring Editor: *Smruti Snigdha*

Senior Editorial Researcher: *Amiya Mahapatra*

Senior Development Editor: *Renu Upadhyay*

Manager—Production Systems: *Satinder S Baveja*

Assistant Manager—Editorial Services: *Sohini Mukherjee*

Assistant General Manager (Marketing)—Higher Education: *Vijay Sarathi*

Senior Graphic Designer—Cover: *Meenu Raghav*

General Manager—Production: *Rajender P Ghansela*

Manager—Production: Reji Kumar

Typeset at Bharati Composers, D-6/159, Sector-VI, Rohini, Delhi 110 085, and printed at

Cover:

To the Memory of
My Parents

Preface

In recent years, the teaching curriculum of Physical Chemistry in many Indian universities has been restructured with greater emphasis on theoretical and conceptual methodology and the applications of the underlying basic concepts and principles. This shift in the emphasis, as I have observed, has unduly frightened undergraduates whose performance in Physical Chemistry has been otherwise generally far from satisfactory. This poor performance is partly because of the non-availability of a comprehensive textbook which also lays adequate stress on the logical deduction and solution of numericals and related problems. Naturally, the students find themselves unduly constrained when they are forced to refer to various books to collect the necessary reading material.

It is primarily to help these students that I have ventured to present a textbook which provides a systematic and comprehensive coverage of the theory as well as of the illustration of the applications thereof.

The present volumes grew out of more than a decade of classroom teaching through lecture notes and assignments prepared for my students of BSc (General) and BSc (Honours). The schematic structure of the book is assigned to cover the major topics of Physical Chemistry in six different volumes. **Volume I** discusses the states of matter and ions in solutions. It comprises five chapters on the gaseous state, physical properties of liquids, solid state, ionic equilibria and conductance. **Volume II** describes the basic principles of thermodynamics and chemical equilibrium in seven chapters, viz., introduction and mathematical background, zeroth and first laws of thermodynamics, thermochemistry, second law of thermodynamics, criteria for equilibrium and A and G functions, systems of variable composition, and thermodynamics of chemical reactions. **Volume III** seeks to present the applications of thermodynamics to the equilibria between phases, colligative properties, phase rule, solutions, phase diagrams of one-, two- and three-component systems, and electrochemical cells. **Volume IV** deals with quantum chemistry, molecular spectroscopy and applications of molecular symmetry. It focuses on atomic structure, chemical bonding, electrical and magnetic properties, molecular spectroscopy and applications of Molecular symmetry. **Volume V** covers dynamics of chemical reactions, statistical and irreversible thermodynamics, and macromolecules in six chapters, viz., adsorption, chemical kinetics, photochemistry, statistical thermodynamics, macromolecules and introduction to irreversible processes. **Volume VI** describes computational aspects in physical chemistry in three chapters, viz., synopsis of commonly used statements in BASIC language, list of programs, and projects.

The study of Physical Chemistry is incomplete if students confine themselves to the ambit of theoretical discussions of the subject. They must grasp the practical significance of the basic theory in all its ramifications and develop a clear perspective to appreciate various problems and how they can be solved.

It is here that these volumes merit mention. Apart from having a lucid style and simplicity of expression, each has a wealth of carefully selected examples and solved illustrations. Further, three types of problems with different objectives in view are listed at the end of each chapter: (1) Revisionary Problems, (2) Try Yourself Problems, and (3) Numerical Problems. Under *Revisionary Problems*, only those problems pertaining to the text are included which should afford an opportunity to the students in self-evaluation. In Try *Yourself Problems*, the problems related to the text but not highlighted therein are provided. Such problems will help students extend their knowledge of the chapter to closely related problems. Finally, unsolved *Numerical Problems* are pieced together for students to practice.

Though the volumes are written on the basis of the syllabi prescribed for undergraduate courses of the University of Delhi, they will also prove useful to students of other universities, since the content of physical chemistry remains the same everywhere. In general, the SI units (*Systeme International d' unite's*), along with some of the common non-SI units such as atm, mmHg, etc., have been used in the books.

Salient Features

- Brief synopsis of commonly used instructions/statements in BASIC language
- Instructions/Statements illustrated through a few preliminary computer programs
- Computer applications in the field of Physical Chemistry highlighted through inclusion of programs

Acknowledgements

I wish to acknowledge my greatest indebtedness to my teacher, late Prof. R P Mitra, who instilled in me the spirit of scientific inquiry. I also record my sense of appreciation to my students and colleagues at Hindu College, University of Delhi, for their comments, constructive criticism and valuable suggestions towards improvement of the book. I am grateful to late Dr Mohan Katyal (St. Stephen's College) and late Prof. V R Shastri (Ujjain University) for the numerous suggestions in improving the book. I would like to thank Sh. M M Jain, Hans Raj College, for his encouragement during the course of publication of the book.

I wish to extend my appreciation to the students and teachers of Delhi University for the constructive suggestions in bringing out this edition of the book. I also wish to thank my children, Saurabh-Urvashi and Surabhi-Jugnu, for many useful suggestions in improving the presentation of the book.

Finally, my special thanks go to my wife, Pratima, for her encouragement, patience and understanding.

Feedback Request

The author takes the entire responsibility for any error or ambiguity, in fact or opinion,that may have found its way into this book. Comments and criticism from readers will, therefore, be highly appreciated and incorporated in subsequent editions.

K L Kapoor

Publisher's Note

McGraw-Hill Education (India) invites suggestions and comments from you, all of which can be sent to *info.india@mheducation.com* (kindly mention the title and author name in the subject line).

Piracy-related issues may also be reported.

Contents

1 Synopsis of Commonly used Statements in BASIC Language

1.1 INTRODUCTION

The BASIC is a high-level computer language which stands for "Beginner's All-purpose Symbolic Instruction Code.

A program in BASIC is a formulation of a given problem in terms of instructions (also called statements) of BASIC language. Each statement is inserted in a new line which may or may not start with a number. In fact, numbering is optional. However, during a branching operation, the statement to which instruction is to be transferred must carry the appropriate number. More than one statement in a line may be inserted. These are separated by colons. However, if such statements are inserted after the statement which uses relation operator or logical operator, these statements are also not executed if the result of the operation is false.

If a statement is not written in accordance to the recommended format, an error message appears on the monitor indicating the nature of error. This may be rectified in order to execute the program successfully.

REM Statement

The REM (from remark) statement is used to add comment or explanatory note wherever required in a BASIC program. It is a nonexecutable statement. Another way of achieving it to add an apostrophe sign in the beginning of the comment statement.

END Statement

The END statement indicates the end of a BASIC program and is inserted as the last statement of the program.

STOP Statement

The STOP statement is used to terminate a BASIC program anywhere within the program. This is equivalent to transferring the control to the END statement.

A brief description of instructions commonly used in BASIC to formulate a computer program for a given problem is in order.

1.2 NUMBERS (OR CONSTANTS) AND STRINGS

Numbers

Numerical quantities are referred to as numbers or constants. These include integral as well as decimal quantities. Examples are 22, 3.14 and 8.314. A scientific number such as 6.022×10^{23} and 1.38×10^{-23} are represented in E notation as 6.022E+23 and 1.38E–23, respectively.

A negative quantity starts with a negative sign (" – ") and if no sign appears, it is understood to be a positive quantity.

Strings

To express certain information in the program, such as name of a molecule, strings are used. These may include any alphabets, numbers and special characters (such as +, –, *, /, $, ..., etc.)

String constants are enclosed within the double quotation marks such as "OXYGEN". The maximum length a string constant can be upto 254 characters.

1.3 VARIABLES

A number or a string may be represented by a name, known as variable. However, the variable name must not be the same as that of a BASIC reserved word (such as INT, ABS, SIN, etc).

The numerical or string variable always starts with an alphabet and may be followed by more alphabets or integers, However, the string variable always ends with the special character "$". The end letter is referred to as suffix. Besides the suffix, "$", the suffixes shown in Table 1 are also used to represent the different types of variables.

Table 1 Suffixes used in BASIC

Suffix	*Example*	*Variable*
%	A%	Integral
! (or no suffix)	A!	Real single-precision
#	A#	Double-precision
$	A$	String

The same variable name with different suffixes represent different type of variables.

A double-precision constant in scientific notation is represented in D notation instead of E notation (e.g. 6.22D + 23).

A single-precision quantity contains *six* or *seven* significant figures while a double-precision quantity contains *sixteen* significant numbers.

Integral quantities lies within the range of –32768 to 32767.

Real quantities lies within the range of 2.9E – 39 to 1.7E + 38.

A variable name can have a maximum of 40 alphabets or digits. It must start with an alphabet and should not include any special characters (such as comma, asterisk, period, etc.) except at the end indicating the type of variable.

1.4 MATHEMATICAL OPERATIONS

The mathematical operations between two quantities are executed by inserting the operators (Table 2) between the two quantities.

Table 2 Operators used in BASIC

Operator	*Explanation*
+	Plus sign for addition
−	Minus sing for subtraction
*	Asterisk for multiplication
/	Slash for division
^	Upper carret for exponentiation
\	Back slash for integer division
MOD	MOD for integer remainder

In integer division and integer remainder, each of the two given number is first rounded to an integer followed by division to yield a truncated quotient or remainder. A few examples to illustrate these two operations are as follows.

Operation	*Answer*	*Operation*	*Answer*
`13 \ 4`	3	`12 MOD 5`	2
`8.7 \2.8`	3	`8.7 MOD 2.8`	0
`8.1\2.7`	2	`8.1 MOD 2.7`	2
`8.4 \ 2.3`	4	`8.4 MOD 2.3`	0

Hierarchy of Operations

If a mathematical statement involves more than one operation, these are executed from left to the right of the expression with the following hierarchy.

(i) Exponentiation, **(ii)** multiplication and division, and **(iii)** addition and subtraction.

Use of Parentheses

In a mathematical statement involving more than one operation, if some operations are to be carried out prior to the hierarchy of operations mentioned above, these are enclosed within the pairs of round parentheses. If there is a nest of pairs of parentheses, the operations within the innermost pairs of parantheses are performed first followed by the operations within the second innermost pairs, and so on. After these evaluations, rest of the operations are carried out from left to the right of the expression following the hierarchy mentioned above.

Enlarging a String Variable

A string variable may be appended by adding string variable(s) to it. For example, if

```
A$ = "Delhi"
B$ = "University"
```

then `C$ = A$ + " " + B$`

causes the string constant `C$` to represent `"Delhi University"`.

1.5 ASSIGNING VALUE OF A VARIABLE

The value of a numerical or string variable can be assigned in the following ways.

```
T = 273              (or LET T = 273)
A$ = "Oxygen"        (or LET A$ = "Oxygen")
```

The value of a string variable is enclosed within the double quotation.

Assigning the Value of a Variable During the Execution of a Programme

For this, either INPUT or READ and DATA statements are used.

INPUT Statement

The INPUT statement goes as follows.

```
INPUT  T
INPUT  A$
```

When the INPUT statement is encountered during the execution of a program, a question mark (?) appears on the console and the execution of the program is stopped until the required data is supplied on the key board followed by pressing the "ENTER" key.

It is not necessary to write separately INPUT statement for each variable. In fact, the values of a list of variables can be supplied by inserting a single INPUT statement as shown in the following.

```
INPUT  T, A$
```

Each variable is separated by a comma, In such a case, when the question mark appears on the console, the values of all variables listed in the INPUT statement have to be supplied together, each one is separated by a comma. The only care to be taken is that there should be one to one correspondence between the types of variables in the list and the supplied values of the variables. In the example given above, the values are supplied as follows.

```
? 12.5, OXYGEN
```

To increase the clarity about the nature of input variable, one can insert a string constant mentioning the nature of variable within a double quotations immediately after the INPUT statement followed by a semicolon and then the name of the variable. For example, if the variable T stands for temperature in kelvin, the variable T in INPUT statement may be inserted as follows.

```
INPUT "Temperature in kelvin, T = "; T
```

where the information within the double quotations mark is the string constant. When this statement is executed, the string constant appears as such on the console followed by a question mark and the execution of the program is stopped until the value of T is supplied. The purpose of semicolon before T in the input statement is to hold cursor immediately after printing the string constant. Only one string constant can be inserted in one INPUT statement.

READ and DATA Statement

The READ and DATA statements go as follows.

```
READ  T
DATA  273
```

```
READ  A$
DATA  OXYGEN
```

The purpose of the DATA statement is to assign appropriate values to the variables listed in the READ statement. It is not necessary to write one DATA statement for each READ statement. In fact, the values of variables listed in sequence in all the READ statements can be supplied in a single DATA statement. In the above example, one can write only one DATA statement instead of two as shown in the following.

```
DATA  273, OXYGEN
```

It is also not necessary to write only one variable in a READ statement. In fact, a list of variables can be inserted one after the other separated by commas. For example,

```
READ  T, A$
DATA  273, Oxygen
```

The DATA statement can be inserted anywhere in the program, prior or after the READ statement(s). The only care to be taken is to have one to one correspondence between the types of variables appeared sequentially in the READ statement(s) and the assigned values of variables in the DATA statement. Reading of data sequentially is carried out by an internal pointer which is advanced to the next data item once a data is read. In fact, there are two such internal pointers, one for the numeric data and the other for string data.

The RESTORE Statement

The RESTORE statement causes the internal pointer for reading a data to reset to the first data item of the proper type so that the subsequent reading of the data starts from the beginning of the DATA items. This way, the same data is reused for the variables in the subsequent READ statements.

1.6 PRINTING VALUE OF A VARIABLE

The values of variables (numeric or string) are printed on the console by the PRINT statement. The list of variables to be printed are inserted after the keyword PRINT. Successive variables are separated by either commas or semicolons.

If commas are used, five variables are printed in one line with equal spacing in between. In fact, each line is divided in five zones of equal length and one output value is printed in each zone. If a comma is also inserted in the last variable in a PRINT statement, then the output of the subsequent PRINT statement, is continued on the same line. The number of such items printed on the same line depends upon the number of free zones in the same line. If the space required to print a variable is larger than the space in a zone, then the subsequent variable is printed in second next zone.

If semicolons are used, the variables are printed with less spacing in between them. This help printing more than five outputs in a line. If a semicolon also appears after the last entry in a PRINT statement, then the output of the subsequent PRINT statement is continued on the same line.

If a PRINT statement does not contain any variable, then a blank line is appeared in between the two lines.

The string constant within a double quotation marks can be inserted before the variable name separated by a semicolon. This provides easy way of spelling the nature of variables to be printed. A PRINT statement may contain more than one quotation marks separated from a variable name or string constant by either a comma or a semicolon. The following example illustrates the above fact.

Suppose T = 273 K and P = 2 atm. These values can be printed as shown in the following.

Statement `PRINT "T = "; T; "K", "P = "; P; " atm"`
Output `T = 273K P = 2 atm`

Note the two entries are separated by a comma thus, the two items are printed in the first two zones of a line. If there appears two commas in between the two entries, then the two items are printed in the first and third zones of a line.

Numerical output quantities are printed as follows.

An integer quantity that contains eight or fewer digits are printed as in integer number. If there are more than eight digits, it is rounded to six significant digits and is printed as a decimal number with an exponent.

A decimal quantity is printed in a decimal notation. If a quantity contains more than six digits (including leading zeros to the right of the decimal point), it is rounded to six digits and is printed in the exponential notation.

Formating Output Data

The numeric data can be printed in a decimal or exponential format by using the key word `PRINT USING`.

For a decimal format, the instruction to be given is

```
PRINT USING "##.###"; A; B; C
```

where hases in the string `"##.###"` indicates the number of digits to be printed before and after the decimal point. Fraction extending beyond the indicated number of digits are rounded. If a number is negative, then the miuns sign is counted within the number of hases before the decimal point. The string is followed by a semicolon and the list of variables separated from each other also by semicolons.

For an exponential format, the instruction to be given is

```
PRINT USING "##.###^^^^"; A; B; C
```

The four carets are for the exponential notation such as E+02 and E−02.

The spacing in the printed values of A, B and C may be created by inserting more hases before the decimal or inserting some blanks before the hases.

If commas are to be inserted before the decimal point after each three digits from the decimal point, the instruction to be given is

```
PRINT USING "######,.##"; A
```

1.7 BRANCHING STATEMENTS

Unconditional Branching Operation

The statements in a `BASIC` program are executed in the same order as they appear. However, the control of execution of a statement can be altered by using the keyword `GOTO` followed by the number of the statement to which control is to be transferred, e.g. `GOTO 5`.

Multiple Branching

In multiple branching, the control of execution of a statement can be directed to different statements. The format of the statement is

`ON` Numeric variable or arithmetic expression `GOTO` List of statement numbers

For example, the statement `ON K GOTO 10, 20, 30` implies the following.

The control is passed over to the statement number `10, 20, 30` depending upon the value of `K` equal to `1, 2` and `3`, respectively. If `K` is not an integer, it is truncated to integer by ignoring the decimal portion.

Conditional Branching — IF ____ THEN Statement

If the execution of a statement depends upon the satisfaction of a relation or condition, then the following statement can be used to achieve the goal.

`IF` relation or condition `THEN` statement or number of statement to be executed

If the relation or condition is not satisfied, the statement next to the `IF___THEN` statement is executed. The following two programs illustrate the use of `IF___THEN` statement.

- Suppose, it is desire to add the given ten numbers. This can be achieved as follows.

```
    SUM = 0
    I = 1
5   INPUT X
    SUM = SUM + X
    I = I + 1
    IF I <= 10 THEN 5
    PRINT SUM
```

- Suppose it is desired to find the roots of a quadratic expression. ($ax^2 + bx + c = 0$). This can be achieved as follows.

```
    DISC = b * b - 4 * a * c
    IF DISC < 0 THEN 10
    SQDISC = DISC ^ 0.5
    ROOT1 = (-b + SQDISC) / (2 * a)
    ROOT2 = (-b - SQDISC) / (2 * a)
    PRINT "ROOT1 = "; ROOT1, "ROOT2 = "; ROOT2
    GOTO 15
10  PRINT "ROOTS ARE IMAGINARY"
15  ____
```

1.8 USE OF LOGICAL OPERATORS

Relation Operators

Relation operators are used to compare two quantities (numbers, variables or strings) so as to direct conditional branching depending upon the condition set in the relation operators. These operators are described in Table 3.

Table 3 Operators used in BASIC

Condition	*Operator*
Equal to	=
Not equal to	< >
Less than	<
Less than or equal to	< =
Greater than	>
Greater than or equal to	> =

Logical Operators

If the conditional branching depends on the outcome of two relation operators (either both are true or one of them is true) then these two operators are connected through logical operators (AND, OR and NOT). The following examples illustrate their use.

(i) `IF X < 10 AND Y > 150 THEN 100`

If both the conditions of `X < 10` and `Y > 150` are satisfied, the control is passed over to the statement number 100, otherwise the instruction is passed over to the statement next to this logical operation.

(ii) `IF M > 100 OR N$ = "Oxygen" THEN 100`

If either of the conditions of `M > 100` or `N$ = "Oxygen"` is satisfied, the control is passed over to the statement 100 otherwise the next statement of this logical operation is executed.

(iii) `IF NOT X < 10 AND Y > 150 THEN 200`

The control is passed over to the statement 200 provided both the conditions of `X < 10` and `Y > 150` are not satisfied. This statement is equivalent to

IF `X >= 10` and `Y <= 150 THEN 200`

IF___THEN___ELSE Statement

In this case, the statement (or statements) following THEN are executed if the given relation or condition is satisfied, otherwise, the statement (or statements) following ELSE is executed. Statements illustrating IF__THEN__ELSE statement are as follows.

```
IF A > 0 THEN K = 1 ELSE K = 2
IF b * b >= 4 * a * c THEN PRINT "ROOTS ARE REAL" ELSE PRINT "ROOTS ARE
IMAGINARY"
IF A > 0 THEN PRINT A: GOTO 10 ELSE B = A: GOTO 50
```

Nested IF___THEN___ELSE Statement

If there are more than one alternative of checking a relation or condition, than one can use the nested IF___THEN ___ELSE statement as illustrated in the following.

```
        VA = 50: CA = 0.1: CB = 0.05: VEQ = VA * CA/CB: VB = 0
5       IF VB = 0 THEN
        H = CA
        ELSEIF VB < VEQ THEN
```

```
        H = (VA * CA - VB * CB)/(VA + VB)
        ELSEIF VB = VEQ THEN
        H = 1.0E-7
        ELSEIF VB > VEQ THEN
        OH = (VB - VEQ) * CB/(VA + VB)
        H = 1E-14 / OH
        END IF
        PRINT H
        IF VB > VEQ + 5 THEN 10
        VB = VB + 0.01
        GOTO 5
10      END
```

In the nested IF____THEN____ELSE statement, the end of the nest is done with the statement END IF.

1.9 USE OF LOOP STATEMENT

If a set of statements is to be repeated many times, the loop of FOR___TO___NEXT is the most convenient way of executing such statements. The loop comprises all statements included between the FOR___TO and the NEXT statements. The following example illustrates such a loop.

```
        K = 1: N = 5: M = 1: FACT = 1
5       FOR I = K TO N STEP M
        FACT = FACT * I
        PRINT I, FACT
10      NEXT I
        _______
        _______
```

From the statement FOR I = K TO N STEP M to the statement NEXT I comprises a loop of I. The execution starts with I = K and all the statements prior to NEXT I are executed. On encountering NEXT I, the instruction is passed back to the start of the loop. The value of I is increased by the step M and is compared with the value of N. If its value is less than or equal to the value of N, the execution of the loop is continued. However, If the value of I is more than N, the excution of the loop is not carried out but the instruction is passed over to the statement following the statement NEXT I.

The loop will not be executed if

1. the values of K and N are equal and the step size M is zero.
2. the value of N is less than the value of K and the step size is positive.

The variables K, N and M can have any positive or negative integral or nonintegral values. The conditions to be satisfied are as follows.

1. The value of N may be equal to K and the value of M is nonzero. In this case, the loop is excuted only once.
2. For a positive value of M, the value of N must be equal to or greater than that of K.

3. For a negative value of M, the value of N must be equal to or less than that of K. In this case, the value of K is decreased by a step M and the execution of the loop is continued until the value of K becomes less than N.
4. Control within a loop can be transferred outside the loop and not vice versa.
5. The running variable I may be used within the loop but its value should not be altered.

Nested Loops

One can use a nest of loops in which a loop is inserted within another loop. The following example illustrates the framework of nested loops.

Here, the K loop is within the J loop which itself is within the I loop. There is another L loop within the I loop. Each inner loop is completely imbeded within an outer loop. Control can be transferred from an inner loop to a statement in an outer loop or to a statement outside the entire nest of loops. The reverse of the above is not permitted.

Conditional Looping

One can generate a conditional looping by using `WHILE` and `WEND` statements. The condition of executing the loop is defined in the `WHILE` statement. So long the condition is satisfied, the loop is executed otherwise not. The end of conditional loop is indicated by the `WEND` statement. The following example of getting real roots of a quadratic expression $ax^2 + bx + c = 0$ illustrates the `WHILE` and `WEND` statements.

```
FOR I = 1 TO 10
INPUT A, B, C
DISC = B ^ 2 - 4 * A * C
WHILE DISC >= 0
ROOT1 = (- B + SQR(DISC))/(2 * A)
ROOT2 = (- B - SQR(DISC))/(2 * A)
PRINT A, B, C, ROOT1, ROOT2
WEND
NEXT I
END
```

1.10 SUBSCRIPTED VARIABLES

Sometimes, a collection of either numerical quantities or strings is known by a single variable name. The individual elements of such a collection is represented by a subscripted variable. Any of such element can be referred to by stating the variable name, followed by the value of subscript enclosed within parantheses. The variable name of a collection of numerical quantities is a single letter. For a collection of strings, it is a single letter followed by a dollar sign.

For one-dimensional collection of elements (known as an array), the subscript is a single integral number and for two-dimensional array, the subscripts are two integral numbers separated by a comma. A subscript is never a negative number. Besides a constant, a subscript may be a variable, arithmetic expression or a function name. If the value of these are noninteger, these are truncated to yield integral value.

DIM statement

`BASIC` automatically assigns 11 elements for a one-dimensional array and these are numbered from 0, 1, . . ., 10. For a two-dimensional array, 121 elements (11 rows and 11 columns) are automatically assigned. However, if the array contains larger number of elements, the size of array should be defined by using the keyword `DIM` (from dimension), followed by one or more array names separated by commas. The size of the array is mentioned by a number enclosed within parentheses immediately after the array name.

Illustration `DIM A(100), B(10, 10), C$(50)`

BASIC assigns 101 elements for A, 11 rows and 11 columns for B and 51 elements for `C$`.

The DIM statement can be inserted anywhere in a BASIC program. However, it is a good practise to insert these in the beginning of the program. Also, an array with lesser than 11 elements (or 11 rows and 11 columns for a two-dimensional array) can be included in the DIM statement. Though, their noninclusion has no affect on the execution of a program, yet their inclusion help assigning lesser storage than the storage automatically assigned in BASIC.

Dimensional arrays are very helpful in the arithmetic operations involving vectors and matrices.

1.11 LIBRARY FUNCTIONS

Some prewritten programs are available in BASIC which allows the direct evaluation of some functions. These are known as library functions. Each function is accessed simply by stating its name followed by relevant arguments within the parentheses. A few such functions are described in Table 4.

ILLUSTRATIONS OF A FEW LIBRARY FUNCTIONS

CHR$ Function

The use of this function may be illustrated by inserting a character on the screen which is not available on the keyboard. For example,

```
PRINT CHR$(94)
```

causes the appearance of the character ↑ on the screen.

Table 4 Some of the Library Functions

Function	*Its Format*	*Description*
ABS	`Y = ABS(X)`	Calculate the absolute value of X
ASC	`Y = ASC(X$)`	Returns ASCII code† of X$ (which is a single character)
ATN	`Y = ATN(X)`	Calculate arctangent of X
CDBL	`Y# = CDBL(X)`	Converts X to double precision
CHR$	`Y$ = CHR$(X)`	Returns a character of ASCII code X
CINT	`Y% = CINT(X)`	Convert X into an integer
COS	`Y = COS(X)`	Calculate cosine of X, X is in radians
CSNG	`Y = CSNG(X#)`	Convert double precision X# into single precision Y
EXP	`Y = EXP(X)`	Calculate the exponential of X
F1X	`Y% = FIX(X)`	Convert X into truncated integer
HEX$	`Y$ = HEX$(X)`	Converts from decimal to hexadecimal
INPUT$(n)	`Y$ = INPUT$(4)`	Return a multicharacter string from the keyboard. Here, n = 4 implies 4-character string.
INT	`Y% = INT(X)`	Returns the largest integer that does not exceed the specified value
LEFT$	`Y$ = LEFT$(X$, 3)`	Returns the left most n character of a string, here n = 3
LEN	`Y = LEN(X$)`	Returns the number of characters in a string
LOG	`Y = LOG(X)`	Calculate natural logarithm of X
MID$	`Y$ = MID$(X$, 2, 3)`	Returns an n-character of a string starting from the mth location, here m = 2, n = 3
OCT$	`Y$ = OCT$(X)`	Converts from decimal to octal
RIGHT$	`Y$ = RIGHT$(X$, 3)`	Returns the right most n(= 3) characters of a string
RND	`Y = RND`	Returns a random number from 0 to 1
SGN	`Y = SGN(X)`	Gives the sign of X, Y is +1 or –1 depending upon positive or negative value of X
SIN	`Y = SIN(X)`	Calculate the sine of X, X is in radians
SPACE$	`PRINT X; SPACE$(5); Y`	Returns a string which is a sequence of 5 blank spaces
SPC	`PRINT X; SPC(5); Y`	5 Blank spaces in a PRINT statement between X and Y
SQR	`Y = SQR(X)`	Calculate square root of X
STR$	`Y$ = STR$(1000)`	Converts a numerical quantity into a string
TAB	`PRINT X; TAB(10); Y`	Printing Y in the specifid position
TAN	`Y = TAN(X)`	Calculate the tangent of X, X is in radians
STRING$(m, n) (m = 2, n = 42)	`Y$ = STRING$(2, 42)`	Returns an m-character string of characters whose ASCII code is n. In the present case, the ASCII code of 42 is *. Therefore, `Y$ = "**"`. This is also equivalent to `STRING$ (2, "*")`.

† See Section 1.13 for ASCII code.

LEFT$ Function

The function `LEFT$(X$, n)` causes the return of a string consisting of `n` leftmost characters of the string `X$`. For example, the program

```
X$ = "INDIA"
FOR I = 1 TO LEN(X$)
PRINT LEFT$(X$, I)
NEXT I
```

causes the printing of the following characters on the screen.

```
I
IN
IND
INDI
INDIA
```

RIGHT$ Function

The function `RIGHT$(X$, n)` causes the return of a string consisting of `n` right most characters of the string `X$`. For example, the program

```
X$ = "INDIA"
FOR I = 1 TO LEN(X$)
PRINT RIGHT$(X$, I)
NEXT I
```

causes the printing of the following characters on the screen.

```
A
IA
DIA
NDIA
INDIA
```

MID$ Function

The function `MID$(X$, n, m)` causes the return of a string consisting of m characters starting from `nth` character of the string `X$`. For example, the program

```
X$ = "VOLUME"
FOR I = 1 TO LEN(X$)
PRINT MID$(X$, I, 1)
NEXT I
```

causes the following printing of the string `X$` in a vertical array, i.e.

```
V
O
L
U
M
E
```

If the value of m (i.e. number of characters to be returned) is not provided, the MID$ statement causes the return of all the characters from the nth character up to the end of the string. For example,

```
PRINT MID$(X$, 5)
```

will case the printing of `ME` if `X$` is `"VOLUME"`.

The `MID$` function is also used in replacing characters in a variable. For example, in a string variable

```
X$ = "MUMBAI UNIVERSITY"
```

if `MUMBAI` is to be replaced by `BARODA`, then the statement to be given is

```
MID$(X$, 1, 6) = "BARODA"
PRINT X$
```

The output of the above statements will be `"BARODA UNIVERSITY"`. Even a simple statement

```
MID$(X$, 1) = "BARODA"
```

will also replace `MUMBAI` by `BARODA`. The replacement starts from the indicated number in the MID$ statement and all characters equivalent to the number of characters of the string constant inserted on the right side are replaced.

RND Function and RANDOMIZE Statement

The RND function (statement to be given is Y = RND) causes the generation of random numbers by using a fixed computational procedure Every time a program containing the RND function causes the generation of the same sequence of random numbers. If it is desireable to generate a different sequence of random numbers, the statement `RANDOMIZE` may be used. Its purpose is to provide a different starting point for the generation of random numbers. This statement is inserted prior to the use of RND function.

INPUT$ Statement

`INPUT$` function returns a string of `n` character from the keyboard. The statement to be given is `Y$ = INPUT$(n)`, where n can have values 1, 2, . . . and so on.

Unlike the `INPUT` statement, the use of this function does not generate a question mark requesting input data. The input string is simply entered from the keyboad without pressing the Enter key. The inserted character is not displayed on the screen. This is helpful in halting the program execution at any stage.

1.12 FUNCTIONS AND SUBROUTINES

A Single-Line Numeric Function

Sometime, the evaluation of a certain expression is required at different places of a program. To avoid writing of the expression repeatedly, the programmer may define his or her own function and then use this function just like a library function wherever it is required in the program. A single-line function is defined as follows

```
DEF FNA(arguments) = Defining expression of the functions
```

where,

DEF (from definition) is the keyword for defining the function, FNA(arguments) is the name of the function. In this, the first two letters FN are fixed and the third letter may be any alphabet from A to Z. The name is followed by nonsubscripted arguments (if any) separated by commas and enclosed within the parentheses. This is followed by equal sign and then the defining expression of the function

Illustrations

```
DFF FNA (X, Y, Z) = SQR(X ^ 2 + Y ^ 2 + Z ^ 2)
DFF FNB (T, X) = R * T * log(X)
```

In the function FNB, R is a variable whose value must be defined before using the function in the program.

Whenever, the function is required it is inserted by its name along with the appropriate set of arguments, separted by commas and enclosed in parentheses. The inserted arguments need not be the same as those in the function definition. These can be any constants, variables or algebraic expressions but their number and nature (numerical or string) must be the same. The inserted arguments are used in place of the corresponding arguments mentioned in the function definition. For example, the function FNA defined above may be used as follows.

```
Y = FNA(A, B, C)
A = FNA(2.0, K(I), 5 * (P + Q))
B = FNA(LOG(P), SQR(P), Q)
```

The above statements are equivalent to the following expressions.

```
Y = SQR(A ^ 2 + B ^ 2 + C ^ 2)
A = SQR(2.0 ^ 2 + K(I) ^ 2 + (5 * (P + Q))^ 2)
B = SQR(LOG(P) ^ 2 + SQR(P) ^ 2 + Q ^ 2)
```

Note that in the function reference, the arguments may be expressed as a subscripted variable, an algebraic expression and a reference to a library function.

A Single-Line String Function

A single-line string function can be defined in a similar way as a single-line numeric function. The only difference is that the name of the function must end with a dollar sign. Example is

```
DFF FNA$ = "MIXING OF COMPONENTS"
```

A numeric and a string function having the same three letters (such as FNA and FNA$) represent different functions and can be used in the same program.

Multiline Function

If the defining expression of a function requires more than one statement, one can incorporate these in a multiline function with the following format.

```
DFF FNA(arguments)
____
____
FNA = ____
END DEF
```

where DFF is the keyword for defining the function, and FNA(arguments) is the name of the function. In this, the first two letters FN are fixed and the third letter may be any alphabet from A to Z. The name is followed by nonsubscripted arguments (if any) separted by commas and enclosed in parenthesis.

Unlike a single-line function, the defining expression of the function is not included in this statement. This is, however, inserted in the following statements, one of which is FNA = ____ which assigns a value to the function name and this value is returned to the main program.

END DEF is the last line of a multiline function which indicates the end of definition of the function.

A multiline function is referred to by the same way as a single-line function.

A multiline function may be illustrated by generating the factorial of a number.

```
    DEF FNA (N)
    FACT = 1
    IF N = 0 THEN 5
    FOR I = 1 TO N
    FACT = FACT * I
    NEXT I
5   FNA = FACT
    END DEF
```

Subroutine

Another way to use some portion of a program at different locations of the main program is to treat the portion as a subroutine. The latter includes a set of statements which starts with a number assigned to the first statement and ends with the keyword RETURN. In the main body of the program, the subroutine is referred to by the keyword GOSUB followed by the number assigned to the first statement of the subroutine. Execution of this statement causes the transfer of the control to the subroutine. The execution of subroutine is continued until the statement of RETURN is encountered which causes the transfer of control back to the statement following GOSUB in the main program.

The following example of generating a factorial of a number illustrates the use of a subroutine.

The arrows shown on the left side indicate only the transfer of control and is not the part of the program.

Nest of Subroutines

One can also use a nest of subroutines in which one subroutine has a reference of another subroutine. The following example illustrates the use of the nested subroutines.

The arrows shown in the program indicate the transfer of control and are not part of the program.

The first arrow indicates the transfer of control to the statement number 100 which indicates the start of first subroutine. During the execution of this subroutine, encountering of `GOSUB 150` causes the transfer of control to the second subroutine which starts with the statement number 150. This is indicated by the second arrow. When the statement `RETURN` is encounted in the second subroutine, the control is transferred to the following statement of `GOSUB 150` in the first subroutine. This is shown by the third arrow. Finally, when the statement `RETURN` is encountered in the first subroutine, the control is transferred back to the following statement of `GOSUB 100` in the main body of the program. This is shown by the fourth arrow. Thereafter, the execution of the remaining statements in the main program is continued.

Please note that if subroutine A references subroutine B, then subroutine B cannot reference subroutine A. However, subroutine B may be referenced from the main part of the program.

1.13 AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

In the computer memory, every thing is stored in binary digits. Each digit, letter and special character is represented by its own unique number.

The conversion from characters to numbers, and vice versa, are carried out automatically within the computer. American standard code for information interchange (`ASCII`) is one of the commonly used coding scheme. Table 5 describes the numerical representations of some of the commonly used characters.

CHANGE Statement

The `CHANGE` statement is used to convert a character into its `ASCII` number and vice versa. The formation of this statement is

`CHANGE` string variable `TO` numeric list

Table 5 A Few Commonly ASCII Representations of Some of the Characters†

Character	*ASCII Value*	*Character*	*ASCII Value*	*Character*	*ASCII Value*
Control		Y	89	Σ	228
Characters	00-31	Z	90	σ	229
Blank	32	[	91	μ	230
!	33	\	92	τ	231
”	34	]	93	Φ	232
#	35	∧	94	θ	233
$	36	–	95	Ω	234
%	37	‘	96	δ	235
&	38	a	97	∞	236
’	39	b	98	ϕ	237
(	40	⋮	⋮	∈	238
)	41	y	121	Π	239
*	42	z	122	≡	240
+	43	{	123	±	241
,	44	¦	124	≥	242
–	45	}	125	≤	243
.	46	~	126	⋮	⋮
/	47	⬠	127	÷	246
0	48	⋮	⋮	≈	247
1	49	1/2	171	°	248
⋮	⋮	1/4	172	·	249
9	57	↓	173	-	250
:	58	<<	174	√	251
;	59	>>	175	η	252
<	60	⋮	⋮	2	253
=	61	⊥	193	⋮	⋮
>	62	⊤	194	blank	255
?	63	⋮	⋮		
@	64	α	224		
A	65	β	225		
B	66	Γ	226		
⋮	⋮	π	227		

†*There are a total of 255 characters.*

This statement causes each character in the string variable to be converted to its numerical equivalent and stored in a numeric list. The first element of the numeric list has a subscript of zero and this indicates the number of encoded characters in the numeric list.

For example if A$ is "LIBRARY", than its conversion to numerical list may be achieved by using the statement.

```
CHANGE A$ TO A
```

The assigned value of the numerical list represented by the variable will be as follows.

```
A(0) = 7        (there are seven characters in A$)
A(1) = 76       (the numerical equivalent of L)
A(2) = 73       (the numerical equivalent of I)
A(3) = 66       (the numerical equivalent of B)
A(4) = 82       (the numerical equivalent of R)
A(5) = 65       (the numerical equivalent of A)
A(6) = 82       (the numerical equivalent of R)
A(7) = 89       (the numerical equivalent of Y)
```

The numerical equivalent of any character in the string A$ may be referred to by the corresponding subscripted variable A.

The inverse operation of a number list to the corresponding character list is also achieved by the CHANGE statement. The format of such a change is

```
CHANGE A TO A$
```

The variable A(0) stores the number of encoded numbers in the string list.

Conversion Using Library Functions

Another way of converting a single character to its ASCII numerical quantity, and vice versa, is to use the library functions ASC and CHR$.

The format of the statements are as follows.

```
A = ASC(L)     ; ASCII equivalent of L is 76. Hence A gets the value of 76
A$ = CHR$(X)   ; If X is 80, then the corresponding character is P. Hence, A$ represents P.
```

The above two library functions can be used in any type of statement (e.g. conditional branching, printing, etc.).

1.14 COMPUTER GRAPHICS

BASIC includes instructions to display the data in graphical form on the monitor.

Graphical Mode

The fundamental elements in the display of graphs are small dots, called pixels (picture elements). To bring the computer to the graphical mode, the statement to be given is

```
SCREEN 1  or  SCREEN 2
```

In SCREEN 1, known as medium-resolution graphics mode, there is a provision of 320 pixels in the horizontal direction and 200 pixels in the vertical direction. The position of a pixel on the monitor is governed by its coordinates. The scheme of coordinates is shown in Fig. 1.

In the scheme shown in Fig. 1, the top left-most corner of the monitor is assigned coordinates (0, 0). The bottom right-most corner of the monitor is assigned coordinates (319, 199).

Fig. 1 Coordinate scheme in a medium-resolution graphics (SCREEN 1)

With this, the coordinates of top right-most and bottom left-most corners of the monitor have the coordinates (319, 0) and (0, 199), respectively. To display a dot at the desired position, the statement to be given is

```
PSET (X, Y)
```

where `X` and `Y` are the coordinates of the desired position of pixels on the monitor.

In `SCREEN 2`, known as high-resolution graphics mode, there is a provision of 640 pixels horizontally and 200 pixels vertically, with the coordinate system shown in Fig. 2.

Fig. 2 Coordinate scheme in a high-resolution graphics (SCREEN 2).

Colour Scheme

The `SCREEN 1` mode has a provision to display graphs in colour whereas in `SCREEN 2`, only black and white format is available. The choice of colour in `SCREEN 1` is specified by the statement

```
COLOR X, Y
```

where X and Y are the two parameters which decide the background colour and the choice of palette, respectively. The parameter X can have any one value from 0 to 15, thus providing with a choice of 16 different colours as shown in Table 6.

Table 6 Colour variables in BASIC

Value of X	*Colour*	*Value of* X	*Colour*
0	black	8	gray
1	blue	9	light blue
2	green	10	light green
3	cyan	11	light cyan
4	red	12	light red
5	magneta	13	light magneta
6	brown	14	yellow
7	white	15	high intensity white

The parameter Y has a value of either 0 or 1. Thus, there are two schemes for the graphical display. These are shown in Table 7.

Table 7 Palletes used in Colour Scheme of BASIC

Pallete 0		*Pallete* 1	
Number	*Colour*	*Number*	*Colour*
0	background colour	0	background colour
1	green	1	cyan
2	red	2	magneta
3	brown	3	white

The choice of colour in either of the above two palettes is indicated along with the graphic statements `PSET, LINE` and `CIRCLE`. In the statement PSET, this number is indicated immediately after the closing parentheses with a comma in between. For example

```
PSET (X, Y), 1
```

If no number is mentioned, it is automatically taken to be 3.

Colour Scheme in Textual Mode

In textual mode (`SCREEN 0`), the COLOR statement involves three parameters specifying the foreground (text) colour, the background colour and the border colour. The COLOR statement is inserted immediately after the statement `SCREEN` which specifies text mode, with colour enabled. For example,

```
SCREEN 0
COLOR 14, 1, 4
```

These statements set yellow text against blue background with a red border.

The textual mode is a default mode when the execution of the program is started. The whole screen is divided into 25 rows and 80 columns.

LINE Statement

It is possible to draw a line between the two points on the screen. The format of the LINE statement is

```
LINE (X1, Y1)-(X2, Y2)
```

where `X1` and `Y1` are the respective horizontal and vertical pixel coordinates of the one point, and `X2` and `Y2` are those of the second point. The coordinates are enclosed in parentheses with a comma in between. The two points are separated by a dash. If the line is to be drawn in a particular colour in SCREEN 1 mode, the number from 0 to 3 of the palette chosen is mentioned immediately after the coordinates of the second point with a comma in between. For example

```
LINE (X1, Y1)-(X2, Y2), 1
```

If no number is mentioned, the computer assigns number 3 automatically.

If after drawing a line, a second line is to be drawn with the coordinates `X2` and `Y2` of the first line as the first point in the second line, this can be done by stating only the coordinates of the second point by the statement shown in the following.

```
LINE -(X3, Y3)
```

This form of the LINE statement is useful in drawing more than are interconnected lines.

It is also possible to draw a complete rectangle by a single LINE statement by adding two commas followed by inserting the symbol B as shown in the following.

```
LINE (X1, Y1)-(X2, Y2), , B
```

If this rectangle is to be drawn in a particular colour of the chosen palette in SCREEN 1 mode, the number from 0 to 3 is inserted in between the two commas such as shown in the following.

```
LINE (X1, Y1)-(X2, Y2), 1, B
```

If no number is mentioned, it is automatically taken to be number 3.

If the rectangle is to filled with the chosen colour, then the symbol BF is inserted instead of B. For example, the statement

```
LINE (X1, Y1)-(X2, Y2), 1, BF
```

generates a green rectangle against a choosen background colour if the palette 0 is mentioned in the COLOR statement.

CIRCLE Statement

It is possible to draw a circle of desired radius around a chosen point. The statement to be given is

```
CIRCLE (X1, Y1), Z, 1
```

where X1 and Y1 are the respective horizontal and vertical pixel-coordinates of the centre of the circle, Z is the radius mentioned as the number of pixels and the last number is the colour of the circle (from 0 to 3) from the palette (either 0 or 1) inserted in the COLOR statement. For example, in palette number 1, the statement

```
CIRCLE (160, 100), 60, 2
```

draw a circle around the point with horizontal and vertical pixel-coordinates equal to 160 and 100, respectively. The radius of the circle is 60 pixels and its colour is magneta. If the colour parameter is not inserted, the default value of 3 is automatically taken.

It is possible to draw an arc of a circle by stating a starting angle and an ending angle measured counterclockwise direction from the right half of the horizontal axis. These are inserted immediately after the parameter of colour separated by commas. Both the inserted angles are expressed in radians for example, the statements

```
SCREEN 1
COLOUR 14, 0
CIRCLE (160, 100), 60, 1, 0, 3.14
```

generates the upper half of a circle of radius 60 pixels around the point (160, 100) pixels in green colour.

If the angles are expressed in negative, these are interpreted as positive but with an additional effect of connecting the end points of the arc with the centre of the circle for example,

```
SCREEN 1
COLOR 14, 0
CIRCLE (160, 100), 60, 1, -3.14, -6.28
```

generates a bottom half of a circle of radius 60 pixels around the centre (160, 100) pixels in green colour with the end points joined to the centre (160, 100).

Besides drawing circles and arcs, the `CIRCLE` statement can also be used to draw ellipses and elliptical arcs. This is achieved by inserting a positive parameter immediately after the angles separated by a comma. The value of the parameter goes as follows.

Value	*Figure*
1	nearly circular
greater than 1	vertical ellipse
less than 1	horizontal ellipse

The eccentricity of the ellipse depends on the value of this parameter. Larger its difference from unity, larger the eccentricity. The radius parameter is half of the major axis of the ellipse. For drawing a complete ellipse, either angles 0 and 2π are inserted or their locations are left blank without disturbing the placement of commas. For example, the following three statements produce the same figure.

```
CIRCLE (160, 100), 80, , , , 0.5
CIRCLE (160, 100), 80, 3, , , 0.5
CIRCLE (160, 100), 80, 3, 0, 6.28, 0.5
```

Summary of the CIRCLE Statement

The complete statement of `CIRCLE` is

PAINT Statement

A figure enclosed in a closed boundary can be filled with a colour matching with the colour of the boundary by using the statement

```
PAINT (X1, Y1), 2
```

where `X1` and `Y1` are the respective horizontal and vertical pixel-coordinates of a point within the closed boundary. The colour number should be the same as that of the closed boundary.

The colour parameter in the `PAINT` statement may be followed by one more parameter which explicitly indicates the colour of the boundary. For example, the statement

```
PAINT (X1, Y1), 3, 2
```

causes the figure with boundary colour 2 to be filled with colour number 3. This statement helps filling a desired figure if the point represented by the pixels `X1` and `Y1` lies within more than one figure.

VIEW Statement

It is possible to choose a portion of the monitor within which the graphical display is to be restricted. This is achieved by the VIEW statement. Its format is

```
VIEW (X1, Y1)-(X2, Y2)
```

where X1 and Y1 are the respective horizontal and vertical pixel coordinates of the left most top corner of the selected portion of the screen and X2 and Y2 are those of the right most bottom corner of the selected portion. For example, the statement

```
VIEW (20, 20)-(300, 180)
```

causes the selection of the following portion of the monitor.

By selecting the appropriate portion of the screen, it is possible to display more than one graphical representation on the same screen.

WINDOW Statement

The WINDOW statement helps replacing the pixel numbering system by the new coordinate system suitable for displaying graph on the full screen or the chosen portion of the screen via VIEW statement. The format of the WINDOW statement is

```
WINDOW (X1, Y1)-(X2, Y2)
```

where X1 and Y1 are the respective horizontal and vertical coordinates of the bottom-left corner of the screen or view portion, and X2 and Y2 are those of the right-top corner. These are shown in the following.

With this coordinate system, the horizontal variation is from X1 to X2 and the vertical variation is from Y1 to Y2. The coordinates of the left-top and bottom-right corners are (X1, Y2) and (X2, Y1), respectively.

The WINDOW statement is normally inserted immediately after the VIEW statement. The graphic statements PSET, LINE and CIRCLE are governed by the coordinate system inserted in the WINDOW statement.

LOCATE Statement

It is possible to position the cursor in the desired location of the screen. This is achieved by the `LOCATE` statement, the format of which is

```
LOCATE X, Y
```

where `X` is the row number and `Y` is the column number. The entire screen has 25 rows (numbering starts from the top of screen) and 40 columns (numbering starts from the left of the screen).

For example, the statements

```
LOCATE 2, 10
PRINT "Plot of rate versus time"
```

causes the cursor to position in the 2nd row and 10th column. Immediately after this, the printing of the given string constant is executed as given in the next statement.

Using the `LOCATE` statement, the cursor can be placed at the desired position on the screen without disturbing any text previously written on the screen.

2 List of Programs

2.1 A FEW ARITHMETIC CALCULATIONS

(i) Calculate the volume (V) of a given amount (n) of an ideal gas at a given temperature (T) and pressure (p). Given: $V = nRT/p$.

Program

```
CLS
INPUT "Amount of gas in moles="; n
INPUT "Temperature in kelvin="; T
INPUT "Pressure in kPa="; p
R = 8.314
5   V = n * R * T / p
10  PRINT "V="; V; "dm^3"
END
```

Output

```
Amount of gas in moles=? .5
Temperature in kelvin=? 298.15
Pressure in kPa=? 101.325
V= 12.23202 dm^3
```

Comment: Instead of statements 5 and 10, a single statement

```
PRINT "V = "; n * R * T / p ; "dm ^ 3"
```

may be used.

(ii) Calculate the pressure (p) of a given amount (n) of O_2 assuming it to follow van der Waals equation of state at the given temperature (T) and volume (V). Given:

Van der Waals Equation $(p + n^2a/V^2)(V - nb) = nRT$

a = 137.802 kPa dm^6 mol^{-2}; b = 0.031 83 dm^3 mol^{-1}

Program

```
CLS
INPUT "Amount of gas="; N
INPUT "Temperature in kelvin"; T
INPUT "Volume in litres="; V
R = 8.314: a = 137.802: b = .03183
p = N * R * T / (V - N * b) - N ^ 2 * a / V ^ 2
PRINT "p="; p; "kPa"
END
```

Output

```
Amount of gas=? 1
Temperature in kelvin? 298.15
Volume in litres=? 22.4
p= 110.5444 kPa
```

(iii) Calculate the root mean square speed $(=\sqrt{3RT/M})$, average speed $(=\sqrt{8RT/\pi M})$ and most probable speed $(=\sqrt{2RT/M})$ of oxygen molecules at a given temperature.

Program

```
CLS
INPUT "Molar mass of the gas in kg/mol="; M
INPUT "Temperature in kelvin="; T
R = 8.314: PI = 3.1415
RMSS = (3 * R * T / M) ^ .5
AVS = (8 * R * T / (PI * M)) ^ .5
MPS = (2 * R * T / M) ^ .5
PRINT "Root mean square speed="; RMSS; "m/s"
PRINT "Average speed="; AVS; "m/s"
PRINT "Most probable speed="; MPS; "m/s"
END
```

Output

```
Molar mass of the gas in kg/mol=? .032
Temperature in kelvin=? 273
Root mean square speed= 461.2878 m/s
Average speed= 424.9989 m/s
Most probable speed= 376.6399 m/s
```

(iv) Calculate the molecular diameter of helium from its van der Waals constant b(= 24 cm^3 mol^{-1}). Given: $b = 4N_A(4\pi r^3/3)$.

Program

```
CLS
INPUT "van der Waals constant,b,in cm3 ="; b
NA = 6.022E+23: PI = 3.1415
R = (3 * b / (16 * NA * PI)) ^ (1 / 3)
D = 2 * R
PRINT "Molecular diameter="; D; "cm"
END
```

Output

```
van der Waals constant,b,in cm3 =? 24
Molecular diameter= 2.669776E-08 cm
```

(v) Calculate the values of molecular diameter (σ), mean free path (λ), number of collisions (Z) made by a single molecule with other molecules per unit time, and number of bimolecular collisions (Z_{11}) per unit volume per unit time for oxygen molecules at 298 K and 101.325 kPa. Given:

$\sigma = 2(3b/16\pi N_A)^{1/3}$; $\lambda = 1/\sqrt{2}\,\pi\sigma^2 N^*$ where $N^* = p/k_B T$;

$Z_1 = u_{av}/\lambda$ where $u_{av} = \sqrt{8RT/\pi M}$ and $Z_{11} = Z_1 N^*/2$. $b = 0.03183$ dm^3 mol^{-1}.

Program

```
CLS
INPUT "Molar mass of oxygen in kg/mol="; M
INPUT "Temperature in kelvin="; T
INPUT "Pressure in Pa"; p
INPUT "van der Waals constant in dm3/mol="; b
R = 8.314: NA = 6.022E+23: PI = 3.1415
kB = R / NA: b = b * .001          'conversion of b into m3/mol
SIGMA = 2 * (3 * b / (16 * PI * NA)) ^ (1 / 3)
NSTAR = p / (kB * T)
UAV = (8 * R * T / (PI * M)) ^ .5
SIGMA2 = SIGMA ^ 2: CON = 2 ^ .5
LAMBDA = 1 / (CON * PI * SIGMA2 * NSTAR)
Z1 = UAV / LAMBDA: Z11 = .5 * Z1 * NSTAR
PRINT "SIGMA="; SIGMA; "m", "LAMBDA="; LAMBDA; "m"
PRINT "Z1="; Z1; "/s", "Z11="; Z11; "/(m3 s)"
END
```

Output

```
Molar mass of oxygen in kg/mol=? .032
Temperature in kelvin=? 298
Pressure in Pa? 101325
van der Waals constant in dm3/mol=? .03183
SIGMA= 2.933256E-10 m        LAMBDA= 1.062226E-07 m
Z1= 4.180207E+09 /s          Z11= 5.14753E+34 /(m3 s)
```

(vi) The bimolecular decomposition of HI is given by the equation $2HI \rightarrow H_2 + I_2$. Assuming a collision diameter of 3.5 nm and an activation energy of 183.9 kJ mol^{-1}, calculate (a) the collision rate of HI, (b) the rate of reaction, and (c) the rate constant of the reaction at 700 K and 1 atm. Given:

$Z = (1/\sqrt{2})\,\pi\sigma^2 u_{av} N^{*2}$ where $u_{av} = \sqrt{8RT/\pi/M}$ and $N^* = N_A p/RT$

Rate of reaction, $r = -(1/2)\mathrm{d}[HI]/\mathrm{d}t = Z\exp(-E_a/RT)/N_A$

Rate constant, $k = r/[HI]^2$ where $[HI] = N^*/N_A$

Program

```
CLS
T = 700: P = 101325: PI = 3.14159: R = 8.314
NA = 6.022E+23: EA = 183.9 * 10 ^ 3: SIGMA = 3.5E-09
M = (1 + 127) * .001
UAV = (8 * R * T / (PI * M)) ^ .5
NSTAR = NA * P / (R * T)
```

```
Z = (1 / 2 ^ .5) * PI * SIGMA ^ 2 * UAV * NSTAR ^ 2
RATE = Z * EXP(-EA / (R * T)) / NA
K = RATE / (NSTAR / NA) ^ 2
PRINT "COLLISION RATE="; Z; "/s"
PRINT "RATE OF REACTION="; RATE; "mol/s"
PRINT "RATE CONSTANT="; K; "dm^6/(mol s)"
END
```

Output

```
COLLISION RATE= 1.017861E+36 /s
RATE OF REACTION= 3.196437E-02 mol/s
RATE CONSTANT= 1.054506E-04 dm^6/(mol s)
```

2.2 AVERAGE AND ROOT MEAN SQUARE AVERAGE OF THE GIVEN NUMBERS

Program

```
CLS
INPUT "Number of data points="; N
SUM = 0: SUM2 = 0
FOR I = 1 TO N
PRINT "X("; I; ")="; : INPUT X
SUM = SUM + X
SUM2 = SUM2 + X ^ 2
NEXT I
AV = SUM / N
RMSA = SQR(SUM2 / N)
PRINT "Average="; AV, "Root mean square average="; RMSA
END
```

Output

```
Number of data points=? 5
X( 1 )=? 5
X( 2 )=? 8
X( 3 )=? 4
X( 4 )=? 12
X( 5 )=? 16
Average= 9    Root mean square average= 10.04988
```

2.3 MAXIMUM / MINIMUM AMONGST THE GIVEN NUMBERS

Program

```
CLS
INPUT "Number of data points="; N
INPUT "Insert the first number,X(1)="; MIN
FOR I = 2 TO N
```

```
   PRINT "X("; I; ")="; : INPUT X
5  IF MIN > X THEN MIN = X
   NEXT I
10 PRINT "Minimum amongst the given numbers is "; MIN
   END
```

Output

```
Number of data points=? 5
Insert the first number,X(1)=? 5
X( 2 )=? 8
X( 3 )=? 4
X( 4 )=? -12
X( 5 )=? 7
Minimum amongst the given numbers is -12
```

Note For finding the maximum insert the operator `"<"` in the `IF __ THEN` statement (numbered 5) and replace minimum by maximum in the `PRINT` statement (numbered 10).

2.4 ASCENDING/DESCENDING ORDER OF THE GIVEN NUMBERS

Program

```
CLS
INPUT "Number of data points="; N
FOR I = 1 TO N
PRINT "X("; I; ")="; : INPUT X(I): Y(I) = X(I)
NEXT I
A$ = INPUT$(1): CLS
FOR I = 1 TO N - 1
FOR J = I TO N
5  IF Y(I) < Y(J) THEN 25
10 DUMMY = Y(I): Y(I) = Y(J): Y(J) = DUMMY
25 NEXT J
   NEXT I
   PRINT : PRINT "Unsorted list of numbers": PRINT
   FOR I = 1 TO N: PRINT X(I); : NEXT I
   PRINT : PRINT : PRINT "Sorted list of numbers": PRINT
   FOR I = 1 TO N: PRINT Y(I); : NEXT I
   END
```

Output

```
Number of data points=? 5
X( 1 )=? 8
X( 2 )=? 5
X( 3 )=? -10
X( 4 )=? 7
X( 5 )=? -5
```

```
Unsorted list of numbers

 8  5 -10  7 -5

Sorted list of numbers

-10 -5  5  7  8
```

Notes

(i) The statement 5 and 10 may be replaced by a single statement

```
IF Y(I) > Y(J) THEN SWAP Y(I), Y(J)
```

(ii) For arranging in the descending order, replace the operator `"<" by ">"`.

2.5 ASCENDING / DESCENDING ORDER OF ALPHABETICAL NAMES

Since the `ASCII` values of alphabets increase from A to Z, the arrangement of names (represented by string variables) in the alphabetical order may be carried out by a program similar to the program involving arrangement of numerical variables in increasing order (Program 2.4). The only change to be carried out is to replace numerical variables by string variables (e.g. replace `X, Y, DUMMY` by `X$, Y$, DUMMY$,` respectively).

2.6 ROOTS OF A QUADRATIC EXPRESSION, $ax^2 + bx + c = 0$

The roots of the expression $ax^2 + bx + c = 0$ are given by the expression

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

provided $b^2 - 4ac \geq 0$. If it is less than zero, then the roots are imaginary.

Program

```
CLS
INPUT "The coefficient a="; a
INPUT "The coefficient b="; b
INPUT "The coefficient c="; c
DEN = 2 * a: REAL = -b / DEN: DISC = b ^ 2 - 4 * a * c
IF DISC < 0 THEN
PRINT "Roots are imaginary"
IMAG = SQR(-DISC) / DEN
PRINT "Real part="; REAL
PRINT "Imaginary part="; IMAG
ELSEIF DISC = 0 THEN
PRINT "Roots are real and equal"
PRINT "Roots are "; REAL, REAL
ELSEIF DISC > 0 THEN
```

```
REAL1 = SQR(DISC) / DEN
ROOT1 = REAL + REAL1: ROOT2 = REAL - REAL1
PRINT "ROOT1="; ROOT1, "ROOT2="; ROOT2
END IF
END
```

Output

Run 1

```
The coefficient a=? 2
The coefficient b=? 4
The coefficient c=? 5
Roots are imaginary
Real part=-1
Imaginary part= 1.224745
```

Run 2

```
The coefficient a=? 2
The coefficient b=? 5
The coefficient c=? 3
ROOT1=-1      ROOT2=-1.5
```

Exercises from Chemistry

1. The concentration of H^+ in a dilute solution of HCl ($c < 10^{-6}$ M) is given by the expression

   $$[H^+]^2 - [HCl]_0\,[H^+] - K_w = 0$$

   Calculate the pH of 10^{-7} M HCl solution at 25 °C. (Hint: $a = 1$, $b = -[HCl]_0$, $c = -K_w$.)

2. The degree of dissociation of a weak acid is given by the expression $c\alpha^2 + K_a\alpha - K_a = 0$. If $[H^+] = c\alpha$, calculate the pH of 10^{-3} M acetic acid ($K_a = 1.80 \times 10^{-5}$ M) solution.

3. The inversion temperature of a van der Waals gas may be computed from the expression

   $$\frac{2a}{RT_i} - \frac{3abp}{R^2T_i^2} - b = 0$$

   This may be rearranged to

   $$(bR^2)T^2 - (2aR)T + 3\,abp = 0$$

   Calculate and display T versus p for N_2 gas for which $a = 141$ kPa dm^6 mol^{-2} and $b = 0.0392$ dm^3 mol^{-1}. Vary p from 100 atm to 300 atm with a step of 50 atm.

2.7 ROOT OF A FUNCTION

Root of *f*(*x*) = 0 by the Bisection Method

If at $x = x_1$ and $x = x_2$, the function $f(x)$ has opposite signs, then there exists at least one value of x between x_1 and x_2 at which $f(x) = 0$. In the bisection method, the value of x is determined by the following computational procedure.

- Choose x_1 and x_2 such that $f(x_1)\,f(x_2) < 0$.
- Take $\bar{x} = (x_1 + x_2)/2$ and evaluate $f(\bar{x})$.
- If $f(x_1)\,f(\bar{x}) < 0$, then the root of $f(x)$ lies between x_1 and $\bar{x}$. If not, then the root lies between $\bar{x}$ and x_2 for which $f(\bar{x})\,f(x_2) < 0$.

- Replace x_2 or x_1 by $\bar{x}$ and repeat the previous step until $|f(\bar{x})|$ or $|x(\text{new}) - x(\text{old})|$ is less than a predecided minimum value.

Illustration To find a real root of $f(x) = x^3 - 5x + 3$. The following program scans the root(s) from XI to XF in the step of XS.

Program

```
CLS
DEF FNA (X) = X ^ 3 - 5 * X + 3
READ XI, XF, XS, XE: DATA -10,10,1,.0001
3 PRINT
'PRINT "Determining X1 & X2 corresponding"
'PRINT "to opposite signs of function"
'PRINT STRING$(50, "-")
'PRINT "      X1          X2          F1          F2"
'PRINT STRING$(50, "-")
FOR I = XI TO XF - XS STEP XS
X1 = I: X2 = I + XS: F1 = FNA(X1): F2 = FNA(X2)
'PRINT USING "  ##.###^^^^"; X1; X2; F1; F2
IF F2 = F1 OR F1 * F2 > 0 THEN 4
X2 = I + XS: GOTO 6
4 NEXT I
'PRINT STRING$(50, "-")
GOTO 15
6 XI = I + XS
'PRINT STRING$(50, "-"): A$ = INPUT$(1): CLS
M = 0
'PRINT STRING$(55, "-")
'PRINT " N        X1           X2           XAV          F2"
'PRINT STRING$(55, "-")
8 M = M + 1
F1 = FNA(X1): X = (X1 + X2) / 2: F2 = FNA(X)
F1F2 = F1 * F2
'PRINT USING "##"; M;
'PRINT USING "   ##.###^^^^"; X1; X2; X; F2
'A$ = INPUT$(1)
IF ABS(X1 - X2) < XE THEN 10
IF F1F2 < 0 THEN X2 = X: GOTO 8
X1 = X: GOTO 8
10 PRINT
'PRINT STRING$(55, "-")
PRINT "ROOT IS "; X1
```

```
A$ = INPUT$(1)
'CLS
GOTO 3
15 PRINT "****SCANNING IS OVER****"
20 END
```

Output

```
ROOT IS -2.490906

ROOT IS  .6566162

ROOT IS  1.834229

****SCANNING IS OVER****
```

Exercises

1. Determine the volume of 0.5 mol of the van der Waals gas CO_2 at 2226 kPa and 298 K. Given: a = 363.76 kPa dm^6 mol^{-2} and b = 42.67 cm^3 mol^{-1}. Scan volume from 0.1 dm^3 to 2 dm^3 in the step of 0.1 dm^3. (**Ans:** 0.499 5117 dm^3)
2. Determine [H$^+$] in 10^{-4} M solution of glycine (K_a = 1.7 × 10^{-10} M) at 20 °C by using the expression

$$[H^+]^3 + K_a[H^+]^2 - (K_w + K_a\,[HG])\,[H^+] - K_a\,K_w = 0$$

Scan [H$^+$] from 1 × 10^{-7} M to 3 × 10^{-7} M in the step of 1 × 10^{-8} M. (**Ans:** 1.649 218 × 10^{-7} M)

Root of $f(x) = 0$ by the Method of False Position

If at $x = x_1$ and $x = x_2$, the function $f(x)$ has opposite signs, then there exists at least one value of x between x_1 and x_2 at which $f(x) = 0$. In the method of false position, the value of x is determined by the following computational procedure.

- Choose x_1 and x_2 such that $f(x_1)\,f(x_2) < 0$.
- Replace the curve between $[x_1, f(x_1)]$ and $[x_2, f(x_2)]$ by a straight line and determine its point of intersection with x-axis.

Equation of straight line $$\frac{y - f(x_1)}{x - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

Point of intersection with x-axis where $y = 0$

$$x_0 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)}$$

- If $f(x_1)\,f(x_0) < 0$, then the root of $f(x)$ lies between x_1 and x_0. If not, then the root lies between x_0 and x_2 for which $f(x_0)\,f(x_2) < 0$.
- Replace x_2 or x_1 by x_0 and repeat the previous step until $|f(x_0)|$ or $|x(\text{new}) - x(\text{old})|$ is less than a predecided minimum value.

Illustration To find a real root of $f(x) = x^3 - 4x + 1$. The following program scans the root(s) from XI to XF in the step of XS.

Program

```
CLS
DEF FNA (X) = X ^ 3 - 4 * X + 1
READ XI, XF, XS, XE: DATA  -10, 10, .1, .0001
3 PRINT
'PRINT "Determining X1 & X2 corresponding to opposite signs ";
'PRINT "of function"
'PRINT STRING$(53, "-")
'PRINT "         X1              X2              F1             F2"
'PRINT STRING$(53, "-")
FOR I = XI TO XF - XS STEP XS
X1 = I: X2 = I + XS: F1 = FNA(X1): F2 = FNA(X2)
'PRINT USING "    ##.###^^^^"; X1; X2; F1; F2: A$ = INPUT$(1)
IF F2 = F1 OR F1 * F2 > 0 THEN 4
X2 = I + XS: GOTO 6
4 NEXT I
'PRINT STRING$(53, "-")
GOTO 15
6 XI = I + XS
'PRINT STRING$(53, "-"): A$ = INPUT$(1): CLS
M = 0
'PRINT STRING$(55, "-")
'PRINT " M          X1             X2              X              FX"
'PRINT STRING$(55, "-")
XP = X1
8 M = M + 1
F1 = FNA(X1): F2 = FNA(X2)
X = (X1 * F2 - X2 * F1) / (F2 - F1)
FX = FNA(X): F1F2 = F1 * F2
'PRINT USING "##"; M; : PRINT USING "    ##.###^^^^"; X1; X2; X; FX
'A$ = INPUT$(1)
IF ABS(XP - X) < XE THEN 10
IF F1F2 < 0 THEN X2 = X: XP = X: GOTO 8
X1 = X: XP = X: GOTO 8
10 PRINT
'PRINT STRING$(55, "-")
PRINT "ROOT IS  "; X
'LINE (X, -10)-(X, 0), 2: A$ = INPUT$(1)
```

```
'A$ = INPUT$(1)
'CLS
GOTO 3
15 PRINT "**** SCANNING IS OVER ****"
20 END
```

Output

```
ROOT IS  -2.114905

ROOT IS   .2541017

ROOT IS   1.860806

**** SCANNING IS OVER ****
```

Improvement of a Root of $f(x) = 0$ by the Newton-Raphson Method

In the Newton-Raphson method, the first improved root of $f(x) = 0$ is written as

$$x_1 = x_0 + h$$

where x_0 is the approximate root of $f(x) = 0$ and h is the small refinement. Writing $f(x_1)$ by Taylor's series, we have

$$f(x_1) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + \cdots$$

Neglecting the second- and higher-order derivatives, we get

$$f(x_1) = 0 = f(x_0) + h\,f'(x_0)$$

which gives $$h = -\frac{f(x_0)}{f'(x_0)}$$

Hence, the refined root is

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

The above procedure may be repeated to get the second refined root and so on. The general expression of refined root is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}; \qquad n = 0, 1, 2, \ldots$$

Illustration

To determine the volume of 0.5 mol of van der Waals gas (carbon dixide) at 2 226 Pa and 298 K. Take the starting volume as obtained from the ideal gas equation of state. Given: a = 363.76 kPa dm^6 mol^{-2}, b = 0.042 67 dm^3 mol^{-1}.

We have

$$f(V) = V^3 - \left(nb + \frac{nRT}{p}\right)V^2 + \frac{n^2a}{p}V - \frac{n^3ab}{p} = 0$$

From this, we get

$$f'(V) = 3\ V^2 - 2\left(nb + \frac{nRT}{p}\right)V + \frac{n^2a}{p}$$

Program

```
CLS
P = 2226: A = 363.76: B = .04267: R = 8.314
T = 298: N = .5
NB = N * B: NRT = N * R * T: N2A = N ^ 2 * A
N3AB = N ^ 3 * A * B
DEF FNA (V) = V ^ 3 - (NB + NRT / P) * V ^ 2 + (N2A / P) * V - N3AB / P
DEF FNB (V) = 3 * V ^ 2 - (NB + NRT / P) * 2 * V + (N2A / P)
X1 = NRT / P
6 N = 0
PRINT STRING$(35, "-")
PRINT " N          X1           X2          FNA"
PRINT STRING$(35, "-")
8 N = N + 1
X2 = X1 - FNA(X1) / FNB(X1)
PRINT N; : PRINT USING "    ##.####"; X1; X2; FNA(X1)
A$ = INPUT$(1)
IF ABS(X2 - X1) < .00001 THEN 10
X1 = X2: GOTO 8
10 PRINT STRING$(35, "-")
20 END
```

Output

```
-----------------------------------
N       X1         X2         FNA
-----------------------------------
1     0.5565     0.5098     0.0153
2     0.5098     0.5000     0.0023
3     0.5000     0.4996     0.0001
4     0.4996     0.4996     0.0000
-----------------------------------
```

Root of *f*(*x*) = 0 by the Muller's Method

In the Muller's method, the function $f(x) = 0$ is approximated by a quadratic equation in the vicinity of its root. If x_i is the approximation to a root of $f(x) = 0$, then the quadratic equation passing through x_i is

$$p = A(x - x_i)^2 + B(x - x_i) + y_i \tag{1}$$

where $y_i = f(x_i)$. The values of A and B are determined by considering two other approximations of the root. If x_{i-1} and x_{i-2} are the roots then

$$y_{i-1} = A\,(x_{i-1} - x_i)^2 + B\,(x_{i-1} - x_i) + y_i$$

$$y_{i-2} = A\,(x_{i-2} - x_i)^2 + B\,(x_{i-2} - x_i) + y_i$$

Solving for A and B, we get

$$A = \frac{y_{i-1} - y_i}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)} + \frac{y_{i-2} - y_i}{(x_{i-2} - x_i)(x_{i-2} - x_{i-1})}$$

$$B = \frac{y_{i-1} - y_i}{x_{i-1} - x_i} - A\,(x_{i-1} - x_i)$$

With these values of A and B, the root of Eq. (1) is considered to be new approximation x_{i+1} of the root of $f(x) = 0$.

Equation (1) provides

$$x_{i+1} - x = \frac{-B \pm \sqrt{B^2 - 4Ay_i}}{2A} \tag{2}$$

The above expression is written in its equivalent form as

$$x_{i+1} - x_i = -\frac{2y_i}{-B \pm \sqrt{B^2 - 4Ay_i}} \tag{3}$$

The sign in the denominator of right-hand side of the above expression is chosen so as to have a larger value.

Considering x_{i+1} as the new approximation of x_i, the values of A and B are evaluated and then their substitution in Eq. (3) provides then next improved approximation of x_i. The procedure is repeated to attain self consistency.

Illustration

To determine the volume of 0.5 mol of van der Waals gas (carbon dioxide) at 2 226 Pa and 298 K.
Given: a = 363.76 kPa dm^6 mol^{-2} and b = 0.042 67 dm^3 mol^{-1}. Take $x_3 = 2$, $x_2 = 0.6$ and $x_1 = 0.2$ as the approximations of the function

$$f(V) = V^3 - \left(nb + \frac{nRT}{p}\right)V^2 + \frac{n^2a}{p}V - \frac{n^3ab}{p} = 0$$

Program

```
CLS
P = 2226: A = 363.76: B = .04267: R = 8.314: T = 298: N = .5
NB = N * B: NRT = N * R * T: N2A = N ^ 2 * A: N3AB = N ^ 3 * A * B
DEF FNA (V) = V ^ 3 - (NB + NRT / P) * V ^ 2 + (N2A / P) * V - N3AB / P
2 INPUT "X1="; X1: INPUT "X2="; X2: INPUT "X3="; X3
A$ = INPUT$(1)
```

```
CLS : Y1 = FNA(X1): Y2 = FNA(X2)
N = 0
PRINT STRING$(47, "-")
PRINT " N         A              B          ROOT         ER"
PRINT STRING$(47, "-")
5 Y3 = FNA(X3)
N = N + 1
TERM1 = (Y2 - Y3) / ((X2 - X1) * (X2 - X3))
TERM2 = (Y1 - Y3) / ((X1 - X3) * (X1 - X2))
A = TERM1 + TERM2
B = (Y2 - Y3) / (X2 - X3) - A * (X2 - X3)
DIS = B ^ 2 - 4 * A * Y3
IF DIS < 0 THEN 8
TERM = SQR(DIS)
TERM1 = B + TERM: TERM2 = B - TERM
IF TERM1 > TERM2 THEN DEN = TERM1 ELSE DEN = TERM2
X4 = X3 - (2 * Y3) / DEN
ER = 100 * (X4 - X3) / X4
PRINT USING "##"; N; : PRINT USING "   ####.####"; A; B; X4; ER
A$ = INPUT$(1)
IF ABS(ER) < .001 THEN 10
X3 = X4
GOTO 5
8 PRINT "ROOT IS IMAGINARY"
PRINT "TRY DIFFERENT VALUES OF X1,X2 AND X3"
GOTO 2
10 PRINT STRING$(47, "-")
END
```

Output

```
X1=? .2
X2=? .6
X3=? 2
-----------------------------------------------
 N       A           B         ROOT        ER
-----------------------------------------------
 1     2.2222      7.2095      0.5653  -253.8160
 2     0.7874      0.3588      0.5071   -11.4655
 3     0.7293      0.2548      0.5005    -1.3257
 4     0.7226      0.2438      0.4997    -0.1632
 5     0.7218      0.2425      0.4996    -0.0203
 6     0.7217      0.2423      0.4996    -0.0025
 7     0.7217      0.2423      0.4996    -0.0003
-----------------------------------------------
```

Method of Successive Approximation

The exact expression of $[H^+]$ in a dilute solution of a weak acid is given by the expression

$$[H^+]^3 + [H^+]^2 K_a - [H^+] \{K_w + K_a [HA]_0\} - K_a K_w = 0$$

Calculate $[H^+]$ in a 10^{-4} M solution of glycine which acts as a weak acid with $K_a = 1.7 \times 10^{-10}$ M at 25 °C.

In the method of successive approximation, the above expression of $[H^+]$ is arranged as

$$[H^+]_{i+1} = [[H^+]_i \{K_w + K_a [HA]_0\} - [H^+]_i^2 K_a + K_a K_w]^{1/3}$$

With a reasonable val ue of $[H^+]$, the right-hand side of the above expression is evaluated to give a new value of $[H^+]$. The procedure is repeated with the refined value of $[H^+]$ till the two successive values tally with each other within the certain range.

Program

```
CLS
INPUT "Ionization constant of the acid="; KA
INPUT "Concentration of the acid="; C
INPUT "Starting concentration of H+="; H
KW = 1E-14
PRINT STRING$(28, "_")
PRINT "ITERATION"; SPC(10); "[H+]"
PRINT STRING$(28, "-")
FOR I = 1 TO 20
HP = (H * (KW + KA * C) - H * H * KA + KA * KW) ^ (1 / 3)
PRINT I, HP: A$ = INPUT$(1)
IF ABS(HP - H) < 1E-10 THEN 5
H = HP
NEXT I
5 PRINT STRING$(28, "_")
PH = -LOG(HP) / LOG(10)
PRINT : PRINT "pH="; : PRINT USING "##.##"; PH
END
```

Output

```
Ionization constant of the acid=? 1.7E-10
Concentration of the acid=? .001
Starting concentration of H+=? 0
```

ITERATION	[H+]
1	1.193483E-08
2	1.290652E-07
3	2.85334E-07
4	3.716845E-07
5	4.059151E-07
6	4.180072E-07
7	4.221157E-07
8	4.234936E-07
9	4.239537E-07
10	4.241071E-07
11	4.241583E-07

```
pH= 6.37
```

Lin-Bairstow's Method for Solving Cubic Equation

See Project 3.1 for the theory and computation by this method

Exercise

The concentration of OH^- in a buffer solution of ammonium chloride (0.25 M) and ammonium hydroxide (0.05 M) containing $Mg(OH)_2$ and $Al(OH)_3$ is given by the expression

$$[OH^-]^3 = (3.6 \times 10^{-6}\ M)[OH^-]^2 + (7.12 \times 10^{-11}\ M^2)[OH^-] + 1.28 \times 10^{-15}\ M^3$$

Calculate the value of $[OH^-]$ by the method of successive approximation.

(Ans. 1.45×10^{-5} M)

2.8 GENERATION OF FIBONACCI NUMBERS

To Generate Fibonacci numbers up to twenty.

The Fibonacci numbers are numbers in which each number is equal to the sum of the previous two numbers. Start with the first two numbers as 1 and 2.

Program

```
CLS
F1 = 1: F2 = 2
PRINT F1, : PRINT F2,
FOR I = 1 TO 20
F = F1 + F2
PRINT F,
F1 = F2: F2 = F
NEXT I
END
```

Output

```
1             2             3             5             8
13            21            34            55            89
144           233           377           610           987
1597          2584          4181          6765          10946
17711         28657
```

2.9 GENERATION OF FACTORIALS

To generate factorials up to twenty.

By definition $n! = 1 \times 2 \times \cdots \times n$ with $0! = 1$

Program

```
CLS
N = 20
FACT = 1
FOR I = 0 TO N
IF I = 0 THEN 5
FACT = FACT * I
```

```
    5 PRINT I, FACT
    NEXT I
    END
```

Output

```
0             1
1             1
2             2
3             6
4             24
5             120
6             720
7             5040
8             40320
9             362880
10            3628800
11            3.99168E+07
12            4.790016E+08
13            6.227021E+09
14            8.717829E+10
15            1.307674E+12
16            2.092279E+13
17            3.556874E+14
18            6.402374E+15
19            1.216451E+17
20            2.432902E+18
```

2.10 GENERATION OF PRIME NUMBERS

To generate prime numbers up to fifty.

A number is a prime if it is not evenly divisible by a smaller integer. By definition, 1 and 2 are prime numbers.

Carry out the integer division of the given number N by all integers from 2 to $N - 1$. If the integer remainder is nonzero for all divisions, the number is a prime number. The divisions can be carried out by using the statement "MOD".

Alternatively, determine the quotient M in a division of the given number N one by one by integers from 2 to INT$(\sqrt{N})$, where "INT" is a library function that determines the largest integer not exceeding $\sqrt{N}$. If M and INT(M) are identical for any division, the giving number is not a prime number.

Programs

```
REM PROGRAM PRIMENO;GENERATION OF PRIME NUMBERS
READ N: DATA 50
REM -----METHOD 1------
CLS : PRINT " 1", : PRINT " 2",
FOR I = 3 TO N
FOR J = 2 TO I - 1
IF I MOD J = 0 THEN 5
NEXT J
```

```
PRINT I,
5 NEXT I
REM -----METHOD 2-----
PRINT
PRINT " 1", : PRINT " 2",
FOR I = 3 TO N
FOR J = 2 TO INT(SQR(I))
R = I / J
RI = INT(R)
IF R = RI THEN 15
NEXT J
PRINT I,
15 NEXT I
END
```

Output

```
1           2           3           5           7
11          13          17          19          23
29          31          37          41          43
47
1           2           3           5           7
11          13          17          19          23
29          31          37          41          43
47
```

2.11 GENERATION OF PASCAL TRIANGLE

A Pascal triangle involves the arrangement of numbers in a triangular form in which a number in a row is sum of the numbers placed on the left and right of the previous row as shown in the following.

```
                1
              1   1
            1   2   1
          1   3   3   1 . . . and so on.
```

The numbers may be generated by the expression ${}^{n}C_{m} = n!/(m!(n - m)!)$ where $m = 0, 1, 2, \ldots, n$.

The following program illustrates the generation of a Pascal triangle in the graphical or textual mode. For graphical mode, the variable `OPT = 1` and for textual mode, `OPT = 2`.

Program

```
REM PROGRAM PASCAL;GENERATION OF PASCAL TRIANGLE
REM OPT=1 FOR GRAPHICAL MODE;OPT=2 FOR TEXTUAL MODE
CLS : READ N, OPT: DATA 9,1
DEF FNF (I)
FACT = 1
IF I = 0 THEN 5
```

```
FOR J = 1 TO I
FACT = FACT * J
NEXT J
5 FNF = FACT
END DEF
ON OPT GOTO 10, 15
10 SCREEN 1: COLOR 14, 0
15 FOR K = 0 TO N
FOR L = 0 TO K
BIN = FNF(K) / (FNF(L) * FNF(K - L))
ON OPT GOTO 20, 25
20 LOCATE K + 2, 20 - K * 2 + L * 4: PRINT BIN;
GOTO 30
25 PRINT TAB(38 - 4 * K + 8 * L); BIN;
30 NEXT L
PRINT
NEXT K
END
```

Output

```
                    1
                  1   1
                1   2   1
              1   3   3   1
            1   4   6   4   1
          1   5   10  10  5   1
        1   6   15  20  15  6   1
      1   7   21  35  35  21  7   1
    1   8   28  56  70  56  28  8   1
  1   9   36  84  126 126 84  36  9   1
```

2.12 ROUNDING OF A NUMBER

To carry out the rounding of a given number for a given number of digits after the decimal point.

Multiply the given number (N) by 10 raised to a power ND (where ND is the number of digits beyond which rounding is required) to give the number N1.

Determine the truncated integer N2 of the number N1 by using the statement "FIX" (or "INT").

Determine the absolute of the difference between N1 and N2 to give the number N3.

Step 1. If N3 is less than 0.5 then print the rounded number which is N2 divided by (10 raised to power ND).

Step 2. If N3 is more than 0.5, add one to N2 if N2 is positive otherwise subtract 1 (if N2 is negative). Print the rounded number which is N2 divided by (10 raised to power ND).

If N3 = 0.5, then determine whether N2 is even or odd. If even then print the rounded number as given in Step 1 otherwise print as given in Step 2.

Program

```
REM ROUNDING NUMBER AFTER DECIMAL
CLS : INPUT "Given number is "; N
INPUT "Number of digits after decimal "; ND
N1 = N * 10 ^ ND: PRINT "N1="; N1: a$ = INPUT$(1)
N2 = FIX(N1): PRINT "N2="; N2: a$ = INPUT$(1)
N3 = ABS(N1 - N2): PRINT "N3="; N3
IF N3 < .5 THEN 15
IF N3 > .5 THEN 10
IF ABS(N2) = ABS(2 * INT(N2 / 2)) THEN 15
10 IF N2 < 0 THEN N2 = N2 - 1 ELSE N2 = N2 + 1
15 PRINT "Rounded number is "; N2 / 10 ^ ND
END
```

Output

```
Run 1
Given number is ? 123.456
Number of digits after decimal ? 2
N1= 12345.6
N2= 12345
N3= .6005859
Rounded number is  123.46

Run 2
Given number is ? 123.55
Number of digits after decimal ? 1
N1= 1235.5
N2= 1235
N3= .5
Rounded number is  123.6
```

2.13 MATRIX MULTIPLICATION

Expression to be used: $C_{ij} = \Sigma_k A_{ik} B_{kj}$

Program

```
REM MATRIX MULTIPLICATION
CLS
INPUT "Number of rows of matrix A="; RA
INPUT "Number of columns of matrix A="; CA
INPUT "Number of rows of matrix B="; RB
INPUT "Number of columns of matrix B="; CB
IF CA <> RB THEN 5
FOR I = 1 TO RA: FOR J = 1 TO CA
PRINT "A("; I; ","; J; ") ="; : INPUT A(I, J)
NEXT J: NEXT I
FOR I = 1 TO RB: FOR J = 1 TO CB
PRINT "B("; I; ","; J; ") ="; : INPUT B(I, J)
```

```
NEXT J: NEXT I
FOR I = 1 TO RA: FOR J = 1 TO CB
C(I, J) = 0
FOR K = 1 TO CA
C(I, J) = C(I, J) + A(I, K) * B(K, J)
NEXT K
NEXT J: NEXT I
PRINT "A MATRIX"
FOR I = 1 TO RA: FOR J = 1 TO CA
PRINT A(I, J); : NEXT J: PRINT : NEXT I
PRINT : PRINT "B MATRIX"
FOR I = 1 TO RB: FOR J = 1 TO CB
PRINT B(I, J); : NEXT J: PRINT : NEXT I
PRINT : PRINT "Multiplied matrix"
FOR I = 1 TO RA: FOR J = 1 TO CB
PRINT C(I, J); : NEXT J: PRINT : NEXT I
GOTO 10
5 PRINT "Matrix multiplication is not possible because"
PRINT "number of columns of A matrix is not equal to"
PRINT "the number of rows of matrix B"
10 END
```

Output

```
Number of rows of matrix A=? 2
Number of columns of matrix A=? 2
Number of rows of matrix B=? 2
Number of columns of matrix B=? 2
A( 1 , 1 ) =? 2
A( 1 , 2 ) =? 2
A( 2 , 1 ) =? 2
A( 2 , 2 ) =? 2
B( 1 , 1 ) =? 2
B( 1 , 2 ) =? 2
B( 2 , 1 ) =? 2
B( 2 , 2 ) =? 2
A MATRIX
 2  2
 2  2

B MATRIX
 2  2
 2  2

Multiplied matrix
 8  8
 8  8
```

Explanation

In the printing of matrices, the insertion of the statement PRINT brings the cursor to the first position of the next line because there is no semicolon immediately after this. This helps printing matrices row-wise.

2.14 VALUE OF A DETERMINANT

Conventional Method

The conventional method of evaluating the value of a determinant is determined from the expression

$$D = a_{11}C_{11} + a_{12}C_{12} + a_{13}C_{13} + \cdots + a_{1N}C_{1N}$$

where C_{ik} is the cofactor of a_{ik} which is obtained by deleting ith row and kth column of the determinant multiplied by $(-1)^{i+k}$. The values of cofactors are determined in a similar manner. Each time, the order of the cofactor is decreased by one. This is continued till the order of cofactor is two.

The value of 2×2 determinant is determined by the expression

$$C = C_{11}C_{22} - C_{12}C_{21}$$

The above procedure is impractical for the larger determinant. However, this can be carried out with the help of a computer.

Program

```
REM PROGRAM DETERMIN;VALUE OF DETERMINANT(CONVENTIONAL)
3 CLS : DIM A(5, 5), B(4, 4), C(3, 3), D(2, 2), X(5, 5), Y(5, 5)
5 DEF FNY = Y(1, 1) * Y(2, 2) - Y(1, 2) * Y(2, 1)
6 INPUT "ORDER OF THE DETERMINANT="; NA
10 FOR I = 1 TO NA: FOR J = 1 TO NA
15 PRINT "A("; I; ","; J; ")="; : INPUT A(I, J): Y(I, J) = A(I, J)
20 NEXT J: NEXT I
25 IF NA = 2 THEN DET1 = FNY: GOTO 235
30 IA = 1: I1 = IA: SUMA = 0
35 FOR JA = 1 TO NA
38 NB = NA - 1: N = NA
40 AIJ = A(IA, JA): J1 = JA
42 FOR I = 1 TO NA: FOR J = 1 TO NA: X(I, J) = A(I, J): NEXT J: NEXT I
45 GOSUB 170
50 IF NB = 2 THEN SUMB1 = FNY: GOTO 155
54 FOR I = 1 TO NB: FOR J = 1 TO NB
56 B(I, J) = Y(I, J): 'PRINT "B("; I; ","; J; ")="; B(I, J),
58 X(I, J) = Y(I, J)
60 NEXT J
62 NEXT I
'A$ = INPUT$(1)
64 IB = 1: I1 = IB: SUMB1 = 0
75 FOR JB = 1 TO NB
76 NC = NB - 1: N = NB
80 BIJ = B(IB, JB): J1 = JB
85 GOSUB 170
```

```
90 IF NC = 2 THEN SUMC = FNY: GOTO 145
92 FOR I = 1 TO NC: FOR J = 1 TO NC
94 C(I, J) = Y(I, J): X(I, J) = Y(I, J)
96 NEXT J
98 NEXT I
100 IC = 1: I1 = IC: SUMC = 0
112 FOR JC = 1 TO NC
116 ND = NC - 1: N = NC
120 CIJ = C(IC, JC): J1 = JC
125 GOSUB 170
130 IF ND = 2 THEN SUMD = FNY
135 SUMC = SUMC + (-1) ^ (IC + JC) * CIJ * SUMD
140 NEXT JC
145 SUMB1 = SUMB1 + (-1) ^ (IB + JB) * BIJ * SUMC
150 NEXT JB
155 SUMA = SUMA + (-1) ^ (IA + JA) * AIJ * SUMB1
160 NEXT JA
165 GOTO 226
170 K = 0
175 FOR II = 1 TO N
180 IF I1 = II THEN 220
185 K = K + 1
190 L = 0
195 FOR JJ = 1 TO N
200 IF J1 = JJ THEN 215
205 L = L + 1
210 Y(K, L) = X(II, JJ)
215 NEXT JJ
220 NEXT II
225 RETURN
226 DET1 = SUMA
235 PRINT "VALUE OF DETERMINANT="; DET1
240 END
```

Output

```
ORDER OF THE DETERMINANT=? 2
A( 1 , 1 )=? 2
A( 1 , 2 )=? 5
A( 2 , 1 )=? 3
A( 2 , 2 )=? 7
VALUE OF DETERMINANT=-1
```

Numerical Method

The numerical method involves the reduction of the determinant to a triangular form. This reduction is carried out by using the principle stated in the following.

The values of a determinant is left unchanged if the entries in a row (or column) are altered by adding to them any constant multiple of the corresponding entries in any other row (or column).

The following expression may be used to reduce a given determinant of the order N in a triangular form.

$$a_{jk} = a_{jk} - a_{ik}\left(\frac{a_{ji}}{a_{ii}}\right)$$

where the variations in the indices i, j and k go as follows.

$i = 1$ to $N - 1$

$j = i + 1$ to N

$k = i$ to N

Finally, the value of determinant is determined by multiplying the diagonal elements.

Program

```
REM PROGRAM DETZERO;MAKING LOWER HALF EQUAL TO ZERO
CLS : DIM A(5, 5)
INPUT "ORDER OF THE DETERMINANT="; N
FOR I = 1 TO N: FOR J = 1 TO N
PRINT "A("; I; ","; J; ")="; : INPUT A(I, J)
NEXT J: NEXT I
FOR I = 1 TO N - 1
FOR J = I + 1 TO N
IF A(I, I) = 0 THEN 5
DUM = A(J, I) / A(I, I)
FOR K = I TO N
A(J, K) = A(J, K) - A(I, K) * DUM
NEXT K
NEXT J
5 NEXT I
KK = 0
FOR I = 2 TO N - 1
IF A(I, I) <> 0 THEN 15
KK = KK + 1
FOR J = I + 1 TO N
IF A(J, I) = 0 THEN 10
FOR K = 1 TO N: SWAP A(I, K), A(J, K): NEXT K
10 NEXT J
15 NEXT I
DET1 = 1
FOR I = 1 TO N
DET1 = (-1) ^ KK * DET1 * A(I, I)
NEXT I
PRINT : PRINT "THE VALUE OF THE DETERMINANT="; DET1
END
```

Output

```
ORDER OF THE DETERMINANT=? 2
A( 1 , 1 )=? 3
A( 1 , 2 )=? 4
A( 2 , 1 )=? 2
A( 2 , 2 )=? 5

THE VALUE OF THE DETERMINANT= 7
```

2.15 INVERSE OF A MATRIX

Conventional Method

The inverse of a nonsingular $N \times N$ matrix $\boldsymbol{A} = [a_{jk}]$ is given by

$$\boldsymbol{A}^{-1} = \frac{1}{\det \boldsymbol{A}} \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ & \vdots & & \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{bmatrix}$$

where A_{jk} is the cofactor of a_{jk} in det $\boldsymbol{A}$. Note that in $\boldsymbol{A}^{-1}$, the cofactor A_{jk} occupies the place of a_{kj} in det $\boldsymbol{A}$.

The above procedure is impractical for the larger determinant. However, this can be carried out with the help of a computer.

Program

```
REM PROGRAM INVCOF;INVERSE OF A MATRIX VIA COFACTOR
4 CLS
15 DIM A(4, 4), B(3, 3), COF(4, 4), C(4, 4)
20 INPUT "ORDER OF THE MATRIX="; N
25 FOR I = 1 TO N: FOR J = 1 TO N
30 PRINT "A("; I; ","; J; ")="; : INPUT A(I, J)
35 NEXT J: NEXT I
50 'CLS
55 PRINT : PRINT "PRINTING OF THE GIVEN MATRIX": PRINT
60 FOR I = 1 TO N: FOR J = 1 TO N
70 PRINT USING "   ###.###"; A(I, J);
75 NEXT J: PRINT : NEXT I
76 A$ = INPUT$(1)
85 REM----TO GENERATE THE COFACTOR MATRIX----
88 PRINT : PRINT "PRINTING OF COFACTOR": PRINT
90 FOR IROW = 1 TO N: FOR JCOL = 1 TO N
100 K = 1
105 FOR I = 1 TO N
110 L = 1
IF I = IROW THEN 150
```

```
120 FOR J = 1 TO N
125 IF J = JCOL THEN 140
130 B(K, L) = A(I, J)
135 L = L + 1
140 NEXT J
145 K = K + 1
150 NEXT I
155 REM TO DETERMINE THE DETERMINANT VALUE OF THE COFACTOR
156 IF N - 1 = 1 THEN
157 DET1 = B(1, 1)
160 ELSEIF N - 1 = 2 THEN
165 DET1 = B(1, 1) * B(2, 2) - B(1, 2) * B(2, 1)
170 ELSE
175 DET1 = B(1, 1) * (B(2, 2) * B(3, 3) - B(3, 2) * B(2, 3))
180 DET1 = DET1 - B(1, 2) * (B(2, 1) * B(3, 3) - B(2, 3) * B(3, 1))
185 DET1 = DET1 + B(1, 3) * (B(2, 1) * B(3, 2) - B(2, 2) * B(3, 1))
205 END IF
210 COF(IROW, JCOL) = (-1) ^ (IROW + JCOL) * DET1
212 PRINT USING "   ###.###"; COF(IROW, JCOL);
215 NEXT JCOL: PRINT : NEXT IROW: A$ = INPUT$(1)
225 REM VALUE OF THE DETERMINANT
230 DET1 = 0
235 I = 1
240 FOR J = 1 TO N
245 DET1 = DET1 + A(I, J) * COF(I, J)
250 NEXT J
255 REM TRANSPOSE OF COFACTOR MATRIX
260 FOR I = 1 TO N: FOR J = I + 1 TO N
270 TEMP = COF(I, J): COF(I, J) = COF(J, I): COF(J, I) = TEMP
285 NEXT J: NEXT I
300 REM DIVISION OF TRANSPOSE OF COFACTOR MATRIX BY
302 REM THE VALUE OF THE DETERMINANT OF THE GIVEN MATRIX
305 FOR I = 1 TO N: FOR J = 1 TO N
315 COF(I, J) = COF(I, J) / DET1
320 NEXT J: NEXT I
365 PRINT : PRINT "PRINTING OF THE INVERSE OF THE GIVEN MATRIX": PRINT
370 FOR I = 1 TO N: FOR J = 1 TO N
380 PRINT USING "   ###.###"; COF(I, J);
385 NEXT J: PRINT : NEXT I
400 REM VERIFICATION OF THE INVERSE MATRIX
```

```
405 FOR I = 1 TO N: FOR J = 1 TO N
415 SUM = 0
420 FOR K = 1 TO N
425 SUM = SUM + A(I, K) * COF(K, J)
430 NEXT K
435 C(I, J) = SUM
436 IF ABS(C(I, J)) < .001 THEN C(I, J) = 0
440 NEXT J: NEXT I
450 PRINT : PRINT "VERIFICATION OF THE INVERSE MATRIX": PRINT
455 FOR I = 1 TO N: FOR J = 1 TO N
465 PRINT USING "   ###.###"; C(I, J);
470 NEXT J: PRINT : NEXT I
485 END
```

Output

```
ORDER OF THE MATRIX=? 2
A( 1 , 1 )=? 2
A( 1 , 2 )=? 3
A( 2 , 1 )=? 4
A( 2 , 2 )=? 5

PRINTING OF THE GIVEN MATRIX

    2.000    3.000
    4.000    5.000

PRINTING OF COFACTOR

    5.000   -4.000
   -3.000    2.000

PRINTING OF THE INVERSE OF THE GIVEN MATRIX

   -2.500    1.500
    2.000   -1.000

VERIFICATION OF THE INVERSE MATRIX

    1.000    0.000
    0.000    1.000
```

Numerical Method (Gauss-Jordan Elimination Method)

The numerical method entitled "Gauss-Jordan Elimination" may be employed to obtain the inverse of a matrix. The procedure of this method involves the following steps.

- Supplement the given matrix with $N \times N$ unit matrix. Thus, the dimensions of the matrix becomes $N \times 2N$.
- Make the lower-half of the original matrix ($N \times N$) equal to zero. This can be achieved by the expression

$$A_{jk} = A_{jk} - A_{ik}\left(\frac{A_{ji}}{A_{ii}}\right)$$

where i varies from 1 to $N - 1$
j varies from $i + 1$ to N;
k varies from 1 to $N2$ where $N2 = 2 * N$.

- Make $A_{N, N}$ equal to one by carrying out the operations

$$A_{Nk} = A_{Nk}/A_{NN}$$

where k varies from N to $N2$.

- Make the upper half of the original matrix equal to zero by carrying out the operations

$$A_{ik} = A_{ik} - A_{jk}\, A_{ij}$$

where i varies from 1 to $N - 1$
j vareis from $i + 1$ to N
k varies from j to $N2$

- The supplemented matrix A_{jk} where
j varies from 1 to N
k varies from $N + 1$ to $N2$
represents the inverse matrix.

Program

```
REM INVERSE OF A MATRIX BY GAUSS-JORDAN ELIMINATION METHOD
5 CLS
20 DIM A(10, 10), B(10, 10), C(10, 10)
25 INPUT "ORDER OF THE MATRIX="; N
26 N2 = N * 2
REM ----ENTERING THE ELEMENTS OF THE MATRIX----
30 FOR I = 1 TO N: FOR J = 1 TO N
40 PRINT "A("; I; ","; J; ")="; : INPUT A(I, J)
45 B(I, J) = A(I, J)
50 NEXT J: NEXT I
60 'CLS
65 PRINT : PRINT "PRINTING OF THE GIVEN MATRIX": PRINT
70 FOR I = 1 TO N: FOR J = 1 TO N
75 PRINT USING "  ###.###"; A(I, J);
80 NEXT J: PRINT : NEXT I
REM ---SUPPLEMENTING GIVEN MATRIX WITH NxN UNIT MATRIX---
```

```
100 FOR I = 1 TO N
105 K = I + N
110 FOR J = N + 1 TO N2
IF J = K THEN A(I, J) = 1 ELSE A(I, J) = 0
125 NEXT J: NEXT I
REM ---MAKING A(I,I) = 1 AND LOWER HALF ELEMENTS = 0----
145 FOR I = 1 TO N - 1
150 DUM = A(I, I)
155 FOR J = 1 TO N2: A(I, J) = A(I, J) / DUM: NEXT J
170 FOR J = I + 1 TO N
175 DUM = A(J, I) / A(I, I)
180 FOR K = 1 TO N2
185 A(J, K) = A(J, K) - A(I, K) * DUM
190 NEXT K: NEXT J: NEXT I
REM ----MAKING A(N,N) EQUAL TO ONE----
210 DUM = A(N, N)
215 FOR J = N TO N2: A(N, J) = A(N, J) / DUM: NEXT J
REM -----MAKE UPPER HALF MATRIX EQUAL TO ZERO----
230 FOR I = 1 TO N - 1: FOR J = I + 1 TO N
240 DUM = A(I, J)
FOR K = J TO N2
250 A(I, K) = A(I, K) - DUM * A(J, K)
255 NEXT K: NEXT J: NEXT I
270 PRINT : PRINT "PRINTING OF THE INVERSE MATRIX": PRINT
275 FOR I = 1 TO N: FOR J = N + 1 TO N2
285 PRINT USING "   ###.###"; A(I, J);
290 NEXT J: PRINT : NEXT I
REM ----VERIFICATION OF THE INVERSE MATRIX----
305 FOR I = 1 TO N: FOR J = 1 TO N
315 SUM = 0
320 FOR K = 1 TO N: SUM = SUM + B(I, K) * A(K, J + N): NEXT K
335 C(I, J) = SUM
340 NEXT J: NEXT I
350 PRINT : PRINT "VERIFICATION OF THE INVERSE MATRIX": PRINT
355 FOR I = 1 TO N: FOR J = 1 TO N
365 PRINT USING "   ###.###"; C(I, J);
370 NEXT J: PRINT : NEXT I
380 END
```

Output

```
ORDER OF THE MATRIX=? 2
A( 1 , 1 )=? 3
A( 1 , 2 )=? 2
A( 2 , 1 )=? 5
A( 2 , 2 )=? 4

PRINTING OF THE GIVEN MATRIX

    3.000    2.000
    5.000    4.000

PRINTING OF THE INVERSE MATRIX

    2.000   -1.000
   -2.500    1.500

VERIFICATION OF THE INVERSE MATRIX

    1.000   -0.000
    0.000    1.000
```

2.16 SOLUTIONS OF LINEAR AND NONLINEAR EQUATIONS

I. Linear Equations

A set of n simultaneous linear equations is of the form

$$a_{11}\,x_1 + \cdots + a_{1n}\,x_n = b_1$$
$$a_{21}\,x_1 + \cdots + a_{2n}\,x_n = b_2$$
$$a_{ni}\,x_1 + \cdots + a_{nn}\,x_n = b_n$$

where the coefficients a_{jk} and the b_j are numbers. In matrix notation, the above equations are represented as

$$\begin{bmatrix} a_{12} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \text{i.e.} \quad \boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$$

One of the methods to determine $\boldsymbol{x}$ is to left-multiply the above expression by $\boldsymbol{A}^{-1}$ to give

$$\boldsymbol{A}^{-1}\,\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{A}^{-1}\,\boldsymbol{b}$$

or

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\,\boldsymbol{b}$$

Thus, left multiplying the $\boldsymbol{b}$ by $\boldsymbol{A}^{-1}$ gives $\boldsymbol{x}$.

Alternatively, the following expressions (known as Cramer's rule) may be used to work out the solutions.

$$x_1 = \frac{\boldsymbol{A}_1}{\boldsymbol{A}},\ x_2 = \frac{\boldsymbol{A}_2}{\boldsymbol{A}}, \ldots, x_n = \frac{\boldsymbol{A}_n}{\boldsymbol{A}}$$

where $\boldsymbol{A}_k$ is the determinant obtained from the determinant $\boldsymbol{A}$ by replacing in $\boldsymbol{A}$ the kth column by the column with the entries in the vector $\boldsymbol{b}$.

Illustration

Solve the equations for x_1, x_2 and x_3 satisfying the relations $3x_1 + 5x_2 + 2x_3 = 8$; $8x_2 + 2x_3 = -7$ and $6x_1 + 2x_2 + 8x_3 = 26$

Program

```
REM PROGRAM EQCRAMR0;DET VIA LOWER HALF ZERO
4 CLS
DIM A(5, 5), A1(5, 5), B(5), X(5), DET2(5)
READ N
FOR I = 1 TO N: FOR J = 1 TO N
READ A(I, J): A1(I, J) = A(I, J)
NEXT J: READ B(I): NEXT I
'DATA 3,2,-1,2,2,1,10,-3,5,-1,1,1,-3
DATA 3,3,5,2,8,0,8,2,-7,6,2,8,26
'DATA 3,9,-2,0,1.5,2,-4,3,4,0,3,7,25.5
PRINT : PRINT "GIVEN DATA"
FOR I = 1 TO N: FOR J = 1 TO N
PRINT A(I, J), : NEXT J: PRINT B(I): NEXT I: PRINT
K1 = 0
IF K1 = 0 THEN 25
16 K1 = K1 + 1
FOR J2 = 1 TO N
FOR JJ = 1 TO N: FOR KK = 1 TO N
IF J2 = KK THEN A(JJ, KK) = B(JJ): GOTO 23
A(JJ, KK) = A1(JJ, KK)
23 NEXT KK: NEXT JJ
25 FOR I = 1 TO N - 1: FOR J = I + 1 TO N
IF A(I, I) = 0 THEN 5
DUM = A(J, I) / A(I, I)
FOR K = I TO N: A(J, K) = A(J, K) - A(I, K) * DUM: NEXT K
NEXT J
5 NEXT I
KK = 0
FOR I = 2 TO N
IF A(I, I) <> 0 THEN 15
KK = KK + 1
FOR J = I + 1 TO N
IF A(J, I) = 0 THEN 10
FOR K = 1 TO N: SWAP A(I, K), A(J, K): NEXT K
10 NEXT J
15 NEXT I
DET1 = 1
FOR I = 1 TO N: DET1 = DET1 * A(I, I): NEXT I
IF K1 = 0 THEN DET = DET1: PRINT "MAIN DETERMINANT="; DET: PRINT : GOTO 16
DET2(J2) = DET1
```

```
PRINT "   D("; J2; ")="; : PRINT USING "####.###"; DET1;
NEXT J2: PRINT
FOR I = 1 TO N
X(I) = DET2(I) / DET
PRINT "   X("; I; ")="; : PRINT USING "####.###"; X(I);
NEXT I
END
```

Output

```
GIVEN DATA
 3            5            2            8
 0            8            2           -7
 6            2            8            26

MAIN DETERMINANT= 144

   D( 1 )= 576.000   D( 2 )=-144.000   D( 3 )=  72.000
   X( 1 )=   4.000   X( 2 )=  -1.000   X( 3 )=   0.500
```

Numerical Methods of Solving Simultaneous Equations

Many numerical methods are available. A few such methods are described in the following.

I. Linear Equations

Gauss Elimination

The matrix $\boldsymbol{A}$ is augmented with the vector $\boldsymbol{b}$. The matrix $\boldsymbol{A}$ along with the vector $\boldsymbol{b}$ is reduced to a triangular form where the lower portion of $\boldsymbol{A}$ involves zero as the matrix elements. This is followed by the determination of $\boldsymbol{x}$ by back substitution. For example, the equations

$$3x_1 + 5x_2 + 2x_3 = 8$$

$$8x_2 + 2x_3 = -7$$

$$6x_1 + 2x_2 + 8x_3 = 26$$

are written as $\begin{bmatrix} 3 & 5 & 2 & 8 \\ 0 & 8 & 2 & -7 \\ 6 & 2 & 8 & 26 \end{bmatrix}$. This is converted into $\begin{bmatrix} 3 & 5 & 2 & 8 \\ 0 & 8 & 2 & -7 \\ 0 & 0 & 6 & 3 \end{bmatrix}$

From which the vector $\boldsymbol{x}$ evaluated by back substition. This is given as

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3}(8 - 5x_2 - 2x_3) = 4 \\ \frac{1}{8}(-7 - 2x_3) = -1 \\ \frac{3}{6} = \frac{1}{2} \end{bmatrix}$$

Program

```
5 REM SIMULTANEOUS EQUATIOS BY GAUSS ELIMINATION METHOD
10 CLS : DIM A(5, 5), B(5), X(5), A$(5)
15 A$(1) = "X1": A$(2) = "X2": A$(3) = "X3": A$(4) = "X4": A$(5) = "X5"
20 PRINT "Number of Equations ="; : READ N
25 FOR I = 1 TO N
35 FOR J = 1 TO N: PRINT "A("; I; J; ")="; : READ A(I, J): NEXT J
45 PRINT "B("; I; ")="; : READ B(I)
50 NEXT I
55 CLS
60 PRINT "Printing of equation parameters": PRINT
65 FOR I = 1 TO N
75 FOR J = 1 TO N: PRINT USING "   ###.##"; A(I, J); : NEXT J
85 PRINT USING "          ###.###"; B(I): PRINT
95 NEXT I
100 REM TO MAKE THE LOWER HALF MATRIX EQUAL TO ZERO
105 FOR I = 1 TO N - 1
110 REM CHECK IF A(I,I)=0 IF SO INTERCHANGE WITH THE ROW HAVING
REM A(K<I) NOT EQUAL TO ZERO
120 K = I
125 IF A(K, K) <> 0 THEN 140
128 K = K + 1
130 GOTO 125
140 IF K = I THEN 185
145 FOR J = 1 TO N: SWAP A(K, J), A(I, J): NEXT J
168 SWAP B(I), B(K)
185 FOR J = I + 1 TO N
190 DUM = A(J, I) / A(I, I)
195 FOR K = 1 TO N: A(J, K) = A(J, K) - A(I, K) * DUM: NEXT K
210 B(J) = B(J) - B(I) * DUM
212 NEXT J
214 PRINT "Step number "; I
215 FOR JJ = 1 TO N: FOR KK = 1 TO N
216 PRINT USING "    ###.##"; A(JJ, KK);
218 NEXT KK: PRINT USING "    ###.##"; B(JJ)
219 NEXT JJ: A$ = INPUT$(1)
220 NEXT I
225 REM CALCULATION OF UNKNOWN VARIABLES
232 IF A(N, N) = 0 THEN 290
235 X(N) = B(N) / A(N, N)
240 FOR I = N - 1 TO 1 STEP -1
245 X(I) = B(I)
246 FOR J = I + 1 TO N: X(I) = X(I) - A(I, J) * X(J): NEXT J
258 X(I) = X(I) / A(I, I)
260 NEXT I
```

```
265 REM PRINTING OF THE VARIABLES
266 PRINT : PRINT "Solution variables": PRINT
270 FOR I = 1 TO N: PRINT "X("; I; ")=";
PRINT USING "###.##"; X(I): NEXT I
285 GOTO 320
290 FOR I = N - 1 TO 1 STEP -1
295 FOR J = I TO N: PRINT "+("; A(I, J); ")*"; A$(J); : NEXT J
310 PRINT "="; B(I)
315 NEXT I
DATA 3, 3,5,2,8,0,8,2,-7,6,2,8,26
'DATA 3,-1,1,2,2,3,-1,1,6,-1,3,4,4
'DATA 2,2,1,4,5,-2,1
'DATA 2,1,1,0,3,-4,1
'DATA 3,-1,1,2,0,3,4,1,0,2,5,3,1
'DATA 3,1,1,1,-1,0,4,6,6,0,1,1,1
'DATA 3,2,2,-4,-1,0,3,2,9,-4,1,3,4.5
''DATA 3,0,2,-1,1,4,-10,3,5,3,-3,0,6
'DATA 3,3,4,6,1,-2,8,-4,2,4,-8,8,-2
'DATA 3,0,4,7,-13,5,-3,4,-23,-1,2,-8,29
'DATA 4,1,1,1,0,3,-3,-17,1,2,1,4,-17,8,-5,1,0,-5,-2,1,1
320 END
```

Output

```
Printing of equation parameters

    3.00    5.00    2.00          8.000

    0.00    8.00    2.00         -7.000

    6.00    2.00    8.00         26.000

Step number  1
     3.00     5.00     2.00     8.00
     0.00     8.00     2.00    -7.00
     0.00    -8.00     4.00    10.00
Step number  2
     3.00     5.00     2.00     8.00
     0.00     8.00     2.00    -7.00
     0.00     0.00     6.00     3.00

Solution variables

X( 1 )=  4.00
X( 2 )= -1.00
X( 3 )=  0.50
```

Exercises

Determine the vlaues of variables in the following equations.

1. $3x + 5y + 2z = 8$; $8y + 2z = -7$; $6x + 2y + z = 26$

2. $-x + y + 2z = 2$; $3x - y + z = 6$; $-x + 3y + 4z = 4$

3. $2x + y = 4$; $5x - 2y = 1$

4. $x + y = 0$; $3x - 4y = 1$

5. $x + y + z = -1$; $4y + 6z = 6$; $y + z = 1$

6. $2x + 2y - 4z = -1$; $3y + 2z = 9$; $-4x + y + 3z = 4.5$

7. $3x + 4y + 6z = 1$; $-2x + 8y - 4z = 2$; $4x - 8y + 8z = -2$

8. $4y + 7z = -13$; $5x - 3y + 4z = -23$; $-x + 2y - 8z = 29$

LU-Factorization (Doolittle's method)

The systems of equations represented by $\boldsymbol{Ax} = \boldsymbol{b}$ is written as

$$(\boldsymbol{LU})\boldsymbol{x} = \boldsymbol{b}$$

where $\boldsymbol{L}$ is lower triangular and $\boldsymbol{U}$ is upper triangular. With this, the solution of $\boldsymbol{Ax} = \boldsymbol{b}$ is reduced in two steps:

$$\boldsymbol{Ly} = \boldsymbol{b} \quad \text{where} \quad \boldsymbol{Ux} = \boldsymbol{y}$$

Firstly, $\boldsymbol{Ly} = \boldsymbol{b}$ is solved for $\boldsymbol{y}$ and then $\boldsymbol{Ux} = \boldsymbol{y}$ is solved for $\boldsymbol{x}$.

For example, the matrix $\boldsymbol{A}$ described on P.57 in the Gauss elimination is written as

$$\boldsymbol{A} = \begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

In the explicit form, the above expression is given by the following expressions.

$$A_{11} = u_{11} \quad (1) \qquad A_{12} = u_{12} \quad (2) \qquad A_{13} = u_{13} \quad (3)$$

$$A_{21} = m_{21}\, u_{11} \quad (4) \qquad A_{22} = m_{21}\, u_{12} + u_{22} \quad (5) \qquad A_{23} = m_{21}\, u_{13} + u_{23} \quad (6)$$

$$A_{31} = m_{31}\, u_{11} \quad (7) \qquad A_{32} = m_{31}\, u_{12} + m_{32}\, u_{22} \quad (8) \qquad A_{33} = m_{31}\, u_{13} + m_{32}\, u_{23} + u_{33} \quad (9)$$

The unknown parameters m_{21}, ... and u_{11}, ... are determined in the order shown above.

Using the above expressions, we will get

$$A = \begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{bmatrix}$$

Firstly, we solve $\boldsymbol{Ly} = \boldsymbol{b}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -7 \\ 26 \end{bmatrix} \quad \text{This gives } \boldsymbol{y} = \begin{bmatrix} y_1 = 8/1 \\ y_2 = -7(0)y_1 = -7 \\ y_3 = 26 - 2y_1 + y_2 = 3 \end{bmatrix}$$

Then, we solved $\boldsymbol{Ux} = \boldsymbol{y}$

$$\begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -7 \\ 3 \end{bmatrix} \quad \text{This gives} \quad \boldsymbol{x} = \begin{bmatrix} x_1 = \frac{1}{3}(8 - 5x_2 - 2x_2) = 4 \\ x_2 = \frac{1}{8}(-7 - 2x_3) = -1) \\ x_3 = \frac{3}{6} = \frac{1}{2} \end{bmatrix}$$

(The generalized expressions of u' s and m's are as follows.

$$u_{1k} = a_{1k} \qquad ; \qquad k = 1, \ldots, n$$

$$m_{j1} = \frac{a_{j1}}{a_{11}} \qquad ; \qquad j = 2, \ldots, n$$

$$u_{jk} = a_{jk} - \sum_{s=1}^{j-1} m_{js} u_{sk} \qquad ; \qquad k = j, \ldots, n \; ; \quad j \geq 2$$

$$m_{jk} = \frac{1}{u_{kk}} \left(a_{jk} - \sum_{s=1}^{k-1} m_{js} u_{sk} \right) \qquad ; \qquad j = k + 1, \ldots, n; \quad k \geq 2$$

Program

```
REM PROGRAM LINEQLU;SOLUTION OF LINEAR EQUATIONS BY FACTORING
CLS : DIM A(5, 5), B(5), M(5, 5), U(5, 5), X(5), Y(5)
READ N: DATA 3
FOR I = 1 TO N: FOR J = 1 TO N
READ A(I, J): NEXT J: READ B(I): NEXT I
FOR I = 1 TO N: FOR J = I TO N
IF I = J THEN M(I, J) = 1 ELSE M(I, J) = 0
NEXT J: NEXT I
```

```
FOR I = 2 TO N: FOR J = 1 TO I - 1: U(I, J) = 0: NEXT J: NEXT I
FOR J = 1 TO N: FOR K = 1 TO N
SUM = 0
IF J = 1 THEN U(J, K) = A(J, K): GOTO 10
IF J <= K THEN 5
FOR L = 1 TO K - 1: SUM = SUM + M(J, L) * U(L, K): NEXT L
M(J, K) = (A(J, K) - SUM) / U(K, K)
GOTO 10
5 FOR L = 1 TO J - 1: SUM = SUM + M(J, L) * U(L, K): NEXT L
U(J, K) = A(J, K) - SUM
10 NEXT K
NEXT J
PRINT "A MATRIX"
FOR I = 1 TO N: FOR J = 1 TO N: PRINT A(I, J),
NEXT J: PRINT B(I): NEXT I: PRINT
PRINT "Lower Matrix"
FOR I = 1 TO N: FOR J = 1 TO N: PRINT M(I, J),
NEXT J: PRINT : NEXT I: PRINT
PRINT "Upper Matrix"
FOR I = 1 TO N: FOR J = 1 TO N: PRINT U(I, J),
NEXT J: PRINT : NEXT I: PRINT
FOR J = 1 TO N
SUM = 0
FOR K = 1 TO J - 1: SUM = SUM + M(J, K) * Y(K): NEXT K
Y(J) = (B(J) - SUM) / M(J, J)
NEXT J
PRINT "Y vector"
FOR J = 1 TO N: PRINT Y(J), : NEXT J: PRINT
FOR J = N TO 1 STEP -1
SUM = 0
FOR K = J + 1 TO N: SUM = SUM + U(J, K) * X(K): NEXT K
X(J) = (Y(J) - SUM) / U(J, J)
NEXT J
PRINT "X vector"
FOR J = 1 TO N: PRINT X(J), : NEXT J
DATA 3, 5, 2, 8, 0, 8, 2, -7, 6, 2, 8, 26
'DATA 1,2,3,17,2,5,8,44,3,8,14,76
'DATA 5,4,1,3.4,10,9,4,8.8,10,13,15,19.2
'DATA 3,2,4,5.8,6,13,11,23.6,12,11,26,30.8
'DATA 8,1,1,-1,18,-2,12,-1,0,-7,2,0,16,2,54,0,1,2,-20,-14
'DATA 4,2,14,14,2,17,-5,-101,14,-5,83,155
END
```

Output

```
A MATRIX
 3            5            2            8
 0            8            2           -7
 6            2            8            26

Lower Matrix
 1            0            0
 0            1            0
 2           -1            1

Upper Matrix
 3            5            2
 0            8            2
 0            0            6

Y vector
 8           -7            3
X vector
 4           -1            .5
```

Gauss-Seidel Method

Gauss-Seidel method for solving linear equations is an iterative method in which we write linear equations

$$
\begin{aligned}
&a_{11}\,x_1 + a_{12}\,x_2 + \cdots + a_{1n}\,x_n = b_1\\
&a_{21}\,x_1 + a_{22}\,x_2 + \cdots + a_{2n}\,x_n = b_2\\
&\vdots\\
&a_{n1}\,x_1 + a_{n2}\,x_2 + \cdots + a_{nn}\,x_n = b_n
\end{aligned}
$$

as

$$
\begin{aligned}
x_1 &= \frac{b_1}{a_{11}} - \left(\frac{a_{12}}{a_{11}}\,x_2 + \frac{a_{13}}{a_{11}}\,x_3 + \cdots + \frac{a_{1n}}{a_{11}}\,x_n\right)\\
x_2 &= \frac{b_2}{a_{22}} - \left(\frac{a_{21}}{a_{22}}\,x_1 + \frac{a_{23}}{a_{22}}\,x_3 + \cdots + \frac{a_{2n}}{a_{22}}\,x_n\right)\\
&\vdots\\
x_n &= \frac{b_n}{a_{nn}} - \left(\frac{a_{n1}}{a_{nn}}\,x_1 + \frac{a_{n2}}{a_{nn}}\,x_2 + \cdots + \frac{a_{n,n-1}}{a_{nn}}\,x_{n-1}\right)
\end{aligned}
$$

Let $x_1^{(1)}$, $x_2^{(1)}$, ..., $x_n^{(1)}$ be the initial approximate solutions of the linear equations. These are substituted in the first equation to yield refined value of $x_1^{(2)}$. This along with the rest of approximate x's are substituted in the second equation to yield $x_2^{(2)}$. This process is continued till all the refined solutions are obtained. The process is once again repeated starting from the first equation and continued till all the refined solutions are obtained. This iteration provides the refined roots $x_1^{(3)}$, $x_2^{(3)}$, ..., $x_n^{(3)}$. This iteration is continued until the roots are obtained within the prescribed accuracy limit.

Since the iterative equations involve division by a_{ii}, this element should not be equal to zero for any value of i. If this is so, the equations should be rearranged so as to provide nonzero value of a_{ii}.

The Gauss-Seidel method converges provided the sum given below is almost equal to one or at least in one of the equations, this value is less than or near to one.

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| \leq 1; \quad (i = 1, 2, \ldots, n)$$

Illustration

Solve the equations

$$x_1 - 0.25\, x_2 - 0.25\, x_3 = 50$$
$$-0.25\, x_1 + x_2 - 0.25\, x_4 = 50$$
$$-0.25\, x_1 + x_3 - 0.25\, x_4 = 25$$
$$-0.25\, x_2 - 0.25\, x_3 + x_4 = 25$$

Program

```
REM Iterative method for the sollution of linear equations
CLS
READ ITER: DATA 10
READ N
PRINT "    ********** A Matrix **********   B Vector"
FOR I = 1 TO N: FOR J = 1 TO N
READ A(I, J): PRINT USING "   ###.##"; A(I, J);
NEXT J
READ B(I): PRINT USING "   ###.##"; B(I)
NEXT I
DATA 4, 1, -.25, -.25, 0, 50, -.25, 1, 0, -.25, 50
DATA -.25, 0, 1, -.25, 25, 0, -.25, -.25, 1, 25
'DATA 4,2,-1,0,0,0,-1,2,-1,0,0
'DATA 0,-1,2,-1,0,0,0,-1,2,1
'DATA 4,10,-2,-1,-1,3,-2,10,-1,-1,15
'DATA -1,-1,10,-2,27,-1,-1,-2,10,-9
'DATA 3,10,2,1,9,2,20,-2,-44,-2,3,10,22
FOR I = 1 TO N: X(I) = 0: NEXT I
PRINT : PRINT STRING$(48, "-")
PRINT "ITER     ";
FOR I = 1 TO N: PRINT "X("; I; ")       "; : NEXT I: PRINT
PRINT STRING$(48, "-")
FOR K = 1 TO ITER
PRINT USING "##"; K;
FOR I = 1 TO N
IF A(I, I) <> 0 THEN 10
FOR J = I + 1 TO N
IF I = J THEN 5
IF A(J, J) <> 0 THEN
FOR L = 1 TO N: SWAP A(I, L), A(J, L): NEXT L
SWAP B(I), B(J)
END IF
```

```
5 NEXT J
10 X(I) = B(I) / A(I, I)
FOR J = 1 TO N
IF I = J THEN 20
X(I) = X(I) - (A(I, J) / A(I, I)) * X(J)
20 NEXT J
PRINT USING "     ####.##"; X(I);
NEXT I: PRINT
NEXT K
PRINT STRING$(48, "-")
END
```

Output

```
********** A Matrix **********  B Vector
 1.00   -0.25   -0.25    0.00   50.00
-0.25    1.00    0.00   -0.25   50.00
-0.25    0.00    1.00   -0.25   25.00
 0.00   -0.25   -0.25    1.00   25.00

------------------------------------------------
ITER    X( 1 )     X( 2 )     X( 3 )     X( 4 )
------------------------------------------------
 1      50.00      62.50      37.50      50.00
 2      75.00      81.25      56.25      59.38
 3      84.38      85.94      60.94      61.72
 4      86.72      87.11      62.11      62.30
 5      87.30      87.40      62.40      62.45
 6      87.45      87.48      62.48      62.49
 7      87.49      87.49      62.49      62.50
 8      87.50      87.50      62.50      62.50
 9      87.50      87.50      62.50      62.50
10      87.50      87.50      62.50      62.50
------------------------------------------------
```

Exercises Solve the variables in the following equations.

1.
$$x + 2y + 3z = 17$$
$$2x + 5y + 8z = 44$$
$$3x + 8y + 14z = 76$$

2.
$$5x + 4y + z = 3.4$$
$$10x + 9y + 4z = 8.8$$
$$10x + 13y + 15z = 19.2$$

3.
$$3x + 2y + 4z = 5.8$$
$$6x + 13y + 11z = 23.6$$
$$12x + 11y + 26z = 30.8$$

4.
$$4x + 2y + 14z = 14$$
$$2x + 17y - 5z = -101$$
$$14x - 5y + 83z = 155$$

II. Nonlinear Equations

Newton-Raphson Method

Let the nonlinear equations may be represented as

$$f(x, y) = 0 \quad \text{and} \quad g(x, y) = 0$$

Let the initial approximate roots of the above two equations may be equal to x_0 and y_0, respectively If $x_0 + h$ and $y_0 + k$ are the correct roots, then we will have

$$f(x_0 + h, y_0 + k) = 0 \quad \text{and} \quad g(x_0 + h, y_0 + k) = 0$$

If the functions are differentiable, then by Taylor's series, the above two functions may be written as

$$f(x_0 + h, y_0 + k) = f_0 + h\frac{\partial f}{\partial x_0} + k\frac{\partial f}{\partial y_0} + \cdots = 0$$

$$g(x_0 + h, y_0 + k) = g_0 + h\frac{\partial g}{\partial x_0} + k\frac{\partial g}{\partial y_0} + \cdots = 0$$

where the subscript 0 represents the values of functions and their derivatives at the initial approximate roots x_0 and y_0. If the higher-order terms in the above two expressions are neglected, then we can write

$$f_0 + h\frac{\partial f}{\partial x_0} + k\frac{\partial f}{\partial y_0} = 0$$

$$g_0 + h\frac{\partial g}{\partial x_0} + k\frac{\partial g}{\partial y_0} = 0$$

In the matrix notations, we have

$$\begin{bmatrix} \partial f/\partial x_0 & \partial f/\partial y_0 \\ \partial g/\partial x_0 & \partial g/\partial y_0 \end{bmatrix} \begin{bmatrix} h \\ k \end{bmatrix} = \begin{bmatrix} -f_0 \\ -g_0 \end{bmatrix}$$

Hence

$$\begin{bmatrix} h \\ k \end{bmatrix} = \begin{bmatrix} \partial f/\partial x_0 & -\partial f/\partial y_0 \\ \partial g/\partial x_0 & \partial g/\partial y_0 \end{bmatrix}^{-1} \begin{bmatrix} -f_0 \\ -g_0 \end{bmatrix}$$

$$= \frac{1}{j(f,g)} \begin{bmatrix} \partial g/\partial y_0 & -\partial f/\partial y_0 \\ -\partial g/\partial x_0 & \partial f/\partial x_0 \end{bmatrix} \begin{bmatrix} -f_0 \\ -g_0 \end{bmatrix}$$

where

$$J(f, g) = \begin{vmatrix} \partial f/\partial x_0 & \partial f/\partial y_0 \\ \partial g/\partial x_0 & \partial g/\partial y_0 \end{vmatrix}$$

Thus

$$h = \frac{1}{J(f,g)} \left(-f_0 \frac{\partial g}{\partial y_0} + g_0 \frac{\partial f}{\partial y_0} \right)$$

$$k = \frac{1}{J(f,g)} \left(f_0 \frac{\partial g}{\partial x_0} - g_0 \frac{\partial f}{\partial x_0} \right)$$

The new roots are $x = x_0 + h$ and $y = x_0 + k$

The above process is repeated till the two roots are obtained within the desired accuracy.

Illustration To obtain the roots of $x^2 - y^2 = 4$ and $x^2 + y^2 = 16$ by the Newton-Raphson method.

Program

```
DEF FNA (X, Y) = X ^ 2 - Y ^ 2 - 4
DEF FNB (X, Y) = 2 * X: DEF FNC (X, Y) = -2 * Y
DEF FND (X, Y) = X ^ 2 + Y ^ 2 - 16
DEF FNE (X, Y) = 2 * X: DEF FNF (X, Y) = 2 * Y
'DEF FNA (X, Y) = X ^ 2 - 3 * X * Y + 7
'DEF FNB (X, Y) = 2 * X - 3 * Y: DEF FNC (X, Y) = -3 * X
'DEF FND (X, Y) = Y - 2 * X - 2
'DEF FNE (X, Y) = -2: DEF FNF (X, Y) = 1
M = 1: XD = 3
5 ON M GOTO 10, 20, 30, 40, 80
10 X0 = -XD: Y0 = -XD: GOTO 50
20 X0 = -XD:  Y0 = XD: GOTO 50
30 X0 = XD: Y0 = -XD: GOTO 50
40 X0 = XD: Y0 = XD
50 PRINT "Given approximate roots are    ";
PRINT "X0="; X0; "   Y0="; Y0
55 F0 = FNA(X0, Y0): G0 = FND(X0, Y0)
DFDX = FNB(X0, Y0): DFDY = FNC(X0, Y0)
DGDX = FNE(X0, Y0): DGDY = FNF(X0, Y0)
DET = DFDX * DGDY - DFDY * DGDX
IF DET = 0 THEN 70
H = (-F0 * DGDY + G0 * DFDY) / DET
K = (-G0 * DFDX + F0 * DGDX) / DET
IF ABS(H) < .0001 AND ABS(K) < .0001 THEN 60
X0 = X0 + H: Y0 = Y0 + K
'PRINT "F0="; F0; "   DFDX="; DFDX; "   DFDY="; DFDY
'PRINT "G0="; G0; "   DGDX="; DGDX; "   DGDY="; DGDY
'PRINT "DET="; DET
'PRINT "H="; H; "  K="; K; "  X0="; X0; "  Y0="; Y0
'A$ = INPUT$(1)
'PRINT
GOTO 55
60 PRINT "Roots of the given equations are    ";
PRINT "X="; : PRINT USING "##.####"; X0;
PRINT "  Y="; : PRINT USING "##.####"; Y0: PRINT
M = M + 1: A$ = INPUT$(1): GOTO 5
70 PRINT "SOLUTION IS NOT POSSIBLE"
80 END
```

Output

```
Given approximate roots are    X0=-3    Y0=-3
Roots of the given equations are    X=-3.1623  Y=-2.4495

Given approximate roots are    X0=-3    Y0= 3
Roots of the given equations are    X=-3.1623  Y= 2.4495

Given approximate roots are    X0= 3    Y0=-3
Roots of the given equations are    X= 3.1623  Y=-2.4495

Given approximate roots are    X0= 3    Y0= 3
Roots of the given equations are    X= 3.1623  Y= 2.4495
```

Exercise

Determine the roots of the equations $x^2 = 3xy - 7$ and $y = 2(x + 1)$

(**Ans:** – 1.9267, – 1.8533; 0.7266, 3.4533)

2.17 BALANCING CHEMICAL EQUATION

The procedure to be adopted is described by considering the chemical equation

$$CH_4 + O_2 \rightarrow CO_2 + H_2O$$

In this equation, the number of different elements appeared are identified. These are C, O and H. A matrix is formed in which rows are identified with elements and columns are identified with the different species appeared in the chemical equation. The elements of the matrix are the number of elements appeared in different chemical species of the chemical equation. In the present case, we have

N ↓	M →	1 CH_4	2 O_2	3 CO_2	4 H_2O
1	C	1	0	1	0
2	O	0	2	2	1
3	H	4	0	0	2

Note That one element is selected from each reactant in sequence until all the elements are inserted.

The above matrix is transformed into an equivalent matrix with all off-diagonal on the left of diagonal elements equal to zero.

The above matrix is thus transformed into the following matrix.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & -4 & 2 \end{bmatrix}$$

From this matrix, the following coefficients are computed by back substitution (see P.57) using the expressions.

$$X_{M-1} = \frac{A_{N,M}}{A_{N,N}} \qquad \text{where } N = 3 \text{ and } M = 4$$

$$X_i = \frac{A_{i,M} - \sum_{j=i+1}^{N} A_{i,j} \times X_j}{A_{i,i}} \qquad \text{where } i \text{ goes from } N - 1 \text{ to } 1$$

The stoiciometric numbers are negative of the above coefficients with the stoichimetric number of Mth species equal to 1. The negative coefficients are those of reactants while the positive ones are for products.

Illustration Balance the chemical equation

$Na_2CO_3 + Br_2 \rightarrow NaBr + NaBrO_3 + CO_2$

Program

```
REM PROGRAM BALEQ;BALANCING CHEMICAL EQUATION
REM No. OF ELEMENTS MUST BE ONE LESS THAN No. OF SPEICES
CLS : DIM A(10, 10), B(10), X(10), M$(10), NE$(10)
PRINT "NUMBER OF ELEMENTS ="; : READ N: PRINT N
PRINT "NUMBER OF SPEICES IN THE EQUATION ="; : READ M: PRINT M
FOR I = 1 TO M: READ M$(I): PRINT I; M$(I), : NEXT I: PRINT
FOR I = 1 TO N: READ NE$(I): NEXT I
FOR I = 1 TO N: PRINT NE$(I): FOR J = 1 TO M
PRINT "A("; I; ","; J; ")="; : READ A(I, J): PRINT A(I, J),
NEXT J: PRINT : NEXT I
A$ = INPUT$(1): CLS
PRINT "Printing of equation parameters": PRINT
FOR I = 1 TO N: FOR J = 1 TO M
PRINT USING "   ###.#"; A(I, J);
NEXT J: PRINT : NEXT I
REM TO MAKE THE LOWER HALF MATRIX EQUAL TO ZERO
FOR I = 1 TO N - 1
REM CHECK IF A(I,I)=0 IF SO INTERCHANGE WITH THE ROW HAVING
REM A(K,I) NOT EQUAL TO ZERO
K = I
10 IF A(K, K) <> 0 THEN 20
K = K + 1
GOTO 10
20 IF K = I THEN 30
FOR J = 1 TO M: SWAP A(K, J), A(I, J): NEXT J
30 FOR J = I + 1 TO N
DUM = A(J, I) / A(I, I)
```

```
FOR K = 1 TO M
A(J, K) = A(J, K) - A(I, K) * DUM
NEXT K: NEXT J
PRINT "Step number "; I
FOR JJ = 1 TO N: FOR KK = 1 TO M
PRINT USING "    ###.#"; A(JJ, KK);
NEXT KK: PRINT : A$ = INPUT$(1)
NEXT JJ
NEXT I
REM CALCULATION OF STOICHIOMETRIC NUMBERS
IF A(N, N) = 0 THEN 290
X(N) = A(N, M) / A(N, N)
290 FOR I = N - 1 TO 1 STEP -1
X(I) = A(I, M)
FOR J = I + 1 TO N: X(I) = X(I) - A(I, J) * X(J): NEXT J
X(I) = X(I) / A(I, I)
NEXT I
MIN = 10
FOR I = 1 TO N
IF ABS(X(I)) < MIN THEN MIN = ABS(X(I))
NEXT I
FOR I = 1 TO N: X(I) = X(I) / MIN: NEXT I
REM PRINTING OF STOICHIOMETRIC NUMBERS
PRINT : PRINT "Stoichimetric numbers": PRINT
FOR I = 1 TO N: PRINT "X("; I; ")=";
PRINT USING "##.#    "; -X(I); : NEXT I
PRINT USING "    ##.#"; 1 / MIN
PRINT : PRINT "Balanced chemical equation is": PRINT
PRINT "0 =";
FOR I = 1 TO M - 1
PRINT "("; : PRINT USING "##.#"; -X(I); : PRINT ")"; M$(I); "+";
NEXT I
PRINT "("; : PRINT USING "##.#"; 1 / MIN; : PRINT ")"; M$(M)
GOTO 40
PRINT "SOLUTION IS NOT UNIQUE"
REM Combustion of methane
'DATA 3,4,CH4,O2,CO2,H2O,C,O,H
'DATA 1,0,1,0,0,2,2,1,4,0,0,2
REM Combustion of pentane
'DATA 3,4,C5H12,O2,CO2,H2O,C,O,H
'DATA 5,0,1,0,0,2,2,1,12,0,0,2:
REM Combustion of sucrose
'DATA 3,4,C12H22O11,O2,CO2,H2O,C,O,H
```

```
'DATA 12,0,1,0,11,2,2,1,22,0,0,2
'DATA 4,5,Zn,HNO3,Zn(NO3)2,N2O,H2O,Zn,H,N,O
'DATA 1,0,1,0,0,0,1,0,0,2,0,1,2,2,0,0,3,6,1,1
'DATA 4,5,Pb3O4,H2SO4,PbSO4,H2O,O2,Pb,O,H,S
'DATA 3,0,1,0,0,4,4,4,1,2,0,2,0,2,0,0,1,1,0,0
'DATA 4,5,P4O11,HNO2,H2O,HPO3,N2O5,P,O,H,N
'DATA 4,0,0,1,0,11,2,1,3,5,0,1,2,1,0,0,1,0,0,2
'DATA 4,5,HNO3,I2,NO2,HIO3,H2O,N,I,O,H
'DATA 1,0,1,0,0,0,2,0,1,0,3,0,2,3,1,1,0,0,1,2
'DATA 4,5,Mg,HNO3,Mg(NO3)2,NH4NO3,H2O,Mg,N,H,O
'DATA 1,0,1,0,0,0,1,2,2,0,0,1,0,4,2,0,3,6,3,1
'DATA 4,5,P4,HNO3,H3PO4,NO2,H2O,P,N,H,O
'DATA 4,0,1,0,0,0,1,0,1,0,0,1,3,0,2,0,3,4,2,1: REM e
'DATA 5,6,Zn,NaNO3,NaOH,NH3,Na2ZnO2,H2O,Zn,Na,N,O,H
'DATA 1,0,0,0,1,0,0,1,1,0,2,0,0,1,0,1,0,0,0,3,1,0,2,1,0,0,1,3,0,2: REM q
'DATA 5,6,KClO3,H2SO4,HClO4,KHSO4,H2O,ClO2,K,S,Cl,O,H
'DATA 1,0,0,1,0,0,0,1,0,1,0,0,1,0,1,0,0,1,3,4,4,4,1,2,0,2,1,1,2,0: REM y
'DATA 4,5,Cu,HNO3,Cu(NO3)2,NO,H2O,Cu,N,H,O
'DATA 1,0,1,0,0,0,1,2,1,0,0,1,0,0,2,0,3,6,1,1: REM o
'DATA 5,6,Au,HNO3,HCl,HAuCl4,NO,H2O,Au,N,Cl,H,O
'DATA 1,0,0,1,0,0,0,1,0,0,1,0,0,0,1,4,0,0,0,1,1,1,0,2,0,3,0,0,1,1: REM r
'DATA 4,5,MnO2,KOH,O2,K2MnO4,H2O,Mn,K,O,H
'DATA 1,0,0,1,0,0,1,0,2,0,2,1,2,4,1,0,1,0,0,2
DATA 4,5,Na2CO3,Br2,NaBr,NaBrO3,CO2,Na,Br,C,O
DATA 2,0,1,1,0,0,2,1,1,0,1,0,0,0,1,3,0,0,3,2
40 END
```

Output

```
NUMBER OF ELEMENTS = 4
NUMBER OF SPEICES IN THE EQUATION = 5
 1 Na2CO3      2 Br2         3 NaBr        4 NaBrO3      5 CO2

Na
A( 1 , 1 )= 2              A( 1 , 2 )= 0              A( 1 , 3 )= 1
A( 1 , 4 )= 1              A( 1 , 5 )= 0
Br
A( 2 , 1 )= 0              A( 2 , 2 )= 2              A( 2 , 3 )= 1
A( 2 , 4 )= 1              A( 2 , 5 )= 0
C
A( 3 , 1 )= 1              A( 3 , 2 )= 0              A( 3 , 3 )= 0
A( 3 , 4 )= 0              A( 3 , 5 )= 1
O
A( 4 , 1 )= 3              A( 4 , 2 )= 0              A( 4 , 3 )= 0
A( 4 , 4 )= 3              A( 4 , 5 )= 2
```

```
Printing of equation parameters

    2.0    0.0    1.0    1.0    0.0
    0.0    2.0    1.0    1.0    0.0
    1.0    0.0    0.0    0.0    1.0
    3.0    0.0    0.0    3.0    2.0
Step number  1
     2.0     0.0     1.0     1.0     0.0
     0.0     2.0     1.0     1.0     0.0
     0.0     0.0    -0.5    -0.5     1.0
     0.0     0.0    -1.5     1.5     2.0
Step number  2
     2.0     0.0     1.0     1.0     0.0
     0.0     2.0     1.0     1.0     0.0
     0.0     0.0    -0.5    -0.5     1.0
     0.0     0.0    -1.5     1.5     2.0
Step number  3
     2.0     0.0     1.0     1.0     0.0
     0.0     2.0     1.0     1.0     0.0
     0.0     0.0    -0.5    -0.5     1.0
     0.0     0.0     0.0     3.0    -1.0

Stoichimetric numbers

X( 1 )=-3.0   X( 2 )=-3.0   X( 3 )= 5.0   X( 4 )= 1.0       3.0

Balanced chemical equation is

0 =(-3.0)Na2CO3+(-3.0)Br2+( 5.0)NaBr+( 1.0)NaBrO3+( 3.0)CO2
```

Exercises

Balance the following chemical equations.

1. $CH_4 + O_2 \rightarrow CO_2 + H_2O$
2. $C_5H_{12} + O_2 \rightarrow CO_2 + H_2O$
3. $C_{12}H_{22}O_{11} + O_2 \rightarrow CO_2 + H_2O$
4. $Zn + HNO_3 \rightarrow Zn(NO_3)_2 + N_2O + H_2O$
5. $Pb_3O_4 + H_2SO_4 \rightarrow PbSO_4 + H_2O + O_2$
6. $P_4O_{11} + HNO_2 + H_2O \rightarrow HPO_3 + N_2O_5$
7. $HNO_3 + I_2 \rightarrow NO_2 + HIO_3 + H_2O$

8. $Mg + HNO_3 \rightarrow Mg(NO_3)_2 + NH_4NO_3 + H_2O$
9. $P_4 + HNO_3 \rightarrow H_3PO_4 + NO_2 + H_2O$
10. $Zn + NaNO_3 + NaOH \rightarrow NH_3 + Na_2ZnO_2 + H_2O$
11. $KClO_3 + H_2SO_4 \rightarrow HClO_4 + KHSO_4 + H_2O + ClO_2$
12. $Cu + HNO_3 \rightarrow Cu(NO_3)_2 + NO + H_2O$
13. $Au + HNO_3 + HCl \rightarrow HAuCl_4 + NO + H_2O$
14. $MnO_2 + KOH + O_2 \rightarrow K_2MnO_4 + H_2O$

2.18 NUMERICAL INTERPOLATION

To carry out the numerical interpolation of $f(x)$ for a given value of x.

Interpolation implies finding the value of $f(x)$ for a given x in between the given x's at which the values of $f(x)$ are known. We consider here a few standard methods available to carry out the interpolation.

Linear Interpolation

This involves interpolation by means of the straight line. The expression to be used is

$$p_1(x) = f_0 + (x - x_0) f[x_0, x_1]$$

where $p_1(x)$ is the estimated value of f at x and $f[x_0, x_1]$ is given by

$$f[x_0, x_1] = \frac{f_0 - f_1}{x_0 - x_1}$$

Quadratic Interpolation

This involves interpolation by means of the equation of at most second degree satisfying the three data points (x_0, f_0), (x_1, f_1) and (x_2, f_2).

The equation to be used is

$$p_2(x) = f_0 + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2]$$

where $f[x_0, x_1, x_2]$, known as the second divided difference, is given by

$$f[x_0, x_1, x_2] = \frac{f(x_0, x_1) - f(x_1, x_2)}{x_0 - x_2}$$

Newton's Divided Difference Interpolation

From the definition of divided differences, we have

$$f[x, x_0] = \frac{f_x - f_0}{x - x_0} \tag{1}$$

This gives $$f_x = f_0 + (x - x_0) f[x, x_0] \tag{2}$$

Again $$f[x, x_0, x_1] = \frac{f[x, x_0] - f[x_0, x_1]}{x_0 - x_1} \tag{3}$$

This gives $$f[x, x_0] = f[x_0, x_1] + (x - x_1) f[x, x_0, x_1] \quad (4)$$

Substituting Eq. (4) in Eq. (2), we get

$$f_x = f_0 + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x, x_0, x_1] \quad (5)$$

Again $$f[x, x_0, x_1, x_2] = \frac{f[x, x_0, x_1] - f[x_0, x_1, x_2]}{x - x_2} \quad (6)$$

This gives $$f[x, x_0, x_1] = f[x_0, x_1, x_2] + (x - x_2) f[x, x_0, x_1, x_2] \quad (7)$$

Substituting Eq. (7) in Eq. (5), we get

$$f_x = f_0 + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) f[x, x_0, x_1, x_2] \quad (8)$$

Continuing this process, we get the general expression

$$f_x = f_0 + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) f[x_0, x_1, x_2, x_3] + \cdots + (x - x_0)(x - x_1) \ldots (x - x_{n-1}) f[x_0, x_1, x_2, \ldots, x_n] \quad (9)$$

where $$f[x_0, x_1] = \frac{f_0 - f_1}{x_0 - x_1}$$

$$f[x_0, x_1, x_2] = \frac{f[x_0, x_1] - f[x_1, x_2]}{(x_0 - x_2)}$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_0, x_1, x_2] - f[x_1, x_2, x_3]}{(x_0 - x_3)}$$

$$\vdots$$

Diagrammatically, the expressions $f[x_0, x_1]$, $f[x_0, x_1, x_2]$, ..., etc, may be represented as follows.

j	x_j	f_j	$f[x_j, x_{j+1}]$	$f[x_j, x_{j+1}, x_{j+2}]$	$f[x_j, x_{j+1}, x_{j+2}, x_{j+3}]$
0	x_0	f_0	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
1	x_1	f_1	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3, x_4]$
2	x_2	f_2	$f[x_2, x_3]$	$f[x_2, x_3, x_4]$	
3	x_3	f_3	$f[x_3, x_4]$		
4	x_4	f_4			

Illustration To determine the value of ln 9.2 Given: ln x, when x = 8, 8.5, 9, and 9.5.

Program

```
REM PROG19A
REM PROGRAM INTERNDD;NEWTON DIVIDED DIFFERENCE INTERPOLATION
CLS : DIM X(20), F(20, 20)
```

```
DEF FNA (X) = LOG(X)
READ N: DATA 3
FOR I = 0 TO N: READ X(I): NEXT I
DATA 8,8.5,9,9.5
READ XX: DATA 9.2
FOR J = 0 TO N: F(0, J) = FNA(X(J))
PRINT "X("; J; ")="; X(J); "  F(0"; ","; J; ")="; F(0, J),
NEXT J: PRINT
FOR I = 1 TO N: FOR J = 0 TO N - I
F(I, J) = (F(I - 1, J + 1) - F(I - 1, J)) / (X(J + I) - X(J))
PRINT "F("; I; ","; J; ")="; F(I, J),
NEXT J: PRINT : NEXT I: PRINT
FX = F(0, 0) + F(1, 0) * (XX - X(0))
FX = FX + F(2, 0) * (XX - X(0)) * (XX - X(1))
FX = FX + F(3, 0) * (XX - X(0)) * (XX - X(1)) * (XX - X(2))
PRINT "X="; XX, "INTERPOLATED VALUE="; FX, "ACTUAL VALUE="; FNA(XX)
END
```

Output

```
X( 0 )= 8   F(0, 0 )= 2.079442          X( 1 )= 8.5   F(0, 1 )= 2.140066
X( 2 )= 9   F(0, 2 )= 2.197225          X( 3 )= 9.5   F(0, 3 )= 2.251292

F( 1 , 0 )= .1212492        F( 1 , 1 )= .1143169        F( 1 , 2 )= .1081343

F( 2 , 0 )=-6.932259E-03    F( 2 , 1 )=-6.182671E-03
F( 3 , 0 )= 4.997253E-04

X= 9.2        INTERPOLATED VALUE= 2.219201              ACTUAL VALUE= 2.219203
```

Netwon's Forward Difference Formula for Regularly Spaced x_j's

If x_j's are regularly spaced such as x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$, . . ., $x_n = x_0 + nh$ and the forward difference of f at x_j are defined as

First forward difference	$\Delta f_j = f_{j+1} - f_j$
Second forward difference	$\Delta^2 f_j = \Delta f_{j+1} - \Delta f_j$
⋮	
kth forward difference	$\Delta^k f_j = \Delta^{k-1} f_{j+1} - \Delta^{k-1} f_j$

then it can be proved by induction that

$$f[x_0, ..., x_k] = \frac{1}{k!h^k}\Delta^k f_0$$

With this, the Netwton's Divided Difference formula can be written as

$$f(x) = \sum_{s=0}^{n} \binom{r}{s} \Delta^s f_0$$

where $r = (x - x_0)/h$

$$\binom{r}{s} = \frac{r(r-1)(r-2)...(r-s+1)}{s!} \qquad ;(s > 0, \text{ integer})$$

with $\binom{r}{0} = 1$

Writing explicitly, the formula is

$$f(x) = f_0 + r\,\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 + \cdots + \frac{r(r-1)\cdots(r-n+1)}{n!}\Delta^n f_0$$

Diagrammatically, the values of $\Delta^n f_j$ may be written as follows.

j	x_j	f_j	Δf_j	$\Delta^2 f_j$	$\Delta^3 f_j$
0	x_0	f_0			
			Δf_0		
1	x_1	f_1		$\Delta^2 f_0$	
			Δf_1		$\Delta^3 f_0$
2	x_2	f_2		$\Delta^2 f_1$	
			Δf_2		
3	x_3	f_3			

Newton's Backward Difference Formula for Regularly Spaced x_j's

The backward difference of f at x_j are defined as

First backward difference $\nabla f_j = f_j - f_{j-1}$

Second backward difference $\nabla^2 f_j = \nabla f_j - \nabla f_{j-1}$

⋮

kth backward difference $\nabla^k f_j = \nabla^{k-1} f_j - \nabla^{k-1} f_{j-1}$

Diagrammatically, the values of $\nabla^k f_j$ may be written as follows.

j	x_j	f_j	∇f_j	$\nabla^2 f_j$	$\nabla^3 f_j$
0	x_0	f_0			
			∇f_1		
1	x_1	f_1		$\nabla^2 f_2$	
			∇f_2		$\nabla^3 f_3$
2	x_2	f_2		$\nabla^2 f_3$	
			∇f_3		
3	x_3	f_3			

The Newton's backward difference formula is

$$f_n(x) = \sum_{s=0}^{n} \binom{r+s-1}{s} \nabla^s f_n; \qquad r = (x - x_n)/h$$

Writing it explicitly, we have

$$f_n(x) = f_n + r\,\nabla f_n + \frac{r(r+1)}{2!}\nabla^2 f_n + \cdots + \frac{r(r+1)\cdots(r+n-1)}{n!}\nabla^n f_n$$

Illustration Determine the interpolated values of sin x at $x = 0.52$ and $x = 0.67$ radian. Given are the values at 0.5, 0.55, 0.6, 0.65 and 0.7.

Program

```
REM NEWTON FORWARD/BACKWARD DIFFERENCE INTERPOLATION
CLS : DIM X(20), F(20, 20)
'DEF FNA (X) = LOG(X) / LOG(10)
DEF FNA (X) = SIN(X)
READ N: DATA 4
READ XI, H: DATA .5,.05
REM Give either of the following statements for
REM the forward(F) or backward(B) interpolation
'READ N$, XX: DATA F,.52
READ N$, XX: DATA B,.67
FOR I = 0 TO N
X(I) = XI + I * H: F(0, I) = FNA(X(I))
PRINT "I="; I; "  X("; I; ")="; X(I), "F(0,"; I; ")="; F(0, I)
NEXT I': PRINT
FOR I = 0 TO N
IF H < 0 THEN 2
IF XX <= X(I) THEN 4
GOTO 3
2 IF XX >= X(I) THEN 4
3 NEXT I
4 IF N$ = "F" THEN M = I - 1 ELSE M = I
PRINT : PRINT "No. of data point used ="; M: PRINT
FOR I = 1 TO N: FOR J = 0 TO N - I
F(I, J) = F(I - 1, J + 1) - F(I - 1, J)
NEXT J: NEXT I
'FOR J = 0 TO N - 1: FOR I = 1 TO N - J
'PRINT "F("; I; J; ")="; F(I, J), : NEXT I: PRINT : NEXT J
R = (XX - X(M)) / H
```

```
'PRINT : PRINT "Details of interpolation"
IF N$ = "F" THEN J1 = N: J2 = -1 ELSE J1 = M: J2 = 1
FOR S = 0 TO J1
IF N$ = "F" THEN J3 = M ELSE J3 = M - S
FACT = 1: BIN = 1
IF S = 0 THEN 5
FOR I = 0 TO S - 1: BIN = BIN * (R + J2 * I): NEXT I
FOR I = 1 TO S: FACT = FACT * I: NEXT I
5 FX = FX + (BIN / FACT) * F(S, J3)
'PRINT "+("; BIN; "/"; FACT; ")("; F(S, J3); ")";
'A$ = INPUT$(1)
NEXT S
'PRINT "="; FX: PRINT
PRINT "X ="; XX; "   Interpolated value ="; FX;
PRINT "   Actual value ="; FNA(XX)
END
```

For Forward Interpolation

```
I= 0   X( 0 )= .5          F(0, 0 )= .4794255
I= 1   X( 1 )= .55         F(0, 1 )= .5226873
I= 2   X( 2 )= .6          F(0, 2 )= .5646425
I= 3   X( 3 )= .65         F(0, 3 )= .6051864
I= 4   X( 4 )= .7          F(0, 4 )= .6442177

No. of data point used = 0

X = .52   Interpolated value = .4968801   Actual value = .4968801
```

For Backward Interpolation

Output

```
I= 0   X( 0 )= .5          F(0, 0 )= .4794255
I= 1   X( 1 )= .55         F(0, 1 )= .5226873
I= 2   X( 2 )= .6          F(0, 2 )= .5646425
I= 3   X( 3 )= .65         F(0, 3 )= .6051864
I= 4   X( 4 )= .7          F(0, 4 )= .6442177

No. of data point used = 4

X = .67   Interpolated value = .620986   Actual value = .620986
```

Centre Difference Interpolation Formulae

Newton's forward and backward interpolation formulae are useful to carry out interpolation near the beginning and end of tabulated values, respectively. If the interpolation is required near the centre of tabulated value, the following scheme is adopted.

X	Y	Δ	Δ^2	Δ^3	Δ^4	Δ^5	Δ^6
X_1	Y_1						
⋮	⋮	ΔY_{K-3}					
X_{K-2}	Y_{K-2}		$\Delta^2 Y_{K-3}$				
		ΔY_{K-2}		$\Delta^3 Y_{K-3}$			
X_{K-1}	Y_{K-1}		$\Delta^2 Y_{K-2}$		$\Delta^4 Y_{K-3}$		
		(B) ΔY_{K-1}	(B)	$\Delta^3 Y_{K-2}$ (B)	(B)	$\Delta^5 Y_{K-3}$ (B)	
X_K	Y_K		$\Delta^2 Y_{K-1}$		$\Delta^4 Y_{K-2}$		$\Delta^6 Y_{K-2}$
		(F) ΔY_K (F)	(F)	$\Delta^3 Y_{K-1}$ (F)	(F)	$\Delta^5 Y_{K-2}$ (F)	
X_{K+1}	Y_{K+1}		$\Delta^2 Y_K$		$\Delta^4 Y_{K-1}$		
		ΔY_{K+1}		$\Delta^3 Y_K$			
X_{K+2}	Y_{K+2}		$\Delta^2 Y_{K+1}$				
⋮	⋮	ΔY_{K+2}					

The above scheme connected through dashed or dotted lines is known as central difference interpolation scheme. Based on this scheme, the following interpolation formula are used.

1. Gauss Forward Formula

The formula is

$$Y_p = Y_K + G_1\,\Delta Y_K + G_2\,\Delta^2 Y_{K-1} + G_3\,\Delta^3 Y_{K-1} + G_4\,\Delta^4 Y_{K-2} \tag{1}$$

where Y_p is the value at X_p. The value of p is given by $p = (X - X_K)/h$.

The unknown in the above expressions are G_1, G_2, ... There are determined as follows.

By definition[†]

$$Y_p = E^p\,Y_K = (1 + \Delta)^p\,Y_K$$

$$= Y_K + p\,\Delta Y_K + \frac{p(p-1)}{2!}\,\Delta^2 Y_K + \frac{p(p-1)(p-2)}{3!}\,\Delta^3 Y_K + \cdots \tag{2}$$

The operation $\Delta^2 Y_{K-1}$ can be expressed in terms of operations on Y_K as shown in the following.

[†]The shift operator is defined by the equation $EY_r = Y_{r+1}$.

We have $E^2 Y_r = E(EY_r) = E(Y_{r+1}) = Y_{r+2}$ and so on. By definition, $\Delta Y_0 = Y_1 - Y_0 = EY_0 - Y_0 = (E - 1)Y_0$ Thus, the identity between operators Δ and E is $\Delta \equiv E - 1$ or $E = 1 + \Delta$

$$\Delta^2 Y_{K-1} = \Delta^2(E^{-1}\ Y_K) = \Delta^2\ (1 + \Delta)^{-1} Y_K$$
$$= \Delta^2\ (1 - \Delta + \Delta^2 - \Delta^3 + \cdots)\ Y_K$$
$$= \Delta^2 Y_K - \Delta^3 Y_K + \Delta^4 Y_K - \Delta^5 Y_K + \cdots \quad (3)$$

Similarly, $$\Delta^3 Y_{K-1} = \Delta^3 Y_K - \Delta^4 Y_K + \Delta^5 Y_K - \Delta^6 Y_K + \cdots \quad (4)$$

Now $$\Delta^4 Y_{K-2} = \Delta^4\ (E^{-2}\ Y_K)$$
$$= \Delta^4\ (1 + \Delta)^{-2}\ Y_K$$
$$= \Delta^4\ (1 - 2\Delta + 3\Delta^2 - 4\Delta^3 + \cdots)\ Y_K$$
$$= \Delta^4\ Y_K - 2\ \Delta^5\ Y_K + 3\Delta^6\ Y_K - 4\Delta^7\ Y_K + \cdots \quad (5)$$

Substituting Eqs (2), (3), (4) and (5) in Eq. (1), we get

$$Y_K + p\ \Delta Y_K + \frac{p(p-1)}{2!}\Delta^2 Y_K + \frac{p(p-1)(p-2)}{3!}\Delta^3 Y_K + \cdots$$

$$= Y_K + G_1\ \Delta Y_K$$
$$+ G_2\ (\quad \Delta^2 Y_K - \Delta^3 Y_K + \Delta^4 Y_K - \Delta^5 Y_K + \cdots)$$
$$+ G_3\ (\quad \Delta^3 Y_K - \Delta^4 Y_K + \Delta^5 Y_K - \Delta^6 Y_K + \cdots)$$
$$+ G_4\ (\quad \Delta^4 Y_K - 2\Delta^5 Y_K + 3\Delta^6 Y_K - \cdots) + \cdots$$

Equating the coefficients of ΔY_K, $\Delta^2 Y_K$, ..., etc. on both sides of the above expression, we get

$$G_1 = p \qquad G_2 = \frac{p(p-1)}{2!}$$

$$G_3 = \frac{(p+1)(p)(p-1)}{3!} \qquad G_4 = \frac{(p+1)(p)(p-1)(p-2)}{4!}$$

2. Gauss Backward Formula

The formula is

$$Y_p = Y_K + G_1'\ \Delta Y_{K-1} + G_2'\ \Delta^2 Y_{K-1} + G_3'\ \Delta^3 Y_{K-2} + G_4'\ \Delta^4 Y_{K-2} + \cdots$$

As derived in Gauss forward formula, the expressions of G' s are

$$G_1' = p \qquad G_2' = \frac{p(p+1)}{2!}$$

$$G_3' = \frac{(p+1)(p)(p-1)}{3!} \qquad G_4' = \frac{(p+2)(p+1)(p)(p-1)}{4!}$$

3. Stirling's Formula

The Stirling's formula involves the mean of Gauss forward and backward formula.

4. Bessel's Formula

The Bessel's formula involves the following scheme.

$\vdots$					
X_K	$\begin{pmatrix} Y_K \\ Y_{K+1} \end{pmatrix}$	ΔY_K	$\begin{pmatrix} \Delta^2 Y_{K-1} \\ \Delta^2 Y_K \end{pmatrix}$	$\Delta^3 Y_{K-1}$	$\begin{pmatrix} \Delta^4 Y_{K-2} \\ \Delta^4 Y_{K-1} \end{pmatrix}$	$\Delta^5 Y_{K-2}$
X_{K+1}						
$\vdots$						

where the brackets involves taking the average of indicated values.

The Bessel's formula is

$$Y_p = \frac{Y_K + Y_{K+1}}{2} + B_1 \Delta Y_K + B_2 \frac{\Delta^2 Y_{K-1} + \Delta^2 Y_K}{2} + B_3 \Delta^3 Y_{K-1} + B_4 \frac{\Delta^4 Y_{K-2} + \Delta^4 Y_{K-1}}{2} + \cdots$$

$$= Y_K + \left(B_1 + \frac{1}{2} \right) \Delta Y_K + B_2 \frac{\Delta^2 Y_{K-1} + \Delta^2 Y_K}{2} + B_3 \Delta^3 Y_{K-1} + B_4 \frac{\Delta^4 Y_{K-2} + \Delta^4 Y_{K-1}}{2} + \cdots$$

Following the above method to determine constants, the expressions of B's are

$$B_1 + \frac{1}{2} = p \qquad B_2 = \frac{p(p-1)}{2!}$$

$$B_3 = \frac{p(p-1)(p-1/2)}{3!} \qquad B_4 = \frac{(p+1)(p)(p-1)(p-2)}{4!}$$

5. Everett's Formula

The Everett's formula uses the following scheme.

X_K	Y_K	—	$\Delta^2 Y_{K-1}$	—	$\Delta^4 Y_{K-2}$	—	$\Delta^6 Y_{K-3}$
X_{K+1}	Y_{K+1}	—	$\Delta^2 Y_K$	—	$\Delta^4 Y_{K-1}$	—	$\Delta^6 Y_{K-2}$

The Everett formula is

$$Y_p = E_0 Y_K + E_2 \Delta^2 Y_{K-1} + E_4 \Delta^4 Y_{K-2} + E_6 \Delta^6 Y_{K-3} + \cdots + F_0 Y_{K+1} + F_2 \Delta^2 Y_K + F_4 \Delta^4 Y_{K-1} + \cdots$$

where the constants are given by the expressions

$$E_0 = q \ (= 1 - p) \qquad F_0 = p$$

$$E_2 = \frac{q(q^2 - 1^2)}{3!} \qquad F_2 = \frac{p(p^2 - 1^2)}{3!}$$

$$E_4 = \frac{q(q^2 - 1^2)(q^2 - 2^2)}{5!} \qquad F_4 = \frac{p(p^2 - 1^2)(p^2 - 2^2)}{5!}$$

Illustration Interpolate the value of $e^{1.17}$. Given the values of e^x where x = 1, 1.05, 1.1, 1.15, 1.2, 1.25 and 1.3.

Program

```
REM Gauss,Stirling,Bessel & Everett Interpolation Formulae
CLS
DEF FNA (X) = EXP(X)
READ XI, H, XG: DATA 1.0, .05, 1.17
K = -1: N = 7
FOR I = 1 TO 10
K = K + 1
IF H < 0 THEN 2
IF XG < (XI + K * H) THEN 5
GOTO 4
2 IF XG > (XI + K * H) THEN 5
4 NEXT I
5 PRINT " X                    Y"
X = XI
FOR I = 1 TO N
A(I) = FNA(X)
PRINT X, : PRINT USING "#.######"; A(I)
X = X + H
NEXT I
A$ = INPUT$(1): PRINT : PRINT " DEL";
FOR I = 1 TO N - 1
B(I) = A(I + 1) - A(I)
PRINT USING "     #.######"; B(I);
NEXT I
A$ = INPUT$(1): PRINT : PRINT "DEL2";
FOR I = 1 TO N - 2
C(I) = B(I + 1) - B(I)
PRINT USING "     #.######"; C(I);
NEXT I
A$ = INPUT$(1): PRINT : PRINT "DEL3";
FOR I = 1 TO N - 3
```

```
D(I) = C(I + 1) - C(I)
PRINT USING "     #.######"; D(I);
NEXT I
A$ = INPUT$(1): PRINT : PRINT "DEL4";
FOR I = 1 TO N - 4
E(I) = D(I + 1) - D(I)
PRINT USING "     #.######"; D(I);
NEXT I
A$ = INPUT$(1): PRINT : PRINT "DEL5";
FOR I = 1 TO N - 5
F(I) = E(I + 1) - E(I)
PRINT USING "     #.######"; F(I);
NEXT I
A$ = INPUT$(1): PRINT : PRINT
M = K - 1: PRINT "M="; M: A$ = INPUT$(1)
p = (XG - (XI + M * H)) / H
T1 = p * (p - 1): T2 = (p + 1) * T1: T3 = T2 * (p - 2)
T4 = p * (p + 1): T5 = T4 * (p - 1): T6 = (p + 2) * T5
GF = A(K) + p * B(K) + (T1 / 2) * C(K - 1)
GF = GF + (T2 / 6) * D(K - 1)
GF = GF + (T3 / 24) * E(K - 2)
GB = A(K) + p * B(K - 1) + (T4 / 2) * C(K - 1)
GB = GB + (T5 / 6) * D(K - 2)
GB = GB + (T6 / 24) * E(K - 2)
BS = A(K) + p * B(K) + (T1 / 2) * (C(K - 1) + C(K)) / 2
BS = BS + (T1 * (p - .5) / 6) * D(K - 1)
BS = BS + (T3 / 24) * (E(K - 2) + E(K - 1)) / 2
q = 1 - p: T7 = q * (q ^ 2 - 1): T8 = p * (p ^ 2 - 1)
EV = q * A(K) + (T7 / 6) * C(K - 1)
EV = EV + (T7 * (q ^ 2 - 4) / 120) * E(K - 2)
EV = EV + p * A(K + 1) + (T8 / 6) * C(K)
EV = EV + (T8 * (p ^ 2 - 4) / 120) * E(K - 1)
PRINT "Value of"; XG; "by Gauss forward formula="; GF
PRINT "Value of"; XG; "by Gauss backward formula="; GB
PRINT "Value of"; XG; "by Stirling formula="; (GF + GB) / 2
PRINT "Value of"; XG; "by Bessel formula="; BS
PRINT "Value of"; XG; "by Everett formula="; EV
PRINT "ACTUAL VALUE OF"; XG; "="; FNA(XG)
END
```

Output

```
 X              Y
 1            2.718282
 1.05         2.857651
 1.1          3.004166
 1.15         3.158192
 1.2          3.320116
 1.25         3.490342
 1.3          3.669296

 DEL    0.139369    0.146515    0.154027    0.161924    0.170226    0.178953
DEL2    0.007145    0.007512    0.007897    0.008302    0.008728
DEL3    0.000367    0.000385    0.000405    0.000426
DEL4    0.000367    0.000385    0.000405
DEL5    0.000001    0.000001

M= 3
Value of 1.17 by Gauss forward formula= 3.221992
Value of 1.17 by Gauss backward formula= 3.221992
Value of 1.17 by Stirling formula= 3.221992
Value of 1.17 by Bessel formula= 3.221992
Value of 1.17 by Everett formula= 3.221992
ACTUAL VALUE OF 1.17 = 3.221992
```

Lagrange's Interpolation Formula

In the Lagrange's interpolation formula, a Lagrange polynomial of degree n for a function $y(x_i)$ is devised so as to satisfy the requirement

$$L_n(x_i) = y(x_i); \qquad i = 0, 1, 2, ..., n$$

Such a polynomial is expressed as

$$L_n = \sum_{i=0}^{n} \left[\prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) \right] y_i \tag{1}$$

For $n = 1$, the polynomial passes through two points (x_0, y_0) and (x_1, y_1) and is given by

$$L_1 = \left(\frac{x - x_1}{x_0 - x_1} \right) y_0 + \left(\frac{x - x_0}{x_1 - x_0} \right) y_1$$

Obviously, for $x = x_0$, $L_1(x_0) = y_0$ and for $x = x_1$, $L_1(x_1) = y_1$

For $n = 2$, the polynomial passes through three points (x_0, y_0), (x_1, y_1) and (x_2, y_2) and is given by

$$L_2 = \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right) y_0 + \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right) y_1 + \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right) y_2$$

Obviously, for $x = x_0$, $L_2(x_0) = y_0$; $x = x_1$, $L_2(x_1) = y_1$ and $x = x_2$, $L_2(x_2) = y_2$.

Thus, Eq. (1) is the required interpolation formula and can be used for equally or unequally spaced x_i.

Illustration To estimate the value of ln 302. Given are the values at 300, 304, 305 and 307.

Program

```
CLS
DIM X(20), Y(20)
DEF FNA (X) = LOG(X)
READ N: DATA 3
PRINT "   X                    Y"
FOR I = 0 TO N
READ X(I): Y(I) = FNA(X(I))
PRINT X(I), Y(I): A$ = INPUT$(1)
NEXT I
DATA 300,304,305,307
READ XG: DATA 302
SUM = 0
FOR I = 0 TO N
TERM = 1
FOR J = 0 TO N
IF I = J THEN 5
TERM = TERM * (XG - X(J)) / (X(I) - X(J))
5 NEXT J
SUM = SUM + TERM * Y(I)
NEXT I
PRINT : PRINT "ESTIMATED VALUE OF "; XG; "= "; SUM
PRINT "ACTUAL VALUE OF "; XG; "= "; FNA(XG)
END
```

Output

```
 X                Y
300           5.703783
304           5.717028
305           5.720312
307           5.726848

ESTIMATED VALUE OF  302 =  5.710428
ACTUAL VALUE OF  302 =  5.710427
```

Hermite's Interpolation Formula

The Hermite's interpolation formula requires both the functional value and its first derivative at each point of interpolation. The expression yields

$$H(x_i) = y_i \quad \text{and} \quad H'(x_i) = y_i'; \qquad i = 0, 1, 2, ..., n$$

The expression of Hermite polynomial is

$$H(x) = \sum_{i=0}^{n} u_i(x)\, y_i + \sum_{i=0}^{n} v_i(x)\, y_i' \tag{1}$$

This satisfies the conditions

$$u_i(x_j) = \delta_{ij} \quad \text{and} \quad u_i'(x) = 0 \quad \text{for all } i \tag{2a}$$

$$v_i(x) = 0 \quad \text{for all } i \quad \text{and} \quad v_i'(x_j) = \delta_{ij} \tag{2b}$$

where δ_{ij} is Kronecker delta; it is equal to 1 for $i = j$ and is equal to zero for $i \neq j$.

The expressions of $u_i(x)$ and $v_i(x)$ are

$$u_i(x) = (a_i x + b_i)\, [l_i(x)]^2 \tag{3a}$$

$$v_i(x) = (c_i x + d_i)\, [l_i(x)]^2 \tag{3b}$$

where $$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) \tag{4}$$

Obviously, $l_i(x_i) = 1$ and $l_i(x_j) = 0$. (5)

For $x = x_i$, Eqs (3a) and (3b) become

$$u_i(x_i) = (a_i x_i + b_i)\, [l_i(x_i)]^2 \tag{6a}$$

$$v_i(x_i) = (c_i x_i + d_i)\, [l_i(x_i)]^2 \tag{6b}$$

which according to the conditions as given by Eqs (2a), (2b) and (5) become

$$a_i x_i + b_i = 1 \quad \text{and} \quad c_i x_i + d_i = 0 \tag{7a}$$

The first derivatives of Eqs (6a) and (6b) are

$$u_i'(x_i) = a_i\, [l_i(x_i)]^2 + (a_i x_i + b_i)\, [2\{l_i(x_i)\}\ \{l_i'(x_i)\}]$$

$$v_i'(x_i) = c_i\, [l_i(x_i)]^2 + (c_i x_i + d_i)\, [2\{l_i(x_i)\}\ \{l_i'(x_i)\}]$$

which according to Eqs (2a), (2b), (5) and (7a) become

$$a_i + 2l_i'(x_i) = 0 \quad \text{and} \quad c_i = 1 \tag{7b}$$

From Eqs (7a) and (7b), we get

$$a_i = -\,2\, l_i'(x_i) \qquad b_i = 1 - a_i x_i = 1 + 2x_i\, [l_i'(x_i)]$$

$$c_i = 1 \qquad d_i = -\,x_i$$

With these, Eqs (3a) and (3b) become

$$u_i(x) = [\{-\,2l_i'(x_i)\}x + 1 + 2\, x_i\, \{l_i'(x_i)\}]\, [l_i(x)]^2$$

$$= [1 - 2\, (x - x_i)\, \{l_i'(x_i)\}]\, [l_i(x)]^2 \tag{8a}$$

$$v_i(x) = [(1)x + (-x_i)]\,[l_i(x)]^2$$
$$= (x - x_i)\,[l_i(x)]^2 \qquad (8b)$$

Finally, Hermite polynomial (Eq.1) becomes

$$H(x) = \sum_{i=0}^{n} [1 - 2(x - x_i)\{l_i'(x_i)\}][l_i(x)]^2\, y_i + \sum_{i=0}^{n} (x - x_i)[l_i(x)]^2\, y_i' \qquad (9)$$

Illustration To estimate the value of ln 303. Given are the values at 300, 304, 305, 307 and 308 .

Program

```
CLS
DIM X(20), Y(20), YD(20)
DEF FNA (X) = LOG(X): DEF FND (X) = 1 / X
READ N: DATA 5
PRINT "   X                    Y               YD"
FOR I = 1 TO N
READ X(I): Y(I) = FNA(X(I)): YD(I) = FND(X(I))
PRINT X(I), Y(I), YD(I): A$ = INPUT$(1)
NEXT I
DATA 300,304,305,307,308
READ XG
DATA 303
SUM = 0: SUMH = 0: TERM1 = 0: TERM2 = 0
FOR I = 1 TO N
NUM = 1: DEN = 1
FOR J = 1 TO N
IF J = I THEN 5
NUM = NUM * (XG - X(J)): DEN = DEN * (X(I) - X(J))
5 NEXT J
TERM = NUM / DEN: SUM = SUM + TERM * Y(I)
SUMD = 0
FOR K = 1 TO N
IF K = I THEN 10
NUM = 1
FOR J = 1 TO N
IF J = I OR J = K THEN 8
NUM = NUM * (X(I) - X(J))
8 NEXT J
SUMD = SUMD + NUM / DEN
10 NEXT K
```

```
T1 = (1 - 2 * (XG - X(I)) * SUMD) * TERM ^ 2 * Y(I)
TERM1 = TERM1 + T1
T2 = (XG - X(I)) * TERM ^ 2 * YD(I)
TERM2 = TERM2 + T2
NEXT I
SUMH = TERM1 + TERM2
PRINT : PRINT "Estimated value of "; XG; "by Lagrange formula = "; SUM
PRINT : PRINT "Estimated value of "; XG; "by Hermite formula = "; SUMH
PRINT : PRINT "Actual value of "; XG; "= "; FNA(XG)
END
```

Output

```
 X              Y            YD
300          5.703783     3.333333E-03
304          5.717028     3.289474E-03
305          5.720312     3.278689E-03
307          5.726848     3.257329E-03
308          5.7301       3.246753E-03

Estimated value of  303 by Lagrange formula =  5.713734

Estimated value of  303 by Hermite formula =  5.713731

Actual value of  303 =  5.713733
```

Spline Interpolation

The given interpolation of a value between the given interval is carried out by dividing the entire interval into various subintervals. For each subinterval, a polynomial of low degree is devised to predict interpolation in that range. The devised polynomial may be linear, quadratic, cubic or of higher order. However, a cubic polynomial is commonly used in engineering applications. We construct linear, quadratic and cubic polynomials to highlight their mode of applications.

Linear Splines Let the given data point be represented as

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

For each subinterval, say $(x_0, y_0) - (x_1, y_1)$, $(x_1, y_1) - (x_2, y_2)$, ..., a linear polynomial is given by the expression

$$s_i(x) = y_{i-1} + m_i (x - x_{i-1}) \tag{1}$$

where $$m_i = (y_i - y_{i-1})/(x_i - x_{i-1}) \tag{2}$$

For example, for the data points (1, 1), (2, 8), (3, 27), we have

Subinterval (1, 1) – (2, 8);

$$s_1(x) = 1 + \frac{8-1}{2-1}\ (x-1) = 7x - 6$$

Subinterval (2, 8) – (3, 27);

$$s_2(x) = 8 + \frac{27-8}{3-2}\ (x-2) = 19x - 30$$

The polynmial $s_1(x)$ is used to predict the interpolation for a value of x between the first subinterval (say, $x = 1.5$) and s_2 is used to predict the value for x between second subinterval (say, $x = 2.5$).

Linear splines are continuous but their slopes are discontinuous.

Quadratic Spline Let the spline in the interval $[x_{i-1}, x_i]$ be represented as $s_i(x)$.

Let $s_i(x)$ and its first derivative $s_i'(x)$ be continuous in the entire range $[x_0, x_n]$.

Since $s_i(x)$ is a quadratic, it follows that $s_i'(x)$ is a linear function and let it be written as

$$s_i'(x) = \frac{1}{h_i}\ [(x_i - x)\ m_{i-1} + (x - x_{i-1})\ m_i] \tag{3}$$

where $h_i = x_i - x_{i-1}$. From the above expression, it follows that

$$m_i = s_i'(x_i) \quad \text{and} \quad m_{i-1} = s_i'(x_{i-1}) \tag{4}$$

Integrating $s_i'(x)$ with respect to x, we get

$$s_i(x) = \frac{1}{h_i}\left[-\frac{(x_i - x)^2}{2}\, m_{i-1} + \frac{(x - x_{i-1})^2}{2}\, m_i\right] + c_i \tag{5}$$

where c_i is constant of integration. Its expression may be derived by putting $x = x_{i-1}$ where $s_i\,(x_{i-1}) = y_{i-1}$. Hence

$$y_{i-1} = \frac{1}{h_i}\left[-\frac{(x_i - x_{i-1})^2}{2}\, m_{i-1} + \frac{(x_{i-1} - x_{i-1})^2}{2}\, m_i\right] + c_i \tag{6}$$

which gives $$c_i = y_{i-1} + \frac{h_i}{2}\ m_{i-1} \tag{7}$$

With this, the expression of spline $s_i(x)$ as given by Eq. (5) becomes

$$s_i(x) = \frac{1}{h_i}\left[-\frac{(x_i - x)^2}{2}\, m_{i-1} + \frac{(x - x_{i-1})^2}{2}\, m_i\right] + y_{i-1} + \frac{h_i}{2}\, m_{i-1} \tag{8}$$

In the above expression, m_{i-1} and m_i are unknown, these are determined based on the fact that the function $s_i(x)$ is continuous at $x = x_i$, for which we have

$$s_i(x_i -) = s_{i+1}(x_i+) \tag{9}$$

For $x_i- = \lim_{h' \to 0} (x_i - h')$, Eq. (8) yields

$$s_i(x_i-) = \frac{1}{h_i}\left[0 + \frac{h_i^2}{2} m_i\right] + y_{i-1} + \frac{h_i}{2} m_{i-1}$$

$$= \frac{h_i}{2}[m_{i-1} + m_i] + y_{i-1} \tag{10}$$

For $x_i+ = \lim_{h' \to 0} (x_i + h')$, we first write the spline $s_{i+1}(x)$ from Eq. (8) by replacing the subscript i by $i + 1$. Thus

$$s_{i+1}(x) = \frac{1}{h_{i+1}}\left[-\frac{(x_{i+1} - x)^2}{2} m_i + \frac{(x - x_i)^2}{2} m_{i+1}\right] + y_i + \frac{h_{i+1}}{2} m_i$$

which gives $$s_{i+1}(x_i+) = \frac{1}{h_{i+1}}\left[-\frac{h_{i+1}^2}{2} m_i + 0\right] + y_i + \frac{h_{i+1}}{2} m_i = y_i \tag{11}$$

Equating Eqs (10) and (11), we get

$$\frac{h_i}{2}[m_{i-1} + m_i] + y_{i-1} = y_i$$

or $$m_{i-1} + m_i = \frac{2}{h_i}(y_i - y_{i-1}); \qquad i = 1, 2, ..., n \tag{12}$$

Equation (12) is a recurrence relation. This given n equation for $n + 1$ unknowns (which are $m_0, m_1, ..., m_n$). The one extra unknown is fixed for selecting a natural spline for which $s_1''(x_1) = 0$. Differentiating Eq. (3), we get

$$s_i''(x) = \frac{1}{h_i}[-m_{i-1} + m_i]$$

Hence, $s_1''(x_1) = 0$ gives

$$0 = \frac{1}{h_i}(-m_0 + m_1)$$

or $$m_0 = m_1 \tag{13}$$

Equation (12) and (13) can be used to determine various m_i's which can be used in Eq. (8) for the required interpolation.

For example, for the data points (1, 1), (2, 8), (3, 27), we have

Subinterval **(1, 1) – (2, 8):** $h_1 = x_1 - x_0 = 2 - 1 = 1$

$$s_1(x) = \frac{1}{h_i}\left[-\frac{(x_1 - x)^2}{2} m_0 + \frac{(x - x_0)^2}{2} m_1\right] + y_0 + \frac{h_1}{2} m_0$$

$$= \left[-\frac{(2-x)^2}{2} m_0 + \frac{(x-1)^2}{2} m_1 \right] + 1 + \frac{1}{2} m_0$$

Subinterval **(2, 8) – (3, 27):** $h_2 = x_2 - x_1 = 3 - 2 = 1$

$$s_2(x) = \frac{1}{h_2}\left[-\frac{(x_2 - x)^2}{2} m_1 + \frac{(x - x_1)^2}{2} m_2 \right] + y_1 + \frac{h_2}{2} m_1$$

$$= \left[-\frac{(3-x)^2}{2} m_1 + \frac{(x-2)^2}{2} m_2 \right] + 8 + \frac{1}{2} m_1$$

The expressions of m_i s in the above expressions as given by Eq. (12) are

$$m_0 + m_1 = \frac{2}{h_1} (y_1 - y_0) = 2\ (8 - 1) = 14$$

$$m_1 + m_2 = \frac{2}{h_2} (y_2 - y_1) = 2\ (27 - 8) = 38$$

Since $m_0 = m_1$, (Eq. 13), we get

$$m_0 = m_1 = 7 \quad \text{and} \quad m_2 = 31$$

Hence, the find expressions are

$$s_1(x) = \left[-\frac{(2-x)^2}{2}(7) + \frac{(x-1)^2}{2}(7) \right] + 1 + \frac{7}{2}$$

$$s_2(x) = \left[-\frac{(3-x)^2}{2}(7) + \frac{(x-2)^2}{2}(31) \right] + 8 + \frac{7}{2}$$

The polynomial $s_1(x)$ is used to predict the interpolation for a value of x between the first interval (say, $x = 1.5$) and $s_2(x)$ is used to predict the value of x between the second subinterval (say, $x = 2.5$).

Cubic Splines For the cubic splines, the second derivative of $s_i(x)$ will be linear and continuous for which, we write

$$s_i''(x) = \frac{1}{h_i} [(x_i - x)\ M_{i-1} + (x - x_{i-1})\ M_i] \tag{14}$$

where $h_i = x_i - x_{i-1}$. Obviously, $s_i''(x_i) = M_i$

Integration twice of Eq. (14) with respect to x gives

$$s_i(x) = \frac{1}{h_i}\left[\frac{(x_i - x)^3}{6} M_{i-1} + \frac{(x - x_{i-1})^3}{6} M_i \right] + c_i\ (x_i - x) + d_i\ (x - x_{i-1}) \tag{15}$$

The constants of integration c_i and d_i are determined from the fact that $s_i(x_i) = y_i$ and $s_i(x_{i-1}) = y_{i-1}$. For $x = x_i$, we have

$$y_i = \frac{1}{h_i}\left[0 + \frac{h_i^3}{6} M_i\right] + 0 + d_i h_i \quad \Rightarrow \quad d_i = \frac{1}{h_i}\left(y_i - \frac{h_i^2}{6} M_i\right)$$

For $x = x_{i-1}$, we have

$$y_{i-1} = \frac{1}{h_i}\left(\frac{h_i^3}{6} M_{i-1} + 0\right) + c_i h_i + 0 \quad \Rightarrow \quad c_i = \frac{1}{h_i}\left(y_{i-1} - \frac{h_i^2}{6} M_{i-1}\right)$$

With these, Eq. (15) becomes

$$s_i(x) = \frac{1}{h_i}\left[\frac{(x_i - x)^3}{6} M_{i-1} + \frac{(x - x_{i-1})^3}{6} M_i + \left(y_{i-1} - \frac{h_i^2}{6} M_{i-1}\right)(x_i - x) + \left(y_i - \frac{h_i^2}{6} M_i\right)(x - x_{i-1})\right] \tag{16}$$

In Eq. (16), the second derivatives, M_i and M_{i-1} are not yet defined. To determine their expressions, the continuity of the first derivatives may be used, for which we write

$$s_i'(x_i-) = s_{i+1}'(x_i+) \tag{17}$$

The first derivative of Eq. (16) is

$$s_i'(x) = \frac{1}{h_i}\left[-\frac{(x_i - x)^2}{2} M_{i-1} + \frac{(x - x_{i-1})^2}{2} M_i - \left(y_{i-1} - \frac{h_i^2}{6} M_{i-1}\right) + \left(y_i - \frac{h_i^2}{6} M_i\right)\right] \tag{18}$$

For $s_i'(x_i-)$, we get

$$s_i'(x_i-) = \frac{1}{h_i}\left[\frac{h_i^2}{2} M_i - \left(y_{i-1} - \frac{h_i^2}{6} M_{i-1}\right) + \left(y_i - \frac{h_i^2}{6} M_i\right)\right]$$

$$= \frac{1}{h_i}(y_i - y_{i-1}) + \frac{h_i}{6} M_{i-1} + \frac{h_i}{3} M_i \tag{19}$$

For $s_{i+1}'(x_i+)$, first we write the expression of $s_{i+1}'(x)$. Replacing i by $i + 1$ in Eq. (18), we get

$$s_{i+1}'(x_i) = \frac{1}{h_{i+1}}\left[-\frac{(x_{i+1} - x)^2}{2} M_i + \frac{(x - x_i)^2}{2} M_{i+1} - \left(y_i - \frac{h_{i+1}^2}{6} M_i\right) + \left(y_{i+1} - \frac{h_{i+1}^2}{6} M_{i+1}\right)\right]$$

Hence, $s_{i+1}'(x_i+)$ is given by

$$s_{i+1}'(x_i+) = \frac{1}{h_{i+1}}\left[-\frac{h_{i+1}^2}{2} M_i - \left(y_i - \frac{h_{i+1}^2}{6} M_i\right) + \left(y_{i+1} - \frac{h_{i+1}^2}{6} M_{i+1}\right)\right]$$

$$= \frac{1}{h_{i+1}}(y_{i+1} - y_i) - \frac{h_{i+1}}{3} M_i - \frac{h_{i+1}}{6} M_{i+1} \tag{20}$$

Equating Eqs (19) and (20), we get

$$\frac{1}{h_i}(y_i - y_{i-1}) + \frac{h_i}{6} M_{i-1} + \frac{h_i}{3} M_i = \frac{1}{h_{i+1}}(y_{i+1} - y_i) - \frac{h_{i+1}}{3} M_i - \frac{h_{i+1}}{6} M_{i+1}$$

or
$$\frac{h_i}{6} M_{i-1} + \frac{1}{3}(h_i + h_{i+1}) M_i + \frac{h_{i+1}}{6} M_{i+1} = \frac{1}{h_{i+1}}(y_{i+1} - y_i) - \frac{1}{h_i}(y_i - y_{i-1})$$

or
$$h_i M_{i-1} + 2(h_i + h_{i+1}) M_i + h_{i+1} M_{i+1} = \frac{6}{h_{i+1}}(y_{i+1} - y_i) - \frac{6}{h_i}(y_i - y_{i-1}) \tag{21}$$

In Eq. (21), $i = 1, 2, ..., n - 1$. Thus, there are $n - 1$ equations, but there are $n + 1$ unknown M's (viz., M_0, M_1, ..., M_n. For the natural cubic splines, the derivatives M_0 and M_n are set equal to zero. From Eq. (21), the values of M_1 to M_{n-1} are determined. These along with $M_0 = 0$ and $M_n = 0$ are used in Eq. (16) to determine the interpolated value of x.

The values of M_1 to M_{n-1} can be represented as follows.

$$\begin{bmatrix} 2(h_1 + h_2) & h_2 & 0 & 0 & \cdots & 0 & 0 \\ h_2 & 2(h_2 + h_3) & h_3 & 0 & \cdots & 0 & 0 \\ 0 & h_3 & 2(h_3 + h_4) & h_4 & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{6}{h_2}(y_2 - y_1) - \frac{6}{h_1}(y_1 - y_0) \\ \frac{6}{h_3}(y_3 - y_2) - \frac{6}{h_2}(y_2 - y_1) \\ \frac{6}{h_4}(y_4 - y_3) - \frac{6}{h_3}(y_3 - y_1) \\ \vdots \\ \frac{6}{h_n}(y_n - y_{n-1}) - \frac{6}{h_{n-1}}(y_{n-1} - y_{n-2}) \end{bmatrix}$$

Writing it as $\boldsymbol{H}\,\boldsymbol{M} = \boldsymbol{R}$, we get

$$\boldsymbol{M} = \boldsymbol{H}^{-1}\,\boldsymbol{R}$$

Thus, detrmining $\boldsymbol{H}^{-1}$ and right multiplying it with $\boldsymbol{R}$ gives $\boldsymbol{M}$ vector.

For equal intervals of h (i.e. $h_i = h_{i+1} = ... = h$), Eq. (21) takes the form of

$$M_{i-1} + 4M_i + M_{i+1} = \frac{6}{h^2}(y_{i+1} - 2y_i + y_{i-1}); \quad (i = 1, 2, ..., n - 1) \tag{22}$$

In the matrix notation, we have

$$\begin{bmatrix} 4 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 4 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 4 & 1 & \ldots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \ldots & 1 & 4 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ y_4 - 2y_3 + y_2 \\ \vdots \\ y_n - 2y_{n-1} + y_{n-2} \end{bmatrix}$$

i.e. $\boldsymbol{H\,M = R}$

From this, we have $\boldsymbol{M = H^{-1}\,R}$

Illustration To determine dy/dx and d^2y/dx^2 for the function $y = \sin x$ at $x = 15°, 35°, 55°$ and $75°$. Given are the function values for 0° to 180° at the regular interval of 18°.

Program

```
CLS
DEF FNA (X) = SIN(X): DEF FNB (X) = COS(X): DEF FNC (X) = -SIN(X)
'DEF FNA (X) = X ^ 3: DEF FNB (X) = 3 * X ^ 2: DEF FNC (X) = 6 * X
DIM X(30), Y(30), M(30), C(30), A(30, 30), A1(30, 30)
READ N: DATA 10
PRINT STRING$(30, "-")
PRINT " I   Angle    Angle         Y"
PRINT "     Degree   Radian": PRINT STRING$(30, "-")
FOR I = 0 TO N
XD = I * 180 / N: X(I) = I * 3.14159 / N
'X(I) = I
Y(I) = FNA(X(I))
PRINT USING "##"; I; : PRINT USING "   ###"; XD;
PRINT USING "   ###.#####"; X(I); Y(I)
NEXT I
PRINT STRING$(30, "-")
H = X(1) - X(0): PRINT "Value of h="; H: A$ = INPUT$(1)
NM1 = N - 1
FOR I = 1 TO NM1
A(I, I) = 4
C(I) = (6 / H ^ 2) * (Y(I + 1) - 2 * Y(I) + Y(I - 1))
FOR J = I + 1 TO NM1
IF J = I + 1 THEN A(I, J) = 1 ELSE A(I, J) = 0
A(J, I) = A(I, J)
NEXT J
NEXT I
'PRINT : PRINT "M MATRIX"
'PRINT STRING$(7.3 * N, "-")
'FOR I = 1 TO NM1: PRINT USING "      ##"; I; : NEXT I
'PRINT "     RHS": PRINT STRING$(7.3 * N, "-")
FOR I = 1 TO NM1
FOR J = 1 TO NM1
A1(I, J) = A(I, J)
'PRINT USING "   #.###"; A(I, J);
NEXT J
'PRINT USING "   ##.####"; C(I): A$ = INPUT$(1)
NEXT I
```

```
'PRINT STRING$(7.3 * N, "-"): A$ = INPUT$(1)
N2 = 2 * NM1
FOR I = 1 TO NM1
K = I + NM1
FOR J = NM1 + 1 TO N2
IF J = K THEN A(I, J) = 1 ELSE A(I, J) = 0
NEXT J
NEXT I
FOR I = 1 TO NM1 - 1
DUM = A(I, I)
FOR J = 1 TO N2: A(I, J) = A(I, J) / DUM: NEXT J
FOR J = I + 1 TO NM1
DUM = A(J, I) / A(I, I)
FOR K = 1 TO N2: A(J, K) = A(J, K) - A(I, K) * DUM: NEXT K
NEXT J
NEXT I
DUM = A(NM1, NM1)
FOR J = NM1 TO N2: A(NM1, J) = A(NM1, J) / DUM: NEXT J
FOR I = 1 TO NM1 - 1
FOR J = I + 1 TO NM1
DUM = A(I, J)
FOR K = J TO N2: A(I, K) = A(I, K) - DUM * A(J, K): NEXT K
NEXT J
NEXT I
M(0) = 0: M(N) = 0: PRINT : PRINT "M(0)=0",
FOR I = 1 TO NM1
SUM = 0
FOR J = 1 TO NM1: SUM = SUM + A(I, J + NM1) * C(J): NEXT J
M(I) = SUM: PRINT "M("; I; ")="; M(I),
NEXT I
PRINT "M("; N; ")=0": PRINT
A$ = INPUT$(1): PRINT STRING$(78, "-")
PRINT " I  Angle    Angle    Estimated  Actual  ";
PRINT " First         Actual  Second       Actual"
PRINT "     Degree  Radian  Value       Value    ";
PRINT " Derivative  Value    Derivative  Value"
PRINT STRING$(78, "-")
FOR J = 15 TO 90 STEP 20
XG = (J / 180) * 3.14159
'FOR J = .5 TO N
'XG = J
```

```
FOR I = 0 TO N
IF XG <= X(I) GOTO 5
NEXT I
5 S = (X(I) - XG) ^ 3 * M(I - 1) / 6
S = S + (XG - X(I - 1)) ^ 3 * M(I) / 6
S = S + (Y(I - 1) - H ^ 2 * M(I - 1) / 6) * (X(I) - XG)
S = S + (Y(I) - H ^ 2 * M(I) / 6) * (XG - X(I - 1))
S = S / H
SFD = -(X(I) - XG) ^ 2 * M(I - 1) / 2
SFD = SFD + (XG - X(I - 1)) ^ 2 * M(I) / 2
SFD = SFD - (Y(I - 1) - H ^ 2 * M(I - 1) / 6)
SFD = SFD + (Y(I) - H ^ 2 * M(I) / 6)
SFD = SFD / H
SSD = (X(I) - XG) * M(I - 1) + (XG - X(I - 1)) * M(I)
SSD = SSD / H
PRINT I; : PRINT USING " ###"; J;
PRINT USING "   ###.####"; XG; S; FNA(XG); SFD; FNB(XG); SSD; FNC(XG)
NEXT J
PRINT STRING$(78, "-")
END
```

Output

```
------------------------------
 I  Angle   Angle       Y
    Degree  Radian
------------------------------
 0    0    0.00000    0.00000
 1   18    0.31416    0.30902
 2   36    0.62832    0.58778
 3   54    0.94248    0.80902
 4   72    1.25664    0.95106
 5   90    1.57080    1.00000
 6  108    1.88495    0.95106
 7  126    2.19911    0.80902
 8  144    2.51327    0.58779
 9  162    2.82743    0.30902
10  180    3.14159    0.00000
------------------------------
Value of h= .314159

M(0)=0        M( 1 )=-.3115662           M( 2 )=-.5926351
M( 3 )=-.8156919           M( 4 )=-.9589032           M( 5 )=-1.008253
M( 6 )=-.9589041           M( 7 )=-.8156923           M( 8 )=-.5926408
M( 9 )=-.3115652           M( 10 )=0
```

I	Angle Degree	Angle Radian	Estimated Value	Actual Value	First Derivative	Actual Value	Second Derivative	Actual Value
1	15	0.2618	0.2588	0.2588	0.9660	0.9659	-0.2596	-0.2588
2	35	0.6109	0.5736	0.5736	0.8192	0.8192	-0.5770	-0.5736
4	55	0.9599	0.8192	0.8192	0.5734	0.5736	-0.8236	-0.8192
5	75	1.3090	0.9659	0.9659	0.2586	0.2588	-0.9671	-0.9659

Inverse Interpolation by the Method of Successive Approximation Applied to Newton's Forward Difference Formula

Newton's forward difference formula is

$$y_r = y_n + r\,\Delta y_n + \frac{r(r-1)}{2!}\Delta^2 y_n + \frac{r(r-1)(r-2)}{6}\Delta^3 y_n + \cdots \tag{1}$$

where $r = (x - x_n)/h$.

In the method of successive approximation, we start with Eq. (1) in which the second and higher differences are ignored. This gives

$$y_r = y_n + r_1\,\Delta y_n \quad \Rightarrow \quad r_1 = (y_r - y_n)/\Delta y_n \tag{2}$$

The next approximation of r is obtained by including the second difference term in Eq. (2) with the use of $r = r_1$. This gives

$$r_2 = \frac{1}{\Delta y_n}\left[y_r - y_n - \frac{r_1(r_1-1)}{2}\Delta^2 y_n\right]$$

This is continued by including each time the next higher difference trem with the use of r_2 in each term. This gives

$$r_3 = \frac{1}{\Delta y_n}\left[y_r - y_n - \frac{r_2(r_2-1)}{2!}\Delta^2 y_n - \frac{r_2(r_2-1)}{3!}\Delta^3 y_n\right]$$

and so on. The process is repeated till the two successive values of r agree with in the required accuracy.

Illustration To determine the value of x in $x^3 = 9$

Program

```
REM INVERSE INTERPOLATION BY METHOD OF SUCCESSIVE APPROXIMATION
REM APPLIED TO NEWTON FORWARD DIFFERENCE FORMULA
CLS : DIM X(20), F(20, 20)
READ N: DATA 4
DEF FNA (X) = X ^ 3
DEF FNB (X) = X ^ (1 / 3)
READ XI, H, YG: DATA 2,1,9
FOR I = 0 TO N
```

```
X(I) = XI + I * H: F(0, I) = FNA(X(I))
PRINT "X("; I; ")="; X(I), "F(0,"; I; ")="; F(0, I)
NEXT I: PRINT : A$ = INPUT$(1)
FOR I = 1 TO N: FOR J = 0 TO N - I
F(I, J) = F(I - 1, J + 1) - F(I - 1, J)
NEXT J: NEXT I
FOR J = 0 TO N - 1: FOR I = 1 TO N - J
PRINT "F("; I; ","; J; ")="; F(I, J),
NEXT I: PRINT : NEXT J: PRINT
RP = 0
FOR I = 0 TO N
IF F(0, 2) < F(0, 1) THEN 7
IF YG <= F(0, I) THEN 4
GOTO 8
7 IF YG >= F(0, I) THEN 4
8 NEXT I
4 M = I - 1: PRINT " Value of M is"; M: PRINT
FOR I = 0 TO 20
A$ = INPUT$(1)
IF I = 0 THEN R0 = YG - F(0, M): R = R0 / F(1, M): GOTO 5
SUM = 0
FOR L = 1 TO I
PROD = 1
FOR J = 0 TO L: PROD = PROD * (R - J): NEXT J
FACT = 1: FOR K = 1 TO L + 1: FACT = FACT * K: NEXT K
TERM = (PROD / FACT) * F(L + 1, M)
SUM = SUM + TERM
NEXT L
R = (R0 - SUM) / F(1, M)
5 PRINT "ITERATION="; : PRINT USING "##"; I;
PRINT USING "     ###.#####"; R
IF ABS(RP - R) < .00001 THEN 10
RP = R
NEXT I
10 PRINT : PRINT "Required value for"; YG; "=";
PRINT USING "###.#####"; X(M) + R * H
PRINT "Actual value for"; YG; "=";
PRINT USING "###.#####"; FNB(YG)
END
```

Output

```
X( 0 )= 2     F(0, 0 )= 8
X( 1 )= 3     F(0, 1 )= 27
X( 2 )= 4     F(0, 2 )= 64
X( 3 )= 5     F(0, 3 )= 125
X( 4 )= 6     F(0, 4 )= 216

F( 1 , 0 )= 19              F( 2 , 0 )= 18              F( 3 , 0 )= 6
F( 4 , 0 )= 0
F( 1 , 1 )= 37              F( 2 , 1 )= 24              F( 3 , 1 )= 6

F( 1 , 2 )= 61              F( 2 , 2 )= 30
F( 1 , 3 )= 91

 Value of M is 0

ITERATION= 0      0.05263
ITERATION= 1      0.07625
ITERATION= 2      0.07886
ITERATION= 3      0.07970
ITERATION= 4      0.07996
ITERATION= 5      0.08004
ITERATION= 6      0.08007
ITERATION= 7      0.08008

Required value for 9 =  2.08008
Actual value for 9 =  2.08008
```

2.19 NUMERICAL INTEGRATION

To carry out the numerical integration by **(i)** rectangular rule, **(ii)** trapezoidal rule, and **(iii)** Simpson's rule.

The integral

$$J = \int_a^6 f(x)\mathrm{d}x$$

may be evaluated numerically by approximating the integrand f by polynomials.

Rectangular Rule

In rectangular rule, the interval of integration is divided into n equal subintervals of length $h = (b - a)/n$ and approximate f in each such interval by a constant function $f(x_j^*)$ where x_j^* is the mid point of the interval. The function is multiplied by h to give the area under the function in the chosen interval. The value of integral is obtained by summing these areas, i.e.

$$J = h[f(x_1^*) + f(x_2^*) + \cdots + f(x_n^*)]$$

Trapezoidal Rule

In trapezoidal rule, the function f is approximated by a piecewise linear function, and the areas of trapezoids are

$$\frac{1}{2}\,[f(a)+f(x_1)]\,h\;;\;\frac{1}{2}\,[f(x_1)+f(x_2)]h\;;\;\ldots\;\frac{1}{2}\,[f(x_{n-1})+f(b)]h$$

On summing, we get

$$J = h\left[\frac{1}{2}f(a)+f(x_1)+f(x_2)+\cdots+f(x_{n-1})+\frac{1}{2}f(b)\right]$$

Simpson Rule

In Simpson's rule, the function f is approximated by piecewise quadratic expressions. The interval of integration is divided into even numbers of equal subintervals such that $h = (b - a)/2n$. The entire interval is thus labelled as

$$x_0\,(=a);\quad x_1\,(=x_o+h)\;;\quad x_2\,(=x_0+2h),\;\ldots\text{ and so on.}$$

The function $f(x)$ in the interval $x_0 \le x \le x_2$ is approximated by the Lagrange polynomial $L_2(x)$ through (x_0, f_0), (x_1, f_1) and (x_2, f_2).

The expression of $L_2(x)$ in the interval x_0 to x_2 is

$$L_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f_2$$

The denominators are $2h^2$, $-h^2$ and $2h^2$, respectively Let $s = (x - x_1)/h$. We will have

$$x - x_0 = x - (x_1 - h) = \left[\frac{(x-x_1)+h}{h}\right]h = (s+1)h$$

$$x - x_1 = sh$$

$$x - x_2 = (s-1)h$$

Thus $$L_2(s) = \frac{1}{2}\,s(x-1)f_0 - (s+1)\,(s-1)f_1 + \frac{1}{2}\,(s+1)sf_2$$

with $s = (x - x_1)/h$, we get

$$\mathrm{d}s = \frac{dx}{h}$$

with limits of integration given as

$$s_0 = \frac{x_0 - x_1}{h} = \frac{x_0 - (x_0 + h)}{h} = -1$$

$$s_1 = \frac{x_2 - x_1}{h} = \frac{(x_1 + h) - x_1}{h} = +1$$

Hence, $$\int_{x_0}^{x_2} f(x)\,\mathrm{d}x = \int_{-1}^{+1} L_2(s)h\,\mathrm{d}s$$

$$= \int_{-1}^{1}\left[\frac{1}{2}s(s-1)\,f_0 - (s+1)(s-1)f_1 + \frac{1}{2}s(s+1)f_2\right]h\,\mathrm{d}s$$

$$= h\left(\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{1}{3}f_2\right)$$

A similar formula holds for the next two subintervals from x_2 to x_4, and so on. If we sum all these expressions, we will get the Simpson's rule as

$$\int_a^b f(x)\,\mathrm{d}x = \frac{h}{3}\,[f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{2n-2} + 4f_{2n-1} + f_{2n}]$$

where $h = (b - a)/2n$ and $f_j = f(x_j)$

Illustration Make a program to carry out the integration

$$\int_A^B 4\pi\left(\frac{M}{2\pi RT}\right)^{3/2} \exp\left(-\frac{Mu^2}{2RT}\right)u^2\mathrm{d}u$$

with $\mathrm{d}u = (B - A)/N$ where $A = 0$, $B = 1500$ and $N = 100$ by the three methods described above.

Program

```
REM PROGRAM INTSTR;INTEGRATION RECTANGULAR,TRAPZOID AND SIMPSON
CLS : M = .032: A = 0: B = 1500: N = 100
SU = (B - A) / N: PI = 3.1415: R = 8.314
DEF FNA (M, T, U)
X = 4 * PI * (M / (2 * R * PI * T)) ^ 1.5
X = X * EXP(-M * U ^ 2 / (2 * R * T)) * U ^ 2
FNA = X
END DEF
PRINT STRING$(52, "_")
PRINT " T/K            Rectangular     Trapezoidal    Simpson"
PRINT STRING$(52, "_")
FOR T = 298 TO 398 STEP 20
INTS = 0: INTR = 0: INTT = 0: U = A - SU
FOR I = O TO N
U = U + SU: F = FNA(M, T, U)
IF I = 0 OR I = N THEN MULT = 1: GOTO 5
IF I = 2 * INT(I / 2) THEN MULT = 2 ELSE MULT = 4
5 INTS = INTS + MULT * F
```

```
UR = U + SU / 2: INTR = INTR + FNA(M, T, UR)
IF I = 0 OR I = N THEN MULT = .5 ELSE MULT = 1
INTT = INTT + MULT * F
NEXT I
INTS = INTS * SU / 3: INTR = SU * INTR: INTT = SU * INTT
PRINT T, INTR, INTT, INTS
NEXT T
PRINT STRING$(52, "_")
END
```

Output

T/K	Rectangular	Trapezoidal	Simpson
298	1.000013	1.000013	1.000012
318	1.000011	1.000009	1.000009
338	1.000006	1.000003	1.000003
358	.999997	.9999918	.9999921
378	.9999809	.9999726	.9999723
398	.9999555	.9999412	.9999416

Double Integration

Consider the integration

$$I = \int_{x_0}^{x_1}\int_{y_0}^{y_1} f(x, y)\, \mathrm{d}y\, \mathrm{d}x$$

The numerical integration of the above expression can be carried out with the following procedure.

Take $x = x_0$ and carry out the numerical integration over y. To the obtained value, add the new value of integration over y with $x = x_0 + h$ and so on.

The above procedure may be outlined mathematically as follows.

$$I = \frac{hk}{N}\left(\sum_{i=0}^{n}\lambda_i \sum_{j=0}^{n}\lambda_j\, f(x_i, y_i)\right)$$

where $x_\mathrm{i} = x_0 + i \times h$ with $h = (x_N - x_0)/N$

$y_j = y_j + j \times k$ with $k = (y_N - y_0)/N$

The symbols λ_i, λ_j and N are constants, their values are as follows.

Rectangular Method

$$\lambda_i = \lambda_j = N = 1$$

Trapezoidal Method

$\lambda_0 = \lambda_N = 1$ and all other λ's = 2

$N = 2 \times 2 = 4$

Simpson method

$\lambda_0 = \lambda_N = 1$
λ's = 2 if i or j is even
λ's = 4 if i or j is odd
$N = 3 \times 3 = 9$

Illustration To carry out the numerical integration

$$I = \int_0^4 \int_{-2}^{+2} (x^2 - xy + y^2)\, \mathrm{d}y\, \mathrm{d}x$$

Program

```
REM Double Integration by Rectangular,Trapezoid & Simpson Methods
REM M=No. of Divisions: XI & XF=Initial & Final X
REM YI & YF=Initial & Final Y
CLS
DEF FNA (X, Y) = X ^ 2 - X * Y + Y ^ 2
N = 0: M = 10
READ XI, XF, YI, YF
DATA 0,4,-2,2
PRINT STRING$(52, "_")
PRINT "  N          Rectangular    Trapezoidal    Simpson"
PRINT STRING$(52, "_")
FOR L = 1 TO M
N = N + 4
XS = (XF - XI) / N: YS = (YF - YI) / N
INTS = 0: INTR = 0: INTT = 0
FOR I = 0 TO N
X = XI + I * XS
IF I = 0 OR I = N THEN MULTS = 1: GOTO 10
IF I = 2 * INT(I / 2) THEN MULTS = 2 ELSE MULTS = 4
10 IF I = 0 OR I = N THEN MULTT = 1 ELSE MULTT = 2
FOR J = 0 TO N
Y = YI + J * YS: F = FNA(X, Y)
IF J = 0 OR J = N THEN MULT = 1: GOTO 15
IF J = 2 * INT(J / 2) THEN MULT = 2 ELSE MULT = 4
15 INTS = INTS + MULTS * MULT * F
INTR = INTR + F
IF J = 0 OR J = N THEN MULT = 1 ELSE MULT = 2
```

```
INTT = INTT + MULTT * MULT * F
NEXT J
NEXT I
INTS = INTS * XS * YS / 9
INTR = XS * YS * INTR: INTT = INTT * XS * YS / 4
PRINT N, INTR, INTT, INTS
NEXT L
PRINT STRING$(52, "_")
END
```

Output

N	Rectangular	Trapezoidal	Simpson
4	200	112	106.6667
8	148.5	108	106.6667
12	133.5309	107.2593	106.6667
16	126.4375	107	106.6667
20	122.304	106.88	106.6667
24	119.5987	106.8147	106.6667
28	117.691	106.7756	106.6667
32	116.2734	106.75	106.6667
36	115.1788	106.7326	106.6666
40	114.308	106.72	106.6667

2.20 NUMERICAL DIFFERENTIATION

The derivative of a function f available in the form of tabulated values is normally carried out by the differentiation of the interpolation formula.

Using Newton's Forward/Backward Difference formula

For the Newton's forward difference formula, we have

$$y = y_m + r\,\Delta y_m + \frac{r(r-1)}{2!}\Delta^2 y_m + \frac{r(r-1)(r-2)}{3!}\Delta^3 y_m + \cdots; \qquad (r = (x - x_m)/h)$$

We have

$$\frac{dy}{dx} = \left(\frac{dy}{dr}\right)\left(\frac{dr}{dx}\right)$$

$$= \frac{1}{h}\left[\Delta y_m + \frac{(r-1)+r}{2!}\Delta^2 y_m + \frac{\{(r-1)(r-2)\}+\{r(r-2)\}+\{r(r-1)\}}{3!}\Delta^3 y_m + \cdots\right]$$

$$= \frac{1}{h}\left[\Delta y_m + \sum_{s=1}^{n} \frac{1}{(s+1)!}\left\{\sum_{i=0}^{s}\left(\prod_{\substack{j=0\\ j\neq i}}^{s}(r-j)\right)\right\}\Delta^{s+1}y_m\right]$$

The expression of second derivative is

$$\frac{\mathrm{d}^2y}{\mathrm{d}x^2} = \frac{1}{h^2}\left[\Delta^2 y_m + \frac{\{(r-2)+(r-1)\}+\{(r-2)+r\}+\{(r-1)+r)\}}{3!}\Delta^3 y_m + \cdots\right]$$

$$= \frac{1}{h^2}\left[\Delta^2 y_m + \sum_{s=2}^{n}\frac{1}{(s+1)!}\left[\sum_{i=0}^{s}\left\{\sum_{\substack{j=0\\ j\neq i}}^{s}\left(\prod_{\substack{k=0\\ k\neq i,j}}^{s}(r-k)\right)\right\}\right]\Delta^{s+1}y_m\right]$$

For the Newton's backward difference formula, we have

$$y = y_m + r\,\nabla y_m + \frac{r(r+1)}{2!}\nabla^2 y_m + \frac{r(r+1)(r+2)}{3!}\nabla^3 y_m + \cdots$$

we have

$$y' = \frac{1}{h}\left[\nabla y_m + \frac{(r+1)+r}{2!}\nabla^2 y_m + \frac{\{(r+1)(r+2)\}+\{r(r+2)\}+\{r(r+1)\}}{3!}\nabla^3 y_m + \cdots\right]$$

$$y'' = \frac{1}{h^2}\left[\nabla^2 y_m + \frac{\{(r+2)+(r+1)\}+\{(r+2)+r\}+\{(r+1)+r\}}{3!}\nabla^3 y_m + \cdots\right]$$

Illustration Determine the first and second derivatives $\mathrm{d}f/\mathrm{d}x$ and $\mathrm{d}^2f/\mathrm{d}x^2$ of $f = \sin x$ (where x is in radians) at $x = 1.25$ and 2.1. Given are the values from 1 to 2.2 at the interval of 0.2.

Program

```
REM NEWTON FORWARD/BACKWARD DIFFERENTIATION
CLS : DIM X(20), F(20, 20), FXD(20), SXD(20)
REM Give either of the following two statements
REM depending upon the forward or backward differentiation
READ N$, XX: DATA F,1.25
'READ N$, XX: DATA B,2.1
DEF FNA (X) = SIN(X): DEF FNB (X) = COS(X): DEF FNC (X) = -SIN(X)
'DEF FNA (X) = EXP(X): DEF FNB (X) = EXP(X): DEF FNC (X) = EXP(X)
READ N: DATA 6
READ XI, H: DATA 1,.2
FOR I = 0 TO N
X(I) = XI + I * H: F(0, I) = FNA(X(I))
```

```
PRINT "I="; I; "    X("; I; ")="; X(I),
PRINT "F(0,"; I; ")="; F(0, I)
NEXT I: PRINT
FOR I = 1 TO N: FOR J = 0 TO N - I
F(I, J) = F(I - 1, J + 1) - F(I - 1, J)
NEXT J: 'PRINT
NEXT I: 'PRINT
'FOR J = 0 TO N - 1: FOR I = 1 TO N - J
'PRINT "F("; I; J; ")="; F(I, J),
'NEXT I: PRINT : NEXT J
'A$ = INPUT$(1)
FOR M = 0 TO N
IF H < 0 THEN 1
IF XX <= X(M) THEN 3
GOTO 2
1 IF XX >= X(M) THEN 3
2 NEXT M
3 PRINT "No. of data point used ="; M
U = (XX - X(M)) / H: J3 = M
IF N$ = "F" THEN J1 = N: J2 = -1 ELSE J1 = M: J2 = 1
FOR S = 0 TO J1
FXD(S) = 0: FACT = 1: FD = 0: SD = 0
SXD(S) = 0
FOR I = 0 TO S
FDN = 1: FD = 0: SD = 0
FOR J = 0 TO S
IF I = J THEN 6
FDN = FDN * (U + J2 * J): SDN = 1
FOR K = 0 TO S
IF K = I OR K = J THEN 4
SDN = SDN * (U + J2 * K)
4 NEXT K
SD = SD + SDN
6 NEXT J
FD = FD + FDN: SXD(S) = SXD(S) + SD
FXD(S) = FXD(S) + FD
NEXT I
NEXT S
FIRST = 0
```

```
'PRINT : PRINT "DETAILS OF FIRST DERIVATIVE": PRINT
'PRINT "[";
IF N$ = "F" THEN K3 = J1 ELSE K3 = J3 - 1
FOR S = 0 TO K3
IF N$ = "F" THEN J4 = 0 ELSE J4 = S + 1
FACT = 1: FOR I = 1 TO S + 1: FACT = FACT * I: NEXT I
'PRINT "+("; FXD(S); "/"; FACT; ")("; F(S + 1, J3 - J4); ")";
'A$ = INPUT$(1)
FIRST = FIRST + (FXD(S) / FACT) * F(S + 1, J3 - J4)
NEXT S
'PRINT "]/"; H; "="; FIRST / H
PRINT "  First derivative at"; XX; "="; FIRST / H
PRINT "  Correct value at"; XX; "="; FNB(XX): PRINT
'PRINT "DETAILS OF SECOND DERIVATIVE": PRINT
'PRINT "[";
FOR S = 2 TO K3
IF N$ = "F" THEN J4 = 0 ELSE J4 = S
FACT = 1: FOR I = 1 TO S: FACT = FACT * I: NEXT I
'PRINT "+("; SXD(S - 1); "/"; FACT; ")+("; F(S, J3 - J4); ") ";
'A$ = INPUT$(1)
SECOND = SECOND + (SXD(S - 1) / FACT) * F(S, J3 - J4)
NEXT S
'PRINT "]/"; H ^ 2; "="; SECOND / H ^ 2
PRINT "  Second derivative at"; XX; "="; SECOND / H ^ 2
PRINT "  Correct value at"; XX; "="; FNC(XX)
END
```

Output of Forward Differentiation

```
I= 0    X( 0 )= 1          F(0, 0 )= .841471
I= 1    X( 1 )= 1.2        F(0, 1 )= .9320391
I= 2    X( 2 )= 1.4        F(0, 2 )= .9854497
I= 3    X( 3 )= 1.6        F(0, 3 )= .9995736
I= 4    X( 4 )= 1.8        F(0, 4 )= .9738476
I= 5    X( 5 )= 2          F(0, 5 )= .9092974
I= 6    X( 6 )= 2.2        F(0, 6 )= .8084964

No. of data point used = 2
  First derivative at 1.25 = .315613
  Correct value at 1.25 = .3153224

  Second derivative at 1.25 =-.9512241
  Correct value at 1.25 =-.9489846
```

Output of Backward Differentiation

```
I= 0    X( 0 )= 1          F(0, 0 )= .841471
I= 1    X( 1 )= 1.2        F(0, 1 )= .9320391
I= 2    X( 2 )= 1.4        F(0, 2 )= .9854497
I= 3    X( 3 )= 1.6        F(0, 3 )= .9995736
I= 4    X( 4 )= 1.8        F(0, 4 )= .9738476
I= 5    X( 5 )= 2          F(0, 5 )= .9092974
I= 6    X( 6 )= 2.2        F(0, 6 )= .8084964

No. of data point used = 6
  First derivative at 2.1 =-.504846
  Correct value at 2.1 =-.504846

  Second derivative at 2.1 =-.8630071
  Correct value at 2.1 =-.8632094
```

Using Lagrange's Interpolation Formula

The Lagrange's interpolation formula is

$$L_n = \sum_{i=0}^{n} \left[\prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) \right] y_{\mathrm{i}}$$

where $L_n\ (x_{\mathrm{i}}) = y\ (x_{\mathrm{i}})$.

The differentiation of L_n gives

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}L_n}{\mathrm{d}x} = \sum_{i=0}^{n} \left[\sum_{\substack{j=0 \\ j \neq i}}^{n} \left\{ \frac{1}{(x_i - x_j)} \prod_{\substack{k=0 \\ k \neq i,j}}^{n} \left(\frac{x - x_k}{x_i - x_k} \right) \right\} \right] y_{\mathrm{i}}$$

The expression of second derivative is

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} = \sum_{i=0}^{n} \left[\sum_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{1}{x_i - x_j} \right) \left\{ \sum_{\substack{k=0 \\ k \neq i,j}}^{n} \left(\frac{1}{x_i - x_k} \right) \left(\prod_{\substack{l=0 \\ l \neq i,j,k}}^{n} \left(\frac{x - x_l}{x_i - x_l} \right) \right) \right\} \right] y_{\mathrm{i}}$$

Illustration Differentiation of ln (x)

Program

```
CLS
REM Differentiation via Lagrange's interpolation formula
DEF FNA (X) = LOG(X): DEF FNB (X) = 1 / X
DEF FNC (X) = -1 / X ^ 2
READ N: DATA 4
```

```
FOR I = 0 TO N
READ X(I): Y(I) = FNA(X(I)): PRINT X(I), Y(I)
NEXT I
DATA 300,302,304,306,308
READ XG: DATA 305
REM Calculation of the first derivative
SUMI = 0
FOR I = 0 TO N
SUMJ = 0
FOR J = 0 TO N
IF J = I THEN 10
TERM = 1 / (X(I) - X(J))
FOR K = 0 TO N
IF K = I OR K = J THEN 5
TERM = TERM * (XG - X(K)) / (X(I) - X(K))
5 NEXT K
SUMJ = SUMJ + TERM
10 NEXT J
SUMI = SUMI + SUMJ * Y(I)
NEXT I
PRINT "Calculated first derivative at"; XG; "="; SUMI
PRINT "Actual value at"; XG; "="; FNB(XG)
REM Calculation of the second derivative
SUMI = 0
FOR I = 0 TO N
SUMJ = 0
FOR J = 0 TO N
IF J = I THEN 25
TERM = 1 / (X(I) - X(J))
SUMK = 0
FOR K = 0 TO N
IF K = I OR K = J THEN 20
TERM1 = 1 / (X(I) - X(K))
SUML = 1
FOR L = 0 TO N
IF L = I OR L = J OR L = K THEN 15
SUML = SUML * (XG - X(L)) / (X(I) - X(L))
15 NEXT L
SUMK = SUMK + SUML * TERM1
20 NEXT K
```

```
SUMJ = SUMJ + SUMK * TERM
25 NEXT J
SUMI = SUMI + SUMJ * Y(I)
NEXT I
A$ = INPUT$(1): PRINT
PRINT "Calculated second derivative at "; XG; "="; SUMI
PRINT "Actual value at "; XG; "="; FNC(XG)
END
```

Output

```
 300           5.703783
 302           5.710427
 304           5.717028
 306           5.723585
 308           5.7301
Calculated first derivative at 305 = 3.278658E-03
Actual value at 305 = 3.278689E-03

Calculated second derivative at  305 =-1.064077E-05
Actual value at  305 =-1.07498E-05
```

2.21 SINE AND COSINE OF AN ANGLE BY SERIES SUMMATION

To determine the values of sin x and cos x by carrying out the summations

$$\sin x = \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m+1}}{(2m+1)!}$$

$$\cos x = \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m}}{(2m)!}$$

where x is expressed in radians. Continue the summation until the term is less than 10^{-4}.

Also display the graphical variations of (i) sin θ versus θ, (ii) cos θ versus θ, (iii) tan θ versus θ.

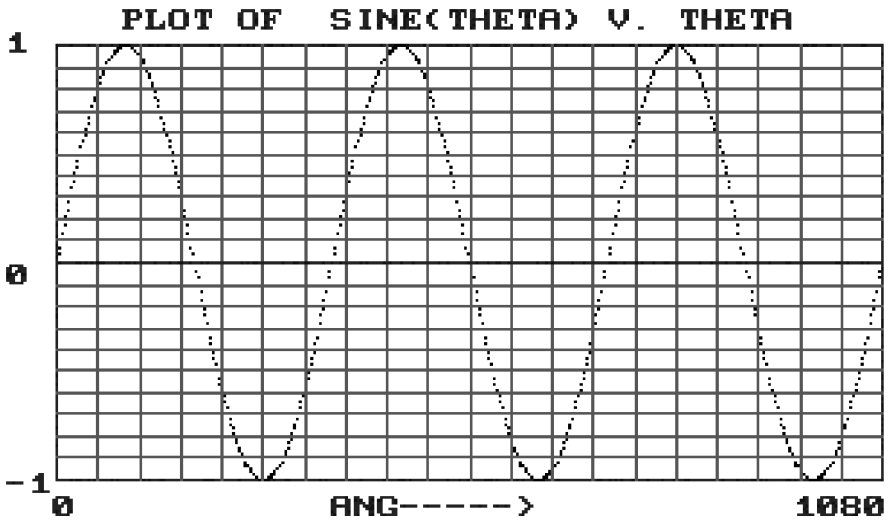
Since θ and $\theta + n(2\pi)$ have the same values of sin θ and cos θ, all angles greater than 2π may be reduced to the principal value lying between 0 and 2π.

Program

```
REM PROGRAM SINCOS;PLOTS OF SINE OR COSINE OF ANGLES
REM OPT$=SINE IF SINE REQUIRED ELSE COSINE IF COSINE REQUIRED
REM OPT=1 IF NUMERICAL VALUES REQUIRED ELSE 2 IF PLOTTING REQUIRED
CLS : READ ANGMAX, OPT, ANGD, OPT$
DATA 1080,2,2 ,SINE
KK = 1
ON OPT GOTO 5, 10
5 PRINT STRING$(30, "-")
PRINT "THETA    ";
IF OPT$ = "SINE" THEN PRINT "SINE";  ELSE PRINT "COSINE";
```

```
PRINT "       LIBRARY ": PRINT STRING$(30, "-")
GOTO 15
10 SCREEN 1: COLOR 15, 0
VIEW (25, 20)-(310, 170)
WINDOW (0, -1)-(ANGMAX, 1): LINE (0, -1)-(ANGMAX, 1), , B
LOCATE 2, 7: PRINT "PLOT OF ";
IF OPT$ = "SINE" THEN PRINT " SINE";  ELSE PRINT "COSINE";
PRINT "(THETA) V. THETA"
LOCATE 3, 2: PRINT "1": LOCATE 13, 2: PRINT "0"
LOCATE 22, 2: PRINT "-1"
15 FOR ANG = 0 TO ANGMAX STEP ANGD
IF ANG > KK * 360 THEN KK = KK + 1
IF KK = 1 THEN THETA = ANG ELSE THETA = ANG - (KK - 1) * 360
SINE = 0: RAD = 3.14159 * THETA / 180
FOR M = 0 TO 12
IF OPT$ = "SINE" THEN K = 2 * M + 1 ELSE K = 2 * M
GOSUB 100
TERM = (-1) ^ M * RAD ^ K / FACT: SINE = SINE + TERM
IF ABS(TERM) < .00001 THEN 50
NEXT M
50 ON OPT GOTO 60, 65
60 IF OPT$ = "SINE" THEN CAL = SIN(RAD) ELSE CAL = COS(RAD)
PRINT ANG; : PRINT USING "     ##.####"; SINE; CAL
a$ = INPUT$(1)
GOTO 70
65 PSET (ANG, SINE), 2
70 FOR LL = 1 TO 100 STEP .001: NEXT LL: NEXT ANG
ON OPT GOTO 80, 90
80 PRINT STRING$(30, "-")
GOTO 200
90 LOCATE 23, 2: PRINT "  0           ANG----->           ";
PRINT USING "####"; ANGMAX
FOR I = 1 TO 19
XX = ANGMAX * I / 20: YY = -1 + 2 * I / 20
LINE (XX, -1)-(XX, 1), 1: LINE (0, YY)-(ANGMAX, YY), 1
NEXT I
LINE (0, 0)-(ANGMAX, 0)
GOTO 200
100 FACT = 1
FOR J = 1 TO K: FACT = FACT * J: NEXT J
RETURN
200 END
```

Output



Exercises

Calculate the value of the following functions by carrying out the indicated summations.

(i) $e^x = \sum_{m=0}^{\infty} \frac{x_m}{m!}$ **(ii)** $\ln(1-x) = -\sum_{m=0}^{\infty} \frac{x_m}{m!}$; $(|x| < 1)$

(iii) $\arctan x = \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m+1}}{(2m+1)}$; $(|x| < 1)$

Also display the graphical variations of these functions for increasing value of x.

2.22 HISTROGRAM OF THE RATIO ${}^nC_m/({}^nC_m)_{\text{maximum}}$

To draw histogram displaying the ratio $\frac{{}^nC_m}{({}^nC_m)_{\text{maximum}}}$ where $m = 0, 1, 2, \ldots, n$

Maximum value of ${}^nC_m$ is obtained when $m = n/2$ for even value of n and when $m = \text{Integer}(n/2) + 1$ for odd value of n.

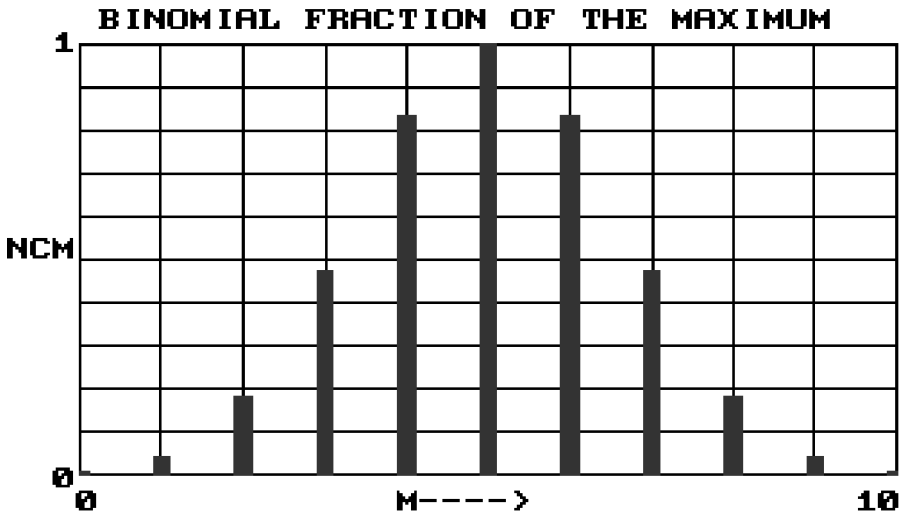
Program

```
REM BINOMIAL;BINOMIAL FRACTIONS
CLS : DIM NCM(25)
READ N: DATA 10
DEF FNA (J)
FACT = 1
IF J = 0 THEN 5
FOR I = 1 TO J: FACT = FACT * I: NEXT I
5 FNA = FACT
```

```
END DEF
NCMMAX = 0
PRINT STRING$(35, "_")
PRINT "   N               M               NCM"
PRINT STRING$(35, "_"): A$ = INPUT$(1)
FOR M = 0 TO N
NCM(M) = FNA(N) / (FNA(M) * FNA(N - M))
IF NCMMAX < NCM(M) THEN NCMMAX = NCM(M)
PRINT N, M, NCM(M)
NEXT M
PRINT STRING$(35, "_"): A$ = INPUT$(1)
SCREEN 1: COLOR 15, 0
VIEW (25, 20)-(310, 170)
WINDOW (0, 0)-(N, 1): LINE (0, 0)-(N, 1), , B
FOR I = 0 TO N
LINE (I, 0)-(I, 1): LINE (0, I * .1)-(N, I * .1)
NEXT I
FOR M = 0 TO N
ND = N / 100: MD = NCM(M) / NCMMAX
LINE (M - ND, 0)-(M + ND, MD), 1, BF
NEXT M
LOCATE 2, 5: PRINT "BINOMIAL FRACTION OF THE MAXIMUM"
LOCATE 3, 3: PRINT "1": LOCATE 12, 1: PRINT "NCM"
LOCATE 22, 3: PRINT "0"
LOCATE 23, 3: PRINT " 0            M---->           "; : PRINT N
END
```

Output

N	M	NCM
10	0	1
10	1	10
10	2	45
10	3	120
10	4	210
10	5	252
10	6	210
10	7	120
10	8	45
10	9	10
10	10	1



2.23 EVALUATION OF THERMODYNAMIC QUANTITIES OF A MONATOMIC GASEOUS SUBSTANCE

To determine the molar thermodynamic quantities (U°_m, $C_{V,m}$, H°_m, S°_m, A°_m and G°_m) of a monatomic fluorine gas at 1 bar and 1000 K.

For fluorine (relative atomic mass: 19), the first two electronic energy levels are as follows.

$(1s)^2(2s)^2(2p)^5$	$^2P_{3/2}$	0
	$^2P_{1/2}$	404 cm^{-1}

For translation contribution, the expressions to be used are

$$U^\circ_m = (3/2)RT \quad ; \quad C_{V,m} = (3/2)\,R$$

$$H^\circ_m = (5/2)RT \quad ; \quad C_{p,m} = (5/2)\,R$$

$$S^\circ_m = R\left[-1.1541 + \frac{3}{2}\ln A_r + \frac{5}{2}\ln\left(\frac{T}{K}\right) - \ln\left(\frac{p}{p^\circ}\right)\right]$$

$$A^\circ_m = U^\circ_m - T\,S^\circ_m \quad : \quad G^\circ_m = H^\circ_m - T\,S^\circ_m$$

For electronic contribution, the expressions to be used are:

$$q_e = \Sigma_i g_i \exp(-\varepsilon_i/kT) \qquad q'_e = \Sigma_i g_i (\varepsilon_i/kT) \exp(-\varepsilon_i/kT)$$

$$q''_e = \Sigma_i g_i (\varepsilon_i/kT)^2 \exp(-\varepsilon_i/kT) - q'_e \qquad (U^\circ_m)_e = (H^\circ_m)_e = RT \ln (q'_e/q_e)$$

$$(C_{p,\mathrm{m}})_\mathrm{e} = (C_{V,\mathrm{m}})_\mathrm{e} = R\left[\frac{q'_\mathrm{e} + q''_\mathrm{e}}{q_\mathrm{e}} - \left(\frac{q'_\mathrm{e}}{q_\mathrm{e}}\right)^2\right] \qquad (S^\circ_\mathrm{m})_\mathrm{e} = R\left[\ln q_\mathrm{e} + \frac{q'_\mathrm{e}}{q_\mathrm{e}}\right]$$

$$(G^\circ_\mathrm{m})_\mathrm{e} = (H^\circ_\mathrm{m})_\mathrm{e} - T(S^\circ_\mathrm{m})_\mathrm{e} \qquad (A^\circ_\mathrm{m})_\mathrm{e} = (U^\circ_\mathrm{m})_\mathrm{e} - T(S^\circ_\mathrm{m})_\mathrm{e}$$

Program

```
REM
REM PROGRAM STMONO;THERMODYNAMIC DATA OF MONOATOMIC GAS
CLS : DIM J(5), WN(5), G(5), EBKT(5)
READ T, Ar, P: DATA 1000,19,1,1.5,0,.5,40400
R = 8.314: H = 6.626E-34: C = 3E+08: KB = R / 6.022E+23
UT = 1.5 * R * T: CVT = 1.5 * R
HT = 2.5 * R * T: CPT = 2.5 * R
ST = R * (-1.1541 + 1.5 * LOG(Ar) + 2.5 * LOG(T) - LOG(P))
AT = UT - T * SM: GT = HT - T * ST
QE = 0: QEP = 0: QEDP = 0
FOR I = 1 TO 2
READ J(I), WN(I): G(I) = 2 * J(I) + 1
EBKT(I) = H * C * WN(I) / (KB * T)
QE = QE + G(I) * EXP(-EBKT(I))
QEP = QEP + G(I) * EBKT(I) * EXP(-EBKT(I))
QEDP = QEDP + G(I) * EBKT(I) ^ 2 * EXP(-EBKT(I))
NEXT I
QEDP = QEDP - QEP
UE = R * T * (QEP / QE)
CVE = R * ((QEP + QEDP) / QE - (QEP / QE) ^ 2)
HE = UE: CPE = CVE
SE = R * (LOG(QE) + QEP / QE)
AE = R * T * LOG(QE): GE = AE
UM = UT + UE: CVM = CVT + CVE
HM = HT + HE: CPM = CPT + CPE
SM = ST + SE: AM = AT + AE: GM = GT + GE
PRINT "FLUORINE ATOM AT 1000 K": PRINT
PRINT "U in J/mol": PRINT "UT ="; UT,
PRINT "UE ="; UE, "UM ="; UM: PRINT
PRINT "Cv in J/(K mol)": PRINT "CVT ="; CVT, "CVE ="; CVE,
PRINT "CVM ="; CVM: PRINT
PRINT "H in J/mol": PRINT "HT ="; HT, "HE ="; HE,
PRINT "HM ="; HM: PRINT
PRINT "Cp in J/(K mol)": PRINT "CPT ="; CPT, "CPE ="; CPE,
```

```
PRINT "CPM ="; CPM: PRINT
PRINT "S in J/(K mol)": PRINT "ST ="; ST, "SE ="; SE,
PRINT "SM ="; SM: PRINT
PRINT "A in J/mol": PRINT "AT ="; AT, "AE ="; AE,
PRINT "AM ="; AM: PRINT
PRINT "G in J/mol": PRINT "GT ="; GT, "GE ="; GE,
PRINT "GM ="; GM
END
```

Output

```
FLUORINE ATOM AT 1000 K

U in J/mol
UT = 12471    UE = 1056.357              UM = 13527.36

Cv in J/(K mol)
CVT = 12.471  CVE = .4802446             CVM = 12.95125

H in J/mol
HT = 20785    HE = 1056.357              HM = 21841.36

Cp in J/(K mol)
CPT = 20.785  CPE = .4802446             CPM = 21.26524

S in J/(K mol)
ST = 170.7026              SE = 14.63102              SM = 185.3336

A in J/mol
AT = 12471    AE = 13574.66              AM = 26045.66

G in J/mol
GT =-149917.6              GE = 13574.66              GM =-136343
```

2.24 EVALUATION OF THERMODYNAMIC QUANTITIES OF A DIATOMIC GASEOUS SUBSTANCE

To determine the molar thermodynamic quantities (U°_m, $C_{V,m}$, H°_m, $C_{p,m}$, S°_m, A°_m and G°_m) for a diatomic gas at 1 bar and 298 K.

For translational contribution, the expressions to be used are

$$U^\circ_m = (3/2)\,RT \quad ; \quad H^\circ_m = (5/2)\,RT$$

$$C_{V,m} = (3/2)\,R \quad ; \quad C_{p,m} = (5/2)\,R$$

$$S^\circ_m = R\left[-1.1541 + \frac{3}{2}\ln(M_r) + \frac{5}{2}\ln\left(\frac{T}{K}\right) - \ln\left(\frac{p}{p_0}\right)\right]$$

$A^\circ_m = U^\circ_m - TS^\circ_m$; $G^\circ_m = H^\circ_m - TS^\circ_m$

For rotational contribution, the expressions to be used are

$\mu_1 = [M_1 M_2 / (M_1 + M_2)]\ (1/N_A)$; $I = \mu r^2$

$\theta_r = h^2/ (8\pi^2 k_B I)$; $q_r = T/\sigma \theta_r$

$U_r = RT$; $C_{V,m} = R$

$H_r = RT$; $C_{p,m} = R$

$S_r = R(\ln q_r + 1)$; $A_r = -RT \ln q_r$

$G_r = A_r$.

For vibrational contribution, the expressions to be used are

$$\theta_v = h\nu/k_B \quad ; \quad q_v = \frac{1}{1 - \exp(-\theta_v / T)}$$

$$U_v = \frac{R\theta_v}{\exp(-\theta_v / T) - 1} \quad ; \quad C_V = \frac{R(\theta_v / T)^2 \exp(\theta_v / T)}{\{\exp(\theta_v / T) - 1\}^2}$$

$H_v = U_v$; $C_p = C_V$

$$S_v = R\left[-\ln\{1 - \exp(-\theta_v / T)\} + \frac{\theta_v / T}{\exp(\theta_v / T) - 1}\right]$$

$A_v = RT \ln [1 - \exp(-\theta_v/T)]$

$G_v = A_v$

For electronic contribution, the expressions to be used are

$q_e = 0$; $q'_e = 0$; $q''_e = 0$

$U_e = 0$; $(C_V)_e = 0$; $H_e = 0$; $(C_P)_e = (C_V)_e$

$S_e = R \ln g_0$; $A_e = -RT \ln g_0$; $G_e = A_e$

Data

Molecule	*Internuclear distance*	σ	ν/Hz	g_0
O_2	121 pm	2	4.74 E + 13	3
H_2	74.17 pm	2	1.32 E + 14	1
F_2	140.9 pm	2	2.77 E + 13	1
Cl_2	198.8 pm	2	1.68 E + 13	1
HCl	127.46 pm	1	8.97 E + 13	1

Illustration Thermodynamic quantities of HCl.

Program

```
REM
REM PROGRAM STDI1;THERMODYNAMIC DATA FOR DIATOMIC GAS-TABULAR
REM M1 & M2 ARE RELATIVE ATOMIC MASSES,DIS IN pm
CLS : READ NAME$, T, M1, M2, P, DIS, SIGMA, NU, G0
PRINT NAME$, "T="; T; "K": M1M2 = M1 + M2
R = 8.314: H = 6.626E-34: C = 3E+08: NA = 6.022E+23: KB = R / NA
UT = 1.5 * R * T: CVT = 1.5 * R: HT = 2.5 * R * T: CPT = 2.5 * R
ST = R * (-1.1541 + 1.5 * LOG(M1M2) + 2.5 * LOG(T) - LOG(P))
AT = UT - T * SM: GT = HT - T * ST
RM = (M1 * M2) / (M1M2 * NA): MI = RM * (DIS * 1E-12) ^ 2
THETR = H ^ 2 / (8 * 3.14159 ^ 2 * MI * .001 * KB)
QR = T / (SIGMA * THETR): UR = R * T: CVR = R: HR = R * T
CPR = R: SR = R * (LOG(QR) + 1): AR = -R * T * LOG(QR): GR = AR
'THETV = H * C * WN0 * 100 / KB
THETV = H * NU / KB: THETVT = THETV / T
EXTHET = EXP(THETVT) - 1: QV = 1 / (1 - EXP(-THETVT))
UV = R * THETV / EXTHET
CVV = R * THETVT ^ 2 * EXP(THETVT) / EXTHET ^ 2: HV = UV
CPV = CVV: SV = R * (-LOG(1 - EXP(-THETVT)) + THETVT / EXTHET)
AV = R * T * LOG(1 - EXP(-THETVT)): GV = AV
QE = 0: QEP = 0: QEDP = 0: UE = 0: CVE = 0
HE = UE: CPE = CVE: SE = R * LOG(G0)
AE = -R * T * LOG(G0): GE = AE
UM = UT + UR + UV + UE: CVM = CVT + CVR + CVV + CVE
HM = HT + HR + HV + HE: CPM = CPT + CPR + CPV + CPE
SM = ST + SR + SV + SE: AM = AT + AR + AV + AE
GM = GT + GR + GV + GE: PRINT
PRINT STRING$(74, "-")
PRINT TAB(14); "TRANSLATIONAL   ROTATIONAL   VIBRATIONAL   ELECTRONIC   TOTAL"
PRINT STRING$(74, "-")
PRINT "U/(J/mol)     "; : PRINT USING "  ######.###"; UT; UR; UV; UE; UM
PRINT "Cv/(J/(K mol))"; : PRINT USING "  ######.###"; CVT; CVR; CVV; CVE; CVM
PRINT "H/(J/mol)     "; : PRINT USING "  ######.###"; HT; HR; HV; HE; HM
PRINT "Cp/(J/(K mol))"; : PRINT USING " ######.###"; CPT; CPR; CPV; CPE; CPM
PRINT "S/(J/(K mol)) "; : PRINT USING "  ######.###"; ST; SR; SV; SE; SM
PRINT "A/(J/mol)     "; : PRINT USING "  ######.###"; AT; AR; AV; AE; AM
PRINT "G/(J/mol)     "; : PRINT USING "  ######.###"; GT; GR; GV; GE; GM
PRINT STRING$(74, "-")
'DATA O2,298,16,16,1,121,2,4.74E+13,3
'DATA H2,298,1,1,1,74.17,2,1.32E+14,1
```

```
'DATA F2,298,19,19,1,140.9,2,2.77E+13,1
'DATA Cl2,298,35,35,1,198.8,2,1.68E+13,1
DATA HCl,298,1,35,1,127.46,1,8.97E+13,1
END
```

Output

```
HCl            T= 298 K

-------------------------------------------------------------------------
              TRANSLATIONAL   ROTATIONAL   VIBRATIONAL   ELECTRONIC   TOTAL
-------------------------------------------------------------------------
U/(J/mol)          3716.358    2477.572         0.019        0.000    6193.949
Cv/(J/(K mol))       12.471       8.314         0.001        0.000      20.786
H/(J/mol)          6193.930    2477.572         0.019        0.000    8671.521
Cp/(J/(K mol))       20.785       8.314         0.001        0.000      29.100
S/(J/(K mol))       153.509      33.095         0.000        0.000     186.604
A/(J/mol)          3716.358   -7384.639        -0.001        0.000   -3668.282
G/(J/mol)        -39551.742   -7384.639        -0.001        0.000  -46936.383
-------------------------------------------------------------------------
```

Exercises

Determine the thermodynamic properties of the following molecules

(a) O_2 at 298 K and 1 atm pressure
Internuclear distance 121 pm
$g_0 = 3$; $\nu_{vib} = 4.74E+13$, $\sigma = 2$

(b) H_2 at 298 K and 1 atm pressure
Internuclear distance 74.17 pm
$g_0 = 1$; $\nu_{vib} = 1.32E+14$; $\sigma = 2$

(c) F_2 at 298 K and 1 atm pressure
Internuclear distance 140.9 pm
$g_0 = 1$; $\nu_{vib} = 2.77E+13$, $\sigma = 2$

(d) Cl_2 at 298 K and 1 atm pressure
Internuclear distance 198.8 om
$g_0 = 1$, $\nu_{vib} = 1.68\,E+13$, $\sigma = 2$

2.25 EVALUATION OF THERMODYNAMIC QUANTITIES OF A TRIATOMIC GASEOUS SUBSTANCE

To determine the molar thermodynamic quantities of a triatomic molecule at a given temperature and pressure.

LINEAR MOLECULE

For Translational Contribution

$$U_m = (3/2)\,RT \qquad ; \qquad H_m = (5/2)\,RT$$

$$S_m = R\left[-1.1541 + 1.5\ln M_r + 2.5\ln\left(\frac{T}{K}\right) - \ln\left(\frac{p}{p^\circ}\right)\right]$$

$$A_m = U_m - TS_m \quad ; \quad G_m = H_m - TS_m$$

For Rotational Contribution

$$I = \frac{1}{m_r}\sum_{i=1}^{N-1}\sum_{i=j+1}^{N} m_i m_j r_{ij} \quad ; \quad m_r = \sum_i m_i$$

$$\theta_r = h^2/8\pi^2 Ik \quad ; \quad q_r = T/\sigma\,\theta_r$$

$$U_r = RT \quad ; \quad H_r = RT \quad ; \quad C_V = R \quad ; \quad C_p = R$$

$$S_r = R \ln q_r + U_r/T \quad ; \quad A_r = -RT \ln q_r \quad ; \quad G_r = A_r$$

For Vibrational Contributions

$$\theta_v = h\nu/k \quad ; \quad q_v = 1/[1 - \exp(-\theta_v/T)]$$

$$U_v = R\theta_v / [\exp(\theta_v/T) - 1] \quad ; \quad U_v' = U_v + \frac{1}{2} N_A h\nu$$

$$C_V = R(\theta_v/T)^2 \exp(\theta_v/T)/[\exp(\theta_v/T) - 1]$$

$$H_v = U_v \quad ; \quad C_p = C_V$$

$$S_v = R \ln q_v + U_v/T \quad ; \quad A_v = RT \ln [1 - \exp(-\theta_v/T)]$$

$$G_v = A_v.$$

Moment of Inertia for Nonlinear Molecules

The moment of inertia (I_xI_y) for a nonlinear molecule may be computed as follow.

$$A = \Sigma_i m_i y_i^2 - \frac{1}{M_r}(\Sigma_i m_i y_i)^2 \quad ; \quad B = \Sigma_i m_i x_i^2 - \frac{1}{M_r}(\Sigma_i m_i x_i)^2$$

$$C = \Sigma_i m_i (x_i^2 + y_i^2) - \frac{1}{M_r}(\Sigma_i m_i x_i)^2 - \frac{1}{M_r}(\Sigma_i m_i y_i)^2$$

$$D = \Sigma_i m_i x_i y_i - \frac{1}{M_r}(\Sigma_i m_i x_i)(\Sigma_i m_i y_i) \quad ; \quad E = 0 \text{ and } F = 0$$

where m_i's are relative atomic masses.

$$I_x I_y = \begin{vmatrix} A & -D & -E \\ -D & B & -F \\ -E & -F & C \end{vmatrix}$$

$$q_r = 8\pi^2 (I_x I_y)^{1/2} (2\pi kT)^{3/2}/\sigma h^3 \quad ; \quad U_r = H_r = 1.5\,RT \quad ; \quad C_p = C_V = 1.5\,R.$$

Rest of the expressions are the same as those of linear molecules.

Illustration Thermodynamic quantities of H_2O

Program

```
REM PROGRAM STTRI1;THERMODYNAMIC DATA FOR A TRIATOMIC GAS-TABULAR
CLS
REM M1 & M2 IN GRAMS;WN0 IN cm-1:DIS IN pm; M$ IS LINEAR OR NONLINEAR
REM OPTNU=0 FOR FREQUNCY,OTHERWISE 1 IF WAVENUMBER IN CM-1
R = 8.314: H = 6.626E-34: C = 3E+08: NA = 6.022E+23: KB = R / NA
READ M$, NAME$, OPTNU, T, P, SIGMA, G0, NC
PRINT M$, NAME$, T; "K": RMM = 0
IF M$ = "LINEAR" THEN NV = 4 ELSE NV = 3
FOR I = 1 TO NC: READ X(I), Y(I), RM(I)
RMM = RMM + RM(I)
NEXT I
RMMIN = 1 / RMM
FOR I = 1 TO NV: READ NU(I): NEXT I
IF OPTNU = 0 THEN 4
FOR I = 1 TO NV: NU(I) = NU(I) * C * 100: NEXT I
4 TERM1 = R * T / (P * 101325)
TERM2 = 2 * 3.14159 * 1.66 * RMM * (KB / 1E-23) * T
TERM2 = TERM2 / (H / 1E-34) ^ 2
QT = TERM1 * (TERM2 * 1E+18) ^ 1.5
UT = 1.5 * R * T: CVT = 1.5 * R
HT = 2.5 * R * T: CPT = 2.5 * R
ST = R * (-1.1541 + 1.5 * LOG(RMM) + 2.5 * LOG(T) - LOG(P))
'ST = R * (LOG(QT / NA) + 2.5)
AT = UT - T * ST: GT = HT - T * ST
IF M$ = "NONLINEAR" THEN 10
MI = 0
FOR I = 1 TO NC
FOR J = I + 1 TO NC
DIS = ((X(J) - X(I)) ^ 2 + (Y(J) - Y(I)) ^ 2) ^ .5: DIS = DIS * 1E-12
TERM = RM(I) * DIS ^ 2: MI = MI + TERM * RM(J)
NEXT J: NEXT I
MI = RMMIN * MI / NA
GOTO 20
10 SY = 0: SY2 = 0: SX = 0: SX2 = 0: SXY = 0: SX2Y2 = 0
FOR I = 1 TO NC
SY2 = SY2 + RM(I) * Y(I) ^ 2: SY = SY + RM(I) * Y(I)
SX2 = SX2 + RM(I) * X(I) ^ 2: SX = SX + RM(I) * X(I)
SX2Y2 = SX2Y2 + RM(I) * (X(I) ^ 2 + Y(I) ^ 2)
SXY = SXY + RM(I) * X(I) * Y(I)
NEXT I
```

```
AA = SY2 - RMMIN * SY ^ 2: BB = SX2 - RMMIN * SX ^ 2
CC = SX2Y2 - RMMIN * (SX ^ 2 + SY ^ 2)
DD = SXY - RMMIN * SX * SY: EE = 0: FF = 0
MI = AA * (BB * CC - FF ^ 2) + DD * (-DD * CC - EE * FF)
MI = MI - EE * (DD * FF + BB * EE): MI = MI * 4.57E-12
GOTO 30
20 THETR = H ^ 2 / (8 * 3.14159 ^ 2 * MI * KB)
THETR = THETR / .001
IF M$ = "NONLINEAR" THEN 30
QR = T / (SIGMA * THETR)
UR = R * T: CVR = R: HR = R * T: CPR = R
GOTO 40
30 QR = 8 * 3.14159 ^ 2 * MI ^ .5
QR = QR * (2 * 3.14159 * (KB / 1E-23) * T) ^ 1.5
QR = QR / (SIGMA * (H / 1E-34) ^ 3)
QR = QR * .01 ^ 1.5: UR = 1.5 * R * T: CVR = 1.5 * R
HR = UR: CPR = CVR
40 SR = R * LOG(QR) + UR / T
AR = -R * T * LOG(QR): GR = AR
FOR I = 1 TO NV
THETV = H * NU(I) / KB: THETVT = THETV / T
EXTHET = EXP(THETVT) - 1: QV = 1 / (1 - EXP(-THETVT))
UV(I) = R * THETV / EXTHET: UVP(I) = UV(I) + .5 * NA * H * NU(I)
CVV(I) = R * THETVT ^ 2 * EXP(THETVT) / EXTHET ^ 2
HV(I) = UV(I): CPV(I) = CVV(I): SV(I) = R * LOG(QV) + UV(I) / T
AV(I) = R * T * LOG(1 - EXP(-THETVT)): GV(I) = AV(I)
NEXT I
UM = UT + UR: UMP = UT + UR: CVM = CVT + CVR
HM = HT + HR: CPM = CPT + CPR
SM = ST + SR: AM = AT + AR: GM = GT + GR
FOR I = 1 TO NV
UM = UM + UV(I): CVM = CVM + CVV(I): UMP = UMP + UVP(I)
HM = HM + HV(I): CPM = CPM + CPV(I)
SM = SM + SV(I): AM = AM + AV(I): GM = GM + GV(I)
NEXT I
PRINT STRING$(79, "-")
PRINT TAB(18); "TRANS    ROT           VIBRATIONS             ELEC   TOTAL"
PRINT STRING$(79, "-")
PRINT "U/(J/mol)       "; : PRINT USING "######.#"; UT; UR;
FOR I = 1 TO NV: PRINT USING " ####.##"; UV(I); : NEXT I
PRINT USING "###.##"; UE; : PRINT USING " #####.##"; UM
'PRINT TAB(40); : FOR I = 1 TO NV
'PRINT USING " #####.##"; UVP(I); : NEXT I
```

```
'PRINT USING "        #####.##"; UMP
'PRINT : PRINT "U is in J/mol": PRINT
'PRINT "UT="; UT, "UR="; UR
'FOR I = 1 TO NV: PRINT "UV("; I; ")="; UV(I), : NEXT I
'PRINT "UE="; UE, "U="; UM
'FOR I = 1 TO NV: PRINT "UVP("; I; ")="; UVP(I), : NEXT I
'PRINT "UP="; UMP: A$ = INPUT$(1)
PRINT "Cv/(J/(K mol))"; : PRINT USING " ####.##"; CVT; CVR;
FOR I = 1 TO NV: PRINT USING "   ###.##"; CVV(I); : NEXT I
PRINT USING "###.##"; CVE; : PRINT USING " #####.##"; CVM
'PRINT : PRINT "CV is in J/(K mol)": PRINT
'PRINT "CVT="; CVT, "CVR="; CVR
'FOR I = 1 TO NV: PRINT "CVV("; I; ")="; CVV(I), : NEXT I
'PRINT "CVE="; CVE, "CV="; CVM: A$ = INPUT$(1)
PRINT "H/(J/mol)       "; : PRINT USING "######.#"; HT; HR;
FOR I = 1 TO NV: PRINT USING " ####.##"; HV(I); : NEXT I
PRINT USING "###.##"; HE; : PRINT USING "######.##"; HM
'PRINT : PRINT "H is in J/mol": PRINT
'PRINT "HT="; HT, "HR="; HR
'FOR I = 1 TO NV: PRINT "HV("; I; ")="; HV(I), : NEXT I
'PRINT "HE="; HE, "H="; HM: A$ = INPUT$(1)
'PRINT : PRINT "CP is in J/(K mol)": PRINT
PRINT "Cp/(J/(K mol))"; : PRINT USING " ####.##"; CPT; CPR;
FOR I = 1 TO NV: PRINT USING "   ###.##"; CPV(I); : NEXT I
PRINT USING "###.##"; CPE; : PRINT USING " #####.##"; CPM
'PRINT "CPT="; CPT, "CPR="; CPR
'FOR I = 1 TO NV: PRINT "CPV("; I; ")="; CPV(I), : NEXT I
'PRINT "CPE="; CPE, "CP="; CPM: A$ = INPUT$(1)
PRINT "S/(J/(K mol)    "; : PRINT USING "#####.##"; ST; SR;
FOR I = 1 TO NV: PRINT USING "   ###.##"; SV(I); : NEXT I
PRINT USING "###.##"; SE; : PRINT USING "######.##"; SM
'PRINT : PRINT "S is in J/(K mol)": PRINT
'PRINT "ST="; ST, "SR="; SR
'FOR I = 1 TO NV: PRINT "SV("; I; ")="; SV(I), : NEXT I
'PRINT "SE="; SE, "S="; SM: A$ = INPUT$(1)
PRINT "A/(J/mol)     "; : PRINT USING "#######.#"; AT; AR;
FOR I = 1 TO NV: PRINT USING " #####.#"; AV(I); : NEXT I
PRINT USING "##.##"; AE; : PRINT USING "#######.#"; AM
'PRINT : PRINT "A is in J/mol": PRINT
'PRINT "AT="; AT, "AR="; AR
'FOR I = 1 TO NV: PRINT "AV("; I; ")="; AV(I), : NEXT I
'PRINT "AE="; AE, "A="; AM: A$ = INPUT$(1)
PRINT "G/(J/mol)     "; : PRINT USING "#######.#"; GT;
```

```
PRINT USING " ######.#"; GR;
FOR I = 1 TO NV: PRINT USING " #####.#"; GV(I); : NEXT I
PRINT USING "##.##"; GE; : PRINT USING "#######.#"; GM
'PRINT : PRINT "G is in J/mol": PRINT
'PRINT "GT="; GT, "GR="; GR
'FOR I = 1 TO NV: PRINT "GV("; I; ")="; GV(I), : NEXT I
'PRINT "GE="; GE, "G="; GM: A$ = INPUT$(1)
PRINT STRING$(79, "-")
DATA NONLINEAR,H2O,0
DATA 298,1,2,1,3
DATA 0,0,16,-75.75,58.65,1,75.75,58.65,1
DATA 1.1E+14,4.78E+13,1.13E+14
'DATA LINEAR,CO2,0
'DATA 1200,1,2,1,3
'DATA 0,0,16,116.2,0,12,232.4,0,16
'DATA 4.03E+13,2E+13,2E+13,7.05E+13
'DATA LINEAR,N2O,1
'DATA 298,1,1,3,3
'DATA 0,0,14,112.82,0,14,231.24,0,16
'DATA 1276.5,589.2,589.2,2223.7
END
```

Output Table

HCl T= 298 K

	TRANSLATIONAL	ROTATIONAL	VIBRATIONAL	ELECTRONIC	TOTAL
U/(J/mol)	3716.358	2477.572	0.019	0.000	6193.949
Cv/(J/(K mol))	12.471	8.314	0.001	0.000	20.786
H/(J/mol)	6193.930	2477.572	0.019	0.000	8671.521
Cp/(J/(K mol))	20.785	8.314	0.001	0.000	29.100
S/(J/(K mol))	153.509	33.095	0.000	0.000	186.604
A/(J/mol)	3716.358	-7384.639	-0.001	0.000	-3668.282
G/(J/mol)	-39551.742	-7384.639	-0.001	0.000	-46936.383

Exercise

Execute the program for CO_2 and N_2O for which data is provided in the program.

2.26 EVALUATION OF EQUILIBRIUM CONSTANT OF $N_2(g) \rightleftharpoons 2N(g)$

To determine the equilibrium constant of the reaction $N_2(g) \rightleftharpoons 2N(g)$ at 5000 K. Given:

$$r_{eq}(N_2) = 110 \text{ pm}, \qquad \nu(N_2) = 7.07 \times 10^{13} \text{ s}^{-1}$$

$D(N_2)$ = 940.3 kJ mol^{-1}.

Degeneracy of ground electronic level of N_2 is 1 and that of N is 4.

The expression to be used is

$$K_p^\circ = \left[\left(\frac{kT}{p^\circ}\right)^{\Sigma\nu_i}\right]\left[e^{-\Delta_t U_0/RT}\right]\left[\prod_i \left(\frac{q_i}{V}\right)^{V_i}\right]$$

For N $\quad q = q_t\, q_e$ where $\frac{q_t}{V} = \left(\frac{2\pi mkT}{h^2}\right)^{3/2}$ and $q_e = g_0 = 4$

For N_2 $\quad q = q_t\, q_r\, q_v\, q_e$

where $\frac{q_t}{V} = \left(\frac{2\pi mkT}{h^2}\right)^{3/2}$; $q_r = \left(\frac{8\pi^2 Ik}{h^2}\right)\frac{I}{\sigma}$; $I = \mu r^2$

$q_v = \frac{1}{1 - e^{-\theta_v/T}}$; $\theta_v = \frac{h\nu}{k}$ and $q_e = g_e = 1$

Program

```
REM PROG25
REM PROGRAM STEQUIL;EQUILIBRIUM CONSTANT STATISTICALLY
REM DIS IN UNIT OF pm
CLS : R = 8.314: KB = 1.38: H = 6.626: C = 3: AU = 1.66
READ NC, T, P
FOR I = 1 TO NC
READ M(I), AT(I), GE(I), NU(I)
IF AT(I) = 1 THEN 5
READ M1(I), M2(I), SIGMA(I), NU0(I), DIS(I), DISE(I)
5 NEXT I
KP = 1
FOR I = 1 TO NC
QT = ((2 * 3.14159 * M(I) * AU * KB * T / H ^ 2) * 1E+18) ^ 1.5
ON AT(I) GOTO 10, 40
10 QE = GE(I): Q(I) = QT * QE
GOTO 80
40 MU = AU * M1(I) * M2(I) / (M1(I) + M2(I))
MI = MU * DIS(I) ^ 2
QR = (8 * 3.14159 ^ 2 * MI * KB / H ^ 2) * T / SIGMA(I)
QR = QR * .000001: THETAV = (H * NU0(I) / KB) * 1E-11
QV = 1 / (1 - EXP(-THETAV / T)): QE = GE(I)
Q(I) = QT * QR * QV * QE
```

```
80 KP = KP * (Q(I) * KB * T / 9.999999E+27) ^ NU(I)
RE = RE + NU(I) * DISE(I)
NEXT I
KP = KP * EXP(RE / (R * T))
PRINT "REACTION: N2=2N", : PRINT "   Kp="; KP
DATA 2,5000,100000,28,2,1,-1,14,14,2,7.07E13,110,9.403E5,14,1,4,2
END
```

Output `REACTION: N2=2N  Kp= 1.15073E-03`

2.27 EVALUATION OF RATE CONSTANT OF $H + HBr \rightarrow H_2 + Br$

To determine the rate constant of the reaction $H + HBr \rightarrow H_2 + Br$ at 300 K. Given:

Barrier height from zero-point level = 5.0 kJ mol^{-1}; r(H – Br) = 141.4 pm ; ν(H – Br) = 7.95×10^3 Hz

Activated complex is linear $\quad H \underset{150\ pm}{\text{———}} H \underset{142\ pm}{\text{———}} Br$

Its $\quad \nu$ (symmetrical stretch) = 7.02×10^{13} Hz

ν (bending mode) = 1.38×10^{13} Hz (doubly degenerate)

Electronic contribution is negligible.

The expression to be used is

$$k = \left(\frac{RT}{h}\right)(e^{-\Delta_r U/RT})\left\{\frac{q_{\neq}/V}{(q_H/V)(q_{HBr}/V)}\right\}$$

$$\frac{q_H}{V} = \frac{q_t}{V} = \left(\frac{2\pi mkT}{h^2}\right)^{3/2}$$

$$\frac{q_{HBr}}{V} = \left(\frac{q_t}{V}\right) q_r\, q_v \qquad \text{where} \qquad \left(\frac{q_t}{V}\right) = \left(\frac{2\pi mkT}{h^2}\right)^{3/2}$$

$$q_r = \frac{8\pi^2 IkT}{h^2} \qquad ; \qquad I = \mu r^2$$

$$q_v = \frac{1}{1 - \exp(\theta_v / T)} \qquad ; \qquad \theta_v = h\nu/k$$

$$\left(\frac{q_{\neq}}{V}\right) = \left(\frac{q_t}{V}\right)(q_r\, q_{v1}\, q_{v2}\, q_{v3})$$

where $\left(\frac{q_t}{V}\right)=\left(\frac{2\pi mkT}{h^2}\right)^{3/2}$

$q_r = \frac{8\pi^2 IkT}{h^2}$; $I = \Sigma_i m_i r_i^2$; (r_i from the centre of mass)

$q_{v1} = \frac{1}{1-\exp(-\theta_1/T)}$; $\theta_1 = h\nu_1/k$

$q_{v2} = \frac{1}{1-\exp(-\theta_2/T)}$; $\theta_2 = h\nu_2/k.$

Program

```
REM PROG26   H+HBr --> H2+Br
REM PROGRAM STTST;RATE OF REACTION STATISTICALLY
CLS
REM M1 & M2 ARE RELATIVE MOLAR MASSES; DIS IN pm
REM WN0 IN cm-1 OTHERWISE IT REPRESENTS FREQUENCY
R = 8.314: H = 6.626: C = 3: NA = 6.022: KB = 1.38: AU = 1.67
PI = 3.14159: READ NC, T, P, BH
FOR I = 1 TO NC: READ NAME$(I), M(I), AT(I), NU(I)
QT = ((2 * PI * M(I) * AU * KB * T / H ^ 2) * 1E+18) ^ 1.5
ON AT(I) GOTO 5, 10, 15
5 Q(I) = QT: GOTO 60
10 READ M1(I), M2(I), SIGMA(I), NU0(I), DIS(I)
MU = AU * M1(I) * M2(I) / (M1(I) + M2(I)): MI = MU * DIS(I) ^ 2
QR = (8 * PI ^ 2 * MI * KB / H ^ 2) * T / SIGMA(I)
QR = QR * .000001: THETAV = (H * NU0(I) / KB) * 1E-11
QV = 1 / (1 - EXP(-THETAV / T)): Q(I) = QT * QR * QV
GOTO 60
15 READ M$(I), SIGMA(I)
IF M$(I) = "LINEAR" THEN NV = 3 ELSE NV = 2
FOR J = 1 TO NC: READ X(J), Y(J), RM(J)
RMM = RMM + RM(J): NEXT J
RMMIN = 1 / RMM
FOR J = 1 TO NV: READ NU0(J): NEXT J
'FOR J = 1 TO NV: READ NU(J): NU0(J) = NU(J) * C * 100: NEXT J
IF M$ = "NONLINEAR" THEN 18
MI = 0
FOR J = 1 TO NC - 1: FOR K = J + 1 TO NC
DIS = ((X(K) - X(J)) ^ 2 + (Y(K) - Y(J)) ^ 2) ^ .5
```

```
DIS = DIS * 1E-12: TERM = RM(K) * DIS ^ 2: MI = MI + TERM
NEXT K
MI = MI * RM(J)
NEXT J
MI = RMMIN * MI / NA
GOTO 20
18 SY = 0: SY2 = 0: SX = 0: SX2 = 0: SXY = 0: SX2Y2 = 0
FOR J = 1 TO NC
SY2 = SY2 + RM(J) * Y(J) ^ 2: SY = SY + RM(J) * Y(J)
SX2 = SX2 + RM(J) * X(J) ^ 2: SX = SX + RM(J) * X(J)
SX2Y2 = SX2Y2 + RM(J) * (X(J) ^ 2 + Y(J) ^ 2)
SXY = SXY + RM(J) * X(J) * Y(J)
NEXT J
AA = SY2 - RMMIN * SY ^ 2: BB = SX2 - RMMIN * SX ^ 2
CC = SX2Y2 - RMMIN * (SX ^ 2 + SY ^ 2)
DD = SXY - RMMIN * SX * SY: EE = 0: FF = 0
MI = AA * (BB * CC - FF ^ 2) + DD * (-DD * CC - EE * FF)
MI = MI - EE * (DD * FF + BB * EE): MI = MI * 4.57E-12
GOTO 30
20 THETR = (H ^ 2 / (8 * PI ^ 2 * MI * KB)) * 1E-19
IF M$ = "NONLINEAR" THEN 30
QR = T / (SIGMA(I) * THETR): GOTO 40
30 QR = 8 * PI ^ 2 * MI ^ .5 * (2 * PI * (KB / 1E-23) * T) ^ 1.5
QR = QR / (SIGMA * (H / 1E-34) ^ 3): QR = QR * .01 ^ 1.5
40 Q(I) = QT * QR
FOR J = 1 TO NV
THETV = H * 1E-34 * NU0(J) / (KB * 1E-23)
THETVT = THETV / T: EXTHET = EXP(THETVT) - 1
QV = 1 / (1 - EXP(-THETVT)): Q(I) = Q(I) * QV
NEXT J
60 PRINT NAME$(I), "Q("; I; ")="; Q(I)
NEXT I
RATE = (R * T / (H * 1E-34)) * EXP(-BH / (R * T))
FOR I = 1 TO NC: RATE = RATE * Q(I) ^ NU(I): NEXT I
PRINT "RATE OF REACTION="; : PRINT USING "##.####^^^^"; RATE;
PRINT "  mol-1 m3 s-1"
DATA 3,300,1,5000
DATA H,1,1,-1
DATA HBr,80.9,2,-1
```

```
DATA 1,79.9,1,7.95E13,141.4
DATA HHBr,81.9,3,1
DATA LINEAR,1
DATA 0,0,1,150,0,1,292,0,79.9
DATA 7.02E13,1.38E13,1.38E13
END
```

Output

```
H              Q( 1 )= 9.842154E+29
HBr            Q( 2 )= 1.758386E+34
HHBr           Q( 3 )= 1.173858E+35
RATE OF REACTION= 3.4394E+06  mol-1 m3 s-1
```

2.28 LEAST SQUARE FITTING FOR $y = a_0 + a_1x$

To carry out the least square fitting of the given data points (x_i, y_i) as per the equation $y = a_0 + a_1x$.

Calculate the standard deviation of computed y_i's from the given y_i's.

Also display the regression line between computed y versus x_i.

The straight line $y = a_0 + a_1x$ is fitted through the given points (x_1, y_1), (x_2, y_2), . . . , (x_n, y_n) so that the sum of squares of $(y_{\text{computed}} - y_{\text{given}})$ is minimum. If the minimum is represented as

$$q = \sum_{i=1}^{N} (y_i - a_0 - a_1x_i)^2$$

then, the condition for q to be minimum is

$$\frac{\partial q}{\partial a_0} = -2\sum_{i=1}^{N} (y_i - a_0 - a_1x_i) = 0 \quad \text{and} \quad \frac{\partial q}{\partial a_1} = -2\sum_{i=1}^{N} x_i(y_i - a_0 - a_1x_i) = 0$$

Rearranging these expressions, we get

$$a_0 N + a_1 \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i \quad \text{and} \quad a_0 \sum_{i=1}^{N} x_i + a_1 \sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} x_i y_i$$

These give

$$a_0 = \frac{\left(\sum_{i=1}^{N} x_i^2\right)\left(\sum_{i=1}^{N} y_i\right) - \left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} x_i y_i\right)}{N\left(\sum_{i=1}^{N} x_1^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2} \quad \text{and} \quad a_1 = \frac{N\left(\sum_{i=1}^{N} x_i y_i\right) - \left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} y_i\right)}{N\left(\sum_{i=1}^{N} x_1^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2}$$

The equation of the type $y = a\,e^{bx}$ can be converted into linear equation by taking its natural logartihm:

$$\ln\, y = \ln\, a + bx$$

Hence, the parameters a and b can be computed by carrying out the least square fitting between ln y and x.

In the following program, the values of x and y are suitably changed by defining the functions FNX and FNY. Suppose the regression line to be fitted is

$$\ln\,(p/\text{kPa}) = -\,\frac{\Delta_{\text{vap}}H_{\text{m}}}{R}\,\frac{1}{T} + I$$

then define FNX (X) = 1/X and FNY (Y) = LOG(Y)

Illustration Least square fitting for the data of Exercise 18 given after the figure in the output.

Program

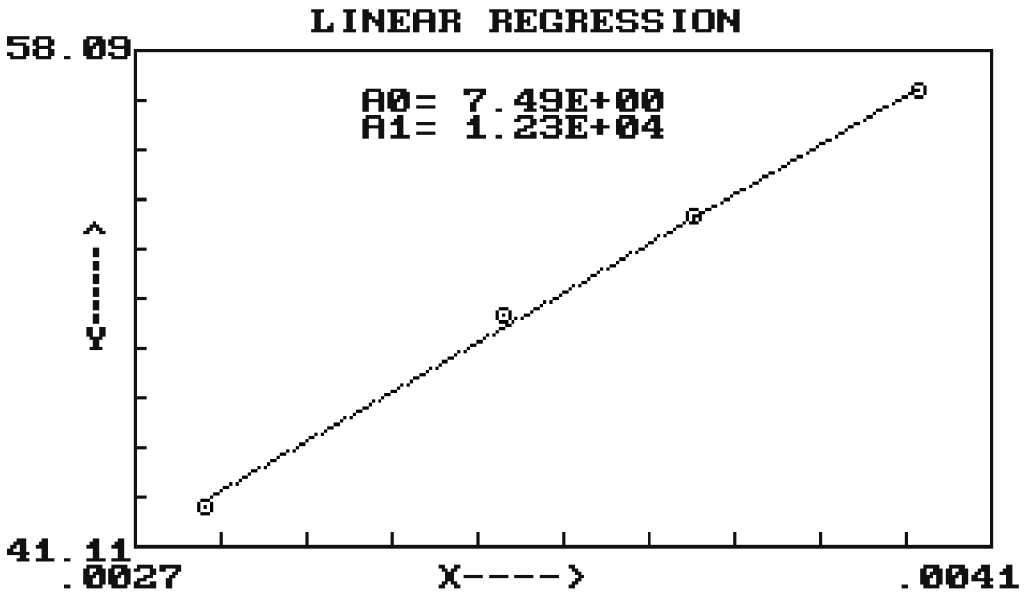
```
REM PROG27
REM PROGRAM LSFLIN;LEAST SQUARE FITTING-LINEAR
CLS : DIM XX(20), YY(20), X(20), Y(20), YCAL(20), YER(20)
REM ------define functions according to the requirement-------
DEF FNX (X) = X: DEF FNY (Y) = (Y - 1) / (Y + 2) * V * 17
READ N: SX = 0: SY = 0: SX2 = 0: SXY = 0: SD = 0
FOR I = 1 TO N
READ XX(I), YY(I), V
X(I) = FNX(1 / XX(I)): Y(I) = FNY(YY(I))
SX = SX + X(I): SX2 = SX2 + X(I) * X(I)
SY = SY + Y(I): SXY = SXY + X(I) * Y(I)
NEXT I
DEN = SX * SX - N * SX2
a0 = (SX * SXY - SX2 * SY) / DEN
a1 = (SX * SY - N * SXY) / DEN
FOR I = 1 TO N
YCAL(I) = a0 + a1 * X(I): YER(I) = YCAL(I) - Y(I)
SD = SD + YER(I) * YER(I)
NEXT I
SD = (SD / N) ^ .5
PRINT "    Y="; : PRINT USING "##.##^^^^"; a0;
PRINT "+("; : PRINT USING "##.##^^^^"; a1; : PRINT ")*X"
PRINT STRING$(45, "-")
PRINT "         X             Y          YCAL          YER"
PRINT STRING$(45, "-")
FOR I = 1 TO N
PRINT USING "   ####.####"; X(I); Y(I); YCAL(I); YER(I)
NEXT I
PRINT STRING$(45, "-"): PRINT
```

```
PRINT "STANDARD DEVIATION="; SD
a$ = INPUT$(1)
XMIN = X(1): YMIN = Y(1): XMAX = X(N): YMAX = Y(N)
FOR I = 1 TO N
IF XMIN > X(I) THEN XMIN = X(I)
IF XMAX < X(I) THEN XMAX = X(I)
IF YMIN > Y(I) THEN YMIN = Y(I)
IF YMAX < Y(I) THEN YMAX = Y(I)
NEXT I
XD = (XMAX - XMIN) / 10: YD = (YMAX - YMIN) / 10
XI = XMIN - XD: XF = XMAX + XD: YI = YMIN - YD: YF = YMAX + YD
SCREEN 1: COLOR 15, 0
VIEW (40, 20)-(310, 170)
WINDOW (XI, YI)-(XF, YF): LINE (XI, YI)-(XF, YF), , B
FOR I = 1 TO N
PSET (X(I), Y(I)), 2: CIRCLE (X(I), Y(I)), XMAX / 400, 2
NEXT I
FOR X1 = XMIN TO XMAX STEP XD / 20
Y1 = a0 + a1 * X1
PSET (X1, Y1), 2
NEXT X1
XD = (XF - XI) / 10: YD = (YF - YI) / 10
FOR I = 1 TO 9
LINE (XI + XD * I, YI)-(XI + XD * I, YI + YD / 4)
LINE (XI, YI + YD * I)-(XI + XD / 8, YI + YD * I)
NEXT I
LOCATE 2, 13: PRINT "LINEAR REGRESSION"
LOCATE 3, 1: PRINT USING "##.##"; YF
LOCATE 5, 15: PRINT "A0="; : PRINT USING "##.##^^^^"; a0
LOCATE 6, 15: PRINT "A1="; : PRINT USING "##.##^^^^"; a1
LOCATE 10, 4: PRINT CHR$(94)
FOR I = 1 TO 3: LOCATE 10 + I, 4: PRINT CHR$(124): NEXT I
LOCATE 14, 4: PRINT "Y"
LOCATE 22, 1: PRINT USING "##.##"; YI
LOCATE 23, 3: PRINT USING ".####"; XI;
PRINT "          X----->              "; : PRINT USING ".####"; XF
REM KINETICS;PROBLEM 11;FNX(X)=X and FNY(Y)=Y
'DATA 7,0,-.602,200,-.652,400,-.703,600,-.759,800,-.818
'DATA 1000,-.873,1200,-.921
REM KINETICS;PROBLEM 13;FNX(X)=X and FNY(Y)=Y
'DATA 8,0,2.23,3,2.20,6,2.18,9,2.15,12,2.13
'DATA 15,2.1,18,2.08,21,2.05
REM CONDUCTANCE;PROBLEM 3
```

```
'DATA   6,16,53.1,32,72.4,64,96.8
'DATA 128,127.7,256,164,512,205.8
DATA 4,250,1.0085,1180,275,1.0071,1305,300,1.0061,1420
DATA 350,1.0045,1670: REM dipole moment;problem 18
END
```

Output

```
   Y= 7.49E+00+( 1.23E+04)*X
-------------------------------------------
      X          Y        YCAL        YER
-------------------------------------------
    0.0040    56.6760    56.8445     0.1685
    0.0036    52.3804    52.3576    -0.0228
    0.0033    48.9855    48.6185    -0.3670
    0.0029    42.5215    42.7428     0.2213
-------------------------------------------

STANDARD DEVIATION= .2305453
```



Exercises for Least Square Fitting

1. The vapour pressure of *n*-propyl alcohol varies with temperature as follows.

t/°C	50	60	70	80
p/kPa	11.626	19.600	31.864	50.129

Carry on the least-square fitting for a straight line

$$\ln\,(p/\text{kPa}) = -\,\frac{\Delta_{\text{vap}}H_{\text{m}}}{R}\,\frac{1}{T} + I$$

to determine the value of $\Delta_{vap}H_m$, where T/K = t/°C + 273.

Here y = ln (p/kPa). Thus FNY (Y) = LOG (Y)

$X = 1/T$. Thus FNX (X) = 1/ (X + 273)

From $a_1 = -\Delta_{vap}H_m/R$, calculate $\Delta_{vap}H_m$. (Ans. 46.16 kJ mol^{-1})

2. The viscosity of diethyl ether varies with temperature as follows.

t/°C	0	20	40
η/millipoise	2.84	2.33	1.97

Determine the value of activation energy of viscous flow through a least square fitting for the equation

$$\ln(\eta/\eta^\circ) = \ln(A/\eta^\circ) + \frac{E}{RT}$$

Here y = ln (η/η°). Thus FNY (Y) = LOG (Y)

$x = 1/T$. Thus FNX (x) = 1/ (x + 273) (Ans. 6.5 kJ mol^1)

3. The following are the conductivities of chloroacetic acid in aqueous solution at 25 °C.

mol dm^{-3}/c	16	32	64	128	256	512
Λ_m / S cm^2 mol^{-1}	53.1	72.4	96.8	127.7	164	205.8

Also given, Λ^∞ = 362 S cm^2 mol^{-1}. Determine the value of K_a of the acid through the least square fitting for the equation

$$\frac{\Lambda_c^2}{\Lambda^\infty(\Lambda^\infty - \Lambda_c)} = \frac{K_a}{c}$$

Here $X = 1/c$. Thus FNX (X) = X

$Y = \dfrac{\Lambda_c^2}{\Lambda^\infty(\Lambda^\infty - \Lambda_c)}$. Thus FNY (Y) = Y ^ 2/ (362 * (362 – Y)) (Ans: 1.46 × 10^{-3} M)

4. The following emfs refer to the cell at 298 K.

Pt | H_2(1 bar) | LiOH(0.01 M), LiCl(m) | AgCl(s) | Ag

m/mol dm^{-3}	0.01	0.02	0.05	0.10	0.20
E/V	1.049 5	1.031 5	1.007 3	0.988 5	0.969 4
μ/mol dm^{-3}	0.02	0.03	0.06	0.11	0.21

Let $y = \dfrac{E_{cell} - 0.225\,V}{0.059\,13\,V} + \log\left(\dfrac{m}{0.01\,\text{mol dm}^{-3}}\right)$

and $x = \mu$

Carry out the least square fitting of the given data points for the equation $y = a_0 + a_1x$.

If a_0 = –log K_w, what is the value of pK_w?

5. Given are the following data for the cell Pt | H_2(g, 1 bar) | HCl (m) | AgCl(s) |Ag

m/mol kg^{-1}	0.009	0.014	0.025	0.055
E/V	0.469 5	0.447 8	0.419 6	0.381 2

Let $y = E + (0.011\,83\ \text{V})\ \log(m/\text{mol kg}^{-1}) - (0.06\text{V})\ \sqrt{m/\text{mol kg}^{-1}}$

and $x = m/\text{mol kg}^{-1}$

Carry out the least square fitting of the given data for the equation $y = a_0 + a_1 x$.

If $a_0 = E^{\circ}_{\text{Cl}^-|\text{AgCl|Ag}}$, what is its value? (Ans. 0.225 V)

6. For the cell $\text{Pt} \mid \text{H}_2(\text{g, 1 bar}) \mid \text{NaOH(1 M), NaCl(1 mol dm}^{-3}) \mid \text{AgCl(s)} \mid \text{Ag}$, the potential at different temperatures are

$t/°\text{C}$	20	25	30
$E°/\text{V}$	1.047 68	1.051 88	1.056 08

Carry out the least square fitting of $E°$ versus T for the equation $E° = a_0 + a_1 T$. Obviously, $a_1 = (\partial E°/\partial T)_p$. Using this value, calculate

$\Delta H°\ (= -nF\,[E° - T(\partial E°/\partial T)_p])$, $\Delta S°\ (= nF(\partial E°/\partial T)_p)$, and $\Delta G°\ (= \Delta H° - T\Delta S°)$.

Also determine K°_{eq} by using the expression $\Delta G° = -RT \ln K^{\circ}_{\text{eq}}$.

7. The data below are for the adsorption of CO on charcoal at 273 K.

p / mmHg	100	200	300	400	500	600	700
υ / cm^3	10.2	18.6	25.5	31.4	36.9	41.6	46.1

Carry out the least square filting of p/υ with p as per the equation $y = a_0 + a_1 x$, where $y = p/\upsilon$ and $x = p$. If $a_1 = 1/\upsilon_{\text{mono}}$ and $a_0 = 1/K\upsilon_{\text{mono}}$, calculate the value of υ_{mono} and K.

(Ans. $\upsilon_{\text{mono}} = 111\ \text{cm}^3$, $K = 0.76\ \text{atm}^{-1}$)

8. The data below are for the adsorption of N_2.

p/p_0	0.05	0.10	0.15	0.20	0.25
υ/cm^3	51.3	58.8	64.0	68.9	74.2

Carry out the least square fitting for the equation $y = a_0 + a_1 x$

where $y = \dfrac{1}{\upsilon\{(p_0/p) - 1\}}$ and $x = \dfrac{p}{p_0}$

If $a_0 = 1/\upsilon_{\text{mono}}C$ and $a_1 = (C - 1)/C\ \upsilon_{\text{mono}}$, calculate the values of C and υ_{mono}.

(Ans. $C = 116.3$, $\upsilon_{\text{mono}} = 57.32\ \text{cm}^3$)

9. The data below are for the adsorption of N_2.

p/p_0	0.05	0.10	0.15	0.20	0.25
υ/cm^3	51.3	58.8	640	68.9	74.2

Carry out the least square fitting for the equation $y = a_0 + a_1 x$ where $y = \log(p/p_0)$ and $x = 1/\upsilon^2$. If σ (area per gram of adsorbent) is related to the constant a_1 by the expression $\sigma = (4.06 \times 10^6\ \text{m}^{-1})\ (-a_1)^{1/2}$, calculate the value of σ. (Ans. $\sigma = 241.8\ \text{m}^2/\text{g}$)

10. The data below show the pressure of CO required for the volume of adsorption to be 10.0 cm^3 at each temperature.

T/K	200	210	220	230	240	250
p/p°	30.0	37.1	45.2	54.0	63.5	73.9

Carry out the least square fitting as per the equation $y = a_0 + a_1 x$ where $y = \log (p/p^\circ)$ and $x = 1/T$. If $a_1 = \Delta_{ads} H/2303\,R$, determine the value of $\Delta_{ads}H$. (Ans. –7 563 J mol^{-1})

11. The following data were obtained for the decomposition of N_2O_5(g).

t/s	0	200	400	600	800	1000	1200
$\frac{[N_2O_5]}{\text{mol dm}^{-3}}$	0.250	0.223	0.198	0.174	0.152	0.134	0.120

Carry out the least square fitting for the equation $y = a_0 + a_1 x$, where $y = \log \{[N_2O_5]/\text{mol dm}^{-3}\}$ and $x = t$/s.

If $a_1 = -k/2.303$, determine the value of rate constant k. (Ans. 6.51×10^4 s^{-1})

12. The following data were obtained on the optical activity of a solution.

t/min	0.0	7.2	36.8	46.8	68.0	∞
θ/degree	24.1	21.4	12.4	10.0	5.5	– 10.7

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = \log \{(\theta_\infty - \theta_t)/ (\theta_\infty - \theta_0)\}$ and $x = t$/min.

If $a_1 = -k/2.303$, determine the value of rate constant. (Ans. 0.012 min^{-1})

13. The following data were obtained for the decomposition of di-tertiarybutyl peroxide in the gas phase at constant volume.

t/min	0	3	6	9	12	15	18	21
p/Torr	169.3	189.2	207.1	224.4	240.2	256.0	265.7	282.6

Carry out the least square fitting as per the equation $y = a_0 + a_1 x$.

where $y = \log (p_t/\text{Torr})$ with $p_t = \frac{3p_0 - p}{2}$ and $x = t$/min

If $a_1 = -k/2.303$, determine the value of rate constant. (Ans. 0.019 3 min^{-1})

14. The following data were obtained for the dimerization of butadiene.

t/min	0	6.12	12.18	17.30	29.18
p_t/Torr	632	606.6	584.2	567.3	535.4
	42.5	60.87	90.05	119.0	176.67
	509.3	482.8	453.3	432.8	405.3

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = 1/(2\,p_t - p_0)$ and $x = t$/min.

If $a_1 = k$, find the value of rate constant k. (Ans. 1.78×10^{-5} Torr^{-1} min^{-1})

15. The following data were obtained in an enzyme catalysed reaction.

$[S] \times 10^4$/mol dm^{-3}	2.5	5.0	10.0	15.0
$r_0 \times 10^6$/mol dm^{-3} min^{-1}	2.2	3.8	5.9	7.1

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = (1/r_0) \times 10^{-6}/\text{mol}^{-1}\ \text{dm}^3\ \text{min}$ and $x = (1/[\text{S}]_0) \times 10^{-4}/\text{mol}^{-1}\ \text{dm}^3$

If $a_0 = K_\text{M}/r_\text{max}$ and $a_1 = 1/r_\text{max}$. Determine the value of K_M and r_max.

(Ans. 1.19×10^{-3} mol dm^{-3}, 1.25×10^{-5} mol dm^{-3} min^{-1})

16. The following data were obtained in an enzyme catalysed reaction.

$[\text{S}] \times 10^4/\text{mol dm}^{-3}$	2.5	5.0	10.0	15.0
$r_0 \times 10^6/\text{mol dm}^{-3}\ \text{min}^{-1}$	2.2	3.8	5.9	7.1

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = (r_0/[\text{S}]_0) \times 10^2/\text{min}^{-1}$ and $x = r_0 \times 10^6/\text{mol dm}^{-3}\ \text{min}^{-1}$

Find x when $y = 0$ and y when $x = 0$.

17. For the decomposition of N_2O_5

T/K	298	308	318	328	338
$10^5\ k_1/\text{s}^{-1}$	1.72	6.65	24.95	75	240

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = \ln k_1$ and $x = 1/T$. If $k_1 = A \exp(-E/RT)$, find the value of A and E.

Hint The equation $k_1 = A \exp(-E/RT)$ is equivalent to $\ln k_1 = \ln A - E/RT$. Hence

$a_0 = \ln A$ and $a_1 = -E/R$.

18. The following data have been reported for NH_3(g).

T/K	250	275	300	350
Relative Permittivity, ε_r	1.008 5	1.007 1	1.006 1	1.004 5
Specific volume/cm^3 g^{-1}	1 180	1 305	1 420	1 670

Calculate the dipole moment of the NH_3 molecule. Carry out the least square fitting for the equation

$$y = a_0 + a_1 x$$

where $y = \dfrac{\varepsilon_\text{r} - 1}{\varepsilon_\text{r} + 2} V_\text{m}$; with $V_\text{m} = V_\text{specific} \times M_{NH_3}$ and $x = 1/T$

From the slope, calculate dipole moment by using the expression

$$p = \left[\frac{9\varepsilon_0 k(\text{slope})}{N_\text{A}}\right]^{1/2}$$

where $\varepsilon_0 = 8.854 \times 10^{-12}\ \text{C}^2\ \text{N}^{-1}\ \text{m}^2$; $k = 1.38 \times 10^{-23}\ \text{J K}^{-1}$ and $N_\text{A} = 6.022 \times 10^{23}\ \text{mol}^{-1}$

19. The following data were obtained for the osmotic pressure of nitrocellulose in acetone at 20 °C.

$10^3\ c/\text{g L}^{-1}$	1.16	3.60	8.38	19.0
Π /(cmH_2O)	0.69	2.56	7.52	25.4

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = (10^{-3})\ (\Pi'/c)/$ erg g^{-1} with $\Pi' = \Pi\ \rho_{H2O}\ g$ and $x = 10^3\ c/$g cm^{-3}

Given: $\rho_{H_2O} = 1$ g /cm^3 ; $g = 980$ cm s^{-2}.

If $a_0 = RT/M_n$, calculate the value of M_n. Given: $R = 8.314 \times 10^7$ erg K^{-1} mol^{-1}.

(Ans. 47 000 g mol^{-1})

20. The relative viscosities of solutions of a sample of polystyrene in toluene were as follows.

$c/10^{-2}$ g cm^{-3}	0.249	0.499	0.999	1.998
η/η_0	1.355	1.782	2.879	6.090

Carry out the least square fitting for the equation $y = a_0 + a_1 x$,

where $y = 10^{-2}\ \eta_{sp}/\ (c/\text{g cm}^{-3})$ with $\eta_{sp} = (\eta/\eta_0) - 1$ and $x = c$

If $\overline{M}_v = (a_0/k)^{1/n}$, calculate $\overline{M}_v$ if $k = 3.7 \times 10^{-2}$ and $n = 0.62$. (Ans. 4.8×10^5 g mol^{-1})

21. The following data were obtained for bovine serum albumin in 0.1 M KI solution in a light scattering experiment using a wavelength of 546 nm (= 546×10^{-7} cm = λ_0).

$10^3\ c_2/$g cm^{-3}	0.936	1.902	2.801	3.701	5.590
$10^4\ \tau/$cm^{-1}	2.88	5.75	8.18	10.59	15.37

Carry out the least square fitting for the equation $y = a_0 + a_1 x$.

where $y = (10^5\ H\ c_2/\tau)\ /$g^{-1} mol and $x = 10^3\ c_2/$g cm^{-3}

Given $H = 32\ \pi^3 (n\ \mathrm{d}n/\mathrm{d}c_2)^2\ /3\ \lambda_0^4\ N_A$ where $n = 1.3342$ and $\mathrm{d}n/\mathrm{d}c_2 = 0.168$ g^{-1} cm^3

If $a_0 = 1/\ (M \times 10^{-5})$, calculate the value of M. (Ans. 1.02×10^5 g mol^{-1})

2.29 POLYNOMIAL FITTING

In a polynomial $y = a_0 + a_1 x + a_2 x^2 + \cdots$ fitting, the sum of squares of errors is

$$q = \Sigma_i\ (y_i - a_0 - a_1 x - a_2 x^2 - \cdots)^2$$

On minimizing through the expressions

$$\frac{\partial q}{\partial a_0} = 0, \qquad \frac{\partial q}{\partial a_1} = 0, \qquad \frac{\partial q}{\partial a_2} = 0, \quad \ldots$$

we get

$$a_0 N + a_1 \Sigma_i x_i + a_2 \Sigma_i x_i^2 + \cdots = \Sigma_i y_i$$

$$a_0 \Sigma_i x_i + a_1 \Sigma_i x_i^2 + a_2 \Sigma_i x^3{}_i + \cdots = \Sigma_i x_i y_i$$

$$a_0 \Sigma_i x_i^2 + a_1 \Sigma_i x_i^3 + a_2 \Sigma_i x_i^4 + \cdots = \Sigma_i x_i^2\ y_i$$

In the matrix form, we have

$$\begin{bmatrix} N & \Sigma_i x_i & \Sigma_i x_i^2 & \dots \\ \Sigma_i x_i & \Sigma_i x_i^2 & \Sigma_i x_i^3 & \dots \\ \Sigma_i x_i^2 & \Sigma_i x_i^3 & \Sigma_i x_i^4 & \dots \\ \vdots & & & \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \Sigma_i y_i \\ \Sigma_i x_i y_i \\ \Sigma_i x_i^2 y_i \\ \vdots \end{bmatrix}$$

Hence, we have

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} N & \Sigma_i x_i & \Sigma_i x_i^2 & \dots \\ \Sigma_i x_i & \Sigma_i x_i^2 & \Sigma_i x_i^3 & \dots \\ \Sigma_i x_i^2 & \Sigma_i x_i^3 & \Sigma_i x_i^4 & \dots \\ \vdots & & & \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_i y_i \\ \Sigma_i x_i y_i \\ \Sigma_i x_i^2 y_i \\ \vdots \end{bmatrix}$$

Thus, finding $\boldsymbol{X}^{-1}$ and then left multiplying it to the $\boldsymbol{XY}$ column vector gives the required vector of the coefficients.

Illustration Least square fitting for the equation $y = a_0 + a_1 x + a_2 x^2$ by the data of Exercise 1 given after the figure in the output.

Program

```
REM PROG28
5 REM LEAST SQUARE FITTING VIA INVERSE MATRIX
REM N IS ORDER OF EQUATION:ND IS NUMBER OF DATA POINTS
CLS : DEF FNX (X) = X: DEF FNY (Y, Z) = Y / Z
10 DIM A(6, 6), B(6), XX(10), A$(6), X(10), Y(10), YCAL(10), YER(10)
DIM X1(10), Y1(10)
15 A$(1) = "X1": A$(2) = "X2": A$(3) = "X3": A$(4) = "X4": A$(5) = "X5"
20 READ N, ND
25 FOR I = 1 TO ND: READ X1(I), Y1(I)
30 X(I) = FNX(X1(I)): Y(I) = FNY(Y1(I), X(I))
35 NEXT I
38 FOR I = 1 TO N: FOR J = I TO N: FOR K = 1 TO ND
42 A(I, J) = A(I, J) + X(K) ^ (I + J - 2)
46 NEXT K: A(J, I) = A(I, J): NEXT J
50 FOR K = 1 TO ND: B(I) = B(I) + X(K) ^ (I - 1) * Y(K): NEXT K
52 NEXT I
55 CLS : PRINT "PRINTING OF EQUATION PARAMETERS": PRINT
65 FOR I = 1 TO N: FOR J = 1 TO N
70 PRINT USING "   ##.####^^^^"; A(I, J);
85 NEXT J: PRINT USING "          ##.####^^^^"; B(I)': PRINT
95 NEXT I
100 REM TO MAKE THE LOWER HALF MATRIX EQUAL TO ZERO
105 FOR I = 1 TO N - 1
110 REM CHECK IF A(I,I)=0 IF SO INTERCHANGE WITH THE ROW HAVING A(K,I)
REM NOT EQUAL TO ZERO
```

```
120 K = I
125 IF A(K, K) <> 0 THEN 140
128 K = K + 1
130 GOTO 125
140 IF K = I THEN 185
145 FOR J = 1 TO N
150 TEMP = A(K, J): A(K, J) = A(I, J): A(I, J) = TEMP
165 NEXT J
168 TEMP = B(I): B(I) = B(K): B(K) = TEMP
185 FOR J = I + 1 TO N
190 DUM = A(J, I) / A(I, I)
195 FOR K = 1 TO N: A(J, K) = A(J, K) - A(I, K) * DUM: NEXT K
210 B(J) = B(J) - B(I) * DUM
212 NEXT J
214 PRINT "step number "; I
215 FOR J = 1 TO N: FOR K = 1 TO N
216 PRINT USING "   ##.####^^^^"; A(J, K);
217 NEXT K: PRINT USING "        ##.####^^^^"; B(J)': PRINT
218 A$ = INPUT$(1): NEXT J: NEXT I
225 REM CALCULATION OF UNKNOWN VARIABLES
232 IF A(N, N) = 0 THEN 290
235 XX(N) = B(N) / A(N, N)
240 FOR I = N - 1 TO 1 STEP -1: XX(I) = B(I)
246 FOR J = I + 1 TO N: XX(I) = XX(I) - A(I, J) * XX(J)
250 NEXT J: XX(I) = XX(I) / A(I, I): NEXT I
265 REM PRINTING OF THE VARIABLES
266 PRINT : PRINT "SOLUTION VARIABLES": PRINT
270 FOR I = 1 TO N
275 PRINT "XX("; I; ")="; : PRINT USING "##.##^^^^"; XX(I)
280 NEXT I
281 A0 = XX(1): A1 = XX(2): A$ = INPUT$(1): CLS
285 GOTO 320
290 FOR I = N - 1 TO 1 STEP -1
295 FOR J = I TO N
300 PRINT "+("; A(I, J); ")*"; A$(J);
305 NEXT J: PRINT "="; B(I): NEXT I
320 PRINT SPC(3); STRING$(50, "-")
322 PRINT "          X            Y           YCAL          YER"
324 PRINT SPC(3); STRING$(50, "-")
326 FOR I = 1 TO ND: YCAL(I) = 0
330 FOR J = 1 TO N: YCAL(I) = YCAL(I) + XX(J) * X(I) ^ (J - 1)
332 NEXT J: YER(I) = YCAL(I) - Y(I): SD = SD + YER(I) ^ 2
336 PRINT USING "    ###.##^^^^"; X(I); Y(I); YCAL(I); YER(I)
338 A$ = INPUT$(1): NEXT I
342 PRINT SPC(3); STRING$(50, "-")
```

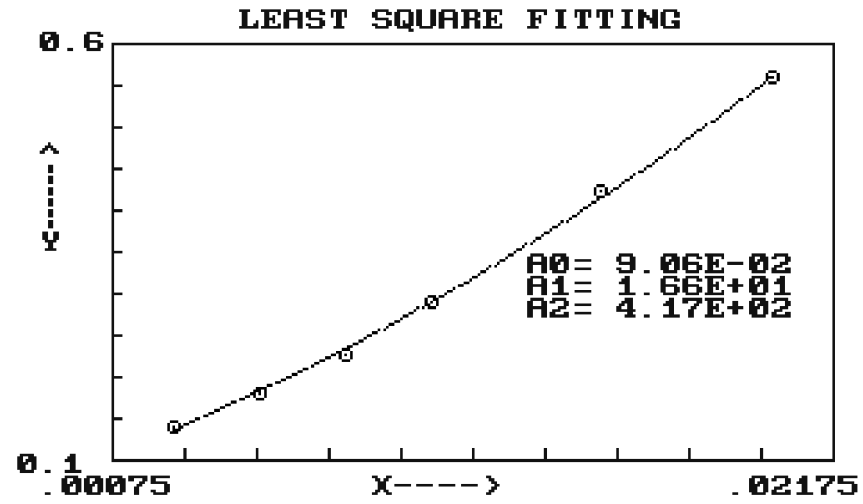
```
344 SD = (SD / N) ^ .5: PRINT "STANDARD DEVIATION="; SD
346 A$ = INPUT$(1)
348 XMIN = X(1): XMAX = X(ND): YMIN = Y(1): YMAX = Y(ND)
350 FOR I = 1 TO ND
352 IF XMIN > X(I) THEN XMIN = X(I)
354 IF YMIN > Y(I) THEN YMIN = Y(I)
356 IF XMAX < X(I) THEN XMAX = X(I)
358 IF YMAX < Y(I) THEN YMAX = Y(I)
360 NEXT I
362 XD = (XMAX - XMIN) / 10: YD = (YMAX - YMIN) / 10
364 XI = XMIN - XD: XF = XMAX + XD: YI = YMIN - YD: YF = YMAX + YD
366 SCREEN 1: COLOR 15, 0
368 VIEW (50, 20)-(310, 170)
370 WINDOW (XI, YI)-(XF, YF): LINE (XI, YI)-(XF, YF), , B
372 FOR I = 1 TO ND
374 PSET (X(I), Y(I)), 2: CIRCLE (X(I), Y(I)), XMAX / 100, 2
376 NEXT I
378 FOR XP = XMIN TO XMAX STEP XD / 20
380 YP = 0
382 FOR J = 1 TO N: YP = YP + XX(J) * XP ^ (J - 1): NEXT J
384 PSET (XP, YP), 2
386 NEXT XP
388 XD = (XF - XI) / 10: YD = (YF - YI) / 10
390 FOR I = 1 TO 9
392 LINE (XI + XD * I, YI)-(XI + XD * I, YI + YD / 4)
394 LINE (XI, YI + YD * I)-(XI + XD / 8, YI + YD * I)
396 NEXT I
398 LOCATE 2, 13: PRINT "LEAST SQUARE FITTING"
400 LOCATE 3, 1: PRINT USING "####.#"; YF
401 FOR I = 1 TO N: LOCATE 12 + I, 26
403 PRINT "A"; : PRINT USING "#"; I - 1;
    PRINT "="; : PRINT USING "##.##^^^^"; XX(I)
404 NEXT I
405 LOCATE 8, 4: PRINT CHR$(94)
406 FOR I = 1 TO 3: LOCATE 8 + I, 4: PRINT CHR$(124): NEXT I
407 LOCATE 12, 4: PRINT "Y"
408 LOCATE 22, 1: PRINT USING "###.#"; YI
410 LOCATE 23, 4: PRINT USING ".#####"; XI;
411 PRINT "           X---->            ";
412 PRINT USING ".#####"; XF
'DATA 4,5,1.4,740,1.8,750,2.3,760,3,750,4,720
DATA 3,6,.02,.0117,.015,.0066,.01,.003,.0075,.00173,.005,.0009,.0025,.00035
'DATA 3,5,298,37.13,398,41.1,498,44.56,598,47.52,698,49.97
414 END
```

Output

```
PRINTING OF EQUATION PARAMETERS

   6.0000E+00   6.0000E-02   8.1250E-04         1.8757E+00
   6.0000E-02   8.1250E-04   1.2937E-05         2.4280E-02
   8.1250E-04   1.2937E-05   2.2445E-07         3.8135E-04
step number  1
   6.0000E+00   6.0000E-02   8.1250E-04       1.8757E+00
  -1.8626E-09   2.1250E-04   4.8125E-06       5.5233E-03
  -2.9104E-11   4.8125E-06   1.1443E-07       1.2735E-04
step number  2
   6.0000E+00   6.0000E-02   8.1250E-04       1.8757E+00
  -1.8626E-09   2.1250E-04   4.8125E-06       5.5233E-03
   1.3080E-11   2.7703E-14   5.4382E-09       2.2663E-06

SOLUTION VARIABLES

XX( 1 )= 9.06E-02
XX( 2 )= 1.66E+01
XX( 3 )= 4.17E+02
```

X	Y	YCAL	YER
20.00E-03	58.50E-02	58.84E-02	34.16E-04
15.00E-03	44.00E-02	43.27E-02	-72.85E-04
10.00E-03	30.00E-02	29.79E-02	-21.48E-04
75.00E-04	23.07E-02	23.82E-02	75.67E-04
50.00E-04	18.00E-02	18.38E-02	38.25E-04
25.00E-04	14.00E-02	13.46E-02	-53.75E-04

```
STANDARD DEVIATION= 7.530529E-03
```



Exercises

1. The osmotic pressure values of various concentrations of polyisobutylene in cyclohexane at 298 K are as follows.

ρ/g cm^{-3}	0.0200	0.0150	0.0100	0.0075	0.0050	0.0025
Π/atm	0.0117	0.0066	0.0030	0.00173	0.0009	0.00035

Carry out the least square fitting as per equation $y = a_0 + a_1x + a_2x^2$, where $y = \Pi/\rho$ and $x = \rho$. If $a_0 = RT/M$, Calculate the value of M.

Given: FNX (X) = X and FNY (Y) = Y/X(I)

(Ans. $a_0 = 9.06 \times 10^{-2}$; $a_1 = 1.66 \times 10$; $a_2 = 4.17 \times 10^2$)

2. Carry out the least square fitting of the following data on molar heat capacity as per the equation $y = a + b(T/\text{K}) + c(T/\text{K})^2$.

T/K	298	398	498	598	698
C_p/ (J K^{-1} mol^{-1})					
Oxygen	29.38	30.24	31.07	31.88	32.67
Carbon dioxide	37.13	41.10	44.56	47.52	49.97

2.30 DISPLAY OF VAN DER WAALS ISOTHERMS OF A GAS

To draw the variation of pressure with volume of a real gas at temperatures lower than its critical temperature.

The van der waals equation for one mole of gas is $\left(p + \frac{a}{V^2}\right)(V - b) = RT$

This gives $p = \frac{RT}{V - b} - \frac{a}{V^2}$

For the given temperature, V is varied from a value lesser than V_c to a value larger than V_c and the corresponding pressure is determined by using the above expression.

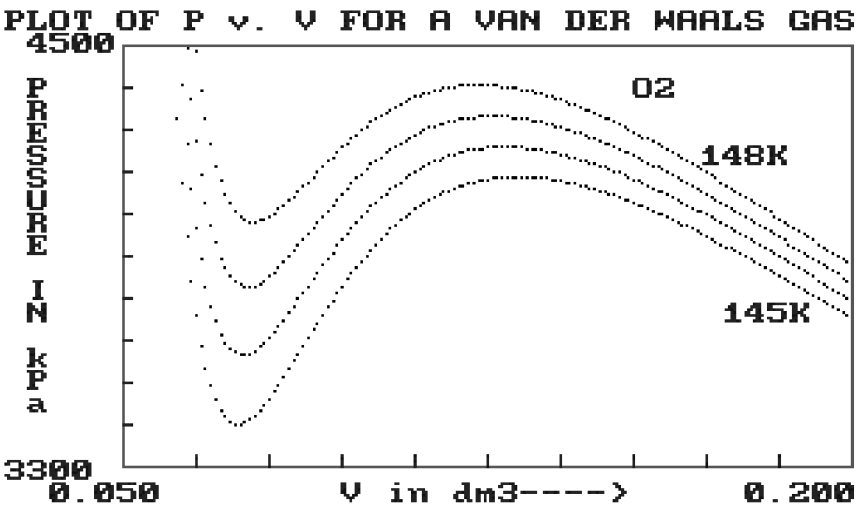
Illustration Display van der Waals isotherm of O_2.

Program

```
REM PROGRAM VANDER;VAN DER WAALS ISOTHERMS
REM PLOT FOR H2O:A in kPA dm3: B in dm3
REM TI ABOUT 10 DEGREE < TC;  BOTH PI,PF < PC:  VI a little < VC
CLS : READ N$, A, B, TI, TF, TS, VI, VF, VS, PI, PF
'DATA H2O,553.639,.03049,560,590,30,.039,.25,.001,2600,20000
'DATA CO2,363.96,.04267,290,292,2,.075,.25,.001,5600,6500
DATA O2,137.802,.03183,145,148,1,.05,.2,.001,3300,4500
'DATA C2H6,556.173,.0638,292,296,4,.11,.50,.001,3400,4500
R = 8.314
SCREEN 1: COLOR 15, 0
VIEW (50, 20)-(310, 170)
```

```
WINDOW (VI, PI)-(VF, PF): LINE (VI, PI)-(VF, PF), 1, B
FOR T = TI TO TF STEP TS: FOR V = VI TO VF STEP VS
P = R * T / (V - B) - A / V ^ 2
PSET (V, P), 2
'PRINT V, P: A$ = INPUT$(1)
FOR KK = 1 TO 100 STEP .001: NEXT KK: NEXT V: NEXT T
VD = (VF - VI) / 10: PD = (PF - PI) / 10
FOR I = 1 TO 9
LINE (VI + VD * I, PI)-(VI + VD * I, PI + PD / 3.5)
LINE (VI, PI + PD * I)-(VI + VD / 10, PI + PD * I)
NEXT I
LOCATE 1, 6: PRINT " PLOT OF P v. V FOR A VAN DER WAALS GAS"
LOCATE 3, 1: PRINT USING "######"; PF
LOCATE 5, 30: PRINT N$
A$ = "PRESSURE IN kPa"
FOR I = 1 TO 15
B$ = MID$(A$, I, 1): LOCATE 4 + I, 3: PRINT B$
NEXT I
LOCATE 8, 33: PRINT USING "###"; TF; : PRINT "K"
LOCATE 15, 34: PRINT USING "###"; TI; : PRINT "K"
LOCATE 22, 1: PRINT USING "#####"; PI
LOCATE 23, 3: PRINT USING "##.###"; VI;
LOCATE 23, 15: PRINT "  V in dm3---->   "; : PRINT USING "##.###"; VF
END
```

Output



Run the program to display the isotherms for H_2O, CO_2 and C_2H_6. The data for these gases are included within the program.

2.31 DISPLAY OF DISTRIBUTION OF MOLECULAR SPEEDS

To draw distribution of molecular speeds amongst gaseous molecules (say, oxygen) at different temperatures by using the Maxwell distribution of molecular speeds. Also find out the speed corresponding to the maximum fraction of molecules and the area under each distribution curve.

The expression to be used is

$$\left(\frac{\mathrm{d}N}{N}\right)\frac{1}{\mathrm{d}u} = 4\pi\left(\frac{M}{2\pi RT}\right)^{3/2} \exp(-Mu^2/2RT)$$

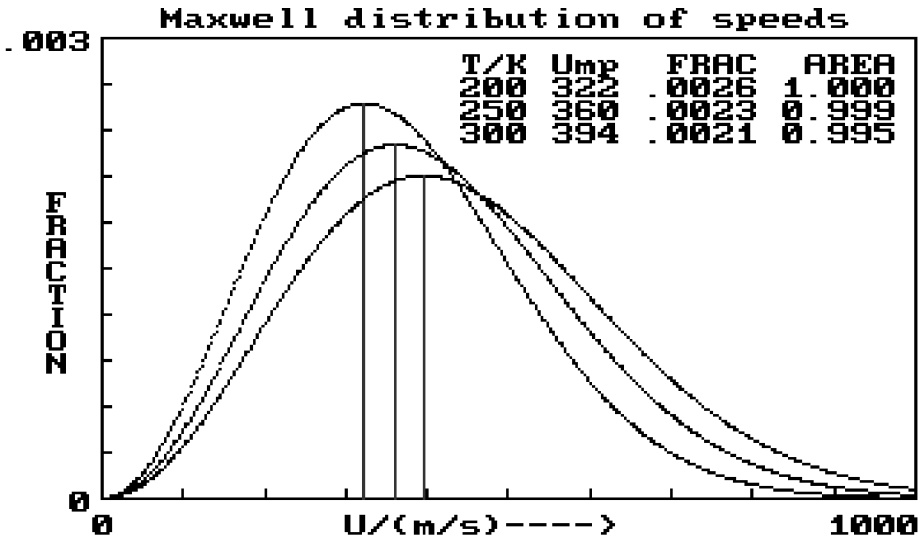
Recommended speeds are from 0 to 1000 m s^{-1}. Recommended temperatures are 200 K, 300 K and 400 K.

Program

```
REM PROGRAM MAXWELL;FRACTION v. SPEED
CLS : DIM T(5), FRMAX(5), UMAX(5), AREA(5)
READ M, TI, TF, TS: DATA .032, 200, 300, 50
I = 0: PI = 3.14159: R = 8.314
SCREEN 1: COLOR 15, 0
VIEW (35, 10)-(318, 170)
WINDOW (0, 0)-(1000, .003): LINE (0, 0)-(1000, .003), , B
FOR TP = TI TO TF STEP TS
I = I + 1: FRMAX(I) = 0: UMAX(I) = 0: T(I) = TP: AREA(I) = 0
US = 2
FOR U = 0 TO 1000 STEP US
TERM1 = M / (2 * PI * R * TP): TERM2 = M * U * U / (2 * R * TP)
FR = 4 * PI * TERM1 ^ 1.5 * EXP(-TERM2) * U * U
AREA(I) = AREA(I) + FR * US
IF FRMAX(I) < FR THEN FRMAX(I) = FR: UMAX(I) = U
PSET (U, FR)
FOR J = 1 TO 100 STEP .1: NEXT J
NEXT U
A$ = "FRACTION"
FOR J = 1 TO 8: B$ = MID$(A$, J, 1)
LOCATE 8 + J, 3: PRINT B$: NEXT J
LOCATE 1, 8: PRINT "Maxwell distribution of speeds"
LOCATE 2, 1: PRINT ".003"
LOCATE 3, 19: PRINT "  T/K Ump  FRAC  AREA"
LOCATE 3 + I, 20: PRINT USING "####"; T(I); UMAX(I);
PRINT USING " .####"; FRMAX(I); : PRINT USING " #.###"; AREA(I)
```

```
LINE (UMAX(I), 0)-(UMAX(I), FRMAX(I)), 1
NEXT TP
LOCATE 22, 4: PRINT "0"
LOCATE 23, 3: PRINT "   0            U/(m/s)---->           1000"
FOR J = 1 TO 9
LINE (J * 100, 0)-(J * 100, .00005)
LINE (0, .0003 * J)-(10, .0003 * J)
NEXT J
END
```

Output



2.32 DISPLAY OF DISTRIBUTION OF MOLECULAR ENERGIES

To draw distribution of molecular energies amongst gaseous molecules (say, oxygen) at different temperatures by using the Maxwell distribution function. Also find out the energy corresponding to the maximum fraction of molecules and the area under each distribution curve.

The expression to be used is

$$\left(\frac{\mathrm{d}N}{N}\right)\frac{1}{\mathrm{d}\varepsilon} = 2\pi\left(\frac{1}{\pi kT}\right)^{3/2} \varepsilon^{1/2} \exp\left(-\varepsilon/kT\right)$$

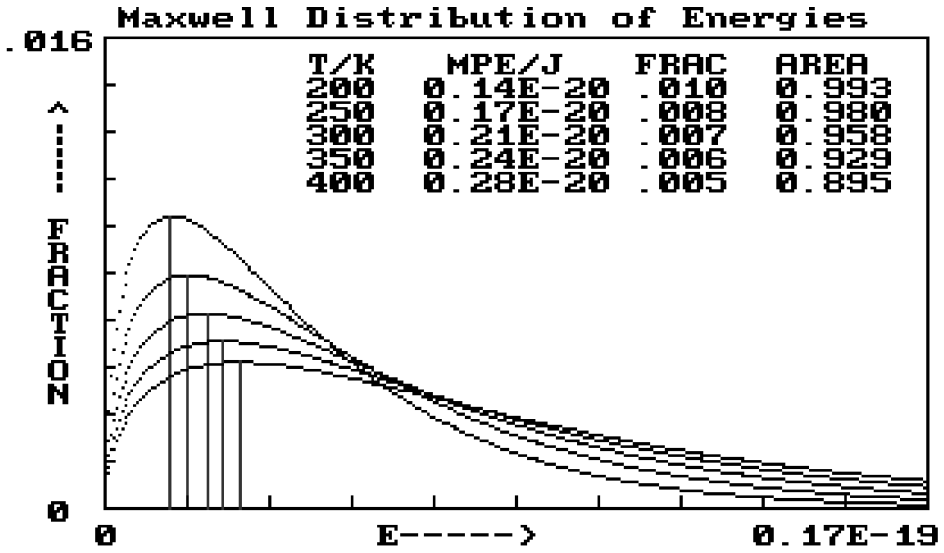
Recommended energies are corresponding to speeds from 0 to 1000 m/s. Recommended temperatures are 200 K, 300 K and 400 K.

Program

```
REM PROGRAM MAXWLEN;FRACTION v. ENERGY
CLS : DIM T(5), FRMAX(5), EMAX(5), AREA(5)
DEF FNE (M, U) = .5 * M * U * U
READ M, TI, TF, TS: DATA .032,200,400,50
```

```
I = 0: PI = 3.14159: R = 8.314: L = 6.022E+23
KB = R / L: MP = M / L: EF = FNE(MP, 800): FRM = .016
SCREEN 1: COLOR 15, 0: VIEW (35, 10)-(315, 170)
WINDOW (0, 0)-(EF, FRM): LINE (0, 0)-(EF, FRM), , B
FOR TP = TI TO TF STEP TS
I = I + 1: FRMAX(I) = 0: EMAX(I) = 0
T(I) = TP: AREA(I) = 0: DE = EF / 300
FOR E = 0 TO EF STEP DE
TERM1 = 1 / (PI * KB * TP): TERM2 = E / (KB * TP)
FR = 2 * PI * TERM1 ^ 1.5 * EXP(-TERM2) * E ^ .5 * DE
AREA(I) = AREA(I) + FR
IF FRMAX(I) < FR THEN FRMAX(I) = FR: EMAX(I) = E
PSET (E, FR)
FOR J = 1 TO 100 STEP .1: NEXT J
NEXT E
A$ = "FRACTION": LOCATE 5, 3: PRINT CHR$(94)
FOR J = 1 TO 3: LOCATE 5 + J, 3: PRINT CHR$(124): NEXT J
FOR J = 1 TO 8: B$ = MID$(A$, J, 1)
LOCATE 9 + J, 3: PRINT B$: NEXT J
LOCATE 1, 6: PRINT "Maxwell Distribution of Energies"
LOCATE 2, 1: PRINT USING ".###"; FRM
LOCATE 3, 13: PRINT " T/K    MPE/J    FRAC   AREA"
LOCATE 3 + I, 13: PRINT USING "####"; T(I);
PRINT USING "   #.##^^^^"; EMAX(I);
PRINT USING " .###"; FRMAX(I);
PRINT USING "   #.###"; AREA(I)
LINE (EMAX(I), 0)-(EMAX(I), FRMAX(I)), 1
LOCATE 23, 4: PRINT " 0              E------>           ";
PRINT USING "#.##^^^^"; EF
NEXT TP
LOCATE 22, 3: PRINT "0"
FOR J = 1 TO 9
LINE (J * EF / 10, 0)-(J * EF / 10, FRM / 40)
LINE (0, FRM * .1 * J)-(EF / 100, FRM * .1 * J)
NEXT J
END
```

Output



2.33 DISPLAY OF POTENTIAL ENERGY OF INTERACTION OF IONS

To display variation of potential energy of interaction between ions having symmetrical-salt structure such as rock salt.

The expression to be used are

Attraction term $$E_1 = -\frac{N_A A z^2}{(4\pi\varepsilon_0) r}$$

Repulsion term $$E_2 = \frac{b}{r^n} \quad \text{where} \quad b = \frac{N_A A z^2 r_0^{n-1}}{(4\pi\varepsilon_0) r}$$

where r_0 is the distance between cation and anion corresponding to the most stable structure of the crystal.

Use the Born exponent $n = 9$ and Madelung constant $A = 1.746$

$$z = 1.6 \times 10^{-19}\ \text{C}; \quad \varepsilon_0 = 8.854 \times 10^{-12}\ \text{C}^2\ \text{N}^{-1}\ \text{m}^{-2}; \quad r_0 = 314\ \text{pm}.$$

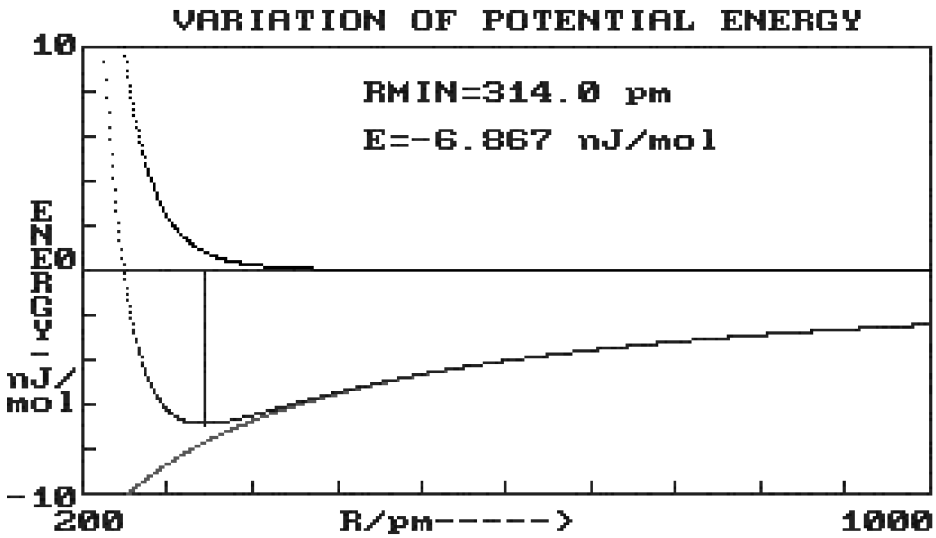
Over-all potential energy, $E = E_1 + E_2$.

Program

```
REM PROG32
REM PROGRAM POTENER;POTENTIAL ENERGY OF INTERACTIONS OF IONS
NA = 6.023: A = 1.746: N = 9: Z = 1.602
R0 = 3.14: MIN = 5: RI = 2: RF = 10
CLS : SCREEN 1: COLOR 15, 0
VIEW (25, 20)-(310, 170): WINDOW (RI, -.1)-(RF, .1)
LINE (RI, -.1)-(RF, .1), , B: LINE (RI, 0)-(RF, 0)
CONS = NA * A * Z ^ 2 / (4 * 3.14159 * 8.854)
```

```
FOR R = RI TO RF STEP .01
TERM1 = -CONS / R: TERM2 = CONS * R0 ^ (N - 1) / (N * R ^ N)
TERM = TERM1 + TERM2
IF MIN > TERM THEN MIN = TERM: RMIN = R
PSET (R, TERM1), 1: PSET (R, TERM2), 2: PSET (R, TERM), 3
FOR J = 1 TO 100 STEP .001: NEXT J
NEXT R
LINE (RMIN, MIN)-(RMIN, 0)
LOCATE 2, 8: PRINT "VARIATION OF POTENTIAL ENERGY"
LOCATE 3, 2: PRINT "10": LOCATE 5, 16: PRINT "RMIN=";
PRINT USING "###.#"; RMIN * 100; : PRINT " pm"
LOCATE 7, 16: PRINT "E="; : PRINT USING "##.###"; MIN * 100;
PRINT " nJ/mol": A$ = "ENERGY-"
FOR I = 1 TO 7: LOCATE 9 + I, 2: PRINT MID$(A$, I, 1): NEXT I
LOCATE 17, 1: PRINT "nJ/": LOCATE 18, 1: PRINT "mol"
LOCATE 12, 3: PRINT "0": LOCATE 22, 1: PRINT "-10"
LOCATE 23, 2: PRINT USING "####"; RI * 100;
PRINT "           R/pm----->            ";
PRINT USING "#####"; RF * 100
FOR I = 1 TO 9
II = -.1 + .02 * I: LINE (2, II)-(2.1, II)
II = 2 + I * (RF - RI) / 10: LINE (II, -1)-(II, -.095)
NEXT I
END
```

Output



2.34 pH TITRATION CURVE OF A STRONG ACID WITH A STRONG BASE

To display the pH of a solution during the titration of a strong acid (volume V_a and molarity M_a) with a strong base (molarity M_b). Also display its first derivative (i.e. the plot of $\partial \text{pH}/\partial V_b$ versus V_b).

The concentration of H^+ in the solution before and after the equivalence point may be computed by the expression

$$[H^+] = \text{ABS}(V_a M_a - V_b M_b) / (V_a + V_b)$$

Hence $\text{pH} = -\log \{[H^+]/\text{mol dm}^{-3}\}$.

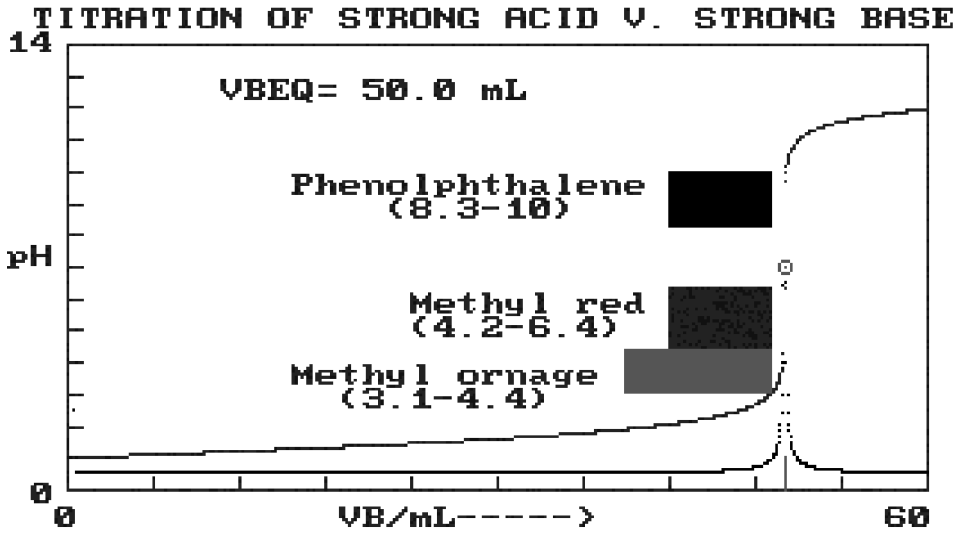
At the equivalence point pH = 7.

Illustration Titration of 50 mL of 0.1M of strong acid with 0.1M of strong base.

Program

```
REM PROGRAM SASBDIR;TITRATION OF STRONG ACID v.STRONG BASE
CLS : READ VA, MA, MB: DATA 50,.1,.1: PHP = 0: VBP = 0
VBEQ = VA * MA / MB: VMAX = VBEQ + 10
SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(310, 170)
WINDOW (0, 0)-(VMAX, 14): LINE (0, 0)-(VMAX, 14), , B
FOR VB = .5 TO VMAX STEP .05
IF VB < VBEQ THEN H = (VA * MA - VB * MB) / (VA + VB)
IF VB = VBEQ THEN H = .0000001
IF VB > VBEQ THEN H = 1E-14 / ABS((VA * MA - VB * MB) / (VA + VB))
PH = -LOG(H) / LOG(10): BC = (PH - PHP) / (VB - VBP) + .5
PSET (VB, PH): PSET (VB, BC), 2
PHP = PH: VBP = VB
FOR KK = 1 TO 100 STEP .1: NEXT KK: NEXT VB
FOR I = 1 TO 9: II = VMAX * .1 * I: LINE (II, 0)-(II, .4): NEXT I
FOR I = 1 TO 13: LINE (0, I)-(1, I): NEXT I
LOCATE 2, 2: PRINT "TITRATION OF STRONG ACID V. STRONG BASE"
LOCATE 3, 1: PRINT "14": LOCATE 12, 1: PRINT "pH"
LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT "0          VB/mL----->         "; : PRINT VMAX
PSET (VBEQ, 7), 1: CIRCLE (VBEQ, 7), .5, 1: VI = VBEQ - 8: VF = VBEQ - 1
LOCATE 5, 10: PRINT "VBEQ="; : PRINT USING "###.#"; VBEQ; : PRINT " mL"
LINE (VI, 8.3)-(VF, 10), 2, BF: LOCATE 9, 13: PRINT "Phenolphthalene"
LOCATE 10, 17: PRINT "(8.3-10)"
LINE (VI, 4.2)-(VF, 6.3), 3, BF: LOCATE 14, 18: PRINT "Methyl red"
LOCATE 15, 18: PRINT "(4.2-6.4)"
LINE (VI - 3, 3.1)-(VF, 4.4), 1, BF: LOCATE 17, 13: PRINT "Methyl ornage"
LOCATE 18, 15: PRINT "(3.1-4.4)"
LINE (VBEQ, 0)-(VBEQ, 1), 1
END
```

Output



2.35 pH TITRATION CURVE OF A WEAK ACID WITH A STRONG BASE

To display the pH of a solution during the titration of a weak acid (volume V_a and molarity M_a) with a strong base (molarity M_b). Also display its first derivative (i.e. the plot of $\partial \text{pH}/\partial V_b$ versus V_b).

Given: $K_a(\text{acid}) = 1.8 \times 10^{-5}$ M.

Start of titration $[H_3O^+]^2 + K_a[H_3O^+] - K_a[HA]_0 = 0$

For a fairly concentrated solution with small dissociation constant of acid, use the expression $[H_3O^+] = \sqrt{K_a[HA]_0}$.

Before the equivalence point $\text{pH} = \text{p}K_a^\circ + \log\left(\dfrac{f}{1-f}\right)$

where f is the fraction of acid neutralized by base.

At the equivalence point $\text{pH} = \dfrac{1}{2}\ (\text{p}K_w^\circ + \text{p}K_a^\circ + \log\ [HA]_0/c^\circ)$

Beyond the equivalence point

$[OH^-]$ = Base added after the equivalence point

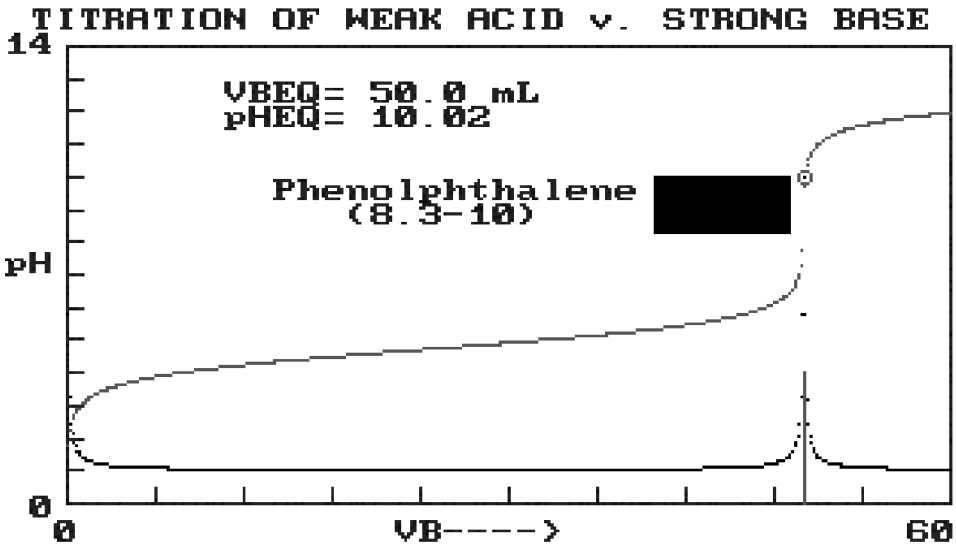
$[H^+] = K_w/[OH^-]$

Illustration Tiration of 0.1 M weak acid ($K_a^\circ = 1.8 \times 10^{-5}$) with 0.1 M strong base

Program

```
REM PROGRAM WASBDIR;WEAK ACID v.BASE TITRATIN DIRECT
CLS : READ VA, MA, MB, KA: DATA 50,.1,.1,1.8E-5
DEF FNA (H) = -LOG(H) / LOG(10): KW = 1E-14
VBEQ = VA * MA / MB: VMAX = VBEQ + 10
SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(310, 170)
WINDOW (0, 0)-(VMAX, 14): LINE (0, 0)-(VMAX, 14), , B
FOR VB = .05 TO VMAX STEP .1
IF VB = 0 THEN
H = (KA * MA) ^ .5: PH = FNA(H)
ELSEIF VB < VBEQ THEN
F = VB * MB / (VA * MA): FRAC = F / (1 - F)
PH = FNA(KA) - FNA(FRAC): PHEQ = PH
ELSEIF VB > VBEQ THEN
OH = (VB - VBEQ) * MB / (VA + VB): H = KW / OH: PH = FNA(H)
END IF
PSET (VB, PH), 1: DERI = (PH - PHP) / (VB - VBP)
PSET ((VB + VBP) / 2, DERI + 1), 2
VBP = VB: PHP = PH
FOR KK = 1 TO 100 STEP .001: NEXT KK: NEXT VB
PH = .5 * (FNA(KW) + FNA(KA) + FNA(MA / 2))
PSET (VBEQ, PH), 2: CIRCLE (VBEQ, PH), .5, 1
LINE (VBEQ, 0)-(VBEQ, 4), 1
LOCATE 2, 2: PRINT "TITRATION OF WEAK ACID v. STRONG BASE"
LOCATE 3, 1: PRINT "14": LOCATE 5, 10: PRINT "VBEQ=";
PRINT USING "###.#"; VBEQ; : PRINT " mL"
LOCATE 6, 10: PRINT "pHEQ="; : PRINT USING "###.##"; PH
LOCATE 12, 1: PRINT "pH": LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT "0                VB---->                ";
PRINT USING "###"; VMAX
FOR I = 1 TO 9: II = VMAX * .1 * I: LINE (II, 0)-(II, .5): NEXT I
FOR I = 1 TO 13: LINE (0, I)-(1, I): NEXT I
LINE (VBEQ - 10, 8.3)-(VBEQ - 1, 10), 2, BF
LOCATE 9, 12: PRINT "Phenolphthalene": LOCATE 10, 15: PRINT "(8.3-10)"
END
```

Output



2.36 DISPLAY OF DISTRIBUTION FUNCTIONS OF A DIPROTIC ACID

To display the distribution functions $[H_2A]/[H_2A]_0$, $[HA^-]/[H_2A]_0$ and $[A^{2-}]/[H_2A]_0$ with the variation of pH during the titration of a dibasic acid, H_2A, with a strong base.

The expressions of distribution functions are:

$$\frac{[H_2A]}{[H_2A]} = \frac{1}{\alpha} \; ; \; \frac{[HA^-]}{[H_2A]_0} = \frac{1}{([H^+]/K_{a1})\alpha} \; ; \; \frac{[A^{2-}]}{[H_2A]_0} = \frac{1}{([H^+]^2/K_{a1}K_{a2})\alpha}$$

where $$\alpha = 1 + \frac{K_{a1}}{[H^+]} + \frac{K_{a1}K_{a2}}{[H^+]^2}$$

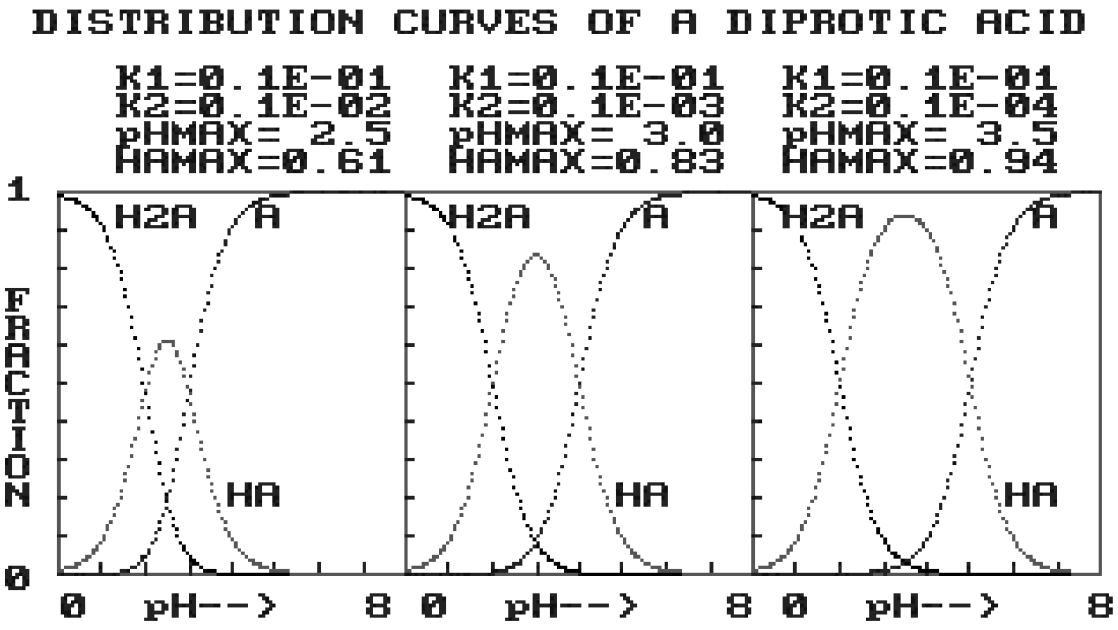
Illustration Display of distribution curves for an acid with $K^\circ_{a1} = 1.0 \times 10^{-2}$ and K°_{a2} varies from 1.0×10^{-3} to 1.0×10^{-5} in the step of 0.1.

Vary pH from 0 to 8.

Program

```
REM PROGRAM DISTRIB;DISTRIBUTION FUNCTIONS OF A DIPROTIC ACID VERSE pH
CLS : READ K1: K2 = K1
DATA .01: PHF = 8
SCREEN 1: COLOR 15, 0
LOCATE 1, 6: PRINT " DISTRIBUTION CURVES OF A DIPROTIC ACID"
LOCATE 8, 1: PRINT "1"
A$ = "FRACTION"
FOR I = 1 TO 8: LOCATE 11 + I, 1: PRINT MID$(A$, I, 1): NEXT I
LOCATE 22, 1: PRINT "0"
FOR J = 1 TO 3
VIEW (15 + 100 * (J - 1), 60)-(115 + 100 * (J - 1), 170)
WINDOW (0, 0)-(PHF, 1): LINE (0, 0)-(PHF, 1), 1, B
```

```
FOR I = 1 TO 9: LINE (0, I / 10)-(.15, I / 10): NEXT I
FOR I = 1 TO 7: II = I * PHF / 8: LINE (II, 0)-(II, .03): NEXT I
'FOR I = 1 TO 9: LINE (0, I / 10)-(PHF, I / 10): NEXT I
'FOR I = 1 TO 7: LINE (I * PHF / 8, 0)-(I * PHF / 8, 1): NEXT I
LOCATE 23, 2 + 13 * (J - 1): PRINT " 0   pH-->   ";
PRINT USING "##"; PHF
K2 = K2 * .1: HAMAX = 0
FOR PH = .01 TO PHF STEP .05
H = 10 ^ (-PH)
H2A = 1 / (1 + K1 / H + K1 * K2 / H ^ 2)
HA = 1 / (H / K1 + 1 + K2 / H)
A = 1 / (H ^ 2 / (K1 * K2) + H / K2 + 1)
PSET (PH, H2A), 2: PSET (PH, HA), 1: PSET (PH, A), 3
IF HAMAX < HA THEN HAMAX = HA: PHMAX = PH
LOCATE 4, 5 + 12 * (J - 1): PRINT "K1="; : PRINT USING "#.#^^^^"; K1
LOCATE 5, 5 + 12 * (J - 1): PRINT "K2="; : PRINT USING "#.#^^^^"; K2
LOCATE 6, 5 + 12 * (J - 1): PRINT "pHMAX="; : PRINT USING "##.#"; PHMAX
LOCATE 7, 5 + 12 * (J - 1): PRINT "HAMAX="; : PRINT USING "#.##"; HAMAX
FOR I = 1 TO 100 STEP .001: NEXT I
NEXT PH
LOCATE 9, 10 + 14 * (J - 1): PRINT "A"'
LOCATE 9, 5 + 12 * (J - 1): PRINT "H2A"
LOCATE 19, 9 + 14 * (J - 1): PRINT "HA"
A$ = INPUT$(1)
NEXT J
END
```



2.37 pH TITRATION CURVE OF A DIPROTIC ACID WITH A STRONG BASE

To display the pH of a solution during the titration of a dibasic acid (volume V_a, molarity M_a) with a strong base (molarity M_b). Also display its first derivative (i.e. the plot of $\partial pH/\partial V_b$ versus V_b).

Given: K_{a1} and K_{a2} of the acid.

At the start of titration $[H_3O^+]^2 + K_{a1}[H_3O^+] - K_{a1}\,[H_2A]_0 = 0.$

For a fairly concentrated acid, use the expression $[H_3O^+] = \sqrt{K_{a1}[H_2A]_0}$

Before the equivalence point $pH = pK^\circ_{a1} + \log\left(\dfrac{f}{1-f}\right)$ where f is the fraction of acid converted to HA^-

At the first equivalence point $pH = \dfrac{1}{2}\,(pK^\circ_{a1} + pK^\circ_{a2})$

After the first equivalence point $pH = pK^\circ_{a2} + \log\left(\dfrac{f}{1-f}\right)$; where f is the fraction of acid HA^- converted to A^{2-}.

At the second equivalence point $pH = \dfrac{1}{2}\,[pK^\circ_w + pK^\circ_{a2} + \log([H_2A]_0/c^\circ)]$

Beyond the second equivalence point $[OH^-]$ = Base added after the equivalence point. $[H^+] = K_w/[OH^-]$

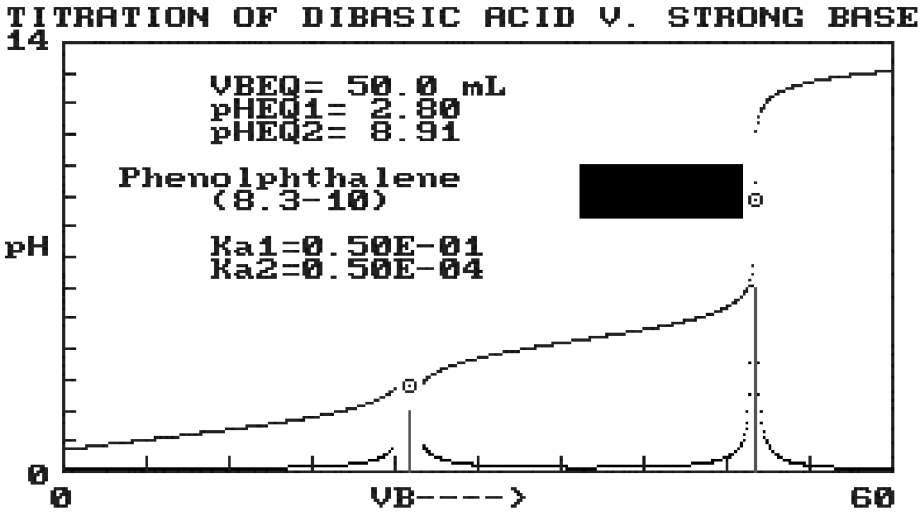
Illustration Titration curve of 25 mL of 1 M diprotic acid ($K^\circ_{a1} = 5 \times 10^{-2}$ and $K^\circ_{a_2} = 5 \times 10^{-5}$ with 1 M of a strong base.

Program

```
REM PROGRAM DASBDIR:TITRATION OF DIBASIC ACID v. BASE
READ VA, MA, MB, KA1, KA2: DATA 25,1,1,5E-2,5E-5: CLS
DEF FNH (H) = -LOG(H) / LOG(10): KW = 1E-14
DEF FNA (C1, C2, K) = (-(C2 + K) + SQR((C2 + K) ^ 2 + 4 * K * C1))/(2 * C1)
VBEQ = 2 * VA * MA / MB: VMAX = VBEQ + 10
SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(310, 170)
WINDOW (0, 0)-(VMAX, 14): LINE (0, 0)-(VMAX, 14), , B
FOR VB = 0 TO VMAX STEP .1
IF VB = 0 THEN
ALPHA = FNA(MA, 0, KA1): H = MA * ALPHA
ELSEIF VB < VBEQ / 2 - 1 THEN
F = VB * MB / (VA * MA)
C1 = MA * (1 - F) * VA / (VA + VB): C2 = MA * F * VA / (VA + VB)
ALPHA = FNA(C1, C2, KA1): H = C1 * ALPHA
ELSEIF VB > VBEQ / 2 - 1 AND VB < VBEQ / 2 + 1 THEN
GOTO 5
```

```
ELSEIF VB > VBEQ / 2 + 1 AND VB < VBEQ THEN
F = (VB - VBEQ / 2) * MB / (VA * MA)
'C1 = MA * (1 - F) * VA / (VA + VB): C2 = MA * F * VA / (VA + VB)
'ALPHA = FNA(C1, C2, KA2): H = C1 * ALPHA
PH = FNH(KA2) - FNH(F / (1 - F)): H = 10 ^ -PH
ELSEIF VB > VBEQ THEN
OH = (VB - VBEQ) / (VA + VB): H = KW / OH
END IF
PH = FNH(H): PSET (VB, PH)
IF VB = 0 THEN 5
DERI = (PH - PHP) / (VB - VBP)
PSET ((VB + VBP) / 2, DERI * 2), 2
VBP = VB: PHP = PH
FOR I = 1 TO 100 STEP .1: NEXT I
5 NEXT VB
PHEQ1 = .5 * (FNH(KA1) + FNH(KA2)): PSET (VBEQ / 2, PHEQ1)
CIRCLE (VBEQ / 2, PHEQ1), VA / 80
PHEQ2 = .5 * (FNH(KW) + FNH(KA2) - FNH(MA * VA / (VA + VBEQ)))
PSET (VBEQ, PHEQ2): CIRCLE (VBEQ, PHEQ2), VA / 80
LINE (VBEQ / 2, 0)-(VBEQ / 2, 2), 1: LINE (VBEQ, 0)-(VBEQ, 6), 1
LOCATE 2, 1: PRINT "TITRATION OF DIBASIC ACID V. STRONG BASE"
LOCATE 3, 1: PRINT "14": LOCATE 5, 10: PRINT "VBEQ=";
PRINT USING "###.#"; VBEQ; : PRINT " mL"
LOCATE 6, 10: PRINT "pHEQ1="; : PRINT USING "##.##"; PHEQ1
LOCATE 7, 10: PRINT "pHEQ2="; : PRINT USING "##.##"; PHEQ2
LOCATE 12, 1: PRINT "pH": LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT "0                VB---->                ";
PRINT USING "###"; VMAX
FOR I = 1 TO 9: II = VMAX * .1 * I: LINE (II, 0)-(II, .5)
'LINE (II, 0)-(II, 14)
NEXT I
FOR I = 1 TO 13: LINE (0, I)-(.03 * VA, I)
'LINE (0, I * 14 / 10)-(VMAX, I * 14 / 10)
NEXT I
LINE (VBEQ - .5 * VA, 8.3)-(VBEQ - 1, 10), 2, BF
LOCATE 9, 6: PRINT "Phenolphthalene"
LOCATE 10, 10: PRINT "(8.3-10)"
LOCATE 12, 10: PRINT "Ka1="; : PRINT USING "#.##^^^^"; KA1
LOCATE 13, 10: PRINT "Ka2="; : PRINT USING "#.##^^^^"; KA2
END
```

Output



2.38 GENERAL TREATMENT OF pH TITRATION CURVE OF A MONOPROTIC ACID

To draw a pH titration curve for the titration between acid (strong or weak) and strong base. Also draw the first derivative and the curve for its buffer effect, $\partial V_b/\partial$pH versus V_b. Locate the pH at the equivalence point.

The expression to be used is

$$V_b = -V_a\left[\frac{[H^+]^3 + K_a[H^+]^2 - (K_aM_a + K_w)[H^+] - K_wK_a}{[H^+]^3 + (M_b + K_a)[H^+]^2 + (K_aM_b - K_w)[H^+] - K_wK_a}\right]$$

For strong acid, use K_a = 1.0E + 07 or higher value

The expression of buffer effect is

$$\frac{\partial V_b}{\partial \text{pH}} = -2.303\,[H^+]\left(\frac{\alpha}{\beta}\right)$$

where $\alpha = 3[H^+]^2\,(V_b + V_a) + 2[H^+]\,(V_bK_a + V_b\,M_b + V_aK_a) + V_bM_bK_a - V_aM_aK_a - V_bK_w - V_aK_w$

$\beta = [H^+]^3 + [H^+]^2\,(K_a + M_b) + [H^+](K_aM_b - K_w) - K_wK_a$

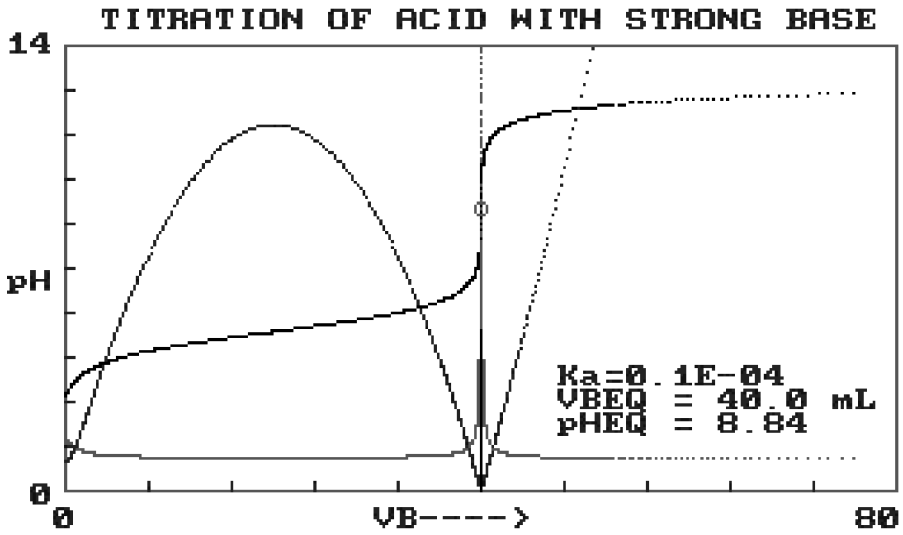
Compute the volume V_b of base to be added in the volume V_a of acid to obtain the known value of H^+(or pH) which lies within the range accessible in the titration. Out side this range, V_b will come out to be negative which should be ignored.

Illustration Titration of 40 mL of 0.1 M of an acid ($K_a^\circ = 1 \times 10^{-5}$) with 0.1 M of a strong base

Program

```
REM PROGRAM TITAB;GENERAL TREATMENT OF TITRATION
CLS : READ VA, MA, KA, MB: DATA 40,.1, 1E-5, .1
DERIMIN = -100: KW = 1E-14
VBEQ = VA * MA / MB: VBMAX = 2 * VBEQ
PHMIN = 0: PHMAX = 12.5
SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(310, 170)
WINDOW (0, 0)-(VBMAX, 14): LINE (0, 0)-(VBMAX, 14), 1, B
FOR PH = PHMIN TO PHMAX STEP .01
H = 10 ^ (-PH)
NUM = H ^ 3 + KA * H ^ 2 - (KA * MA + KW) * H - KW * KA
DEN = H ^ 3 + (MB + KA) * H ^ 2 + (KA * MB - KW) * H - KW * KA
VB = -VA * NUM / DEN
IF VB < 0 THEN 100
PSET (VB, PH), 2
PHDVB = (PH - PH1) / (VB - VB1): PSET (VB, 1 + PHDVB), 1
ALPHA = 3 * H ^ 2 * (VB + VA) + 2 * H * (VB * KA + VB * MB + VA * KA)
ALPHA = ALPHA + VB * MB * KA - VA * MA * KA - VB * KW - VA * KW
BETA = H ^ 3 + H ^ 2 * (KA + MB) + H * (KA * MB - KW) - KA * KW
DERI = -2.303 * H * ALPHA / BETA
IF DERIMIN < DERI THEN DERIMIN = DERI: VBEQ = VB: PHEQ = PH
PSET (VB, -DERI / 2), 3: PH1 = PH: VB1 = VB
100 FOR KK = 1 TO 100 STEP .05: NEXT KK: NEXT PH
CIRCLE (VBEQ, PHEQ), .5, 1
LOCATE 2, 5: PRINT "TITRATION OF ACID WITH STRONG BASE"
LOCATE 3, 1: PRINT "14": LOCATE 17, 25: PRINT "Ka=";
PRINT USING "#.#^^^^"; KA: LOCATE 18, 25: PRINT "VBEQ =";
PRINT USING "###.#"; VBEQ; : PRINT " mL"
LOCATE 19, 25: PRINT "pHEQ ="; : PRINT USING "##.##"; PHEQ
LOCATE 13, 1: PRINT "pH": LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT "0                VB---->                ";
PRINT USING "###"; VBMAX: VBS = VBMAX / 10: PHS = 1.4
FOR I = 1 TO 9: LINE (I * VBS, 0)-(I * VBS, .3)
LINE (0, I * PHS)-(VBS / 10, I * PHS): NEXT I
LINE (VBEQ, 0)-(VBEQ, 5), 2
END
```

Output



2.39 GENERAL TREATMENT OF pH TITRATION CURVE OF A DIPROTIC ACID

To display pH variation when a diprotic acid is titrated against a strong base.

The expression to be used is

$$V_b = -V_a\left(\frac{\alpha}{\beta}\right)$$

where

$\alpha = [H^+]^4 + [H^+]^3 K_{a1} + [H^+]^2 (K_{a1}K_{a2} - K_{a1}M_a - K_w) + [H^+] (-2K_{a1}K_{a2}M_a - K_wK_{a1}) - K_{a1}K_{a2}K_w$

$\beta = [H^+]^4 + [H^+]^3 (K_{a1} + M_b) + [H^+]^2 (K_{a1}K_{a2} + K_{a1}M_b - K_w) + [H^+] (K_{a1}K_{a2}M_b - K_{a1}K_w) - K_{a1}K_{a2}K_w.$

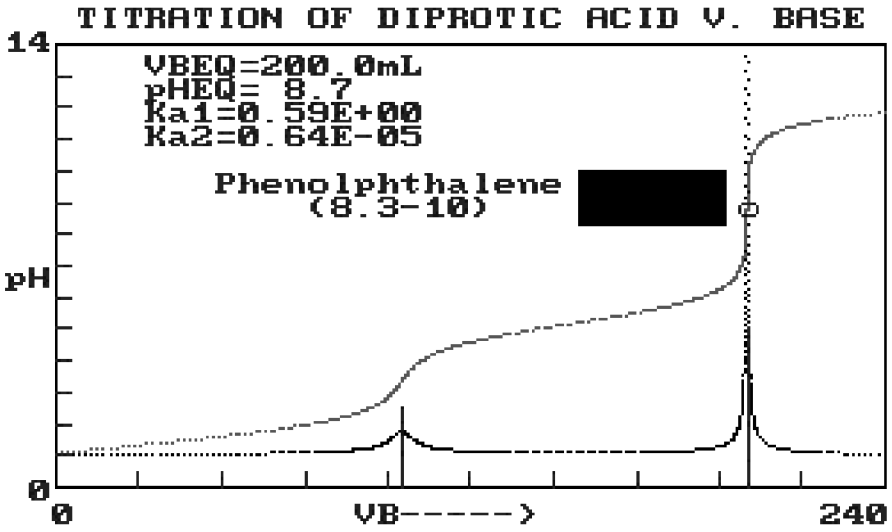
Also, draw the first derivative dpH/dV_b.

Program

```
REM PROGRAM TITDIA;GENERAL TREATMENT OF TITRATION OF DIBASIC ACID
CLS : READ VA, MA, MB, KA1, KA2: DATA 50,.1,.05,5.9E-1,6.4E-6
KW = 1E-14
VBEQ = 2 * VA * MA / MB: VMAX = VBEQ + VBEQ / 5
SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(310, 170)
WINDOW (0, 0)-(VMAX, 14): LINE (0, 0)-(VMAX, 14), 2, B
FOR PH = 0 TO 12 STEP .02
H = 10 ^ (-PH)
NUM = H ^ 4 + H ^ 3 * KA1 + H ^ 2 * (KA1 * KA2 - KA1 * MA - KW)
NUM = NUM + H * (-2 * KA1 * KA2 * MA - KW * KA1) - KA1 * KA2 * KW
DEN = H ^ 4 + H ^ 3 * (KA1 + MB) + H ^ 2 * (KA1 * KA2 + KA1 * MB - KW)
DEN = DEN + H * (KA1 * KA2 * MB - KA1 * KW) - KA1 * KA2 * KW
VB = -VA * NUM / DEN
IF VB < 0 THEN 100
```

```
PSET (VB, PH), 1
IF VBEQ - VB > .0001 THEN PHEQ = PH
PHDVB = (PH - PH1) / (VB - VB1)
PSET (VB, 1 + PHDVB / .2), 2: VB1 = VB: PH1 = PH
100 FOR KK = 1 TO 100 STEP .001: NEXT KK: NEXT PH
CIRCLE (VBEQ, PHEQ), VMAX / 100
FOR I = 1 TO 9: LINE (VMAX * .1 * I, 0)-(VMAX * .1 * I, .5): NEXT I
FOR I = 1 TO 13: LINE (0, I)-(VMAX / 60, I): NEXT I
LOCATE 2, 5: PRINT "TITRATION OF DIPROTIC ACID V. BASE"
LOCATE 3, 2: PRINT "14"
LOCATE 4, 8: PRINT "VBEQ="; : PRINT USING "###.#"; VBEQ; : PRINT "mL"
LOCATE 5, 8: PRINT "pHEQ="; : PRINT USING "##.#"; PHEQ
LOCATE 6, 8: PRINT "Ka1="; : PRINT USING "#.##^^^^"; KA1
LOCATE 7, 8: PRINT "Ka2="; : PRINT USING "#.##^^^^"; KA2
LOCATE 13, 2: PRINT "pH": LOCATE 22, 3: PRINT "0"
LOCATE 23, 3: PRINT " 0           VB----->         "; : PRINT VMAX
LINE (VBEQ / 2, 0)-(VBEQ / 2, 2.5): LINE (VBEQ, 0)-(VBEQ, 5)
LINE (VBEQ - VMAX / 5, 8.3)-(VBEQ - VMAX / 40, 10), 2, BF
LOCATE 9, 11: PRINT "Phenolphthalene": LOCATE 10, 15: PRINT "(8.3-10)"
END
```

Output



2.40 CONDUCTIVITY TITRATION CURVE OF A STRONG ACID WITH A STRONG BASE

Draw the conductivity titration curve of a strong acid solution of concentration c_a(volume of solution is V_a) with a strong base solution of concentration c_b (its concentration must be ten times or more than ten times the concentration c_a). The volume of base added each time is V_b.

Expression to be used before the equivalence point. $\kappa = \left(\frac{V_a}{V_a + V}\right)\left[\kappa_a - c_b\left(\frac{V}{V_a}\right)(\lambda_{H^+} - \lambda_{Na^+})\right]$

After the equivalence point, the expression to be used is $\kappa = \left(\frac{V_a + V_e}{V_a + V}\right)\kappa_e + \left(\frac{V - V_e}{V_a + V}\right)\kappa_b$

κ_a is the conductivity of acid solution (= Λ^∞(HCl) c_a)

κ_b is the conductivity of base solution (= Λ^∞(NaOH) c_b)

κ_e is the conductivity of the solution at the equivalence point (= Λ^∞(NaCl) * c(NaCl))

V_e is the volume of base added at the equivalence point.

$\lambda^\infty_{H^+}$ = 349.8 S cm^2 mol^{-1}

$\lambda^\infty_{Na^+}$ = 50.11 S cm^2 mol^{-1}

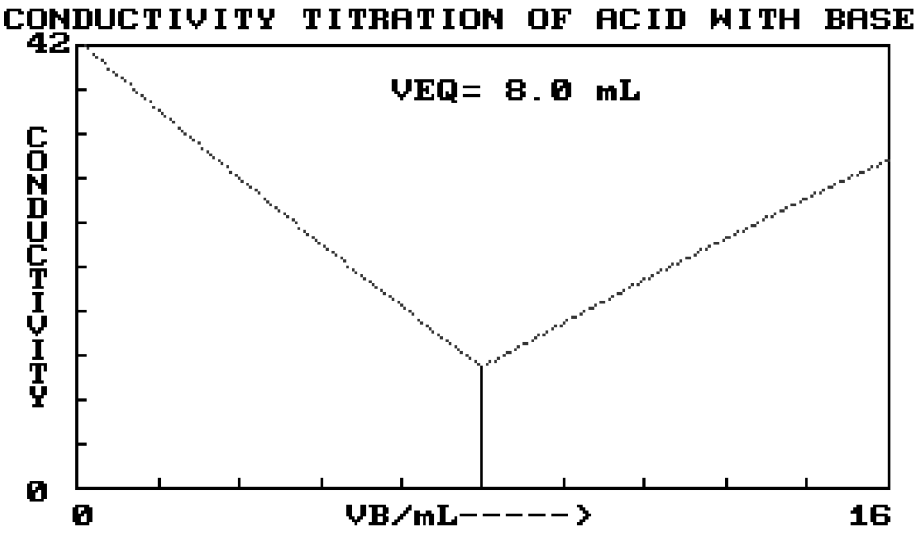
Illustration Conductirity titration curve of 100 mL of 0.1 M strong acid with 1 M of a strong base.

Program

```
REM PROGRAM TITCOND;CONDUCTOMETRIC TITRATION
CLS : READ VA, CA, CB: DATA 80,.1,1.0
VEQ = VA * CA / CB
LH = 349.8: LCl = 76.34: LNa = 50.11: LOH = 197.6
KA = (LH + LCl) * CA: KB = (LNa + LOH) * CB
KS = (LNa + LCl) * (CA * VA / (VA + VEQ))
VF = INT(2 * VEQ): VS = VA / 1000
KM = INT(KA)
SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(310, 170)
WINDOW (0, 0)-(VF, KM): LINE (0, 0)-(VF, KM), , B
FOR V = 0 TO VF STEP VS
IF V = 0 THEN
K = KA
ELSEIF V < VEQ THEN
K = (VA / (VA + V)) * (KA - CB * (V / VA) * (LH - LNa))
ELSEIF V = VEQ THEN
K = KS
ELSEIF V > VEQ THEN
K = ((VA + VEQ) / (VA + V)) * KS + ((V - VEQ) / (VA + V)) * KB
END IF
PSET (V, K), 1
FOR KK = 1 TO 100 STEP .001: NEXT KK: NEXT V
FOR I = 1 TO 9
LINE (I * VF / 10, 0)-(I * VF / 10, KM / 50)
LINE (0, I * KM / 10)-(VF / 100, I * KM / 10)
```

```
NEXT I
LOCATE 2, 1: PRINT "CONDUCTIVITY TITRATION OF ACID WITH BASE"
LOCATE 3, 1: PRINT USING "###"; KM
LOCATE 5, 18: PRINT "VEQ="; : PRINT USING "##.#"; VEQ; : PRINT " mL"
A$ = "CONDUCTIVITY"
FOR I = 1 TO 12
B$ = MID$(A$, I, 1): LOCATE 6 + I, 2: PRINT B$
NEXT I
LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT " 0              VB/mL----->             ";
PRINT USING "###"; VF: LINE (VEQ, 0)-(VEQ, KS), 2
END
```

Output



2.41 DISPLAY OF A SIMPLE EUTECTIC PHASE DIAGRAM

For an ideal system, the variation of solubility of a component with temperature as given by van't Hoff relation is

$$\ln\ x = -\frac{\Delta_{\text{fus}}H^{\circ}_{\text{m}}}{R}\left(\frac{1}{T}-\frac{1}{T^{*}}\right)$$

where x is the saturation solubility in amount fraction of the component and T^{*} is the freezing point of the pure component. For a binary system, the two solubility curves can be drawn by using the above expression with the appropriate value of $\Delta_{\text{fus}}H^{\circ}_{\text{m}}$. Either one can vary temperature to compute amount fraction or vice versa. The two curves are drawn till they meet each other which is the eutectic point of the system.

Table 1 Includes the data on melting points and enthalpy of fusion for a few compounds.

Table 1 Melting Points and Enthalpy of Fusion for a Few Compouds

Compound	$\frac{m.pt}{°C}$	$\Delta_{fus}H/J\ mol^{-1}$
Naphthalene	79.9	19 066
Benzoic acid	121.8	17 304
Cinnamic acid	133	22 602
Resorcinol	110	21 263
Sodium Chloride	801	28 870
Sodium sulphate	881	24 060

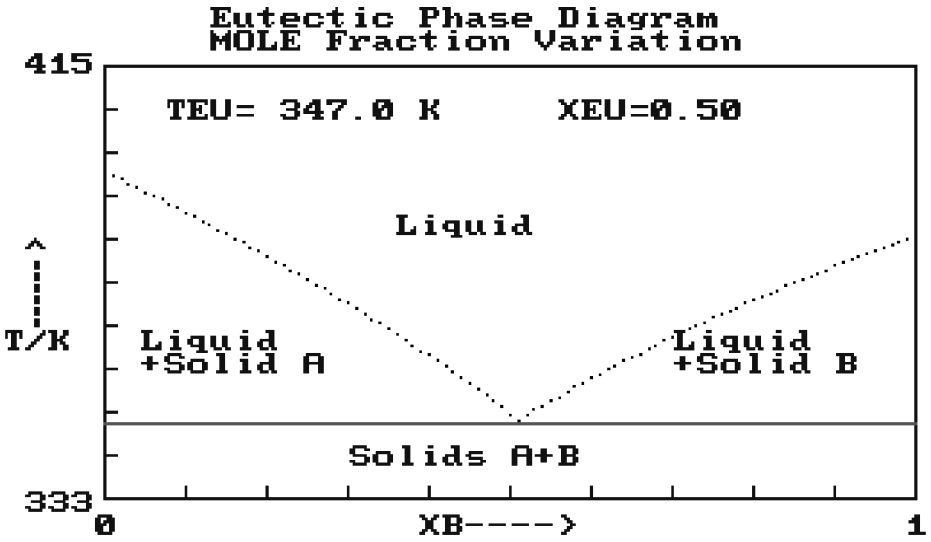
Illustration Phase diagram of resorcianol and cinnamic acid.

Program (Variation of Mole Fraction)

```
REM PROGRAM EUTECT;EUTECTIC PHASE DIAGRAM VIA MOLE FRACTION
READ TA, HA, TB, HB
R = 8.314
IF TA > TB THEN TMAX = TA + 20 ELSE TMAX = TB + 20
IF TA < TB THEN TMIN = TA - 50 ELSE TMIN = TB - 50 'FOR NACL TB-250
SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(315, 170)
WINDOW (0, TMIN)-(1, TMAX): LINE (0, TMIN)-(1, TMAX), , B
FOR X = .01 TO .99 STEP .01
T1 = -(R / HA) * LOG(1 - X) + 1 / TA: T1 = 1 / T1
T2 = -(R / HB) * LOG(X) + 1 / TB: T2 = 1 / T2
IF T1 > T2 THEN PSET (X, T1), 2: TE = T2: XE = X ELSE PSET (X, T2), 2
FOR I = 1 TO 100 STEP .001: NEXT I
NEXT X
LOCATE 5, 8: PRINT "TEU="; : PRINT USING "####.#"; TE;
PRINT " K"; : PRINT "      XEU="; : PRINT USING "#.##"; XE
TS1 = (TMAX - TMIN) / 10: TS = TS1 / 4
FOR I = 1 TO 9
LINE (I * .1, TMIN)-(I * .1, TMIN + TS)
LINE (0, TMIN + TS1 * I)-(.015, TMIN + TS1 * I)
NEXT I
LOCATE 1, 10: PRINT "Eutectic Phase Diagram"
LOCATE 2, 10: PRINT "MOLE Fraction Variation"
LINE (0, TE)-(1, TE), 1: LOCATE 3, 1: PRINT USING "####"; TMAX
LOCATE 10, 18: PRINT "Liquid"
FOR I = 1 TO 3: LOCATE 15 - I, 2: PRINT STRING$(1, 124): NEXT I
LOCATE 11, 2: PRINT CHR$(94): LOCATE 15, 1: PRINT "T/K"
LOCATE 15, 6: PRINT " Liquid": LOCATE 15, 30: PRINT "Liquid"
LOCATE 16, 7: PRINT "+Solid A": LOCATE 16, 30: PRINT "+Solid B"
LOCATE 20, 16: PRINT "Solids A+B"
LOCATE 22, 1: PRINT USING "####"; TMIN
```

```
LOCATE 22, 6: PRINT "     0               XB---->                1"
REM Benzoic acid-Naphthalene:TE=330.6 K;XE=.64
'DATA 395,17304,353,19066
REM Benzoic acd-Cinnamic acid;TE=358.8 K;XE=.41
'DATA 395,17304,406,22602
REM Benzoic acid-Resorniol;TE=348 K;XE=.51
DATA 395,17304,383,21263
REM Resorcinol-Cinnamic acid;TE=356.3 K;XE=.39
'DATA 383,21263,406,22602
REM NaCl-Na2SO4;TE=892.5 K;XE=.48
'DATA 1074,28870,1152,24060
END
```

Output



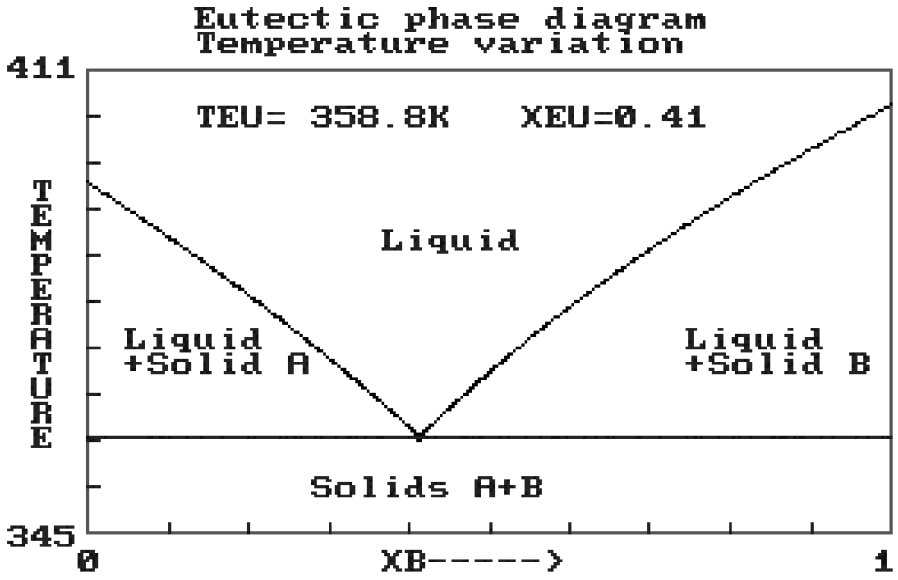
Excercise Determine eutectic temperature and composition for each of the following pairs 1. Naphthalene-Benzoic acid, 2. Benzoic acid-Cinnamic acid, 3. Benzoic acid-Resorcinol., and 4. NaCl-Na_2SO_4.

Program (Variation of Temperature)

```
REM PROGRAM EUTECX;EUTECTIC DIAGRAM VIA TEMPERATURE DETERMINATION
READ TA, HA, TB, HB
'DATA 1074,28870,1152,24060:REM NaCl-NaSO4
DATA 395,17304,406,22602: REM Benzoic acid-Cinnamic acid
'DATA 353,19066,395,17304: REM Naphthalene-Benzoic acid
'DATA 383,21263,395,17304: REM Resorcinol-Benzoic acid
```

```
'DATA 383,21263,406,22602: REM Resorcinol-Cinnamic acid
CLS : R = 8.314
IF TA > TB THEN TMAX = TA + 5 ELSE TMAX = TB + 5
IF TA < TB THEN TMIN = TA - 50 ELSE TMIN = TB - 50
SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(310, 170)
WINDOW (0, TMIN)-(1, TMAX): LINE (0, TMIN)-(1, TMAX), 1, B
FOR T = TMAX TO TMIN STEP -.2
IF T > TA THEN 5
XA = 1 - EXP((-HA / R) * (1 / T - 1 / TA)): PSET (XA, T), 2
5 IF T > TB THEN 15
XB = EXP((-HB / R) * (1 / T - 1 / TB)): PSET (XB, T), 2
15 IF XA < XB THEN TE = T: XE = XA
IF XB < XE THEN 20
FOR I = 1 TO 100 STEP .01: NEXT I
NEXT T
20 LOCATE 1, 10: PRINT "Eutectic phase diagram"
LOCATE 2, 10: PRINT "Temperature variation"
LOCATE 5, 10: PRINT "TEU="; : PRINT USING "####.#"; TE;
PRINT "K    XEU="; : PRINT USING "#.##"; XE
TS = (TMAX - TMIN) / 50: TS1 = (TMAX - TMIN) / 10
FOR I = 1 TO 9
LINE (I * .1, TMIN)-(I * .1, TMIN + TS)
LINE (0, TMIN + TS1 * I)-(.015, TMIN + TS1 * I)
NEXT I
LOCATE 3, 1: PRINT USING "####"; TMAX
LINE (0, TE)-(1, TE): B$ = "TEMPERATURE"
FOR I = 1 TO 11: X$ = MID$(B$, I, 1): LOCATE 7 + I, 3
PRINT X$: NEXT I
LOCATE 10, 18: PRINT "Liquid"
LOCATE 14, 7: PRINT "Liquid": LOCATE 14, 31: PRINT "Liquid"
LOCATE 15, 7: PRINT "+Solid A": LOCATE 15, 31: PRINT "+Solid B"
LOCATE 20, 15: PRINT "Solids A+B"
LOCATE 22, 1: PRINT USING "####"; TMIN
LOCATE 22, 5: PRINT "     0              XB----->               1"
END
```

Output



2.42 DISPLAY OF A PHASE DIAGRAM EXHIBITING A COMPLETE SERIES OF SOLID SOLUTION

In a complete series of solid solution, the amount fractions of solid and liquid solutions in equilibrium are given by the expressions

$$x_s = \frac{\exp(-a) - 1}{\exp(-a) - \exp(-b)} \quad \text{and} \quad x_l = x_s \exp(-b)$$

where $$a = (\Delta_{fus} H^{\circ}_{A,m} / R)\left(\frac{1}{T} - \frac{1}{T^*_A}\right) \quad \text{and} \quad b = (\Delta_{fus} H^{\circ}_{B,m} / R)\left(\frac{1}{T} - \frac{1}{T^*_B}\right)$$

Note The above expressions are also applicable to ideal liquid solution $\rightleftharpoons$ ideal vapours, where ΔH_A and ΔH_B, respectively, represent enthalpy of vaporization of the components A and B. For liquids following Trouton's rule, $\Delta H_A / R T^*_A = \Delta H_B / R T^*_B = 10.6$.

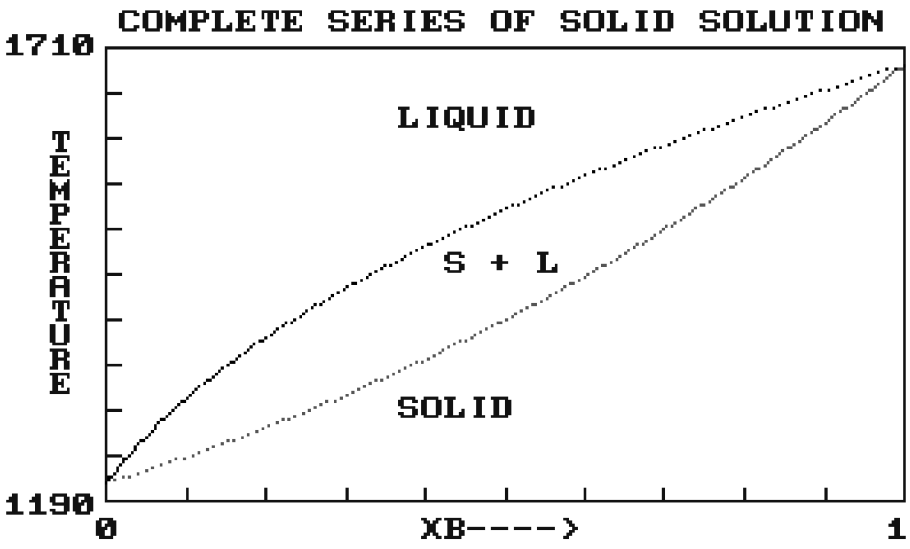
Illustration Phase diagram for the solid $A(\Delta H_A = 31\,800$ J mol^{-1}, $T^*_A = 1\,210$ K) and solid $B(\Delta H_B = 50\,800$ J mol^{-1}, $T^*_B = 1\,690$ K)

Program

```
REM PROGRAM SOLDIR;DIRECT PLOTS OF SOLID-LIQUID SOLUTIONS
H1 = 31800: T1 = 1210: H2 = 50800: T2 = 1690: R = 8.314
SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(310, 170)
IF T1 < T2 THEN
HA = H1: TA = T1: HB = H2: TB = T2
ELSE HA = H2: TA = T2: HB = H1: TB = T1
END IF
```

```
TMIN = TA - 20: TMAX = TB + 20
WINDOW (0, TMIN)-(1, TMAX): LINE (0, TMIN)-(1, TMAX), , B
FOR T = TA TO TB STEP 3
A = HA / R * (1 / T - 1 / TA)
B = HB / R * (1 / T - 1 / TB)
XS = (EXP(-A) - 1) / (EXP(-A) - EXP(-B))
XL = EXP(-B) * XS
PSET (XS, T), 1: PSET (XL, T), 2
FOR KK = 1 TO 100 STEP .001: NEXT KK: NEXT T
LOCATE 2, 6: PRINT "COMPLETE SERIES OF SOLID SOLUTION"
LOCATE 3, 1: PRINT USING "####"; TMAX
LOCATE 6, 18: PRINT "LIQUID "
LOCATE 12, 20: PRINT "S + L": A$ = "TEMPERATURE"
FOR I = 1 TO 11: LOCATE 6 + I, 3: PRINT MID$(A$, I, 1): NEXT I
LOCATE 18, 18: PRINT "SOLID"
LOCATE 22, 1: PRINT USING "####"; TMIN
LOCATE 23, 4: PRINT " 0                XB---->                1"
FOR I = 1 TO 9
LINE (I * .1, TMIN)-(I * .1, TMIN + 15)
TS = TMIN + (TMAX - TMIN) * I / 10: LINE (0, TS)-(.02, TS)
NEXT I
END
```

Output



2.43 DISPLAY OF THE CONCENTRATIONS OF A, B AND C IN THE CONSECUTIVE REACTIONS A → B → C

To display the concentrations of A, B and C in the consecutive reactions

$$A \xrightarrow{k_1} B \qquad ; \qquad B \xrightarrow{k_2} C$$

in which (i) $k_1 > k_2$ and (ii) $k_1 < k_2$.

The concentrations A, B and C are given by the expressions

$$[A] = [A]_0 \, e^{-k_1 t}$$

$$[B] = [A]_0 \left(\frac{k_1}{k_2 - k_1} \right) (e^{-k_1 t} - e^{-k_2 t})$$

$$[C] = [A]_0 \left[1 - \frac{1}{k_2 - k_1} \left(k_2 e^{-k_1 t} - k_1 e^{-k_2 t} \right) \right]$$

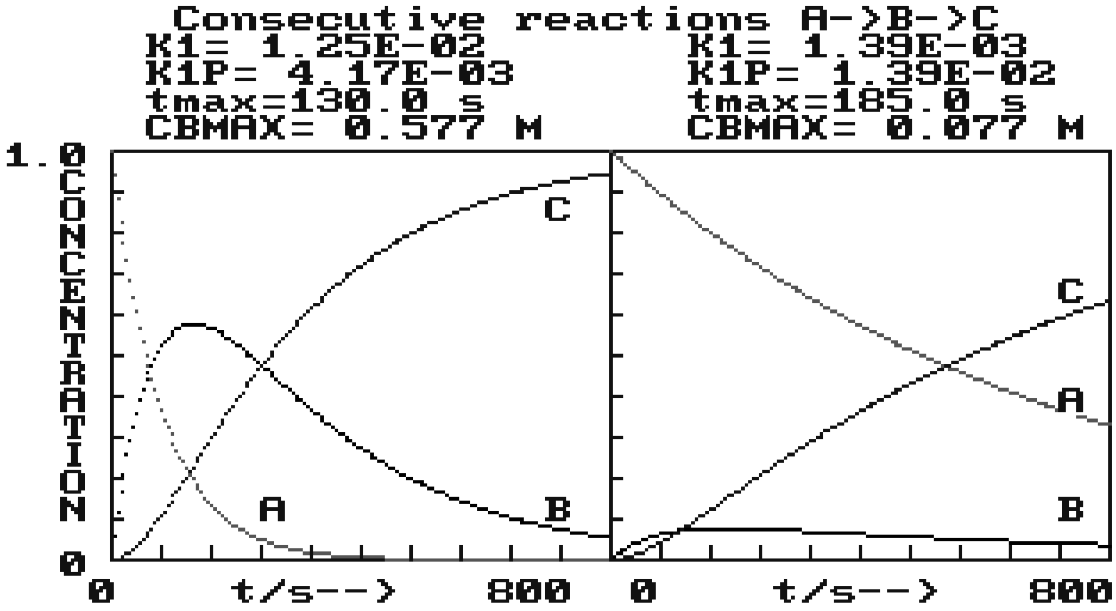
Illustration Display the variations of [A], [B] and [C] with time for (i) $k_1 = (45/3\,600)\ s^{-1}$ and $k_1' = (15/3\,600)\ s^{-1}$, and (ii) $k_1 = (5/3\,600)\ s^{-1}$ and $k_1' = (50/3\,600)\ s^{-1}$.

Program

```
REM PROGRAM KINABC;COSECUTIVE REACTIONS
CLS : CA0 = 1: TF = 800
SCREEN 1: COLOR 15, 0
FOR J = 1 TO 2
ON J GOTO 5, 10
5 K1 = 45 / 3600: K1P = 15 / 3600
GOTO 15
10 K1 = 5 / 3600: K1P = 50 / 3600
15 CBMAX = 0
VIEW (30 + 140 * (J - 1), 50)-(170 + 140 * (J - 1), 170)
WINDOW (0, 0)-(800, 1): LINE (0, 0)-(800, 1), , B
FOR T = 0 TO TF STEP 5
CA = CA0 * EXP(-K1 * T)
TERM = EXP(-K1 * T) - EXP(-K1P * T)
CB = CA0 * (K1 / (K1P - K1)) * TERM
TERM = K1P * EXP(-K1 * T) - K1 * EXP(-K1P * T)
CC = CA0 * (1 - (1 / (K1P - K1)) * TERM)
PSET (T, CA), 1: PSET (T, CB), 2: PSET (T, CC), 3
FOR I = 1 TO 1000 STEP .01: NEXT I
IF CBMAX < CB THEN CBMAX = CB: TMAX = T
NEXT T
FOR I = 1 TO 9
LINE (TF * I / 10, 0)-(TF * I / 10, .03)
LINE (0, CA0 * I / 10)-(15, CA0 * I / 10)
```

```
NEXT I
LOCATE 2, 7: PRINT "Consecutive reactions A->B->C"
LOCATE 7, 1: PRINT USING "#.#"; CA0
LOCATE 3, 6 + 19 * (J - 1): PRINT "K1="; : PRINT USING "##.##^^^^"; K1
LOCATE 4, 6 + 19 * (J - 1): PRINT "K1P="; : PRINT USING "##.##^^^^"; K1P
LOCATE 5, 6 + 19 * (J - 1): PRINT "tmax="; : PRINT USING "###.#"; TMAX;
PRINT " s": LOCATE 6, 6 + 19 * (J - 1): PRINT "CBMAX=";
PRINT USING "##.###"; CBMAX; : PRINT " M"
B$ = "CONCENTRATION"
FOR I = 1 TO 13: C$ = MID$(B$, I, 1)
LOCATE I + 7, 3: PRINT C$: NEXT I
LOCATE 22, 3: PRINT "0"
LOCATE 23, 2 + 19 * (J - 1): PRINT "   0     t/s-->    ";
PRINT USING "###"; TF
ON J GOTO 20, 25
20 LOCATE 20, 10: PRINT "A": LOCATE 20, 20
PRINT "B": LOCATE 9, 20: PRINT "C"
GOTO 30
25 LOCATE 20, 38: PRINT "B": LOCATE 16, 38
PRINT "A": LOCATE 12, 38: PRINT "C"
30 A$ = INPUT$(1)
NEXT J
END
```

Output



2.44 DISPLAY OF WAVE FUNCTIONS OF A PARTICLE IN A BOX

To display the first few wave functions and their square for a particle in a one-dimensional box.

The wave functions are given by

$$\Psi_n = \sqrt{\frac{2}{l}} \sin\left(\frac{n\pi}{l} x\right) \quad ; \quad n = 1, 2, 3, \ldots$$

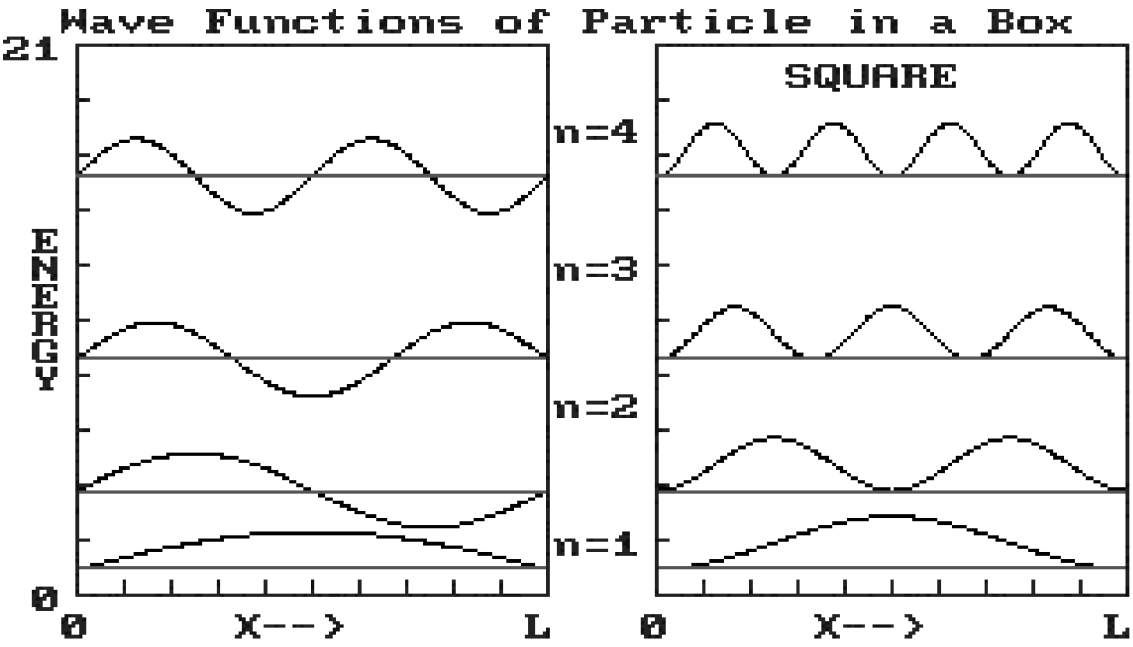
Take $l = 1$ and vary x from 0 to l. Take y-axis as the energy of the particle in a box.

Program

```
REM PROGRAM WFBOX;WAVE FUNCTIONS OF PARTICLE IN A BOX
CLS : L = 1: M = 4: YM = M ^ 2 + 5: A$ = "ENERGY"
SCREEN 1: COLOR 15, 0
FOR N = 1 TO M: FOR X = 0 TO L STEP .005
REM -------Plot of wave functions-------
VIEW (20, 10)-(150, 170)
WINDOW (0, 0)-(L, YM): LINE (0, 0)-(L, YM), , B
Y = SQR(2 / L) * SIN(N * 3.14159 * X / L)
LINE (0, N ^ 2)-(L, N ^ 2), 1:  PSET (X, Y + N ^ 2), 2
FOR I = 1 TO 9: II = I * L * .1: LINE (II, 0)-(II, .5)
JJ = YM * .1 * I: LINE (0, JJ)-(.02, JJ)
'LINE (II, 0)-(II, YM): LINE (0, JJ)-(L, JJ)
NEXT I
REM ------Plot of square of wave functions------
VIEW (180, 10)-(310, 170)
WINDOW (0, 0)-(L, YM): LINE (0, 0)-(L, YM), , B
FOR I = 1 TO 9: II = I * L * .1: LINE (II, 0)-(II, .5)
YY = YM * .1 * I: LINE (0, YY)-(.02, YY)
'LINE (II, 0)-(II, YM)
' LINE (0, YY)-(L, YY)
NEXT I
Y2 = (SQR(2 / L) * SIN(N * 3.14159 * X / L)) ^ 2
LINE (0, N ^ 2)-(L, N ^ 2), 1: PSET (X, Y2 + N ^ 2), 2
FOR KK = 1 TO 100 STEP .1: NEXT KK: NEXT X: NEXT N
LOCATE 1, 3: PRINT "Wave Functions of Particle in a Box"
LOCATE 3, 28: PRINT "SQUARE"
FOR I = 1 TO 6: LOCATE 8 + I, 2: PRINT MID$(A$, I, 1): NEXT I
LOCATE 22, 2: PRINT "0": LOCATE 2, 1: PRINT USING "##"; YM
```

```
LOCATE 23, 1: PRINT "   0      X-->       L    0      X-->       L"
'LOCATE 5, 20: PRINT "n=5": LOCATE 11, 20: PRINT "n=4"
'LOCATE 16, 20: PRINT "n=3": LOCATE 19, 20: PRINT "n=2"
'LOCATE 21, 20: PRINT "n=1"
FOR I = 1 TO M: LOCATE (1 + 16 / M) * I, 20
PRINT "n="; : PRINT USING "#"; M + 1 - I: NEXT I
END
```

Output



2.45 DISPLAY OF WAVE FUNCTIONS OF A HARMONIC OSCILATOR

To display the first few wave functions of a harmonic oscillator.

The wave functions of harmonic oscillator are given by

$$\Psi_\upsilon = \left(\frac{\sqrt{\alpha}}{2^\upsilon \upsilon! \sqrt{\pi}}\right) H_\upsilon \exp(-y^2/2)$$

where $y = \sqrt{\alpha} x$

$$\alpha = \frac{4\pi^2 m v_0}{h} \quad \text{where} \quad v_0 = \frac{1}{2\pi}\sqrt{\frac{k_f}{\mu}}$$

H is Hermite polynomials. The first few Hermite polynomials are

$H_0 = 1$ $\qquad H_4 = 16y^4 - 48y^2 + 12$

$H_1 = 2y$ $\qquad H_5 = 32y^5 - 160y^3 + 120y$

$H_2 = 4y^2 - 2$ $\qquad H_6 = 64y^6 - 480y^4 + 720y^2 - 120$

$H_3 = 8y^3 - 12y$

Alternatively, Hermite polynomials may be generated by using the recurrsion expression.

$$H_{\upsilon+1} = 2yH_\upsilon + 2\upsilon H_{\upsilon-1} \quad \text{with} \quad H_0 = 1 \quad \text{and} \quad H_1 = 2y$$

Plot the wave functions for CO for which $M_C = 0.012$ kg mol^{-1} and $M_O = 0.016$ kg mol^{-1}, The expression of the reduced mass is

$$\mu = \frac{M_1 M_2}{(M_1 + M_2)} \frac{1}{N_A}$$

Vary x from $-x_{max}$ to x_{max} where x_{max} is given by the expression

$$x_{max} = \left[\frac{(2\upsilon_{max} + 1)hv_0}{k_f}\right]^{1/2} + = \left[\frac{(2\upsilon_{max} + 1)\,h}{(2\pi)^2 v_o\ \mu}\right]^{1/2}$$

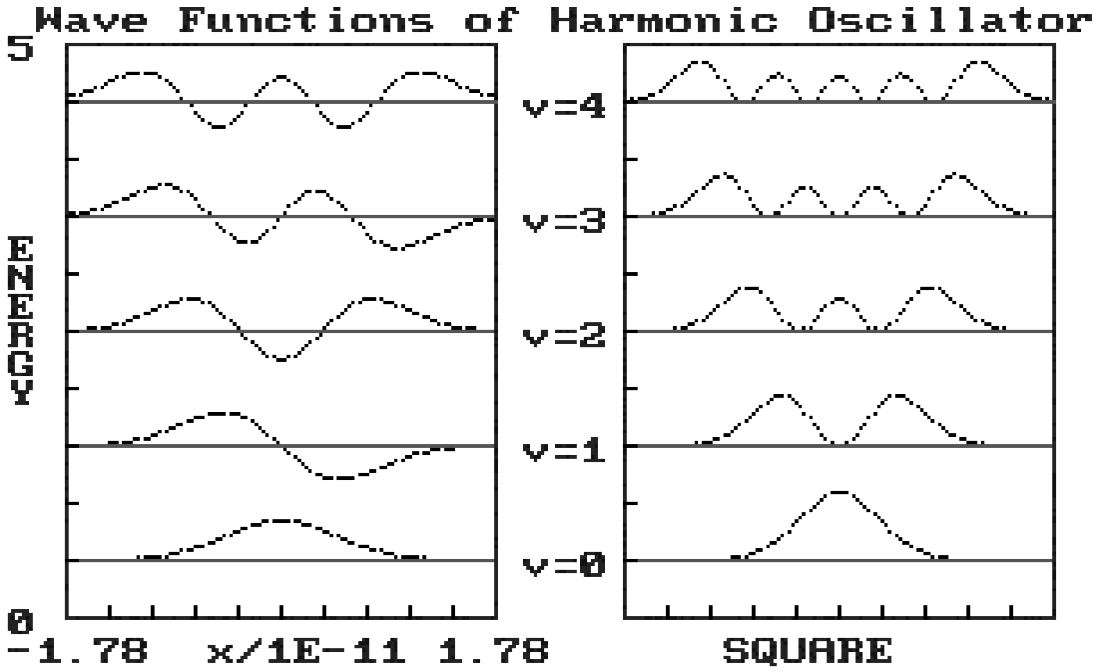
with $\upsilon_{max} = 5$. Plot the wave functions for $\upsilon = 0$, 1, 2, 3 and 4.

Program

```
REM PROGRAM WFOSCIL1;WAVE FUNCTIONS OF HARMONIC OSCILLATOR
CLS : DIM HP(20)
READ VMAX: DATA 4: A$ = "ENERGY"
NA = 6.022E+23: PI = 3.14159: M1 = .012: M2 = .016
M = M1 * M2 / ((M1 + M2) * NA): NU0 = 6.5E+13: VM1 = VMAX + 1
AL = M * NU0 * 4 * PI ^ 2 / 6.626E-34: XM1 = VMAX + 3
XM = (2 * XM1 * 6.626E-34 / ((2 * PI) ^ 2 * NU0 * M)) ^ .5
XS = XM / 50: HP(0) = 1
SCREEN 1: COLOR 15, 0
FOR V = 0 TO VMAX
FACT = 1
IF V = 0 OR V = 1 THEN 4
FOR I = 2 TO V: FACT = FACT * I: NEXT I
4 FOR X = -XM TO XM STEP XS
Y = AL ^ .5 * X: HP(1) = 2 * Y
IF V = 0 OR V = 1 THEN 6
FOR I = 1 TO V - 1
HP(I + 1) = 2 * Y * HP(I) - 2 * I * HP(I - 1)
NEXT I
```

```
6 H = HP(V)
REM -----PLOTS OF WAVE FUNCTIONS--------
VIEW (24, 10)-(144, 170)
WINDOW (-XM, 0)-(XM, VM1): LINE (-XM, 0)-(XM, VM1), , B
SAI = (AL ^ .5 / (2 ^ V * FACT * PI ^ .5)) ^ .5
SAI = SAI * H * EXP(-Y ^ 2 / 2) * (-1) ^ V
LINE (-XM, V + .5)-(XM, V + .5), 1
PSET (X, (V + .5) + SAI * .000001), 2
FOR I = 1 TO 9: XX = -XM + 2 * XM * .1 * I: YY = VM1 * .1 * I
LINE (XX, 0)-(XX, .1), 2
LINE (-XM, YY)-(-XM + 1E-12, YY), 2
'LINE (XX, 0)-(XX, VM1): LINE (-XM, YY)-(XM, YY)
NEXT I
REM------PLOTS OF SQUARE OF WAVE FUNCTIONS------
VIEW (180, 10)-(300, 170)
WINDOW (-XM, 0)-(XM, VM1): LINE (-XM, 0)-(XM, VM1), , B
PSET (X, (V + .5) + SAI ^ 2 * 5E-12), 2
NEXT X
LINE (-XM, V + .5)-(XM, V + .5), 1
FOR I = 1 TO 1000 STEP .001: NEXT I
NEXT V
LOCATE 1, 3: PRINT "Wave Functions of Harmonic Oscillator"
LOCATE 2, 1: PRINT VM1
FOR I = 1 TO 6: LOCATE 8 + I, 2: PRINT MID$(A$, I, 1): NEXT I
LOCATE 22, 2: PRINT "0"
LOCATE 23, 2: PRINT USING "##.##"; -XM * 1E+11;
PRINT "  x/1E-11 "; : PRINT USING "#.##"; XM * 1E+11
LOCATE 23, 27: PRINT "SQUARE"
FOR I = 1 TO 9
XX = -XM + 2 * XM * .1 * I: YY = VM1 * .1 * I
LINE (XX, 0)-(XX, .1), 2
LINE (-XM, YY)-(-XM + 1E-12, YY), 2
'LINE (XX, 0)-(XX, VM1): LINE (-XM, YY)-(XM, YY)
NEXT I
FOR I = 0 TO VMAX
LOCATE 4 + (20 / VM1) * I, 20: PRINT "v=";
PRINT USING "#"; VMAX - I: NEXT I
END
```

Output



2.46 DISPLAY OF *r*–DEPENDENT WAVE FUNCTIONS OF A HYDROGEN ATOM

To display the variations of $R_{n,l}$ v. r/a_0, $R^2_{n,l}$ v. r/a_0 and $r^2R^2_{n,l}$ v. r/a_0 for the following wave functions of hydrogen atom. Also determine the value(s) of r/a_0 at which the plots of R^2 and r^2R^2 exhibit maximum and the value of r_{av} (= $< R|r|R >$).

Wave function $R_{1,0}$ $$R_{1,0} = 2\left(\frac{Z}{a_0}\right)^{3/2} \exp(-Zr/a_0)$$

Wave function $R_{2,0}$ $$R_{2,0} = \left(\frac{Z}{2a_0}\right)^{3/2}\left(2 - \frac{Zr}{a_0}\right) \exp(-Zr/2a_0)$$

Wave function $R_{2,1}$ $$R_{2,1} = \frac{1}{\sqrt{3}}\left(\frac{Z}{2a_0}\right)^{3/2}\left(\frac{Zr}{a_0}\right) \exp(-Zr/2a_0)$$

Wave function $R_{3,0}$ $$R_{3,0} = \frac{2}{3}\left(\frac{Z}{3a_0}\right)^{3/2}\left(3 - \frac{2Zr}{a_0} + \frac{2Z^2r^2}{9a_0^2}\right) \exp(-Zr/3a_0)$$

Wave function $R_{3,1}$
$$R_{3,1} = \frac{2\sqrt{2}}{9}\left(\frac{Z}{3a_0}\right)^{3/2}\left(\frac{2Zr}{a_0} - \frac{Z^2r^2}{3a_0^2}\right)\exp(-Zr/3\ a_0)$$

Wave function $R_{3,2}$
$$R_{3,2} = \frac{4}{27\sqrt{10}}\left(\frac{Z^2r^2}{a_0^2}\right)\exp(-Zr/3\ a_0)$$

Wave function $R_{4,0}$
$$R_{4,0} = \frac{1}{768}\left(\frac{Z}{a_0}\right)^{3/2}\left[192 - 144\left(\frac{Zr}{a_0}\right) + 24\left(\frac{Zr}{a_0}\right)^2 - \left(\frac{Zr}{a_0}\right)^3\right]\exp\left(-\frac{Zr}{4a_0}\right)$$

Wave function $R_{4,1}$
$$R_{4,1} = \frac{3}{768\sqrt{15}}\left(\frac{Z}{a_0}\right)^{3/2}\left[80 - 20\left(\frac{Zr}{a_0}\right) + \left(\frac{Zr}{a_0}\right)^2\right]\left(\frac{Zr}{a_0}\right)\exp\left(-\frac{Zr}{4a_0}\right)$$

Wave function $R_{4,2}$
$$R_{4,2} = \frac{1}{768\sqrt{5}}\left(\frac{Z}{a_0}\right)^{3/2}\left[12 - \left(\frac{Zr}{a_0}\right)\right]\left(\frac{Zr}{a_0}\right)^2\exp\left(-\frac{Zr}{4a_0}\right)$$

Wave function $R_{4,3}$
$$R_{4,3} = \frac{1}{768\sqrt{35}}\left(\frac{Z}{a_0}\right)^{3/2}\left(\frac{Zr}{a_0}\right)^3\exp\left(-\frac{Zr}{4a_0}\right)$$

Summary of Maxima The values of r at which the plots (of the output) exhibit maxima and the values of $< r >$ for the above mentioned orbitals are shown in Table 2.46.1.

Table 2.46.1 Summary of maxima in R^2, r^2R^2 and r_{av} plots

Orbital	R^2	r^2R^2	r^*_{av}
1s		1	1.5
2s	4.0	0.75 5.25	6.0
2p	2.0	4.0	5.0
3s	3.6 11.6	0.80 4.30 13.20	13.5
3p	1.8 10.3	3.0 12.1	12.5
3d	6.0	9.0	10.5
4s	3.5 10.0 22.8	0.8 4.1 10.7 24.7	24.0
4p	1.7 8.8 21.6	2.9 9.7 23.6	23.0
4d	6.8 21.2	5.1 19.0	21.0
4f	12.0	16.0	18.0

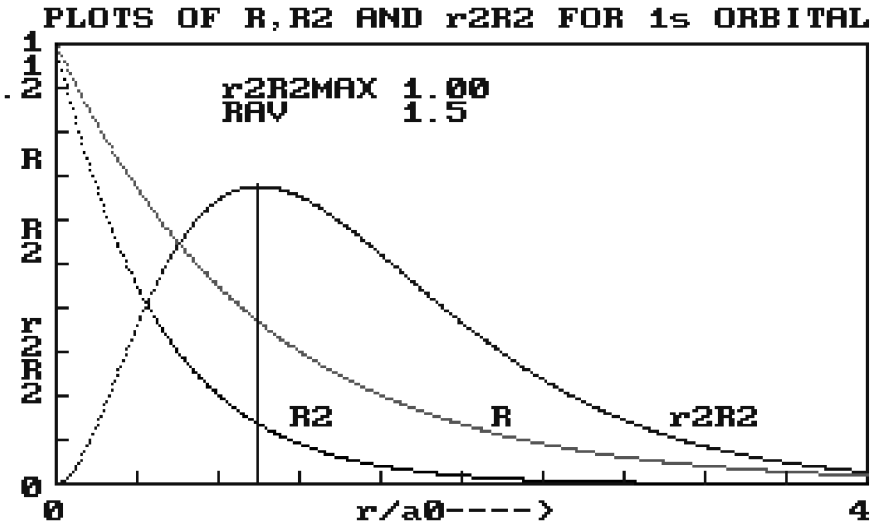
* Theoretically, $< r > = [3n^2 - l\,(l + 1)]\,(a_0/2)$.

Program for $R_{1,0}$

```
REM PROGRAM PLOT1S;PLOTS OF R,R2,r2R2 FOR 1s ORBITAL
CLS : SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(310, 170)
WINDOW (0, 0)-(4, 1): LINE (0, 0)-(4, 1), , B
```

```
AVR = 0: AVD = 0
FOR R = 0 TO 6 STEP .01
SAI = EXP(-R): SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
IF RADMAX < RAD THEN RADMAX = RAD: RMAX = R
PSET (R, SAI), 1: PSET (R, SAI2), 2: PSET (R, 5 * RAD), 3
AVR = AVR + RAD * R: AVD = AVD + RAD
'FOR KK = 1 TO 100 STEP .0005: NEXT KK
NEXT R
FOR I = 1 TO 9: LINE (I * .4, 0)-(I * .4, .03)
LINE (0, I * .1)-(.05, I * .1): NEXT I
A$ = "R   R2   r2R2": LINE (RMAX, 0)-(RMAX, 5 * RADMAX)
LOCATE 2, 3: PRINT "PLOTS OF R,R2 AND r2R2 FOR 1s ORBITAL"
LOCATE 3, 2: PRINT "1": LOCATE 4, 2: PRINT "1"
LOCATE 5, 1: PRINT ".2": LOCATE 5, 8
PRINT "    r2R2MAX"; : PRINT USING " #.##"; RMAX
LOCATE 6, 11: PRINT "RAV"; : PRINT USING "      #.#"; AVR / AVD
FOR I = 1 TO 11: B$ = MID$(A$, I, 1)
LOCATE 7 + I, 2: PRINT B$: NEXT I
LOCATE 19, 14: PRINT "R2": LOCATE 19, 23
PRINT "R": LOCATE 19, 31: PRINT "r2R2"
LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT "0                r/a0---->                4"
END
```

Output

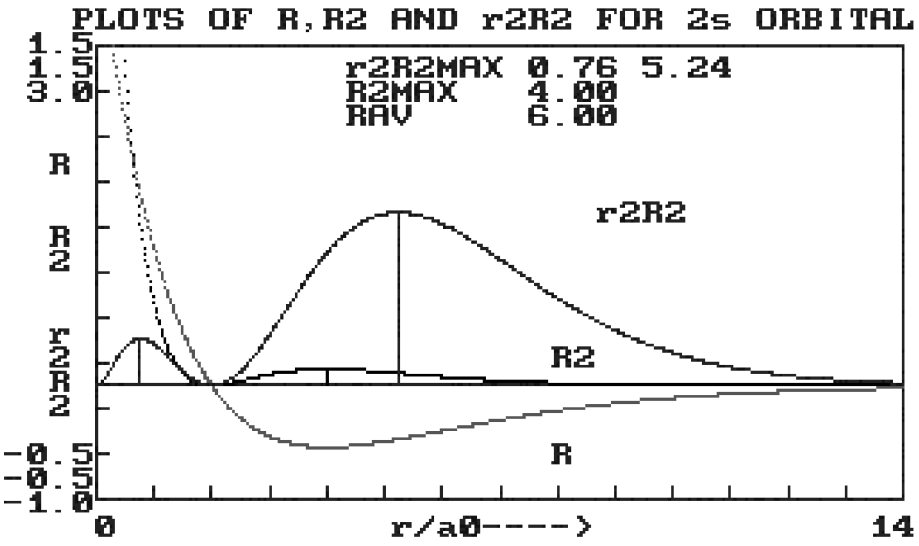


Program for $R_{2,0}$

```
REM PROGRAM PLOT2S;PLOTS OF R,R2,r2R2 FOR 2s ORBITAL
CLS : R1 = 0: R2 = 14: S1 = -1: S2 = 3: RADMAX1 = 0
RI = 0: R2MAX = 100
SCREEN 1: COLOR 15, 0: VIEW (32, 20)-(315, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0): J = 1: K = 1
AVR = 0: AVD = 0
FOR R = 0 TO R2 + 4 STEP .02
SAI = (2 - R) * EXP(-R / 2): SAI2 = SAI ^ 2
RAD = R ^ 2 * SAI2
ON K GOTO 3, 4
3 IF R2MAX > SAI2 THEN R2MAX = SAI2: GOTO 5
R2MAX = SAI2: K = 2: GOTO 5
4 IF R2MAX > SAI2 THEN 5
R2MAX = SAI2: MR2 = R
5 ON J GOTO 10, 30
10 IF RADMAX > RAD THEN 20
RADMAX = RAD: RADMAX1 = RAD: RI1 = R
GOTO 30
20 J = 2
30 IF RADMAX1 > RAD THEN 40
RADMAX1 = RAD: RI2 = R
40 PSET (R, 2 * SAI), 1: PSET (R, 2 * SAI2), 2
PSET (R, RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .1: NEXT I
NEXT R
LINE (RI1, 0)-(RI1, RADMAX): LINE (RI2, 0)-(RI2, RADMAX1)
LINE (MR2, 0)-(MR2, 2 * R2MAX), 2
SS = (S2 - S1) * .1
FOR I = 1 TO 9: SS1 = S1 + I * SS
LINE (R1, SS1)-(R1 + .2, SS1): NEXT I
FOR I = 1 TO R2: LINE (I, S1)-(I, S1 + .1): NEXT I
A$ = " R   R2   r2R2"
LOCATE 2, 4: PRINT "PLOTS OF R,R2 AND r2R2 FOR 2s ORBITAL"
LOCATE 3, 2: PRINT USING "#.#"; S2 / 2
LOCATE 4, 2: PRINT USING "#.#"; S2 / 2
LOCATE 4, 15: PRINT " r2R2MAX"; : PRINT USING " #.##"; RI1; RI2
LOCATE 5, 2: PRINT USING "#.#"; S2
LOCATE 5, 15: PRINT " R2MAX"; : PRINT USING "    #.##"; MR2
LOCATE 6, 16: PRINT "RAV"; : PRINT USING "      #.##"; AVR / AVD
```

```
FOR I = 1 TO 13: B$ = MID$(A$, I, 1)
LOCATE 6 + I, 3: PRINT B$: NEXT I
LOCATE 10, 27: PRINT "r2R2": LOCATE 16, 25
PRINT "R2": LOCATE 20, 25: PRINT "R"
LOCATE 20, 1: PRINT USING "##.#"; S1 / 2
LOCATE 21, 1: PRINT USING "##.#"; S1 / 2
LOCATE 22, 1: PRINT USING "##.#"; S1
LOCATE 23, 3: PRINT "   0              r/a0---->               ";
PRINT USING "##"; R2
END
```

Output

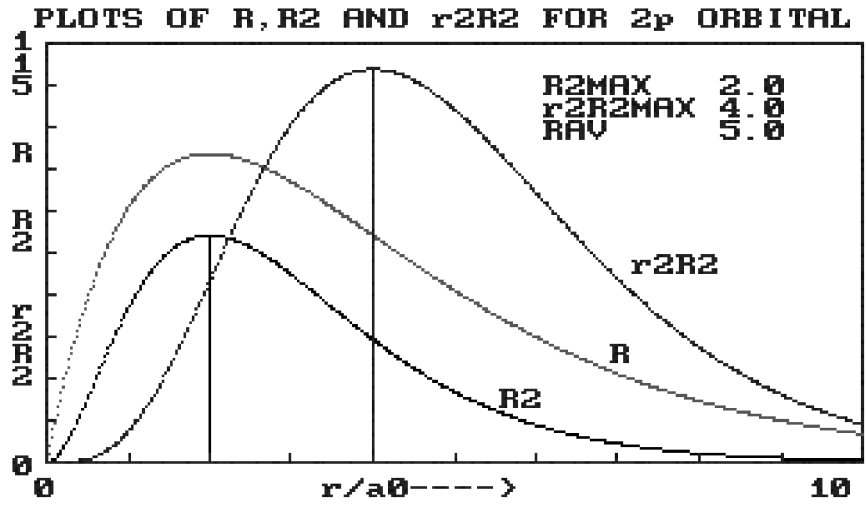


Program for $R_{2,1}$

```
REM PROGRAM PLOT2P;PLOTS OF R,R2,r2R2 FOR 2p ORBITAL
R1 = 0: R2 = 10: S1 = 0: S2 = 5: RADMAX = 0: RI = 0: R2MAX = 0
CLS : SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(315, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0)
AVR = 0: AVD = 0
FOR R = 0 TO R2 + 6 STEP .02
SAI = R * EXP(-R / 2): SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
IF R2MAX > SAI2 THEN 10
R2MAX = SAI2: RMAX = R
10 IF RADMAX > RAD THEN 40
RADMAX = RAD: RI1 = R
40 PSET (R, 5 * SAI), 1: PSET (R, 5 * SAI2), 2: PSET (R, RAD), 3
```

```
'FOR I = 1 TO 100 STEP .1: NEXT I
AVR = AVR + R * RAD: AVD = AVD + RAD
NEXT R
LINE (RMAX, 0)-(RMAX, R2MAX * 5), 2: LINE (RI1, 0)-(RI1, RADMAX)
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9: LINE (I * RS, S1)-(I * RS, S1 + .1)
LINE (R1, S1 + I * SS)-(R1 + .1, S1 + I * SS)
NEXT I: A$ = " R   R2   r2R2"
LOCATE 2, 3: PRINT "PLOTS OF R,R2 AND r2R2 FOR 2p ORBITAL"
LOCATE 3, 1: PRINT USING "##"; S2 / 5
LOCATE 4, 1: PRINT USING "##"; S2 / 5
LOCATE 5, 1: PRINT USING "##"; S2
LOCATE 5, 26: PRINT "R2MAX"; : PRINT USING "    #.#"; RMAX
LOCATE 6, 26: PRINT "r2R2MAX"; : PRINT USING " #.#"; RI1
LOCATE 7, 26: PRINT "RAV"; : PRINT USING "      #.#"; AVR / AVD
FOR I = 1 TO 13: B$ = MID$(A$, I, 1)
LOCATE 6 + I, 2: PRINT B$: NEXT I
LOCATE 22, 1: PRINT USING "##"; S1
LOCATE 23, 3: PRINT "0               r/a0---->                ";
PRINT USING "##"; R2
LOCATE 13, 30: PRINT "r2R2": LOCATE 17, 29
PRINT "R": LOCATE 19, 24: PRINT "R2"
END
```

Output

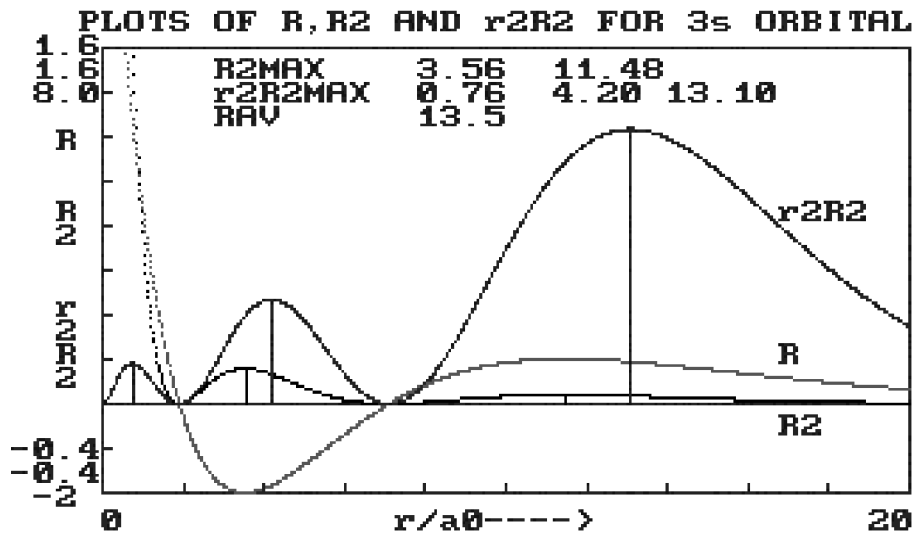


Program for $R_{3,0}$

```
REM PROG45D
REM PROGRAM PLOT3S;PLOTS OF R,R2,r2R2 FOR 3s ORBITAL
CLS : DIM r2R2MX(3), RMAX(3), R2MAX(2), RMX(2)
R1 = 0: R2 = 20: S1 = -2: S2 = 8: RI = 0: R2MX = 1000: r2R2MAX = 0
SCREEN 1: COLOR 15, 0: VIEW (32, 20)-(318, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0)
KK = 1: J = 0: K = 1: JJ = 0
AVR = 0: AVD = 0
FOR R = 0 TO R2 + 9 STEP .02
SAI = (3 - 2 * R + 2 * R ^ 2 / 9) * EXP(-R / 3): SAI2 = SAI ^ 2
RAD = R ^ 2 * SAI2
ON KK GOTO 3, 4
3 IF R2MX > SAI2 THEN R2MX = SAI2: GOTO 9
R2MX = SAI2: KK = 2: GOTO 9
4 IF R2MX < SAI2 THEN R2MX = SAI2: GOTO 9
JJ = JJ + 1: R2MAX(JJ) = SAI2: RMX(JJ) = R: KK = 1
9 ON K GOTO 10, 15
10 IF r2R2MAX > RAD THEN 20
r2R2MAX = RAD: GOTO 25
15 IF r2R2MAX > RAD THEN 25
r2R2MAX = RAD: K = 1
GOTO 10
20 J = J + 1: r2R2MX(J) = r2R2MAX: RMAX(J) = R: K = 2
25 PSET (R, 5 * SAI), 1: PSET (R, 5 * SAI2), 2: PSET (R, RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .1: NEXT I
NEXT R
FOR I = 1 TO J: LINE (RMAX(I), 0)-(RMAX(I), r2R2MX(I)): NEXT I
FOR I = 1 TO JJ: LINE (RMX(I), 0)-(RMX(I), R2MAX(I) * 5), 2:
NEXT I
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9
LINE (I * RS, S1)-(I * RS, S1 + .2)
LINE (R1, S1 + I * SS)-(R1 + .2, S1 + I * SS)
NEXT I
A$ = "R   R2   r2R2"
LOCATE 2, 4: PRINT "PLOTS OF R,R2 AND r2R2 FOR 3s ORBITAL"
LOCATE 3, 2: PRINT USING "#.#"; S2 / 5
```

```
LOCATE 4, 2: PRINT USING "#.#"; S2 / 5
LOCATE 4, 10: PRINT "R2MAX ";
FOR I = 1 TO JJ: PRINT USING "   ##.##"; RMX(I); : NEXT I
LOCATE 5, 2: PRINT USING "#.#"; S2
LOCATE 5, 10: PRINT "r2R2MAX";
FOR I = 1 TO J: PRINT USING " ##.##"; RMAX(I); : NEXT I
LOCATE 6, 10: PRINT "RAV"; : PRINT USING "       ##.#"; AVR / AVD
FOR I = 1 TO 11: B$ = MID$(A$, I, 1)
LOCATE 6 + I, 3: PRINT B$: NEXT I
LOCATE 10, 35: PRINT "r2R2": LOCATE 16, 35
PRINT "R": LOCATE 19, 35: PRINT "R2"
LOCATE 20, 1: PRINT USING "##.#"; S1 / 5
LOCATE 21, 1: PRINT USING "##.#"; S1 / 5
LOCATE 22, 2: PRINT USING "##"; S1
LOCATE 23, 3: PRINT "   0              r/a0---->              ";
PRINT USING "##"; R2
END
```

Output



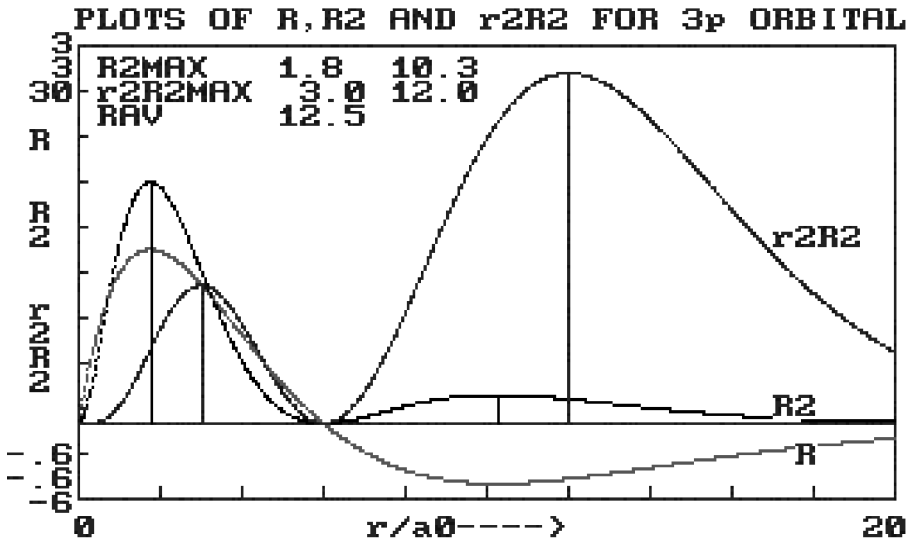
Program for $R_{3,1}$

```
REM PROGRAM PLOT3P;PLOTS OF R,R2,r2R2 FOR 3p ORBITAL
CLS : DIM R2MAX(5), RMX(5), r2R2MAX(5), R1MAX(5)
R1 = 0: R2 = 20: S1 = -6: S2 = 30: SF = 10: RF = 10: RI = 0
SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(315, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
```

```
LINE (R1, 0)-(R2, 0)
R2MX = 0: r2R2MX = 0: J = 0: K = 1: J1 = 0: K1 = 1
AVR = 0: AVD = 0
FOR R = 0 TO R2 + 8 STEP .02
SAI = (2 * R - R ^ 2 / 3) * EXP(-R / 3)
SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
ON K1 GOTO 5, 6
5 IF R2MX > SAI2 THEN 8
R2MX = SAI2: GOTO 9
6 IF R2MX < SAI2 THEN 7
R2MX = SAI2: GOTO 5
7 R2MX = SAI2: K1 = 1: GOTO 9
8 J1 = J1 + 1
R2MAX(J1) = R2MX: RMX(J1) = R: K1 = 2
9 ON K GOTO 10, 15
10 IF r2R2MX > RAD THEN 20
r2R2MX = RAD: GOTO 25
15 IF r2R2MX > RAD THEN 25
r2R2MX = RAD: K = 1
GOTO 10
20 J = J + 1
r2R2MAX(J) = r2R2MX: R1MX(J) = R: K = 2
25 PSET (R, RF * SAI), 1: PSET (R, RF * SAI2), 2
PSET (R, RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .1: NEXT I
NEXT R
LINE (RI1, 0)-(RI1, r2R2MX)
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9
LINE (I * RS, S1)-(I * RS, S1 + 1)
LINE (R1, S1 + I * SS)-(R1 + .2, S1 + I * SS)
NEXT I
A$ = "R   R2   r2R2"
LOCATE 2, 4: PRINT "PLOTS OF R,R2 AND r2R2 FOR 3p ORBITAL"
LOCATE 3, 2: PRINT USING "##"; S2 / RF
LOCATE 4, 2: PRINT USING "##"; S2 / RF
LOCATE 5, 2: PRINT USING "##"; S2
LOCATE 4, 5: PRINT "R2MAX";
FOR I = 1 TO J: PRINT USING "   ##.#"; RMX(I);
```

```
LINE (RMX(I), 0)-(RMX(I), R2MAX(I) * RF), 2: NEXT I
LOCATE 5, 5: PRINT "r2R2MAX";
FOR I = 1 TO J: PRINT USING " ##.#"; R1MX(I);
LINE (R1MX(I), 0)-(R1MX(I), r2R2MAX(I)): NEXT I
LOCATE 6, 5: PRINT "RAV"; : PRINT USING "      ##.#"; AVR / AVD
FOR I = 1 TO 11: B$ = MID$(A$, I, 1)
LOCATE 6 + I, 2: PRINT B$: NEXT I
LOCATE 11, 35: PRINT "r2R2": LOCATE 18, 35
PRINT "R2": LOCATE 20, 36: PRINT "R"
LOCATE 20, 1: PRINT USING "#.#"; S1 / RF
LOCATE 21, 1: PRINT USING "#.#"; S1 / RF
LOCATE 22, 2: PRINT USING "##"; S1
LOCATE 23, 4: PRINT "0              r/a0---->                ";
PRINT USING "##"; R2
END
```

Output

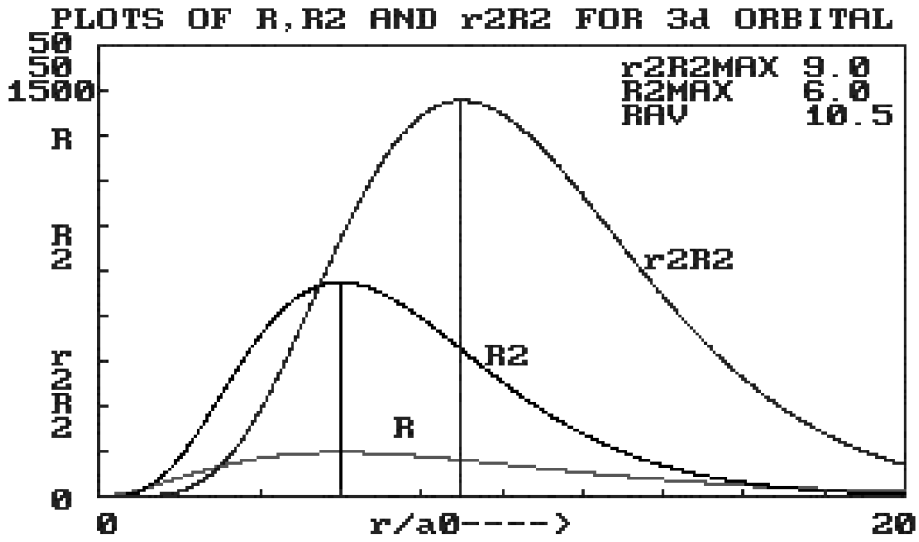


Program for $R_{3,2}$

```
REM PROG45F
REM PROGRAM PLOT3D;PLOTS OF R,R2,r2R2 FOR 3d ORBITAL
CLS : R1 = 0: R2 = 20: S1 = 0: S2 = 1500: RF = 30
RADMAX = 0: R2MAX = 0: RI = 0
SCREEN 1: COLOR 15, 0: VIEW (32, 20)-(315, 170)
```

```
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0): AVR = 0: AVD = 0
FOR R = 0 TO R2 + 5 STEP .02
SAI = R ^ 2 * EXP(-R / 3): SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
IF RADMAX > RAD THEN 40
RADMAX = RAD: MRAD = R
IF R2MAX > SAI2 THEN 40
R2MAX = SAI2: MR2 = R
40 PSET (R, RF * SAI), 1: PSET (R, RF * SAI2), 2: PSET (R, RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .1: NEXT I
NEXT R
LINE (MRAD, 0)-(MRAD, RADMAX)
LINE (MR2, 0)-(MR2, RF * R2MAX), 2
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9: LINE (I * RS, S1)-(I * RS, S1 + 20)
LINE (R1, S1 + I * SS)-(R1 + .2, S1 + I * SS)
NEXT I
A$ = "R    R2    r2R2"
LOCATE 2, 3: PRINT "PLOTS OF R,R2 AND r2R2 FOR 3d ORBITAL"
LOCATE 3, 2: PRINT USING "##"; S2 / RF
LOCATE 4, 2: PRINT USING "##"; S2 / RF
LOCATE 5, 1: PRINT USING "####"; S2
LOCATE 4, 28: PRINT "r2R2MAX"; : PRINT USING " #.#"; MRAD
LOCATE 5, 28: PRINT "R2MAX"; : PRINT USING "    #.#"; MR2
LOCATE 6, 28: PRINT "RAV"; : PRINT USING "      ##.#"; AVR / AVD
FOR I = 1 TO 13: B$ = MID$(A$, I, 1)
LOCATE 6 + I, 3: PRINT B$: NEXT I
LOCATE 12, 29: PRINT "r2R2": LOCATE 16, 22
PRINT "R2": LOCATE 19, 18: PRINT "R"
LOCATE 22, 2: PRINT USING "##"; S1
LOCATE 23, 3: PRINT "   0             r/a0---->                ";
PRINT USING "##"; R2
END
```

Output

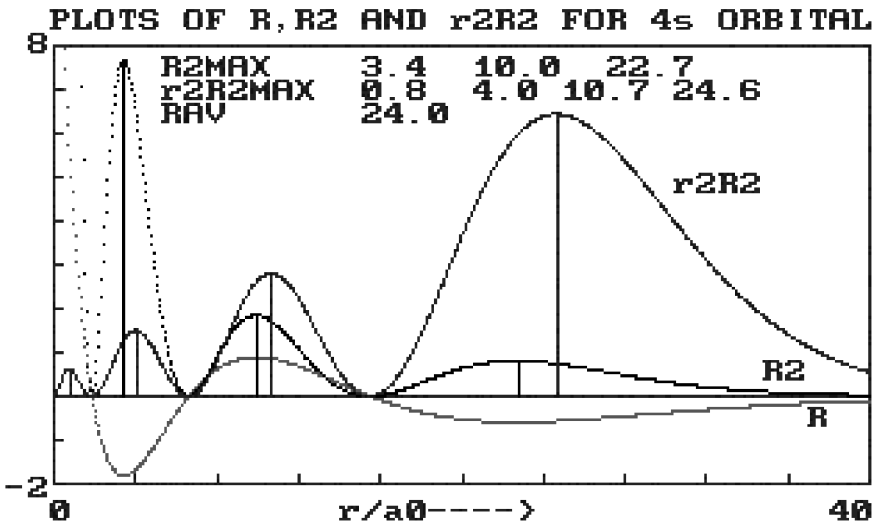


Program for $R_{4,0}$

```
REM PLOTS OF R,R2,r2R2 FOR 4s ORBITAL
CLS : DIM r2R2MX(4), RMAX(4), R2MAX(3), RMX(3)
R1 = 0: R2 = 40: S1 = -2: S2 = 8: R2MX = 1000: r2R2MAX = 0
SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(318, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0)
KK = 1: J = 0: K = 1: JJ = 0: AVR = 0: AVD = 0
FOR R = 0 TO R2 + 13 STEP .05
SAI = (192 - 144 * R + 24 * R ^ 2 - R ^ 3)
SAI = SAI * (1 / 768) * EXP(-R / 4)
SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
ON KK GOTO 3, 4, 5
3 IF R2MX > SAI2 THEN R2MX = SAI2: GOTO 9
R2MX = SAI2: KK = 2: GOTO 9
4 IF R2MX > SAI2 THEN R2MX = SAI2: GOTO 9
R2MX = SAI2: KK = 3: GOTO 9
5 IF R2MX < SAI2 THEN R2MX = SAI2: GOTO 9
JJ = JJ + 1: R2MAX(JJ) = SAI2: RMX(JJ) = R: KK = 1
9 ON K GOTO 10, 15
10 IF r2R2MAX > RAD THEN 20
r2R2MAX = RAD: GOTO 25
15 IF r2R2MAX > RAD THEN 25
r2R2MAX = RAD: K = 1
GOTO 10
20 J = J + 1: r2R2MX(J) = r2R2MAX: RMAX(J) = R: K = 2
```

```
25 PSET (R, 55 * SAI), 1: PSET (R, 7000 * SAI2), 2
PSET (R, 100 * RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .5: NEXT I
NEXT R
FOR I = 1 TO J: LINE (RMAX(I), 0)-(RMAX(I), 100 * r2R2MX(I)):
NEXT I
FOR I = 1 TO JJ: LINE (RMX(I), 0)-(RMX(I), 7000 * R2MAX(I)), 2:
NEXT I
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9
LINE (I * RS, S1)-(I * RS, S1 + .2)
LINE (R1, S1 + I * SS)-(R1 + .4, S1 + I * SS)
NEXT I
LOCATE 2, 4: PRINT "PLOTS OF R,R2 AND r2R2 FOR 4s ORBITAL"
LOCATE 4, 9: PRINT "R2MAX ";
FOR I = 1 TO JJ: PRINT USING "   ##.#"; RMX(I); : NEXT I
LOCATE 3, 3: PRINT USING "#"; S2
LOCATE 5, 9: PRINT "r2R2MAX";
FOR I = 1 TO J: PRINT USING " ##.#"; RMAX(I); : NEXT I
LOCATE 6, 9: PRINT "RAV"; : PRINT USING "       ##.#"; AVR / AVD
LOCATE 9, 32: PRINT "r2R2"
LOCATE 17, 36: PRINT "R2": LOCATE 19, 38: PRINT "R"
LOCATE 22, 2: PRINT USING "##"; S1
LOCATE 23, 3: PRINT " 0               r/a0---->                ";
PRINT USING "##"; R2
END
```

Output

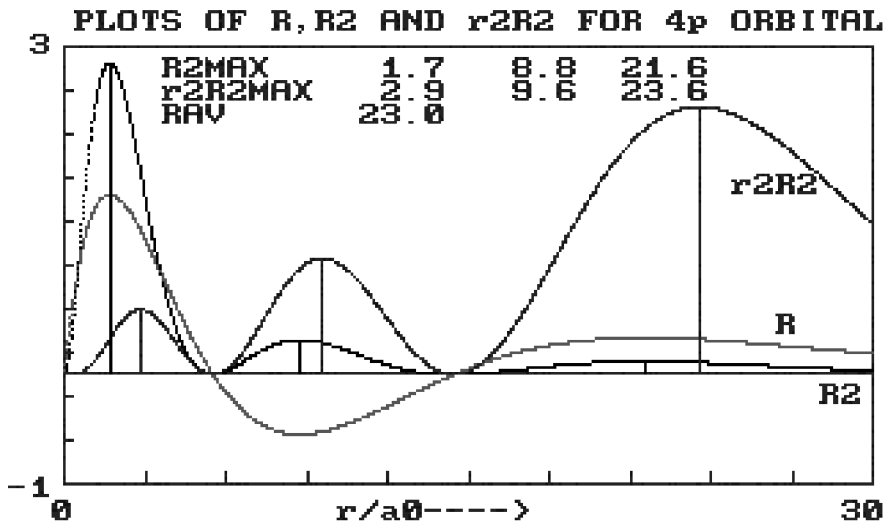


Program for $R_{4,1}$

```
REM PLOTS OF R,R2,r2R2 FOR 4p ORBITAL
CLS : DIM R2MAX(5), RMX(5), r2R2MAX(5), R1MAX(5)
R1 = 0: R2 = 30: S1 = -1: S2 = 3: SF = 10
SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(315, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0)
R2MX = 0: r2R2MX = 0: J = 0: K = 1: J1 = 0: K1 = 1
AVR = 0: AVD = 0
FOR R = 0 TO R2 + 17 STEP .02
SAI = 3 / (768 * SQR(15)) * (80 - 20 * R + R ^ 2) * R * EXP(-R
/ 4)
SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
ON K1 GOTO 5, 6
5 IF R2MX > SAI2 THEN 8
R2MX = SAI2: GOTO 9
6 IF R2MX < SAI2 THEN 7
R2MX = SAI2: GOTO 5
7 R2MX = SAI2: K1 = 1: GOTO 9
8 J1 = J1 + 1
R2MAX(J1) = R2MX: RMX(J1) = R: K1 = 2
9 ON K GOTO 10, 15
10 IF r2R2MX > RAD THEN 20
r2R2MX = RAD: GOTO 25
15 IF r2R2MX > RAD THEN 25
r2R2MX = RAD: K = 1: GOTO 10
20 J = J + 1
r2R2MAX(J) = r2R2MX: R1MX(J) = R: K = 2
25 PSET (R, 30 * SAI), 1: PSET (R, 950 * SAI2), 2
PSET (R, 38 * RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .5: NEXT I
NEXT R
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9
LINE (I * RS, S1)-(I * RS, S1 + .1)
LINE (R1, S1 + I * SS)-(R1 + .3, S1 + I * SS)
NEXT I
LOCATE 2, 4: PRINT "PLOTS OF R,R2 AND r2R2 FOR 4p ORBITAL"
LOCATE 3, 2: PRINT USING "#"; S2
```

```
LOCATE 4, 8: PRINT "R2MAX   ";
FOR I = 1 TO J: PRINT USING "   ##.#"; RMX(I);
LINE (RMX(I), 0)-(RMX(I), 950 * R2MAX(I)), 2: NEXT I
LOCATE 5, 8: PRINT "r2R2MAX";
FOR I = 1 TO J: PRINT USING "   ##.#"; R1MX(I);
LINE (R1MX(I), 0)-(R1MX(I), 38 * r2R2MAX(I)): NEXT I
LOCATE 6, 8: PRINT "RAV "; : PRINT USING "      ##.#"; AVR / AVD
LOCATE 9, 34: PRINT "r2R2": LOCATE 15, 36
PRINT "R": LOCATE 18, 38: PRINT "R2"
LOCATE 22, 1: PRINT USING "##"; S1
LOCATE 23, 3: PRINT "0              r/a0---->                  ";
PRINT USING "##"; R2
END
```

Output



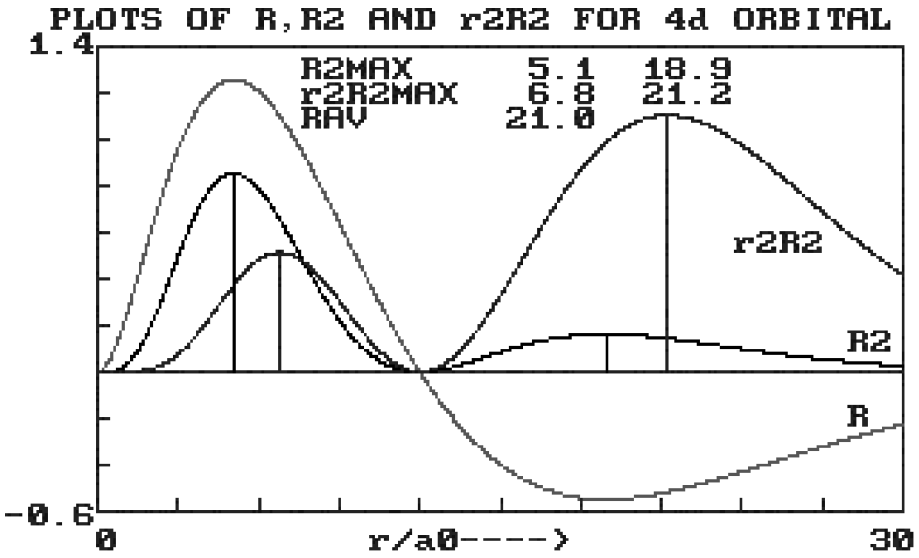
Program for $R_{4,2}$

```
REM PLOTS OF R,R2,r2R2 FOR 4d ORBITAL
DIM RMX(5), R2MAX(5), RDMX(5), RDMAX(5)
CLS : R1 = 0: R2 = 30: S1 = -.6: S2 = 1.4
RADMAX = 0: R2MX = 0: K1 = 1: K2 = 1
SCREEN 1: COLOR 15, 0: VIEW (32, 20)-(315, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0): AVR = 0: AVD = 0
```

```
FOR R = 0 TO R2 + 15 STEP .02
SAI = 1 / (768 * SQR(5)) * (12 - R) * R ^ 2 * EXP(-R / 4)
SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
ON K1 GOTO 5, 6
5 IF R2MX > SAI2 THEN 8
R2MX = SAI2: GOTO 9
6 IF R2MX < SAI2 THEN 7
R2MX = SAI2: GOTO 5
7 R2MX = SAI2: K1 = 1: GOTO 9
8 J1 = J1 + 1
R2MAX(J1) = R2MX: RMX(J1) = R: K1 = 2
9 ON K2 GOTO 15, 16
15 IF RADMAX > RAD THEN 18
RADMAX = RAD: GOTO 19
16 IF RADMAX < RAD THEN 17
RADMAX = RAD: GOTO 15
17 RADMAX = RAD: K2 = 1: GOTO 19
18 J2 = J2 + 1
RDMAX(J2) = RADMAX: RDMX(J2) = R: K2 = 2
19 PSET (R, 43 * SAI), 1
PSET (R, 1000 * SAI2), 2: PSET (R, 17 * RAD), 3
AVR = AVR + R * RAD: AVD = AVD + RAD
'FOR I = 1 TO 100 STEP .5: NEXT I
NEXT R
FOR I = 1 TO J1
LINE (RMX(I), 0)-(RMX(I), 1000 * R2MAX(I)), 2
NEXT I
FOR I = 1 TO J2
LINE (RDMX(I), 0)-(RDMX(I), 17 * RDMAX(I))
NEXT I
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9: LINE (I * RS, S1)-(I * RS, S1 + SS / 4)
LINE (R1, S1 + I * SS)-(R1 + RS / 8, S1 + I * SS)
NEXT I
LOCATE 2, 3: PRINT "PLOTS OF R,R2 AND r2R2 FOR 4d ORBITAL"
LOCATE 3, 1: PRINT USING "##.#"; S2
LOCATE 4, 14: PRINT "R2MAX   ";
FOR I = 1 TO J1: PRINT USING "   ##.#"; RMX(I); : NEXT I
LOCATE 5, 14: PRINT "r2R2MAX";
FOR I = 1 TO J2: PRINT USING "   ##.#"; RDMX(I); : NEXT I
```

```
LOCATE 6, 14: PRINT "RAV"; : PRINT USING "       ##.#"; AVR / AVD
LOCATE 11, 33: PRINT "r2R2": LOCATE 15, 38: PRINT "R2"
LOCATE 18, 38: PRINT "R"
LOCATE 22, 1: PRINT USING "##.#"; S1
LOCATE 23, 3: PRINT "   0              r/a0---->                ";
PRINT USING "##"; R2
END
```

Output

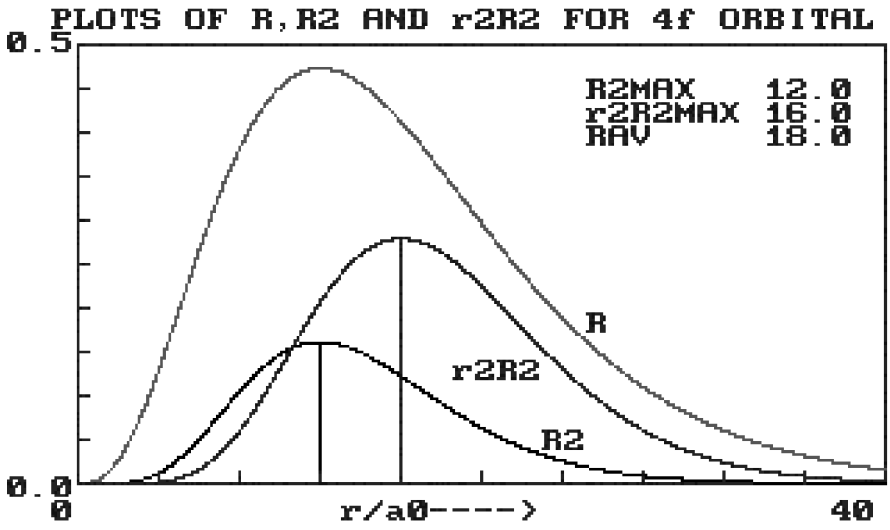


Program for $R_{4,3}$

```
REM PLOTS OF R,R2,r2R2 FOR 4f ORBITAL
R1 = 0: R2 = 40: S1 = 0: S2 = .5: RADMAX = 0: R2MAX = 0
CLS : SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(315, 170)
WINDOW (R1, S1)-(R2, S2): LINE (R1, S1)-(R2, S2), , B
LINE (R1, 0)-(R2, 0): AVR = 0: AVD = 0
FOR R = 0 TO R2 + 1 STEP .02
SAI = 1 / (768 * SQR(35)) * R ^ 3 * EXP(-R / 4)
SAI2 = SAI ^ 2: RAD = R ^ 2 * SAI2
IF R2MAX > SAI2 THEN 10
R2MAX = SAI2: RMAX = R
10 IF RADMAX > RAD THEN 40
RADMAX = RAD: RI1 = R
40 PSET (R, 25 * SAI), 1: PSET (R, 450 * SAI2), 2
PSET (R, 4 * RAD), 3
```

```
'FOR I = 1 TO 100 STEP .5: NEXT I
AVR = AVR + R * RAD: AVD = AVD + RAD
NEXT R
LINE (RMAX, 0)-(RMAX, R2MAX * 450), 2
LINE (RI1, 0)-(RI1, 4 * RADMAX)
RS = (R2 - R1) * .1: SS = (S2 - S1) * .1
FOR I = 1 TO 9: LINE (I * RS, S1)-(I * RS, S1 + SS / 4)
LINE (R1, S1 + I * SS)-(R1 + RS / 8, S1 + I * SS)
NEXT I
LOCATE 2, 3: PRINT "PLOTS OF R,R2 AND r2R2 FOR 4f ORBITAL"
LOCATE 3, 1: PRINT USING "#.#"; S2
LOCATE 5, 27: PRINT "R2MAX"; : PRINT USING "    ##.#"; RMAX
LOCATE 6, 27: PRINT "r2R2MAX"; : PRINT USING " ##.#"; RI1
LOCATE 7, 27: PRINT "RAV"; : PRINT USING "      ##.#"; AVR / AVD
LOCATE 22, 1: PRINT USING "#.#"; S1
LOCATE 23, 3: PRINT "0              r/a0---->               ";
PRINT USING "##"; R2
LOCATE 15, 27: PRINT "R"
LOCATE 17, 21: PRINT "r2R2": LOCATE 20, 25: PRINT "R2"
END
```

Output



2.47 DISPLAY OF ANGULAR DEPENDENCE OF p AND d ORBITALS

To display two dimensional angular dependence of p and d orbitals.

The angular dependence of p and d orbitals are as follows.

p_x $\sin\theta\cos\phi$; keep ϕ constant at 0 and π, vary θ from 0 to π

p_y $\sin\theta\sin\phi$; keep ϕ constant at $\pi/2$ and $3\pi/2$ vary θ from 0 to π

Note Both p_x and p_y plottings involve sin θ versus θ. Only orientation will be different. p_x will be along x-axis ($\phi = 0$ and π) and p_y will be along y-axis ($\phi = \pi/2$) and $3\pi/2$)

p_z $\cos\theta$; vary θ from 0 to π ; maximum along z-axis.

d_{xy} $(\sin\theta\cos\phi)(\sin\theta\sin\phi) \equiv \sin^2\theta\sin 2\phi$; vary ϕ from 0 to π and keep θ constant. Maximum along the line with $\theta = 90°$ and $\phi = 45°$ and thus lie in between x and y axis.

d_{xz} $(\sin\theta\cos\phi)(\cos\theta) \equiv \sin 2\theta\cos\phi$; vary θ from 0 to π and keep ϕ constant. Maximum along $\phi = 0$ and $\theta = \pi/4$ and thus lie in between x and z axis.

d_{yz} $(\sin\theta\sin\theta)(\cos\theta) \equiv \sin 2\theta\sin\phi$; vary θ from 0 to π and keep ϕ constant at $\pi/2$. Maximum along $\phi = \pi/2$ and $\theta = \pi/4$ and thus lie in between y and z axis.

Note d_{xy}, d_{xz} and d_{yz} involve the plotting of sin 2θ versus θ. Only orientation will be different.

$d_{x^2-y^2}$ $\sin^2\theta\cos^2\phi - \sin^2\theta\sin^2\phi \equiv \sin^2\theta\cos 2\phi$ vary ϕ from 0 to π and keep θ constant at $\pi/2$.

Maximum lies along $\theta = \pi/2$ and $\phi = 0$ and $\pi/2$ and thus lie along x and y axis.

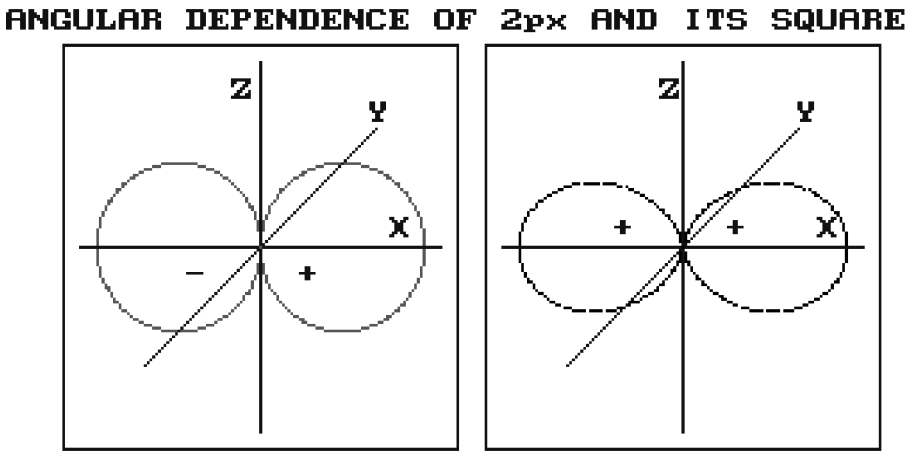
d_{z2} $3\cos^2\theta - 1$; vary θ from 0 to π. Maximum lies along z-axis where $\theta = 0$.

Program for $2p_x$

```
REM PROGRAM ANG2PX
REM ANGLE THETA IS TAKEN TO BE 0 and 180 degrees
CLS : SCREEN 1: COLOR 15, 0
SC = 1.2: SX = SC - .1: SY = SC - .5
FOR AN = 0 TO 360 STEP 1
TH = (22 / 7) * AN / 180
XX = SIN(TH): X = ABS(XX) * COS(TH): Y = ABS(XX) * SIN(TH)
VIEW (20, 20)-(160, 160)
WINDOW (-SC, -SC)-(SC, SC): LINE (-SC, -SC)-(SC, SC), , B
LINE (-SX, 0)-(SX, 0): LINE (0, -SX)-(0, SX)
LINE (-SY, -SY)-(SY, SY)
PSET (Y, X), 1
X1 = SIN(TH) ^ 2 * COS(TH): Y1 = SIN(TH) ^ 3
VIEW (170, 20)-(310, 160)
LINE (-SC, -SC)-(SC, SC), , B: LINE (-SX, 0)-(SX, 0)
LINE (0, -SX)-(0, SX): LINE (-SY, -SY)-(SY, SY)
```

```
PSET (Y1, X1), 2
'FOR I = 1 TO 1000 STEP .1: NEXT I
NEXT AN
LOCATE 1, 4: PRINT "ANGULAR DEPENDENCE OF 2px AND ITS SQUARE"
LOCATE 5, 11: PRINT "Z": LOCATE 5, 30: PRINT "Z"
LOCATE 6, 17: PRINT "Y": LOCATE 6, 36: PRINT "Y"
LOCATE 11, 18: PRINT "X": LOCATE 11, 37: PRINT "X"
LOCATE 13, 9: PRINT "-": LOCATE 13, 14: PRINT "+"
LOCATE 11, 28: PRINT "+": LOCATE 11, 33: PRINT "+"
END
```

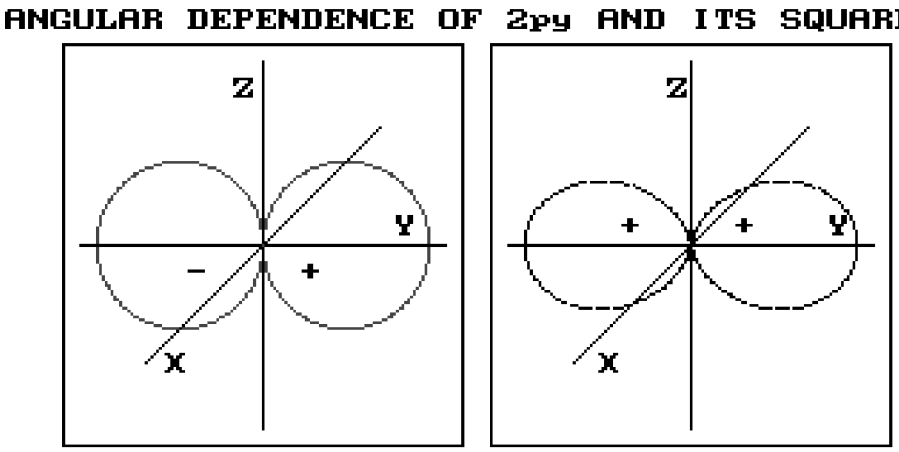
Output



Program for $2p_y$

```
REM PROGRAM ANG2PY
REM ANGLE PHI IS TAKEN TO BE 90 and 270 degrees
CLS : SCREEN 1: COLOR 15, 0
SC = 1.2: SX = SC - .1: SY = SC - .5
FOR AN = 0 TO 360 STEP 1
TH = (22 / 7) * AN / 180
XX = SIN(TH): X = ABS(XX) * COS(TH): Y = ABS(XX) * SIN(TH)
VIEW (20, 20)-(160, 160)
WINDOW (-SC, -SC)-(SC, SC): LINE (-SC, -SC)-(SC, SC), , B
LINE (-SX, 0)-(SX, 0): LINE (0, -SX)-(0, SX)
```

```
LINE (-SY, -SY)-(SY, SY)
PSET (Y, X), 1': PSET (-Y, -X), 1
X1 = SIN(TH) ^ 2 * COS(TH): Y1 = SIN(TH) ^ 3
VIEW (170, 20)-(310, 160): LINE (-SC, -SC)-(SC, SC), , B
LINE (-SX, 0)-(SX, 0): LINE (0, -SX)-(0, SX)
LINE (-SY, -SY)-(SY, SY)
PSET (Y1, X1), 2': PSET (-Y1, -X1), 2
FOR I = 1 TO 1000 STEP .001: NEXT I
NEXT AN
LOCATE 1, 4: PRINT "ANGULAR DEPENDENCE OF 2py AND ITS SQUARE"
LOCATE 5, 11: PRINT "Z": LOCATE 5, 30: PRINT "Z"
LOCATE 17, 8: PRINT "X": LOCATE 17, 27: PRINT "X"
LOCATE 11, 18: PRINT "Y": LOCATE 11, 37: PRINT "Y"
LOCATE 13, 9: PRINT "-": LOCATE 13, 14: PRINT "+"
LOCATE 11, 28: PRINT "+": LOCATE 11, 33: PRINT "+"
END
```

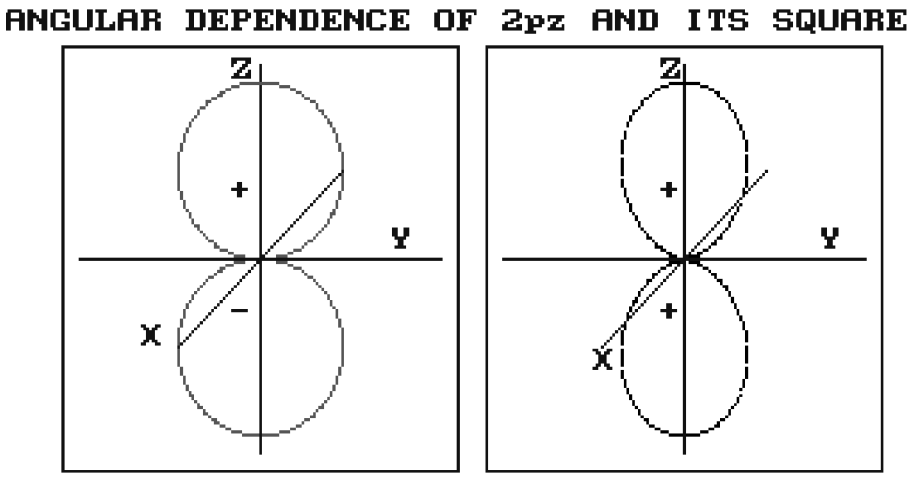


Program for $2p_z$

```
REM PROGRAM ANG2PZ
CLS : SCREEN 1: COLOR 15, 0
SC = 1.2: SX = SC - .1: SY = -.5
FOR AN = 0 TO 360 STEP 1
TH = (22 / 7) * AN / 180
Z = ABS(COS(TH)) * COS(TH): Y = ABS(COS(TH)) * SIN(TH)
VIEW (20, 20)-(160, 160): WINDOW (-SC, -SC)-(SC, SC)
LINE (-SC, -SC)-(SC, SC), , B: LINE (-SX, 0)-(SX, 0)
LINE (0, -SX)-(0, SX): LINE (-SY, -SY)-(SY, SY)
```

```
PSET (Y, Z), 1
X1 = COS(TH) ^ 3: Y1 = COS(TH) ^ 2 * SIN(TH)
VIEW (170, 20)-(310, 160)
LINE (-SC, -SC)-(SC, SC), , B: LINE (-SX, 0)-(SX, 0)
LINE (0, -SX)-(0, SX): LINE (-SY, -SY)-(SY, SY)
PSET (Y1, X1), 2
FOR I = 1 TO 1000 STEP .01: NEXT I
NEXT AN
LOCATE 1, 4: PRINT "ANGULAR DEPENDENCE OF 2pz AND ITS SQUARE"
LOCATE 4, 11: PRINT "Z": LOCATE 4, 30: PRINT "Z"
LOCATE 9, 11: PRINT "+": LOCATE 9, 30: PRINT "+"
LOCATE 14, 11: PRINT "-": LOCATE 14, 30: PRINT "+"
LOCATE 15, 7: PRINT "X": LOCATE 16, 27: PRINT "X"
LOCATE 11, 18: PRINT "Y": LOCATE 11, 37: PRINT "Y"
END
```

Output

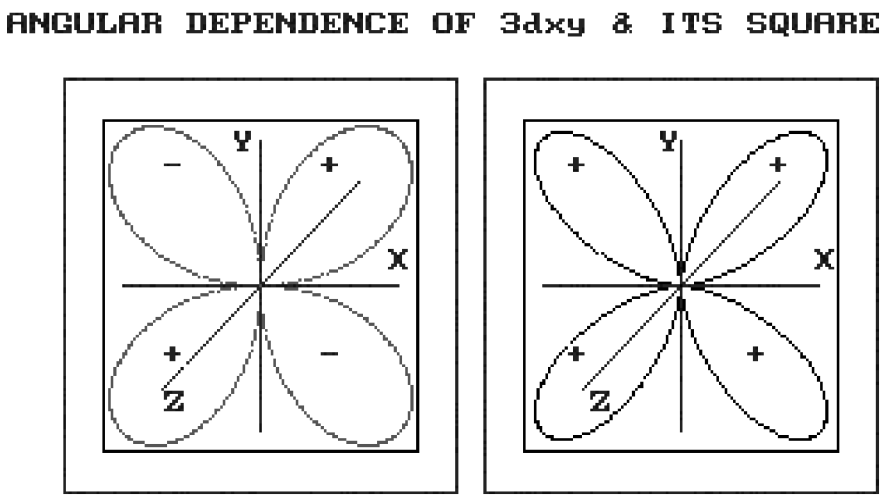


Program for $3d_{xy}$

```
REM PROGRAM ANG3DXY
Z = 1: Z1 = Z - .2: Z2 = Z - .3
CLS : SCREEN 1: COLOR 15, 0
FOR AN = 0 TO 360 STEP .5
VIEW (20, 30)-(160, 170)
WINDOW (-Z, -Z)-(Z, Z): LINE (-Z, -Z)-(Z, Z), , B
LINE (-Z2, 0)-(Z2, 0): LINE (0, -Z2)-(0, Z2)
LINE (-.5, -.5)-(.5, .5): LINE (-Z1, -Z1)-(Z1, Z1), 2, B
```

```
TH = (22 / 7) * AN / 180: XX = SIN(2 * TH)
Y = ABS(XX) * COS(TH): X = ABS(XX) * SIN(TH)
PSET (X, Y), 1
VIEW (170, 30)-(310, 170)
LINE (-Z, -Z)-(Z, Z), , B
LINE (-Z2, 0)-(Z2, 0): LINE (0, -Z2)-(0, Z2)
LINE (-.5, -.5)-(.5, .5): LINE (-Z1, -Z1)-(Z1, Z1), 2, B
Y1 = XX ^ 2 * COS(TH): X1 = XX ^ 2 * SIN(TH)
PSET (X1, Y1), 2
'FOR I = 1 TO 100 STEP .001: NEXT I
NEXT AN
LOCATE 1, 4: PRINT "ANGULAR DEPENDENCE OF 3dxy & ITS SQUARE"
LOCATE 7, 11: PRINT "Y": LOCATE 7, 30: PRINT "Y"
LOCATE 8, 8: PRINT "-": LOCATE 8, 15: PRINT "+"
LOCATE 8, 26: PRINT "+":  LOCATE 8, 35: PRINT "+"
LOCATE 18, 8: PRINT "Z": LOCATE 18, 27: PRINT "Z"
LOCATE 12, 18: PRINT "X": LOCATE 12, 37: PRINT "X"
LOCATE 16, 8: PRINT "+": LOCATE 16, 15: PRINT "-"
LOCATE 16, 26: PRINT "+": LOCATE 16, 34: PRINT "+"
END
```

Output

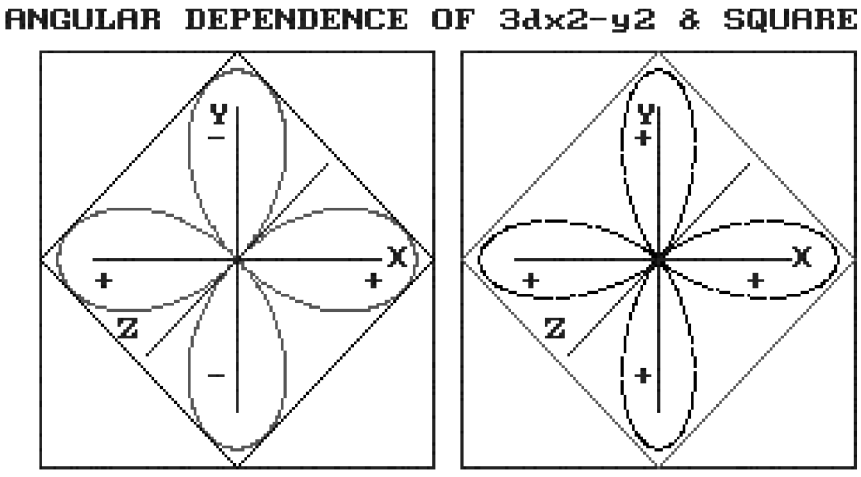


Program for $3d_{x^2-y^2}$

```
REM PROGRAM ANGX2-Y2
Z = 1.1: Z1 = Z - .2: Z2 = Z - .3
CLS : SCREEN 1: COLOR 15, 0
FOR AN = 0 TO 360 STEP .5
```

```
VIEW (20, 30)-(160, 170)
WINDOW (-Z, -Z)-(Z, Z): LINE (-Z, -Z)-(Z, Z), , B
LINE (-Z2, 0)-(Z2, 0): LINE (0, -Z2)-(0, Z2)
LINE (-.5, -.5)-(.5, .5): LINE (-Z, 0)-(0, Z), 2
LINE -(Z, 0), 2: LINE -(0, -Z), 2: LINE -(-Z, 0), 2
TH = (22 / 7) * AN / 180
XX = COS(2 * TH): Y = ABS(XX) * COS(TH): X = ABS(XX) * SIN(TH)
PSET (X, Y), 1
VIEW (170, 30)-(310, 170): LINE (-Z, -Z)-(Z, Z), , B
LINE (-Z2, 0)-(Z2, 0): LINE (0, -Z2)-(0, Z2)
LINE (-.5, -.5)-(.5, .5): LINE (-Z, 0)-(0, Z), 1
LINE -(Z, 0), 1: LINE -(0, -Z), 1: LINE -(-Z, 0), 1
Y1 = XX ^ 2 * COS(TH): X1 = XX ^ 2 * SIN(TH)
'FOR I = 1 TO 100 STEP .5: NEXT I
PSET (X1, Y1), 2
NEXT AN
LOCATE 2, 3: PRINT " ANGULAR DEPENDENCE OF 3dx2-y2 & SQUARE"
LOCATE 7, 11: PRINT "Y": LOCATE 7, 30: PRINT "Y"
LOCATE 16, 7: PRINT "Z": LOCATE 16, 26: PRINT "Z"
LOCATE 13, 19: PRINT "X": LOCATE 13, 37: PRINT "X"
LOCATE 14, 6: PRINT "+": LOCATE 14, 18: PRINT "+"
LOCATE 14, 25: PRINT "+": LOCATE 14, 35: PRINT "+"
LOCATE 8, 11: PRINT "-": LOCATE 18, 11: PRINT "-"
LOCATE 8, 30: PRINT "+": LOCATE 18, 30: PRINT "+"
END
```

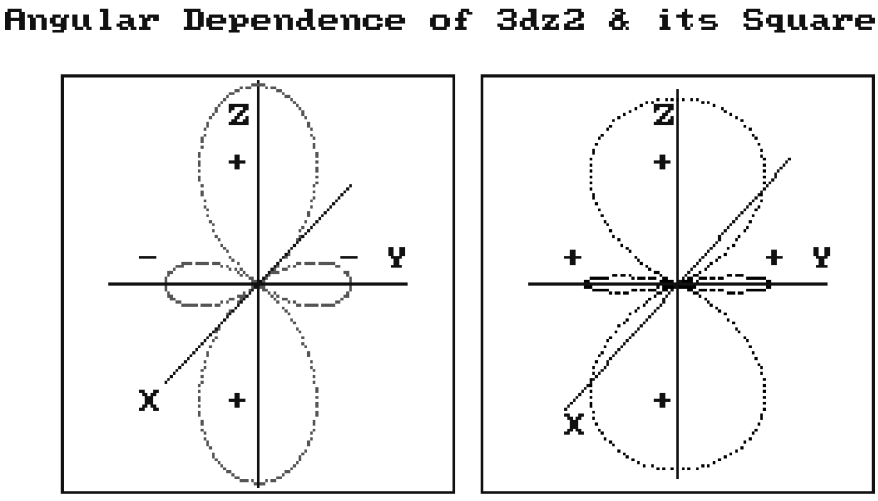
Output



Program for $3d_{z^2}$

```
REM PROGRAM ANG3DZ2
Z = 2.1: Z1 = Z - .5: Z2 = Z - .05
CLS : SCREEN 1: COLOR 15, 0
FOR AN = 0 TO 360 STEP 1
VIEW (20, 30)-(160, 170): WINDOW (-Z, -Z)-(Z, Z)
LINE (-Z, -Z)-(Z, Z), , B: LINE (-Z1, 0)-(Z1, 0)
LINE (0, -Z2)-(0, Z2): LINE (-1, -1)-(1, 1)
TH = (22 / 7) * AN / 180
XX = 3 * COS(TH) ^ 2 - 1
Y = ABS(XX) * COS(TH): X = ABS(XX) * SIN(TH)
PSET (X, Y), 1
VIEW (170, 30)-(310, 170): WINDOW (-Z, -4.5)-(Z, 4.5)
LINE (-Z, -4.5)-(Z, 4.5), , B: LINE (-Z1, 0)-(Z1, 0)
LINE (0, -4.2)-(0, 4.2): LINE (-1.2, -2.7)-(1.2, 2.7)
Y1 = XX ^ 2 * COS(TH): X1 = XX ^ 2 * SIN(TH)
PSET (X1, Y1), 2
'FOR I = 1 TO 1000 STEP .01: NEXT I
NEXT AN
LOCATE 1, 4: PRINT "Angular Dependence of 3dz2 & its Square"
LOCATE 6, 11: PRINT "Z": LOCATE 6, 30: PRINT "Z"
LOCATE 8, 11: PRINT "+": LOCATE 8, 30: PRINT "+"
LOCATE 18, 7: PRINT "X": LOCATE 19, 26: PRINT "X"
LOCATE 12, 7: PRINT "-": LOCATE 12, 16: PRINT "-"
LOCATE 12, 26: PRINT "+": LOCATE 12, 35: PRINT "+"
LOCATE 12, 18: PRINT "Y": LOCATE 12, 37: PRINT "Y"
LOCATE 18, 11: PRINT "+": LOCATE 18, 30: PRINT "+"
END
```

Output



2.48 DISPLAY OF SHAPES AND ORIENTATIONS OF sp AND sp^2 HYBRID ORBITALS

To display the shapes and orientations sp and sp^2 hybrid orbitals.

The wave functions of sp hybrid orbitals are

$$\Psi_{sp_1} = \frac{1}{\sqrt{2}}(1 + \sqrt{3}\cos\theta)$$

$$\Psi_{sp_2} = \frac{1}{\sqrt{2}}(1 - \sqrt{3}\cos\theta)$$

It is suffice to calculate the values of $1 + \sqrt{3}\cos\theta$ and $1 - \sqrt{3}\cos\theta$ for θ varying from 0 to 360°. The data points are shown by the statement `PSET (X, Y)` and X and Y are the *x*-and *y*-components of the data point. These are obtained by multiplying the data point by cos θ and sin θ, respectively.

The wave function of sp^2 hybrid orbitals are

$$\Psi_{sp_1^2} = \frac{1}{\sqrt{3}} + \sqrt{\frac{2}{3}}(\sqrt{3}\sin\theta\cos\phi)$$

$$\Psi_{sp_2^2} = \frac{1}{\sqrt{3}} - \frac{1}{\sqrt{6}}\sqrt{3}\sin\theta\cos\phi + \frac{1}{\sqrt{2}}\sqrt{3}\sin\theta\sin\phi$$

$$\Psi_{sp_3^2} = \frac{1}{\sqrt{3}} - \frac{1}{\sqrt{6}}\sqrt{3}\sin\theta\cos\phi - \frac{1}{\sqrt{2}}\sqrt{3}\sin\theta\sin\phi$$

These orbitals may be shown on XY-plane for which $\theta = 90°$. With this, the expressions of sp^2 orbitals become

$$\Psi_{sp_1^2} = \frac{1}{\sqrt{3}} + \sqrt{2}\cos\phi$$

$$\Psi_{sp_2^2} = \frac{1}{\sqrt{3}} - \frac{1}{\sqrt{2}}\cos\phi + \sqrt{\frac{3}{2}}\sin\phi$$

$$\Psi_{sp_2^3} = \frac{1}{\sqrt{3}} - \frac{1}{\sqrt{2}}\cos\phi - \sqrt{\frac{3}{2}}\sin\phi\,.$$

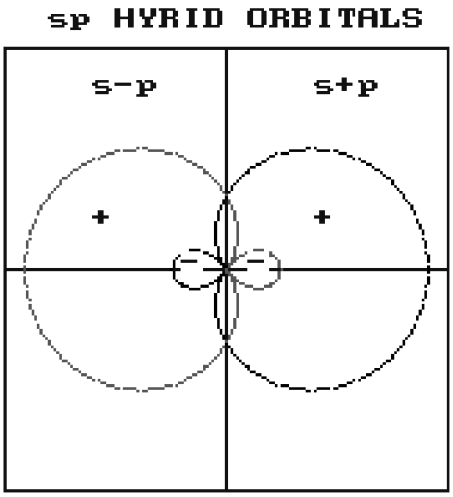
The angle ϕ is varied from 0 to 360° and the data points are displayed by the statement `PSET` (X, Y), where X and Y are the components of the data point which are obtained by multiplying respectively by cos ϕ and sin ϕ.

Progarm for sp Hybrid Orbitals

```
REM PROGRAM HYBRIDSP
CLS : SCREEN 1: COLOR 15, 0
SC = 3
```

```
VIEW (80, 30)-(240, 190)
WINDOW (-SC, -SC)-(SC, SC): LINE (-SC, -SC)-(SC, SC), , B
LINE (-SC, 0)-(SC, 0): LINE (0, -SC)-(0, SC)
FOR AN = 0 TO 360 STEP 1
TH = 3.1416 * AN / 180
XX = ABS(1 + 3 ^ .5 * COS(TH)): XX1 = ABS(1 - 3 ^ .5 * COS(TH))
Y = XX * COS(TH): X = XX * SIN(TH)
Y1 = XX1 * COS(TH): X1 = XX1 * SIN(TH)
PSET (Y, X), 2: PSET (Y1, X1), 1
FOR KK = 1 TO 100 STEP .1: NEXT KK
NEXT AN
LOCATE 3, 13: PRINT "sp HYRID ORBITALS"
LOCATE 12, 15: PRINT "+": LOCATE 12, 25: PRINT "+"
LOCATE 14, 22: PRINT "-": LOCATE 14, 19: PRINT "-"
LOCATE 6, 15: PRINT "s-p": LOCATE 6, 25: PRINT "s+p"
END
```

Output

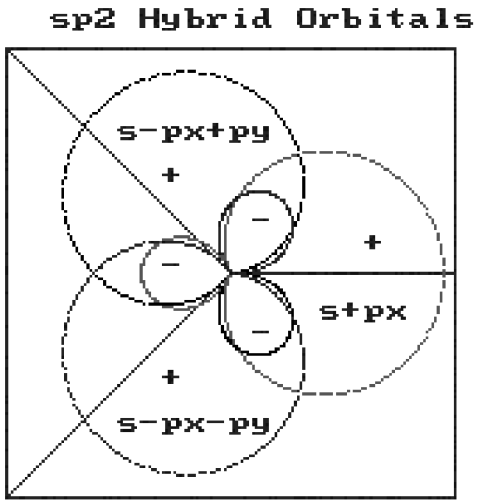


Program for sp^2 Hybrid Orbitals

```
REM PROGRAM HYBRDSP2
CLS : SC = 2.1
SCREEN 1: COLOR 15, 0
VIEW (80, 30)-(240, 190): WINDOW (-SC, -SC)-(SC, SC)
LINE (-SC, -SC)-(SC, SC), , B: LINE (0, 0)-(SC, 0)
LINE (0, 0)-(-SC, SC): LINE (0, 0)-(-SC, -SC)
```

```
FOR AN = 0 TO 360 STEP 1
PHI = 3.1416 * AN / 180
XX1 = ABS(1 / 3 ^ .5 + 2 ^ .5 * COS(PHI))
XX2 = ABS(1 / 3 ^ .5 - 1 / 2 ^ .5 * COS(PHI) + 1.5 ^ .5 * SIN(PHI))
XX3 = ABS(1 / 3 ^ .5 - 1 / 2 ^ .5 * COS(PHI) - 1.5 ^ .5 * SIN(PHI))
Y1 = XX1 * COS(PHI): X1 = XX1 * SIN(PHI)
PSET (Y1, X1), 1: 'a$ = INPUT$(1)
Y2 = XX2 * COS(PHI): X2 = XX2 * SIN(PHI)
PSET (Y2, X2), 2:
'IF XX3 < 0 THEN XX3 = -XX3
Y3 = XX3 * COS(PHI): X3 = XX3 * SIN(PHI)
PSET (Y3, X3), 3:
'FOR KK = 1 TO 100 STEP .1: NEXT KK
NEXT AN
LOCATE 3, 13: PRINT "sp2 Hybrid Orbitals"
LOCATE 14, 18: PRINT "-": LOCATE 13, 27: PRINT "+"
LOCATE 12, 22: PRINT "-": LOCATE 19, 18: PRINT "+"
LOCATE 10, 18: PRINT "+": LOCATE 17, 22: PRINT "-"
LOCATE 8, 16: PRINT "s-px+py"
LOCATE 16, 25: PRINT "s+px"
LOCATE 21, 16: PRINT "s-px-py"
END
```

Output



2.49 DISPLAY OF ENERGIES OF BONDING AND ANTI-BONDING MOLECULAR ORBITALS OF H_2^+

To display the variation of electronic energies of bonding and antibonding molecular orbitals of H_2^+ with the variation of internuclear distance and to determine the minimum in the energy of bonding molecular orbital and the corresponding internuclear distance.

The expressions to be used are

$$E_+ = \frac{1}{R} + \frac{J+K}{1+S} \quad \text{and} \quad E_- = \frac{1}{R} + \frac{J-K}{1-S}$$

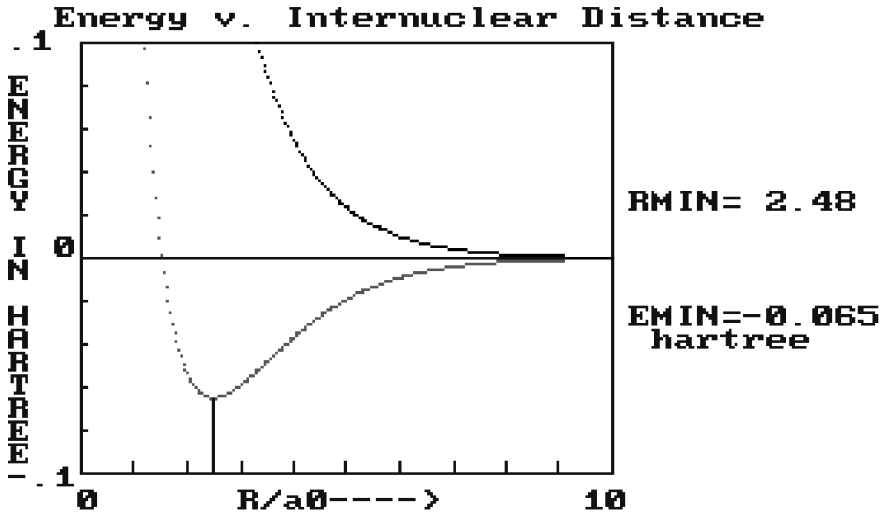
where $J = -\frac{1}{R}\,[1-(1+R)\,e^{-2R}]$; $K = -(1+R)\,e^{-R}$; $S = \left(1+R+\frac{R^2}{3}\right)e^{-R}$

$R = r/a_0$.

Program

```
REM PROGRAM POTENGH2;POTENTIAL ENERGY VARIATION OF H2
CLS : E1MIN = 0
SCREEN 1: COLOR 15, 0
VIEW (25, 20)-(210, 170): WINDOW (0, -.1)-(10, .1)
LINE (0, -.1)-(10, .1), , B: LINE (0, 0)-(10, 0)
FOR R = 10 TO 1 STEP -.04
REP = 1 / R
J = -(1 / R) * (1 - (1 + R) * EXP(-2 * R))
K = -(1 + R) * EXP(-R)
S = (1 + R + R ^ 2 / 3) * EXP(-R)
E1 = REP + (J + K) / (1 + S)
E2 = REP + (J - K) / (1 - S)
IF E1MIN > E1 THEN E1MIN = E1: RMIN = R
'FOR J = 1 TO 100 STEP .001: NEXT J
PSET (R, E1), 1: PSET (R, E2), 2
NEXT R
FOR I = 1 TO 9: II = -.1 + I * .02
LINE (I, -.1)-(I, -.095): LINE (0, II)-(.1, II)
NEXT I
LINE (RMIN, -.1)-(RMIN, E1MIN)
LOCATE 2, 3: PRINT "Energy v. Internuclear Distance"
LOCATE 3, 1: PRINT ".1"
LOCATE 10, 28: PRINT "RMIN="; : PRINT USING "##.##"; RMIN
LOCATE 15, 28: PRINT "EMIN=";
PRINT USING "##.###"; E1MIN; : LOCATE 16, 28: PRINT " hartree"
LOCATE 12, 3: PRINT "0": LOCATE 22, 1: PRINT "-.1"
A$ = "ENERGY IN HARTREE"
FOR I = 1 TO 17: LOCATE 4 + I, 1: PRINT MID$(A$, I, 1): NEXT I
LOCATE 23, 1: PRINT "    0        R/a0---->        10"
END
```

Output



2.50 DISPLAY OF r-DEPENDENCE OF MOLECULAR ORBITALS OF H_2^+

To display the *r*-dependent of bonding and antibonding molecular orbitals of H_2^+.

The expressions of bonding and antibonding molecular orbitals of H_2^+ under zero overlap approximation are

$$R_+ = \frac{1}{\sqrt{2}} [R_{1s(H_a)} + R_{1s(H_b)}] \qquad \text{and} \qquad R_- = \frac{1}{\sqrt{2}} [R_{1s(H_a)} - R_{1s(H_b)}]$$

where $$R_{1,0} = 2\left(\frac{1}{a_0}\right)^{3/2} \exp(-r/a_0)$$

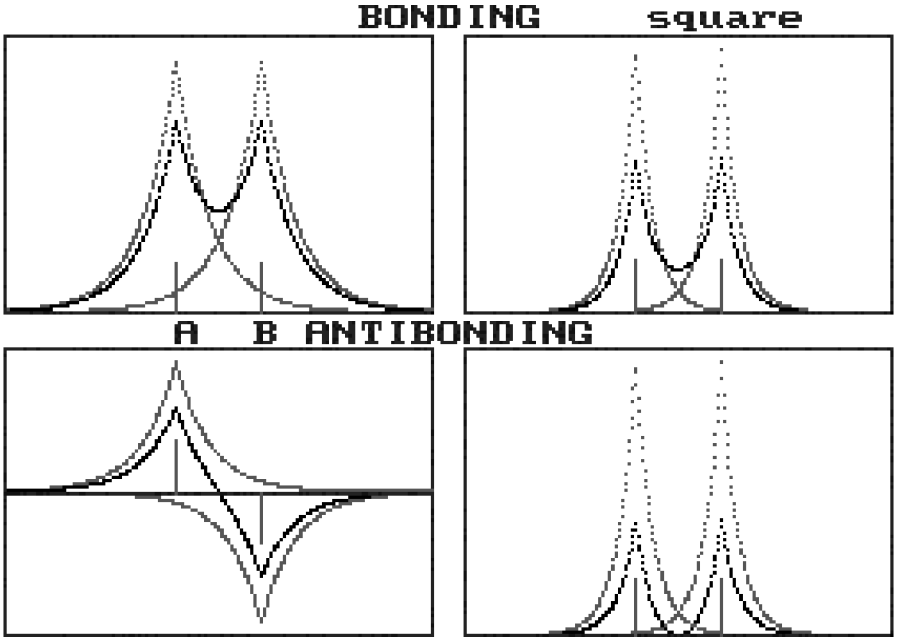
In atomic units $R_{1,0} = 2 \exp(-r)$

Program

```
REM PROGRAM ORBITH2;BONDING & ANTIBONDING WAVE FUNCTIONS OF H2
CLS : SCREEN 1: COLOR 15, 0
RI1 = 5: RI2 = 7.5: DR = .03: IMAX = 12.5
FOR I = 0 TO IMAX STEP DR
XMAX = 2.2: R1 = ABS(RI1 - I): R2 = ABS(RI2 - I)
X1 = 2 * EXP(-R1): X2 = 2 * EXP(-R2)
X1PX2 = (X1 + X2) / 2 ^ .5: X1PX22 = X1PX2 ^ 2
REM-------PLOTS OF WAVE FUNCTIONS--------
VIEW (20, 9)-(150, 93)
```

```
WINDOW (0, 0)-(IMAX, XMAX): LINE (0, 0)-(IMAX, XMAX), , B
PSET (I, X1), 1: PSET (I, X2), 1: PSET (I, X1PX2), 2
XF = -XMAX + 2.6: LINE (5, -XMAX)-(5, XF), 1
LINE (7.5, -XMAX)-(7.5, XF), 1
REM-------PLOTS OF SQUARE OF WAVE FUNCTIONS------
VIEW (160, 9)-(290, 93)
WINDOW (0, 0)-(IMAX, XMAX + 2)
LINE (0, 0)-(IMAX, XMAX + 2), , B
PSET (I, X1 ^ 2), 1: PSET (I, X2 ^ 2), 1: PSET (I, X1PX22), 2
XF = -XMAX + 3: LINE (5, -XMAX)-(5, XF), 1
LINE (7.5, -XMAX)-(7.5, XF), 1
REM--------PLOTS OF WAVE FUNCTIONS--------
X1MX2 = (X1 - X2) / 2 ^ .5: X1MX22 = X1MX2 ^ 2
VIEW (20, 104)-(150, 191)
WINDOW (0, -XMAX)-(IMAX, XMAX)
LINE (0, -XMAX)-(IMAX, XMAX), , B
PSET (I, X1), 1: PSET (I, -X2), 1: PSET (I, X1MX2), 2
LINE (0, 0)-(IMAX, 0)
XF = -XMAX + 3: LINE (5, 0)-(5, XF), 1
LINE (7.5, 0)-(7.5, -XF), 1
REM-------PLOTS OF SQUARE OF WAVE FUNCTIONS-------
VIEW (160, 104)-(290, 191)
WINDOW (0, 0)-(IMAX, XMAX + 2)
LINE (0, 0)-(IMAX, XMAX + 2), , B
PSET (I, X1 ^ 2), 1: PSET (I, X2 ^ 2), 1: PSET (I, X1MX22), 2
LINE (5, 0)-(5, XF), 1: LINE (7.5, 0)-(7.5, XF), 1
'FOR J = 1 TO 100 STEP .001: NEXT J
NEXT I
LOCATE 1, 17: PRINT "BONDING": LOCATE 13, 10
PRINT "A  B": LOCATE 13, 15: PRINT "ANTIBONDING"
LOCATE 1, 28: PRINT "square"
END
```

Output



2.51 DISPLAY OF RADIANT ENERGY OF A BLACK BODY

To display the radiant energy emitted by a black body at temperatures 1000 K and 1200 K as given by the following expressions.

(i) Variation in terms of frequency.

$$\frac{\varepsilon_\nu dN_\nu}{V} = \frac{8\pi h}{c^3} \frac{V^3}{\exp(h\nu / kT) - 1} d\nu$$

Frequency ν to be varied from 0 to 15×10^{13} Hz.

(ii) Variation in terms of wavelength

$$\frac{E_\lambda dN_\lambda}{V} = -\frac{8\pi hc}{\lambda^5} \frac{1}{\exp(hc / \lambda kT) - 1} d\lambda$$

Wavelength λ to be varied from 0 to 2×10^{-6} m.

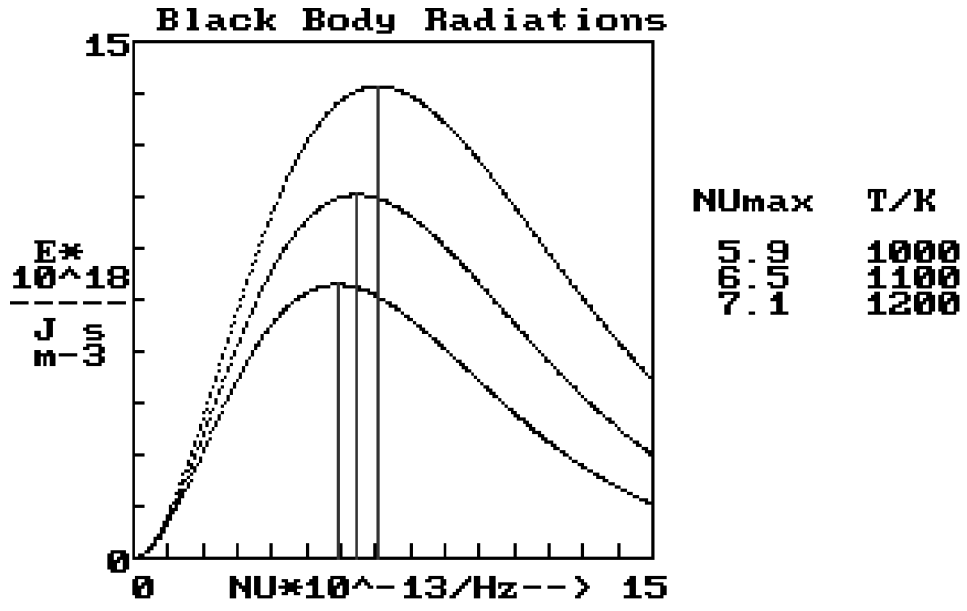
Also verify that $T\lambda_{max}$ = constant.

Program for frequency variation

```
REM PROGRAM BBRAD;BLACK BODY RADITION
CLS : C = 3E+08: K = 8.314 / 6.022E+23: H = 6.626E-34
EMAX = 0: J = 0: PI = 22 / 7
```

```
SCREEN 1: COLOR 15, 0: VIEW (40, 10)-(210, 170)
WINDOW (0, 0)-(15, 15): LINE (0, 0)-(15, 15), , B
FOR T = 1000 TO 1200 STEP 100
FOR NUP = .1 TO 15 STEP .05
NU = NUP * 1E+13
ENU = (8 * PI * H) * (NU / C) ^ 3 / (EXP(H * NU / (K * T)) - 1)
ENU = ENU * 1E+18
IF EMAX < ENU THEN EMAX = ENU: NUMAX = NUP
PSET (NUP, ENU)
FOR I = 1 TO 1000 STEP 10: NEXT I
NEXT NUP
LOCATE 8, 29: PRINT "NUmax  T/K"
LOCATE 10 + J, 29: PRINT USING "##.#"; NUMAX
LOCATE 10 + J, 34: PRINT USING "  ####"; T
J = J + 1: LINE (NUMAX, 0)-(NUMAX, EMAX), 1
NEXT T
LOCATE 1, 7: PRINT "Black Body Radiations"
LOCATE 2, 4: PRINT "15"
LOCATE 10, 2: PRINT "E*": LOCATE 11, 1: PRINT "10^18"
LOCATE 12, 1: PRINT "-----": LOCATE 13, 2: PRINT "J s"
LOCATE 14, 2: PRINT "m-3": LOCATE 22, 5: PRINT "0"
LOCATE 23, 2: PRINT "     0    NU*10^-13/Hz--> 15"
FOR I = 1 TO 9: SC = 15 * I * .1: LINE (0, SC)-(.3, SC): NEXT I
FOR I = 1 TO 14: LINE (I, 0)-(I, .4): NEXT I
END
```

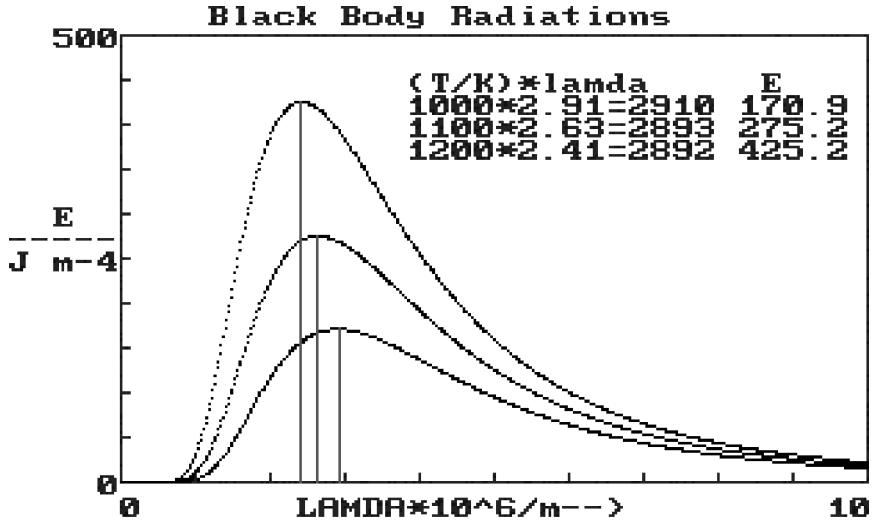
Output



Program for Wave length Variation

```
REM BLACK BODY RADIATION-LAMBDA
C = 3E+08: K = 1.38E-23: H = 6.626E-34: EMAX = 0: J = 0
CLS : SCREEN 1: COLOR 15, 0: VIEW (40, 10)-(310, 170)
WINDOW (0, 0)-(10, 500): LINE (0, 0)-(10, 500), , B
FOR T = 1000 TO 1200 STEP 100
FOR LAMP = .01 TO 10 STEP .02
LAM = LAMP * .000001
TERM1 = 8 * (22 / 7) * H * C / LAM ^ 5
TERM2 = 1 / (EXP(H * C / (LAM * K * T)) - 1)
ELAM = TERM1 * TERM2
IF EMAX < ELAM THEN EMAX = ELAM: LAMMAX = LAMP
PSET (LAMP, ELAM), 2
FOR I = 1 TO 1000 STEP 10: NEXT I
NEXT LAMP
LINE (LAMMAX, 0)-(LAMMAX, EMAX), 1
LOCATE 4, 19: PRINT "(T/K)*lamda      E"
LOCATE 5 + J, 19: PRINT USING "####"; T; : PRINT "*";
'LOCATE 10 + J, 27:
PRINT USING "#.##"; LAMMAX;
PRINT "="; : PRINT USING "####"; T * LAMMAX
LOCATE 5 + J, 33: PRINT USING " ###.#"; EMAX
a$ = INPUT$(1): J = J + 1
NEXT T
LOCATE 1, 10: PRINT "Black Body Radiations"
LOCATE 2, 3: PRINT "500"
LOCATE 10, 1: PRINT "   E"
LOCATE 11, 1: PRINT "-----": LOCATE 12, 1: PRINT "J m-4"
LOCATE 22, 5: PRINT "0"
LOCATE 23, 2: PRINT "     0       LAMDA*10^6/m-->           10"
FOR I = 1 TO 9
SC = 10 * I * .1: SC1 = 500 * I * .1
LINE (SC, 0)-(SC, 10): LINE (0, SC1)-(.1, SC1)
NEXT I
END
```

Output



2.52 DISPLAY OF MASS FRACTION OF *k*mer of a Polymer

To display the mass fraction of *k*mer against k for different values of p as per the expression

$$w_k = kp^{k-1}(1-p)^2$$

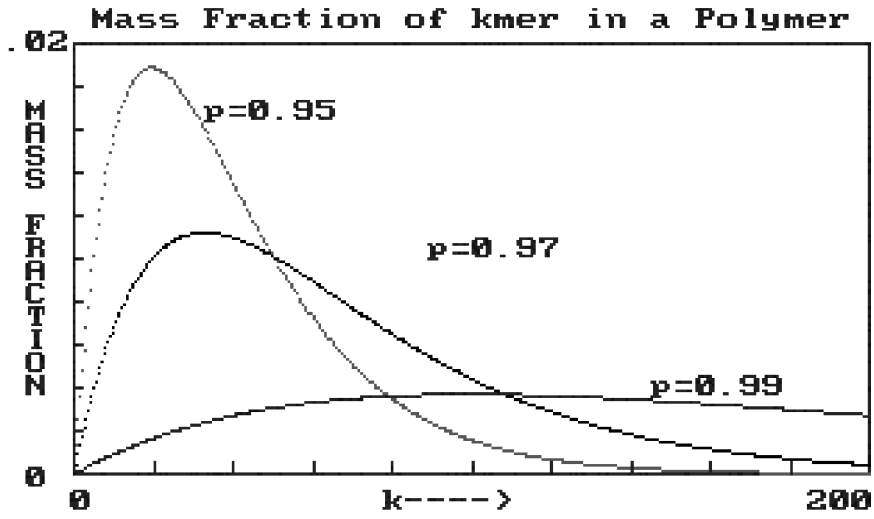
Take p = 0.95, 0.97 and 0.99. Also show that the area under each curve is equal to 1.

Program

```
REM MASS FRACTION OF kmer IN A POLYMER
CLS : SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(310, 170)
WINDOW (0, 0)-(200, .02): LINE (0, 0)-(200, .02), , B
J = 0
FOR P = .95 TO .99 STEP .02
FOR K = 0 TO 200 STEP .5
W = K * P ^ (K - 1) * (1 - P) ^ 2
PSET (K, W), J + 1
FOR I = 1 TO 100 STEP .1: NEXT I
NEXT K
LOCATE 6 * (J + 1), 10 * (J + 1): PRINT "p=";
PRINT USING "#.##"; P: J = J + 1
NEXT P
LOCATE 2, 5: PRINT "Mass Fraction of kmer in a Polymer"
LOCATE 3, 1: PRINT ".02": A$ = "MASS FRACTION"
FOR I = 1 TO 13
B$ = MID$(A$, I, 1): LOCATE 5 + I, 2: PRINT B$
NEXT I
```

```
LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT " 0                k---->                200"
FOR I = 1 TO 9
SC = 200 * I * .1: SC1 = .02 * I * .1
LINE (SC, 0)-(SC, .0005): LINE (0, SC1)-(2, SC1)
NEXT I
END
```

Output



2.53 DISPLAY OF VARIATION OF HEAT CAPACITY OF SOLIDS AT CONSTANT VOLUME

To display the variation of molar heat capacity at constant volume of a monatomic solid with the variation in temperature

The expression for Einstein solids is

$$\frac{C_{V,\mathrm{m}}}{R} = \frac{3(\theta_\mathrm{E}/T)^2 \mathrm{e}^{\theta_\mathrm{E}/T}}{(\mathrm{e}^{\theta_\mathrm{E}/T} - 1)^2}$$

Vary T/θ_E from 0.01 to 2

The expression for Debye solids is

$$\frac{C_{V,\mathrm{m}}}{R} = 3\left[4D - \frac{3x_\mathrm{D}}{\mathrm{e}^{x_\mathrm{D}} - 1}\right]$$

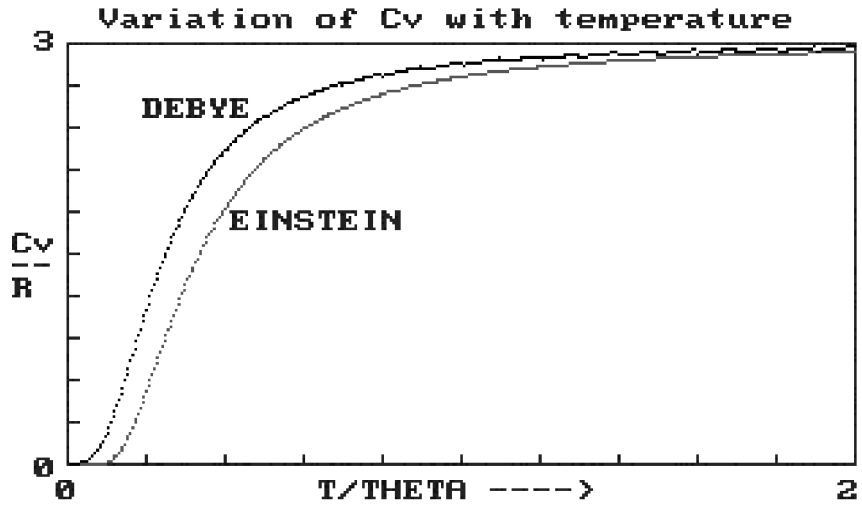
where $x_\mathrm{D} = \theta_\mathrm{D}/T$ and $D = \dfrac{3}{x_\mathrm{D}^3}\displaystyle\int_0^{x_\mathrm{D}} \frac{x^3}{\mathrm{e}^x - 1}\,\mathrm{d}x$

Take $\theta_\mathrm{E} = 1$, $\theta_\mathrm{D} = 1$ and vary T/θ from 0.01 to 2.

Program

```
REM PROGRAM Cv PLOTS OF SOLIDS
CLS : SCREEN 1: COLOR 15, 0: VIEW (20, 20)-(310, 170)
WINDOW (0, 0)-(2, 3): LINE (0, 0)-(2, 3), , B
THETAE = 1: THETAD = 1
FOR T = .01 TO 2 STEP .005
IT = THETAE / T
CV = 3 * IT ^ 2 * EXP(IT) / ((EXP(IT) - 1) ^ 2)
PSET (1 / IT, CV), 1
XD = THETAD / T: SUM = 0: DX = XD / 1000
FOR X = .01 TO XD STEP DX
SUM = SUM + X ^ 3 / (EXP(X) - 1) * DX
NEXT X
D = (3 / XD ^ 3) * SUM
CV = 3 * (4 * D - 3 * XD / (EXP(XD) - 1))
PSET (1 / XD, CV), 2
NEXT T
LOCATE 2, 5: PRINT "Variation of Cv with temperature"
LOCATE 3, 2: PRINT "3"
LOCATE 6, 7: PRINT "DEBYE": LOCATE 11, 11: PRINT "EINSTEIN"
LOCATE 12, 1: PRINT "Cv": LOCATE 13, 1
PRINT "--": LOCATE 14, 1: PRINT "R": LOCATE 22, 2: PRINT "0"
LOCATE 23, 2: PRINT " 0              T/THETA ---->              2"
FOR I = 1 TO 9
LINE (I * .2, 0)-(I * .2, .05): LINE (0, I * .3)-(.03, I * .3)
NEXT I
END
```

Output



2.54 DISPLAY OF UNIT CELLS OF A CUBIC SYSTEM

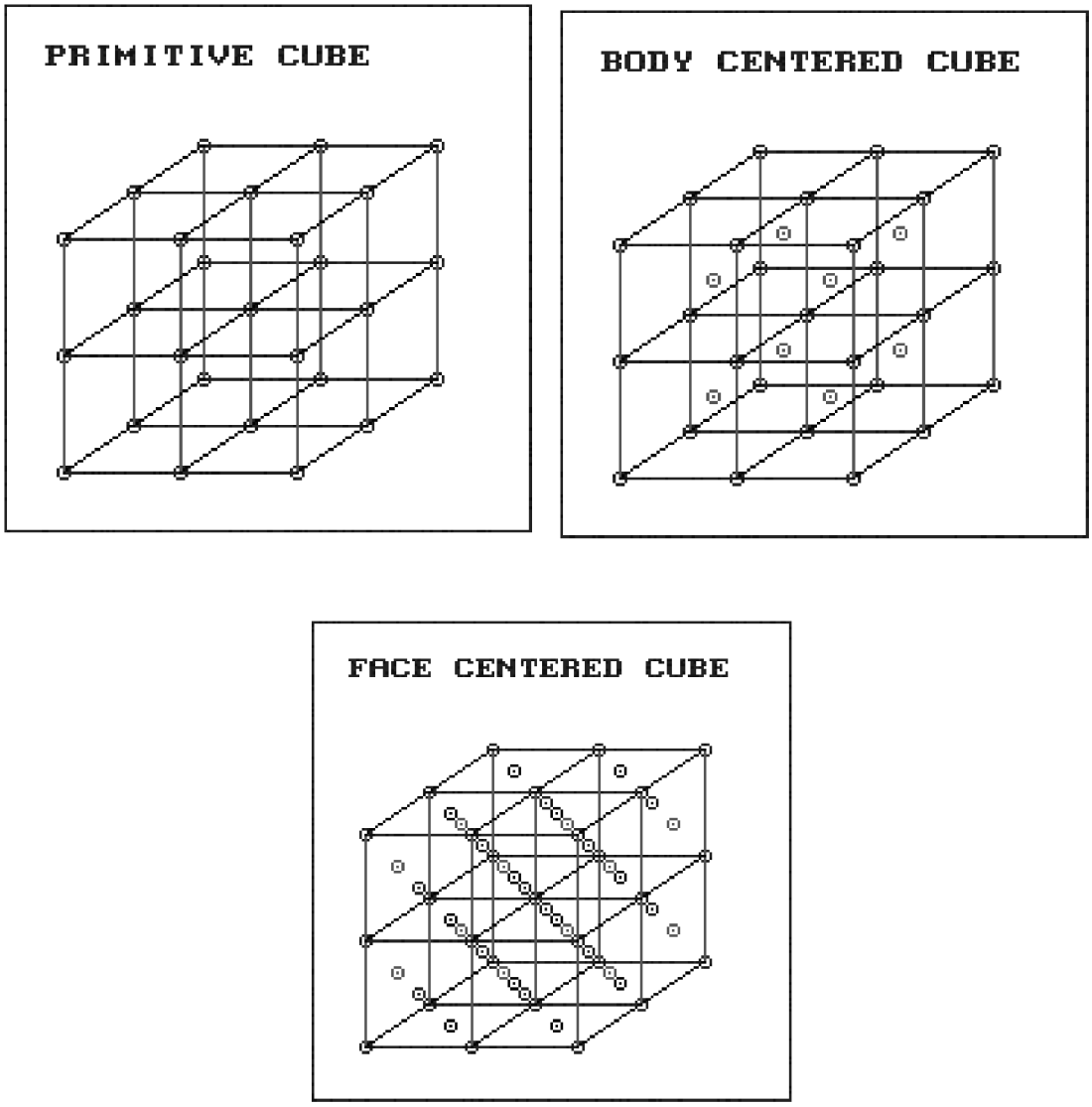
A cubic system has three Bravais unit cells–Primitive, Body-Centred and Face-Centred cubic unit cells.

Program

```
REM PROGRAM CUBIC;BRAVIS LATTICE OF CUBIC SYSTEM
REM No. of units(i)horizontally NH,(ii)vertically NV,(iii) back NB
READ NH, NV, NB: DATA 2,2,2
CLS : OPT = 1: NT = NH + NV + NB - 1
A = .6: B = .4: X = .5: Y = NT
NHM1 = NH - 1: NHP1 = NH + 1: NVM1 = NV - 1
NVP1 = NV + 1: NBM1 = NB - 1: NBP1 = NB + 1
SCREEN 1: COLOR 15, 0: VIEW (10, 10)-(190, 190)
WINDOW (X, X)-(Y, Y): LINE (X, X)-(Y, Y), , B
4 IF INT(OPT) = 4 THEN 25
CLS : LINE (X, X)-(Y, Y), , B
REM------LOCATING LATTICE POINTS------
     REM I: Number of unit cells plus 1 along x-axis
REM J: Number of unit cells plus 1 along y-axis
REM K: Number of unit cells plus 1 along z-axis
FOR K = 1 TO NBP1: K1 = (K - 1) * A: K2 = (K - 1) * B
FOR J = 1 TO NVP1: JJ = J + K2
FOR I = 1 TO NHP1: II = I + K1
PSET (II, JJ): CIRCLE (II, JJ), .01 * NT
NEXT I: NEXT J: NEXT K
LOCATE 4, 4
ON OPT GOTO 5, 10, 15, 25
5 PRINT "PRIMITIVE CUBE": GOTO 20
REM ------SPOTTING BODY CENTERED POINTS------
10 PRINT "BODY CENTERED CUBE"
FOR K = 1 TO NB: K1 = (K - 1) * A: K2 = (K - 1) * B
FOR J = 1 TO NV: JJ = J + K2 + .5 + B / 2
FOR I = 1 TO NH: II = I + K1 + .5 + A / 2
PSET (II, JJ), 1: CIRCLE (II, JJ), .01 * NT, 1
NEXT I: NEXT J: NEXT K
GOTO 20
REM ------SPOTTING FACE CENTERED POINTS------
REM Points on front faceing faces
15 PRINT "FACE CENTERED CUBE "
FOR K = 1 TO NBP1: K1 = (K - 1) * A: K2 = (K - 1) * B
FOR J = 1 TO NV: JJ = J + .5 + K2
```

```
FOR I = 1 TO NH: II = I + .5 + K1
PSET (II, JJ), 3: CIRCLE (II, JJ), .01 * NT, 3
NEXT I: NEXT J: NEXT K
REM Points on top faceing faces
FOR K = 1 TO NB: K1 = (2 * K - 1) * A: K2 = (2 * K - 1) * B
FOR J = 1 TO NVP1: JJ = J + K2 / 2
FOR I = 1 TO NH: II = I + .5 + K1 / 2
PSET (II, JJ), 2: CIRCLE (II, JJ), .01 * NT, 2
NEXT I: NEXT J: NEXT K
REM Points on the side faces
FOR K = 1 TO NB: K1 = (2 * K - 1) * A: K2 = (2 * K - 1) * B
FOR J = 1 TO NV: JJ = J + .5 + K2 / 2
FOR I = 1 TO NHP1: II = I + K1 / 2
PSET (II, JJ), 1: CIRCLE (II, JJ), .01 * NT, 1
NEXT I: NEXT J: NEXT K
REM ------DRAWING LINES HORIZONTALLY------
REM I : Number of unit cells along x-axis
REM J : Number of unit cells along y-axis plus 1
REM K : Number of unit cells along z-axis:initial K=0
20 FOR I = 1 TO NH: FOR J = 1 TO NVP1: FOR K = 0 TO NB
LINE (I + K * A, J + K * B)-(I + K * A + 1, J + K * B)
NEXT K: NEXT J: NEXT I
REM ------DRAWING LINES VERTICALLY------
REM I :Number of unit cells along x-axis plus 1
REM J :Number of unit cells along y-axis
REM K :Number of unit cells along z-axis:Initial K=O
FOR I = 1 TO NHP1: FOR J = 1 TO NV: FOR K = 0 TO NB
LINE (I + K * A, J + K * B)-(I + K * A, J + K * B + 1), 1
NEXT K: NEXT J: NEXT I
REM ------DRAWING LINES SLANTING------
REM I: Number of unit cells along x-axis plus 1
REM J: Number of unit cells along y-axis plus 1
REM K: Number of unit cells along z-axis
FOR I = 1 TO NHP1: FOR J = 1 TO NVP1: FOR K = 1 TO NB
LINE (I, J)-(I + K * A, J + K * B), 2
NEXT K: NEXT J: NEXT I
OPT = OPT + 1: A$ = INPUT$(1): GOTO 4
25 END
```

Output



2.55 DISPLAY OF 100, 110, 111 AND 010 PLANES OF A PRIMITIVE CUBIC SYSTEM

To display of 100, 110, 111 and 010 planes of a cubic lattice.

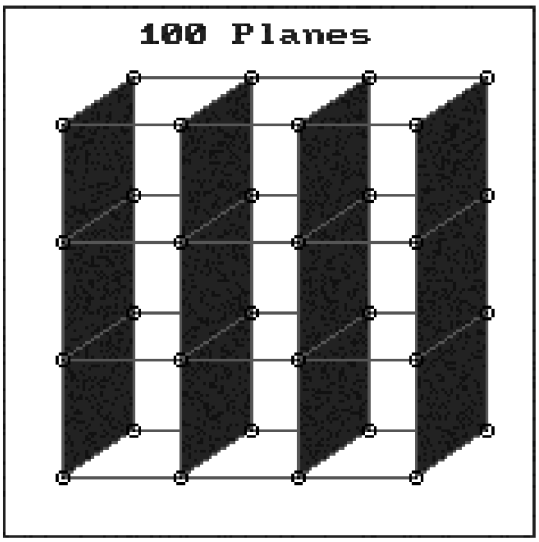
Program for 100 planes

```
REM 100 PLANES OF CUBIC SYSTEM
READ NH, NV: DATA 3,3: NB = 1: NT = NH + 2
```

```
CLS : A = .6: B = .4: X = .5: Y = NT
NHM1 = NH - 1: NHP1 = NH + 1
NVM1 = NV - 1: NVP1 = NV + 1
NBM1 = NB - 1: NBP1 = NB + 1
SCREEN 1: COLOR 15, 0: VIEW (10, 10)-(190, 190)
WINDOW (X, X)-(Y, Y): LINE (X, X)-(Y, Y), , B
NBF = NB: L = 1: GOSUB 100
L = 3: GOSUB 200
L = 3: GOSUB 300
FOR I = 1.3 TO NH + 1.4: FOR J = 1.7 TO NV + .8
PAINT (I, J), 3: NEXT J: NEXT I
NBF = NBM1: L = 1: GOSUB 100
L = 1: GOSUB 200
L = 1: GOSUB 300
LOCATE 3, 8: PRINT "100 Planes"
REM SPOTTING LATTICE POINTS
REM I : Number of unit cells plus 1 along x-axis
REM J : Number of unit cells plus 1 along y-axis
REM K : Number of unit cells plus 1 along z-axis
FOR K = 1 TO NBP1: K1 = (K - 1) * A: K2 = (K - 1) * B
FOR J = 1 TO NVP1: JJ = J + K2
FOR I = 1 TO NHP1: II = I + K1
PSET (II, JJ), 2: CIRCLE (II, JJ), .01 * NT, 2
NEXT I: NEXT J: NEXT K
GOTO 400
REM DRAWING HORIZONTAL LINES
REM I : Number of unit cells along x-axis
REM J : Number of unit cells along y-axis PLUS 1
REM K : Number of unit cells along z-axis : Initial K=0
100 FOR I = 1 TO NH: FOR J = 1 TO NVP1: FOR K = 0 TO NBF
LINE (I + K * A, J + K * B)-(I + K * A + 1, J + K * B), L
NEXT K: NEXT J: NEXT I
RETURN
REM DRAWING VERTICAL LINES
'REM I :number of unit cells along x-axis PLUS 1
'REM J :number of unit cells along y-axis
'REM K :number of unit cells along z-axis : Initial K=0
200 FOR I = 1 TO NHP1: FOR J = 1 TO NV: FOR K = 0 TO NB
LINE (I + K * A, J + K * B)-(I + K * A, J + K * B + 1), L
NEXT K: NEXT J: NEXT I
```

```
RETURN
REM DRAWING SLANTING LINES
REM I: number of unit cells along x-axis plus 1
REM J: number of unit cells along y-axis plus 1
REM K: number of unit cells along z-axis
300 FOR I = 1 TO NHP1: FOR J = 1 TO NVP1: FOR K = 1 TO NB
LINE (I, J)-(I + K * A, J + K * B), L
NEXT K: NEXT J: NEXT I
RETURN
400 END
```

Output

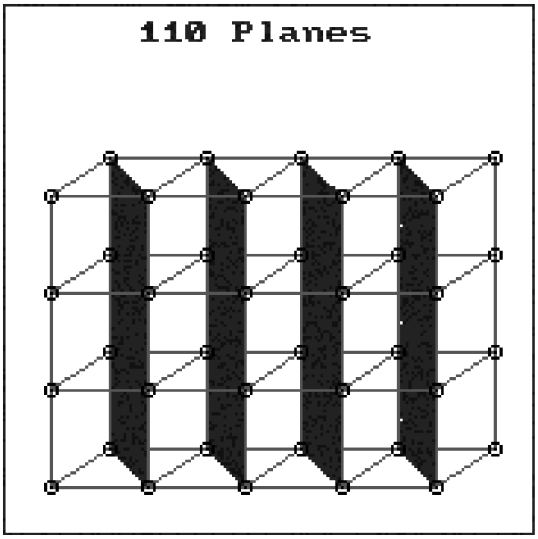


Program for 110 planes

```
REM 110 PLANES OF CUBIC SYSTEM
READ NH, NV: NB = 1: DATA 4,3: NT = NH + 2
CLS : a = .6: B = .4: X = .5: Y = NT
NHM1 = NH - 1: NHP1 = NH + 1
NVM1 = NV - 1: NVP1 = NV + 1
NBM1 = NB - 1: NBP1 = NB + 1
SCREEN 1: COLOR 15, 0: VIEW (10, 10)-(190, 190)
WINDOW (X, X)-(Y, Y): LINE (X, X)-(Y, Y), , B
FK = NB: GOSUB 100
L = 3: GOSUB 200
GOSUB 300
FOR K = 1 TO NB: FOR I = 1 TO NH: FOR J = 1 TO NVP1
LINE (I + K * a, J + K * B)-(I + 1 + (K - 1) * a, J + (K - 1)
* B), 3
```

```
NEXT J: NEXT I: NEXT K
FOR K = 1 TO NB: FOR I = 1 TO NH: FOR J = 1 TO NV
PAINT (I + K * a + .5 - a / 2, J + K * B + .5 - B / 2), 3
NEXT J: NEXT I: NEXT K
FK = NBM1: GOSUB 100
L = 1: GOSUB 200
REM ------SPOTTING LATTICE POINTS------
REM I : Number of unit cells plus 1 along x-axis
REM J : Number of unit cells plus 1 along y-axis
REM K : Number of unit cells plus 1 along z-axis
FOR K = 1 TO NBP1: K1 = (K - 1) * a: K2 = (K - 1) * B
FOR J = 1 TO NVP1: FOR I = 1 TO NHP1
PSET (I + K1, J + K2), 2: CIRCLE (I + K1, J + K2), .01 * NT, 2
NEXT I: NEXT J: NEXT K
GOTO 400
REM ------DRAWING HORIZONTAL LINES------
REM I : number of unit cells along x-axis
REM J : number of unit cells along y-axis plus 1
REM K : number uf unit cells along z-axis :inital K=0
100 FOR I = 1 TO NH: FOR J = 1 TO NVP1: FOR K = 0 TO FK
LINE (I + K * a, J + K * B)-(I + K * a + 1, J + K * B), 1
NEXT K: NEXT J: NEXT I
RETURN
REM ------DRAWING VERTICAL LINES------
REM I :number of unit cells along x-axis plus 1
REM J :number of unit cells along y-axis
REM K :number of unit cells along z-axis :initial K=0
200 FOR I = 1 TO NHP1: FOR J = 1 TO NV: FOR K = 0 TO NB
LINE (I + K * a, J + K * B)-(I + K * a, J + K * B + 1), L
NEXT K: NEXT J: NEXT I
RETURN
REM ------DRAWING SLANTING LINES------
REM I: number of unit cells along x-axis plus 1
REM J: number of unit cells along y-axis plus 1
REM K: number of unit cells along z-axis
300 FOR I = 1 TO NHP1: FOR J = 1 TO NVP1: FOR K = 1 TO NB
LINE (I, J)-(I + K * a, J + K * B), 1
NEXT K: NEXT J: NEXT I
RETURN
400 LOCATE 3, 8: PRINT "110 Planes"
END
```

Output

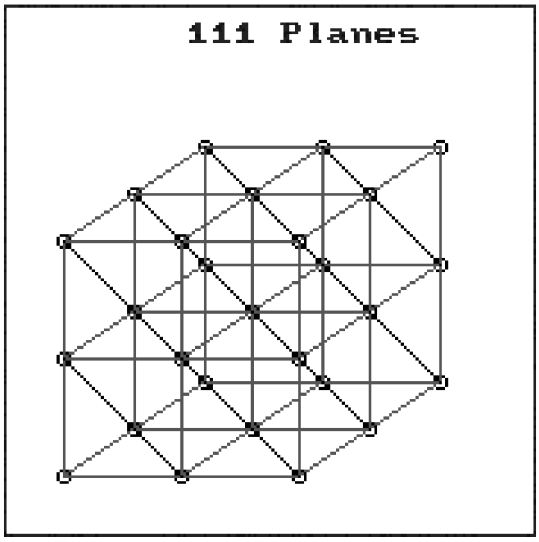


Program for 111 Planes

```
REM 111 PLANES OF CUBIC SYSTEM
READ NH, NV, NB: DATA 2,2,2: NT = NH + 3
CLS : NHP1 = NH + 1: NHM1 = NH - 1
NVP1 = NV + 1: NVM1 = NV - 1
NBP1 = NB + 1: NBM1 = NB - 1
A = .6: B = .4: X = .5: Y = NT
SCREEN 1: COLOR 15, 0: VIEW (10, 10)-(190, 190)
WINDOW (X, X)-(Y, Y): LINE (X, X)-(Y, Y), , B
REM -------SPOTTING LATTICE POINTS---------
REM I represents number of unit cells plus 1 along x-axis
REM J represents number of unit cells plus 1 along y-axis
REM K represents number of unit cells plus 1 along z-axis
FOR K = 1 TO NBP1: K1 = (K - 1) * A: K2 = (K - 1) * B
FOR J = 1 TO NVP1: FOR I = 1 TO NHP1
PSET (I + K1, J + K2), 2: CIRCLE (I + K1, J + K2), .01 * Y, 2
NEXT I: NEXT J: NEXT K
REM --------DRAWING HORIZONTAL LINES--------
REM I : number of unit cells along x-axis
REM J : number of unit cells plus 1 along y-axis
REM K:nnmber of unit cells along z-axis:initial K=0
FOR I = 1 TO NH: FOR J = 1 TO NVP1: FOR K = 0 TO NB
LINE (I + K * A, J + K * B)-(I + K * A + 1, J + K * B), 1
NEXT K: NEXT J: NEXT I
```

```
REM --------DRAWING VERTICAL LINES--------
REM I :number of unit cells plus 1 along x-axis
REM J :number of unit cells along y-axis
REM K :number of unit cells along z-axis :initial K=0
FOR I = 1 TO NHP1: FOR J = 1 TO NV: FOR K = 0 TO NB
LINE (I + K * A, J + K * B)-(I + K * A, J + K * B + 1), 1
NEXT K: NEXT J: NEXT I
REM ---------DRAWING SLANTING LINES--------
REM I: number of unit cells plus 1 along x-axis
REM J: number of unit cells plus 1 along y-axis
REM K: number of unit cells along z-axis
FOR I = 1 TO NHP1: FOR J = 1 TO NVP1: FOR K = 1 TO NB
LINE (I, J)-(I + K * A, J + K * B), 1
NEXT K: NEXT J: NEXT I
REM --------DRAWING 111 LINES--------
FOR K = 2 TO NH + NV + NB
IF K <= NHP1 THEN X1 = K: Y1 = 1: GOTO 5
IF K - NHP1 <= NB THEN
X1 = NHP1 + (K - NHP1) * A
Y1 = 1 + (K - NHP1) * B
GOTO 5
END IF
IF K - NHP1 - NB <= NV THEN
X1 = NHP1 + NB * A
Y1 = 1 + NB * B + (K - NHP1 - NB)
END IF
5 IF K <= NVP1 THEN X2 = 1: Y2 = K: GOTO 10
IF K - NVP1 <= NB THEN
X2 = 1 + (K - NVP1) * A
Y2 = NVP1 + (K - NVP1) * B
GOTO 10
END IF
IF K - NVP1 - NB <= NH THEN
X2 = 1 + NB * A + (K - NVP1 - NB)
Y2 = NVP1 + NB * B
END IF
10  LINE (X1, Y1)-(X2, Y2), 2
A$ = INPUT$(1)
NEXT K
LOCATE 3, 10: PRINT "111 Planes"
END
```

Output

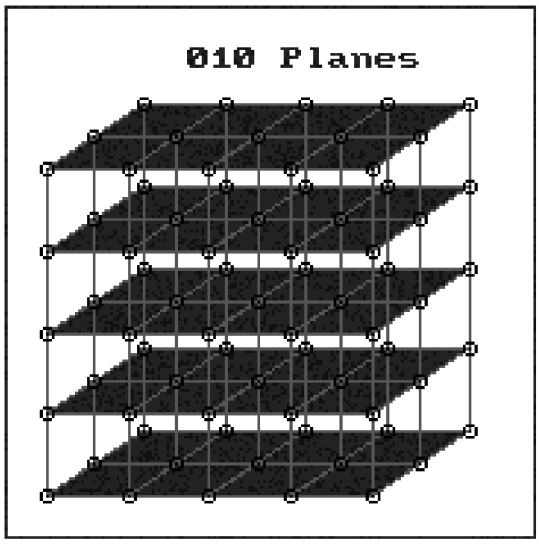


Program for 010 Planes

```
REM 010 PLANES OF CUBIC SYSTEM
READ NH, NV: DATA 4,4: NB = 2
CLS : a = .6: B = .4: X = .5: Y = NH + 3
NHM1 = NH - 1: NHP1 = NH + 1
NVM1 = NV - 1: NVP1 = NV + 1
NBM1 = NB - 1: NBP1 = NB + 1
SCREEN 1: COLOR 15, 0: VIEW (10, 10)-(190, 190)
WINDOW (X, X)-(Y, Y): LINE (X, X)-(Y, Y), , B
REM -------DRAWING HORIZONTAL LINES-------
L = 3: GOSUB 100
REM -------DRAWING SLANTING LINES---------
L = 3: GOSUB 300
REM -------DRAWING 010 PLANES-------------
FOR I = 1 TO NH: II = I + .8: FOR J = 1 TO NVP1: JJ = J + .2
FOR K = 1 TO NB: K1 = (K - 1) * a: K2 = (K - 1) * B
PAINT (II + K1, JJ + K2), 3
NEXT K: NEXT J: NEXT I
REM ----DRAWING HORIZONTAL,VERTICAL & SLANTING LINES----
L = 1: GOSUB 100: L = 1: GOSUB 200: L = 1: GOSUB 300
GOTO 400
REM -----SUBROUTINE FOR DRAWING HORIZONTAL LINES-----
100 FOR I = 1 TO NH: FOR J = 1 TO NVP1: FOR K = 0 TO NB
LINE (I + K * a, J + K * B)-(I + K * a + 1, J + K * B), L
NEXT K: NEXT J: NEXT I
```

```
RETURN
REM -----SUBROUTINE FOR DRAWING VERTICAL LINES-------
200 FOR I = 1 TO NHP1: FOR J = 1 TO NV: FOR K = 0 TO NB
LINE (I + K * a, J + K * B)-(I + K * a, J + K * B + 1), L
NEXT K: NEXT J: NEXT I
RETURN
REM -----SUBROUTINE FOR DRAWING SLANTING LINES------
300 FOR I = 1 TO NHP1: FOR J = 1 TO NVP1: FOR K = 1 TO NB
LINE (I, J)-(I + K * a, J + K * B), L
NEXT K: NEXT J: NEXT I
RETURN
400 LOCATE 4, 10: PRINT "010 Planes"
REM -------SPOTTING LATTICE POINTS--------
FOR K = 1 TO NBP1: K1 = (K - 1) * a: K2 = (K - 1) * B
FOR J = 1 TO NVP1: FOR I = 1 TO NHP1
PSET (I + K1, J + K2), 2: CIRCLE (I + K1, J + K2), .01 * Y, 2
NEXT I: NEXT J: NEXT K
END
```

Output



2.56 DISPLAY OF JOULE-THOMSON COEFFICIENT AND INVERSION TEMPERATURE

(a) To display the variation of $b/C_{p,\,\mathrm{m}}$, $2a/C_{p,\,\mathrm{m}}\,RT$ and μ_{JT} with temperature for N_2 gas.
The values of b = 0.0392 $\mathrm{dm^3\ mol^{-1}}$, a = 141 kPa $\mathrm{dm^6\ mol^{-2}}$ and $C_{p,\,\mathrm{m}}$ = 34.35 $\mathrm{J\ K^{-1}\ mol^{-1}}$. The expression of μ_{JT} is

$$\mu_{\mathrm{JT}} = \frac{1}{C_{p,\mathrm{m}}}\left(\frac{2a}{RT} - b\right)$$

To display the variation of inversion temperature of N_2 with pressure of the gas.
The expression to be used is

$$T_i = \frac{a \pm \sqrt{a^2 - 3ab^2 p}}{bR}$$

(b) To display the variation of second virial coefficient B with temperature for N_2 gas and to locate its inversion temperature by drawing a slope line to B versus T passing from the origin.

The expression of the second virial coefficient is $B = b - \frac{a}{RT}$

The slope of B versus T at the inversion temperature is

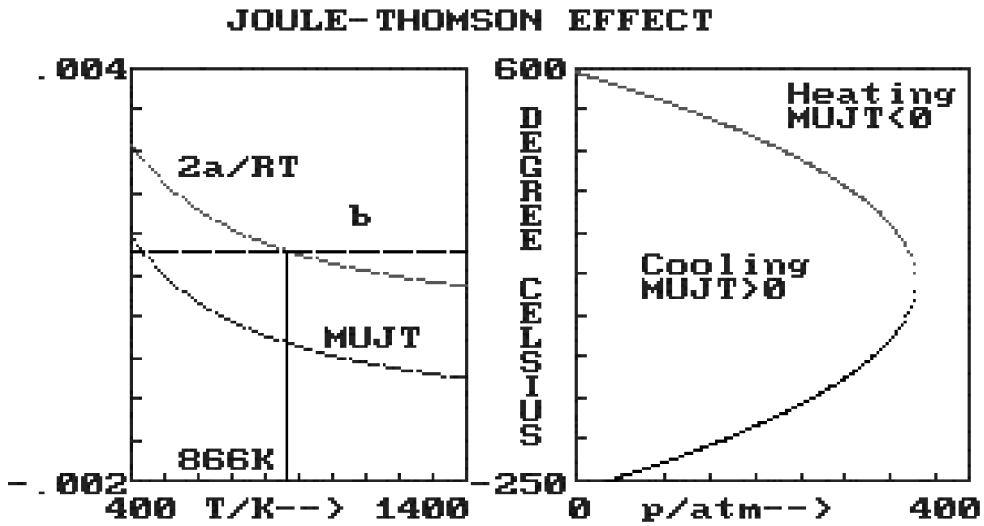
$$\left(\frac{\partial B}{\partial T}\right)_p = \frac{a}{RT_i^2}$$

Program for μ_{JT}

```
REM PROG58
REM JOULE THOMSON COEFFICIENT
CLS : READ A, B: DATA 140.842,.03913: R = 8.314
SCREEN 1: COLOR 15, 0: VIEW (40, 35)-(150, 170)
WINDOW (400, -.002)-(1400, .004)
LINE (400, -.002)-(1400, .004), , B
FOR T = 400 TO 1400 STEP 10
TERM1 = 2 * A / (29.099 * R * T): TERM2 = B / 29.099
MUJT = TERM1 - TERM2
PSET (T, TERM1), 1: PSET (T, TERM2), 2: PSET (T, MUJT)
NEXT T
MU = 2 * A / (R * B): LINE (MU, -.002)-(MU, .0013), 2
LOCATE 5, 2: PRINT ".004": LOCATE 9, 8: PRINT "2a/RT"
LOCATE 11, 15: PRINT "b": LOCATE 16, 14: PRINT "MUJT"
LOCATE 21, 8: PRINT USING "###"; MU; : PRINT "K"
LOCATE 22, 1: PRINT "-.002"
LOCATE 23, 5: PRINT "400 T/K--> 1400"
FOR I = 1 TO 9
II = 400 + 100 * I: LINE (II, -.002)-(II, -.0019)
JJ = -.002 + .0006 * I: LINE (400, JJ)-(430, JJ)
NEXT I
VIEW (186, 35)-(315, 170)
WINDOW (0, -250)-(350, 600): LINE (0, -250)-(350, 600), , B
FOR P1 = 0 TO 400 STEP 2
P = P1 * 101.325
DISC = A ^ 2 - 3 * A * B ^ 2 * P
```

```
IF DISC < 0 THEN 5
DISC = SQR(DISC): BR = B * R
R1 = (A + DISC) / BR - 273: R2 = (A - DISC) / BR - 273
PSET (P1, R1), 1: PSET (P1, R2), 2
5 NEXT P1
LOCATE 3, 10: PRINT "JOULE-THOMSON EFFECT"
LOCATE 5, 21: PRINT "600"
LOCATE 6, 33: PRINT "Heating": LOCATE 7, 33: PRINT "MUJT<0"
LOCATE 13, 27: PRINT "Cooling": LOCATE 14, 27: PRINT "MUJT>0"
FOR I = 1 TO 9
II = 40 * I: LINE (II, -250)-(II, -235)
JJ = -250 + 85 * I: LINE (0, JJ)-(8, JJ)
NEXT I
A$ = "DEGREE CELSIUS"
FOR I = 1 TO LEN(A$): LOCATE 6 + I, 22: PRINT MID$(A$, I, 1):
NEXT I
LOCATE 22, 20: PRINT "-250"
LOCATE 23, 23: PRINT " 0   p/atm-->    400"
END
```

Output

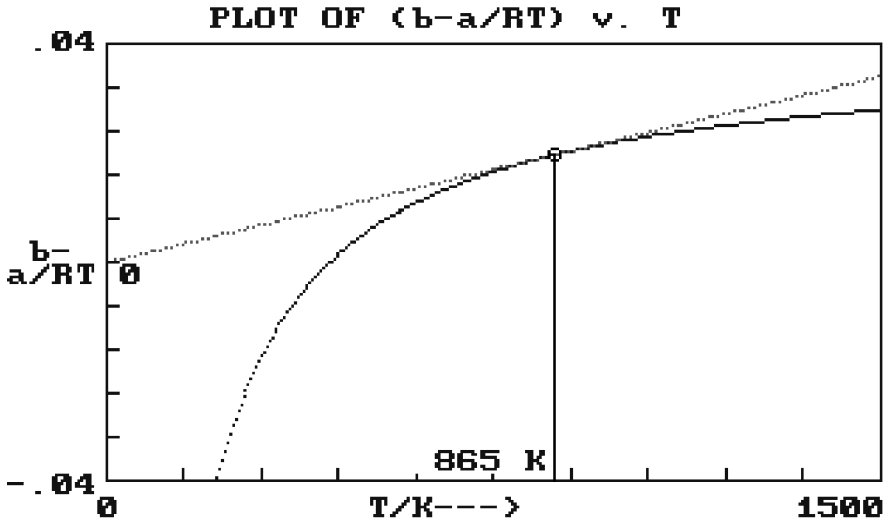


Program for Inversion Temperature

```
REM INVERSION TEMPERATURE VIA SLOPE LINE
B = .0392: A = 141: R = 8.314: TMAX = 1500: YMAX = .04
CLS : SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(310, 170)
WINDOW (0, -YMAX)-(TMAX, YMAX): LINE (0, -YMAX)-(TMAX, YMAX), , B
```

```
FOR T = 50 TO TMAX STEP 5
X = A / (R * T): B1 = B - X: PSET (T, B1)
IF X > B1 THEN TI = T: XX = X
'FOR I = 1 TO 100 STEP .01: NEXT I
NEXT T
FOR T = 5 TO TMAX STEP 10
Z = A * T / (R * TI ^ 2): PSET (T, Z), 1
FOR I = 1 TO 100 STEP .001: NEXT I
NEXT T
CIRCLE (TI, XX), 10, 2: LINE (TI, -YMAX)-(TI, XX), 2
FOR I = 1 TO 9
X1 = TMAX * .1 * I: Y1 = -YMAX + 2 * YMAX * .1 * I
LINE (X1, -YMAX)-(X1, -YMAX + YMAX * .05)
LINE (0, Y1)-(20, Y1)
NEXT I
LOCATE 2, 10: PRINT "PLOT OF (b-a/RT) v. T"
LOCATE 3, 2: PRINT USING ".##"; YMAX
LOCATE 12, 1: PRINT " b-": LOCATE 13, 1: PRINT "a/RT"
LOCATE 13, 6: PRINT "0": LOCATE 21, 19: PRINT TI; "K"
LOCATE 22, 1: PRINT USING "#.##"; -YMAX
LOCATE 23, 4: PRINT " 0              T/K--->              ";
PRINT USING "#####"; TMAX
END
```

Output



2.57 CALCULATION OF INTERNUCLEAR DISTANCE FROM ROTATIONAL CONSTANT

The rotational constant B is given by

$$B = \frac{h}{8\pi^2 Ic}$$

where h is Planck's constant (= 6.626×10^{-34} J s), c is speed of light (= 3×10^8 m s^{-1})
and I is moment of inertia of diatomic molecule (= μr^2)

The expression of reduced mass is

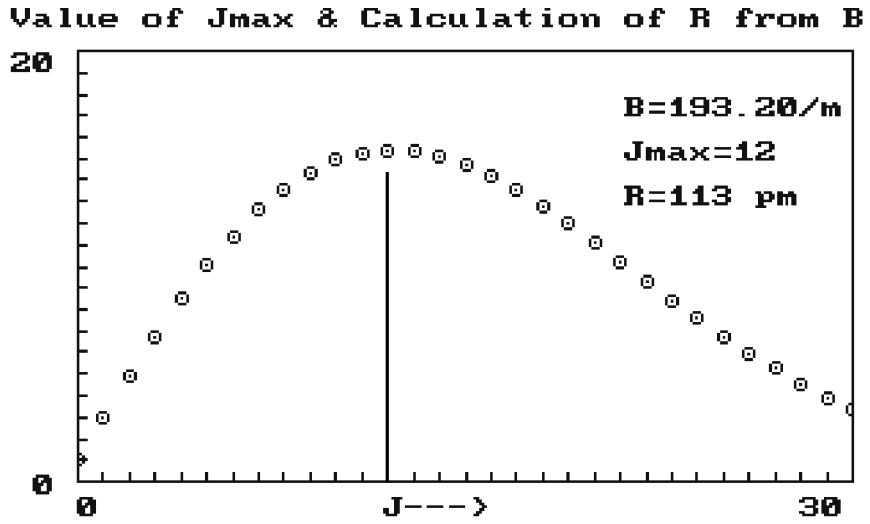
$$\mu = \frac{m_1 m_2}{m_1 + m_2} = \frac{M_1 M_2}{(M_1 + M_2) N_A}$$

Illustration Plot $g_J \exp(-E_J/kT)$ versus J at 298 K where $g_J = 2J + 1$ and $E_J = BhcJ(J + 1)$. Determine the value of J_{max} and internuclear distance for CO for which $B = 193.2$ m^{-1}.

Program

```
CLS : READ B: DATA 193.2
T = 298: NA = 6.023E+23: M1 = .012: M2 = .016
H = 6.626E-34: C = 1E+08: K = 1.38E-23: N1 = 30: N2 = 20: NM = 0
DEF FNE = B * H * C * J * (J + 1)
SCREEN 1: COLOR 15, 0: VIEW (32, 15)-(315, 170)
WINDOW (0, 0)-(N1, N2): LINE (0, 0)-(N1, N2), , B
FOR J = 0 TO N1 STEP 1
N = (2 * J + 1) * EXP(-FNE / (K * T)): PSET (J, N)
CIRCLE (J, N), N1 / 150
IF NM < N THEN NM = N: JM = J
NEXT J
FOR J = 0 TO 30 STEP 1
N = (2 * J + 1) * EXP(-FNE / (K * T)): PSET (J, N)
NEXT J
MI = 6.626 / (8 * 3.14159 ^ 2 * 3 * B)
MU = M1 * M2 / ((M1 + M2) * NA): R = SQR(MI / MU) * 1E-21
LOCATE 1, 2: PRINT "Value of Jmax & Calculation of R from B"
LOCATE 3, 1: PRINT N2: LOCATE 5, 30: PRINT "B=";
PRINT USING "###.##"; B; : PRINT "/m": LOCATE 7, 30
PRINT "Jmax="; : PRINT USING "##"; JM: LOCATE 9, 30
PRINT "R="; : PRINT USING "###"; R * 1E+12; : PRINT " pm"
FOR I = 1 TO N1:  LINE (I, 0)-(I, .4): NEXT I
FOR I = 1 TO N2: LINE (0, I)-(.3, I): NEXT I
LOCATE 22, 3: PRINT "0": LOCATE 23, 2
PRINT "   0"; SPC(13); "J--->"; SPC(13); : PRINT N1
LINE (JM, 0)-(JM, NM - 1), 2
END
```

Output



Exercise To display graphically the populations (n_J) of hydrogen bromide in the various rotational levels relative to that in the lowest rotational level. From these populations, determine the rotational level having maximum population. Also determine its interatomic distance. Given:

$$n_J = [2J + 1] \exp[-BhcJ(J + 1) / kT]$$

$$B = h / 8\pi^2 Ic \quad \text{where} \quad I = \mu r^2$$

h is Planck's constant and c is the speed of light. Take $B = 845\ m^{-1}$

2.58 PLOT OF MORSE POTENTIAL OF A DIATOMIC MOLECULE

The Morse potential is given by

$$V = D\,[1 - \exp\{a(r_{eq} - r)\}]^2$$

where D is the dissociation energy, r_{eq} is the equilibrium bond distance
and a is a constant $(= \sqrt{k_f/2D})$, with k_f as the force constant.

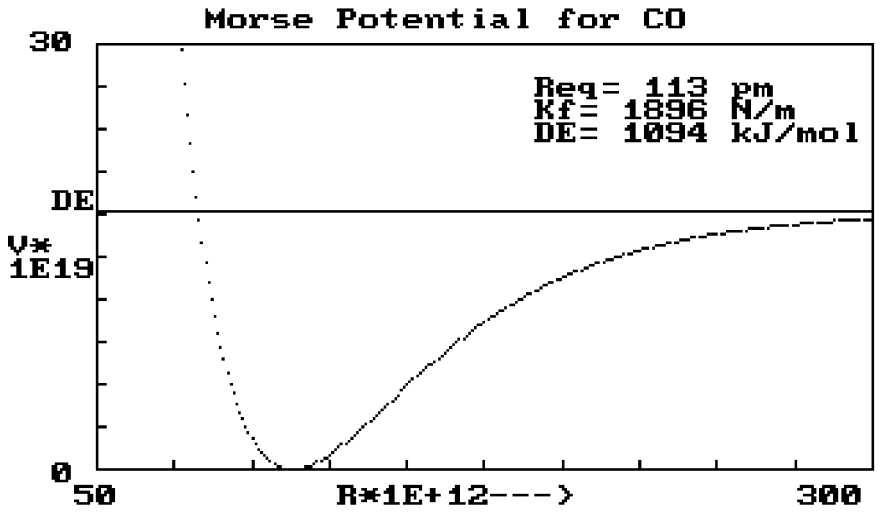
Illustration To draw Morse potential for CO. Given: $r_{eq} = 113$ pm, D = 1094 kJ mol^{-1} $k_f = 1896$ N m^{-1}.

Program

```
REM Kf=Force constantin N/m: DE=Dissociation energy in kJ/mol
REM Req=equilibrium bond distance in picometer
CLS : NA = 6.022E+23
READ N$, Kf, DE, Req: DATA CO,1896,1094,113
RMIN = 50: RMAX = 300: VMIN = 0: VMAX = 30
SCREEN 1: COLOR 15, 0: VIEW (32, 20)-(315, 170)
WINDOW (RMIN, VMIN)-(RMAX, VMAX)
LINE (RMIN, VMIN)-(RMAX, VMAX), , B
D = 1000 * DE / NA: A = SQR(Kf / (2 * D))
FOR R = 50 TO 300 STEP 1
```

```
V = D * (1 - EXP(A * (Req - R) * 1E-12)) ^ 2
V = V * 1E+19: PSET (R, V)
NEXT R
LOCATE 2, 10: PRINT "Morse Potential for "; N$
LOCATE 3, 1: PRINT VMAX
LOCATE 5, 25: PRINT "Req="; Req; "pm"
LOCATE 6, 25: PRINT "Kf="; Kf; "N/m"
LOCATE 7, 25: PRINT "DE="; DE; "kJ/mol"
LOCATE 12, 1: PRINT "V*": LOCATE 10, 3: PRINT "DE"
LOCATE 13, 1: PRINT "1E19": LOCATE 22, 2: PRINT VMIN
LOCATE 23, 3: PRINT RMIN; "          R*1E+12--->          "; RMAX
LINE (RMIN, D * 1E+19)-(RMAX, D * 1E+19)
VS = (VMAX - VMIN) * .1: RS = (RMAX - RMIN) * .1
FOR I = 1 TO 9
LINE (RMIN + RS * I, VMIN)-(RMIN + RS * I, VMIN + VS * .2)
LINE (RMIN, VMIN + VS * I)-(RMIN + RS * .1, VMIN + VS * I)
'LINE (RMIN + RS * I, VMIN)-(RMIN + RS * I, VMAX)
'LINE (RMIN, VMIN + VS * I)-(RMAX, VMIN + VS * I)
NEXT I
END
```

Output



Exercise Plot Morse potential for hydrogen bromide. Given: k_f = 411.9 N m^{-1}, r_{eq} = 142 pm and $D = 6.280 \times 10^{-19}$ J.

2.59 SOLVENT EXTRACTION

The fraction f of mass unextracted of a solute in a solution after it is treated with extracting solvent is given by

$$f = \left(1 + \frac{V_2/n}{K_d V_1}\right)^n$$

where V_1 is the volume of solution, V_2 is the volume of solvent used for extraction and

n is the number of times the extraction is carried out.

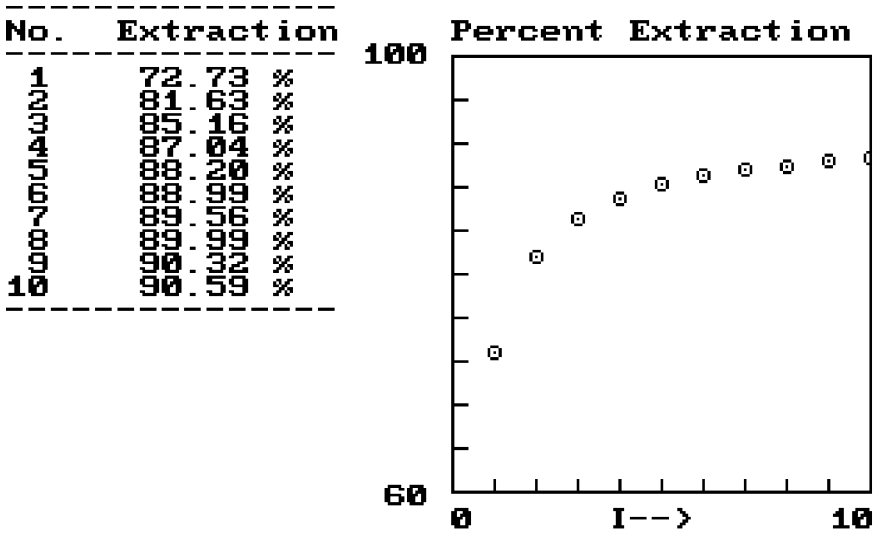
The expression of K_d is

$$K_d = \frac{\text{mass of solute in solution}}{\text{mass of solute in the extracting solvent}}$$

Illustration To display the percent extraction from the volume V_1 (= 50 mL) of a solution when it is treated with V_2 / n (= 20 mL/n) volume of extracting solvent where n is the number of times the extraction is carried out. Given: $K_d = 0.15$

Program

```
REM VI = Volume of solution: V2P = Volume of extracting solvent:
REM N = Number of extractions: KD = Distribution constant
READ V1, V2P, KD, N
DATA 50,20,.15,10
IP = 60: FP = 100
SCREEN 1: COLOR 15, 0: VIEW (160, 20)-(310, 170)
WINDOW (0, IP)-(N, FP): LINE (0, IP)-(N, FP), , B
PRINT STRING$(15, "-"): PRINT "No.  Extraction"
PRINT STRING$(15, "-")
FOR I = 1 TO N
V2 = V2P / I: UNEXT = (KD * V1 / (KD * V1 + V2)) ^ I
PERCENT = (1 - UNEXT) * 100
PRINT USING "##"; I;
PRINT USING "    ###.##"; PERCENT; : PRINT " %": 'PRINT
PSET (I, PERCENT), 2: CIRCLE (I, PERCENT), .1
NEXT I
PRINT STRING$(15, "-")
FOR I = 1 TO N: LINE (I, IP)-(I, IP + 1): NEXT I
FOR I = IP TO FP STEP 4: LINE (0, I)-(.3, I): NEXT I
LOCATE 2, 21: PRINT "Percent Extraction"
LOCATE 3, 16: PRINT FP: LOCATE 22, 17: PRINT IP
LOCATE 23, 21: PRINT "0        I-->       "; N
END
```



Exercise Display that the multistage solvent extraction is more profitable as compared to a single stage extraction. Also display graphically the percent of solute extracted in each stage of extraction.

Use the expression:

$$f = \left(1 + \frac{V_2/n}{k_d V_1}\right)^n$$

where f is the fraction of mass unextracted,

V_1 is the volume of solution,

V_2 is the volume of solvent used for extraction

n is the number of extractions

$$K_d = \frac{\text{mass of solute in solution}}{\text{mass of solute in the extracting solvent}}$$

Use the following data:

V_1 = 100 mL ; V_2 = 20 mL ; K_d = 0.20 ; n = 1 to 10

2.60 PARA/ORTHO VARIATIONS OF H_2 AT DIFFERENT TEMPERATURES

The ratio para : ortho of H_2 is given by

$$\frac{N_{\text{para}}}{N_{\text{ortho}}} = \frac{\sum_{\text{even } J} (2J+1) \exp\left(-\theta_r J(J+1)/T\right)}{(3) \sum_{\text{odd } J} (2J+1) \exp\left(-\theta_r J(J+1)/T\right)}$$

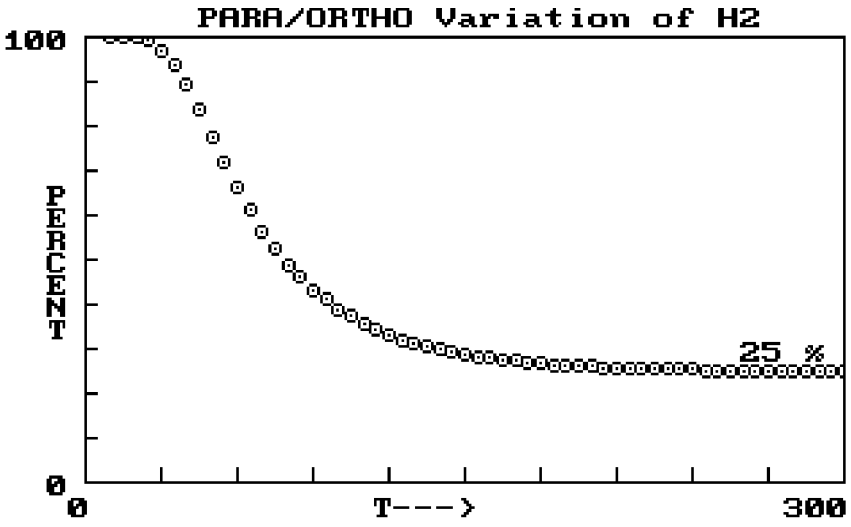
For H_2, θ_r = 86 K.

Illustration Plot the variation of para/ortho ratio of H_2 from 10 K to 300 K in the step of 10 K.

Program

```
CLS
THETA = 86
SCREEN 1: COLOR 15, 0: VIEW (30, 10)-(310, 170)
WINDOW (0, 0)-(300, 100): LINE (0, 0)-(300, 100), , B
FOR T = 10 TO 300 STEP 5
THETAT = THETA / T: SUM1 = 0
FOR J = 0 TO 6 STEP 2
TERM = (2 * J + 1) * EXP(-THETAT * J * (J + 1))
SUM1 = SUM1 + TERM
NEXT J
SUM2 = 0
FOR J = 1 TO 7 STEP 2
TERM = (2 * J + 1) * EXP(-THETAT * J * (J + 1))
SUM2 = SUM2 + 3 * TERM
NEXT J
RATIO = SUM1 / SUM2
PERCENT = RATIO / (RATIO + 1) * 100
PSET (T, PERCENT): CIRCLE (T, PERCENT), 2
NEXT T
FOR I = 1 TO 9: LINE (I * 30, 0)-(I * 30, 3)
LINE (0, I * 10)-(4, I * 10): NEXT I
LOCATE 1, 10: PRINT "PARA/ORTHO Variation of H2 "
LOCATE 2, 1: PRINT "100": A$ = "PERCENT"
FOR I = 1 TO 7: LOCATE 8 + I, 3: PRINT MID$(A$, I, 1): NEXT I
LOCATE 22, 3: PRINT "0": LOCATE 16, 35: PRINT "25 %"
LOCATE 23, 2: PRINT "  0               T--->                300"
END
```

Output



2.61 POTENTIOMETRIC TITRATIONS

The principle underlying the potentiometric titrations is the Nernst equation according to which the potential of an electrode depends upon the concentrations of species involved in the half-cell reaction. During the titration, the concentrations of these species vary, which in turn, varies the potential of the electrode.

The computation of half-cell potential is illustrated with the following example.

Illustration

Compute the variation in potential of the electrode during the titration of 50 mL of 0.1 M Fe^{2+} ions with 0.1 M Ce^{4+} ions. Given: $E°(Fe^{3+}, Fe^{2+}|\,Pt) = 0.7$ V and $E°(Ce^{4+}, Ce^{3+}|\,Pt) = 1.459$ V.

Expressions to be used

Before the equivalence point

$$E = E^{\circ}_{Fe^{3+},Fe^{2+}|Pt} - \left(\frac{RT}{F}\right)\ln\left(\frac{[Fe^{2+}]}{[Fe^{3+}]}\right)$$

$$= E^{\circ}_{Fe^{3+},Fe^{2+}|Pt} - \left(\frac{RT}{F}\right)\ln\left(\frac{1-f}{f}\right)$$

where f is the fraction of Fe^{2+} ions converted to Fe^{3+} ions.

At the equivalence point

$$E = \frac{1}{2}\left(E^{\circ}_{Fe^{3+},Fe^{2+}|Pt} + E^{\circ}_{Ce^{4+},Ce^{3+}|Pt}\right)$$

After the equivalance point

$$E = E^{\circ}_{Ce^{4+},Ce^{3+}|Pt} - \left(\frac{RT}{F}\right)\ln\left(\frac{[Ce^{3+}]}{[Ce^{4+}]}\right)$$

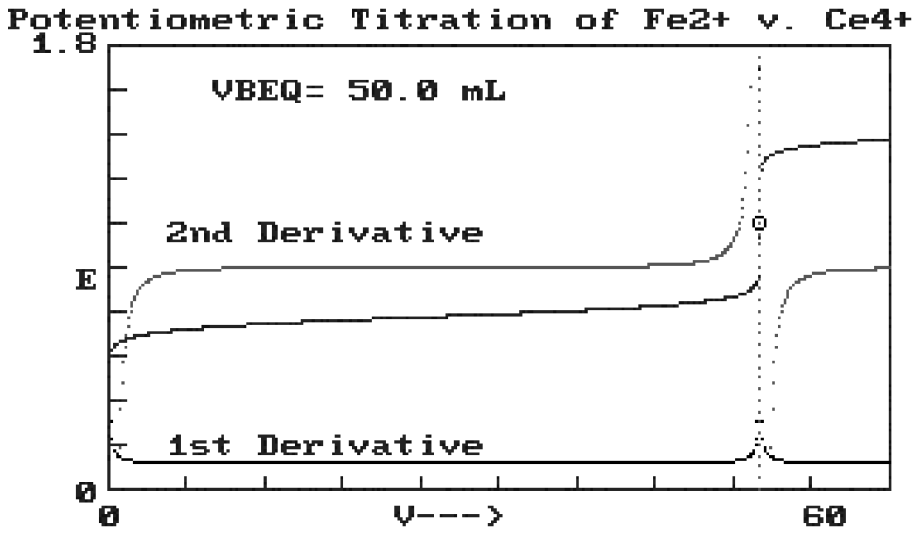
$$= E^{\circ}_{Ce^{4+},Ce^{3+}|Pt} - \left(\frac{RT}{F}\right)\ln\left(\frac{M_{Fe^{2+}}\,V_{Fe^{2+}}}{(V-V_{eq})M_{Ce^{4+}}}\right);\ ([Ce^{3+}]_{eq} = [Fe^{2+}]_0)$$

where V is the total volume of Ce^{4+} ions added.

Program

```
CLS
REM MF=Molarity OF Fe2+: MC=Molarity of Ce4+
REM VF=Volume of Fe2+: PF=Final Potential
REM EF=E(Fe2+/Fe): ECE=E(Ce4+/Ce)
READ MF, MC, VF: DATA .1,.1,50
VBEQ = MF * VF / MC: V2 = VBEQ + 10: PF = 1.8
CONS = 8.314 * 298 / 96500: EF = .7: ECE = 1.459
SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(310, 170)
LOCATE 1, 5: PRINT "Potentiometric Titration of Fe2+ v. Ce4+"
LOCATE 3, 1: PRINT PF
FOR V = .1 TO V2 STEP .1
WINDOW (0, 0)-(V2, PF): LINE (0, 0)-(V2, PF), , B
PSET (VBEQ, .5 * (EF + ECE)), 2
```

```
CIRCLE (VBEQ, .5 * (EF + ECE)), V2 / 150, 2
FOR I = 1 TO 9: II = V2 * .1 * I: JJ = I * PF * .1: IP = V2 / 50
LINE (II, 0)-(II, .05): LINE (0, JJ)-(IP, JJ)
'LINE (II, 0)-(II, PF): LINE (0, JJ)-(V2, JJ)
NEXT I
IF V < VBEQ THEN
F = V / VBEQ: E = EF - CONS * LOG((1 - F) / F)
GOTO 50
ELSEIF V > VBEQ THEN
E = ECE - CONS * LOG(MF * VF / ((V - VBEQ) * MC))
END IF
50 PSET (V, E)
FD = (E - EP) / (V - VP): PSET ((V + VP) / 2, FD + .1), 2
SD = (FD - FDP) / (V - VP)
WINDOW (0, -1)-(V2, 1): PSET ((V + VP) / 2, SD * 20), 1
EP = E: VP = V: FDP = FD
NEXT V
FOR I = -1 TO 1 STEP .05: PSET (VBEQ, I), 1: NEXT I
LOCATE 5, 10: PRINT "VBEQ="; : PRINT USING "###.#"; VBEQ;
PRINT " mL": LOCATE 11, 8: PRINT "2nd Derivative"
LOCATE 13, 4: PRINT "E": LOCATE 20, 8
PRINT "1st Derivative": LOCATE 22, 4: PRINT "0"
LOCATE 23, 3: PRINT "   0                V--->                "; V2
END
```



Comment

The Fe^{2+} versus Ce^{4+} titration can be carried out in the presence of *N*-phenylanthranilic acid (E° = 1.08 V) indicator and not in the presence of diphenylamine (E° = 0.76 V) unless phosphoric acid is present in the titrating solution (this has an effect of reducing $E^\circ(Fe^{3+}, Fe^{2+}|\,Pt)$) so that the steep rise in the potential at the equivalance point encompasses E° of diphenylamine indicator.

Exercise Draw potentiometric titration plot of 50 mL (= V) of 0.1 M Cl^- ions against 0.1 M Ag^+ ions. Also draw the first and second derivatives of E versus $V(Ag^+)$ plot.

Use the following expressions.

Before the equivalence point

$$E_{Cl^-|AgCl|Ag} = E^\circ_{Cl^-|AgCl|Ag} - \frac{RT}{F} \ln \left\{ \frac{(0.1)V}{V + \upsilon} (1 - f) \right\}$$

where f is the fraction of Cl^- ions precipitated and υ is the volume of Ag^+ ions added.

$$E^\circ_{Cl^-|AgCl|Ag} = 0.22 \text{ V}.$$

At the equivalence point

$$E_{Cl^-|AgCl|Ag} = E^\circ_{Cl^-|AgCl|Ag} - \frac{RT}{F} \ln ([Cl^-]/\text{M})$$

where $[Cl^-] = \sqrt{K_{sp}(AgCl)}$ with $K_{sp} = 1.24 \times 10^{-5}$ M

After the equivalence point

$$E_{Ag^+|Ag} = E^\circ_{Ag^+|Ag} - \ln \frac{RT}{F} \ln \left(\frac{1}{[Ag^+]/\text{M}} \right)$$

where $E^\circ_{Ag^+|Ag} = 0.799$ V.

2.62 VAPOUR PRESSURE OF WATER VIA ENTROPY AND FREE ENERGY CONSIDERATIONS

For the reaction

$$H_2O(l) \rightarrow H_2O(g)$$

we have

$$\Delta_r H^\circ_{298.15K} = \Delta_f H^\circ(H_2O, g) - \Delta_f H^\circ(H_2O, l)$$

$$\Delta_r S^\circ_{298.15K} = S^\circ(H_2O, g) - S^\circ(H_2O, l)$$

$$\Delta_r C_p = C_p(H_2O, g) - C_p(H_2O, l)$$

Assuming $\Delta_r C_p$ to be independent of temperature, we will have

$$\Delta_r H^\circ_T = \Delta_r H^\circ_{298.15K} + \Delta_r C_p \, (T - 298.15 \text{ K})$$

$$\Delta_r S^\circ_T = \Delta_r S^\circ_{298.15K} + \Delta_r C_p \ln \left(\frac{T}{298.15 \text{ K}} \right)$$

$$\Delta_r G_T^\circ = \Delta_r H_T^\circ - T\,\Delta_r S_T^\circ$$

For the reaction to be at equilibrium at 1 atm external pressure, we will have

$$\Delta_r G_T^\circ = 0 \quad \Rightarrow \quad (\Delta_r S_T^\circ)_{eq} = \frac{\Delta_r H_T^\circ}{T}$$

For the reaction to be at equilibrium, we must change $\Delta_r S_T^\circ$ to $(\Delta_r S_T^\circ)_{eq}$. If this change is carried out only by changing the vapour pressure of water, we will have

$$(\Delta_r S_T^\circ)_{eq} - \Delta_r S_T^\circ = -R \ln \left(\frac{p_{eq}}{1 \text{ bar}}\right) \tag{1}$$

where $(\Delta_r S_T^\circ)_{eq}$ and $\Delta_r S_T^\circ$ are the entropy changes at pressure p_{eq} (vapour pressure of water) and 1 bar, respectively. This equation may be used to calculate the equilibrium vapour pressure of water.

From the free energy consideration, we must have

$$\Delta_r G^\circ = -RT \ln K_p^\circ \tag{2}$$

where $K_p^\circ = p_{H_2O(g)}/p^\circ$.

Equations (1) and (2) are identical as shown in the following.

$$(\Delta_r S_T^\circ)_{eq} - \Delta_r S_T^\circ = -R \ln \left(\frac{p_{eq}}{1 \text{ bar}}\right)$$

or
$$\frac{\Delta_r H_T^\circ}{T} - \Delta_r S_T^\circ = -R \ln \left(\frac{p_{eq}}{1 \text{ bar}}\right)$$

or
$$\Delta_r H_T^\circ - T\,\Delta_r S_T^\circ = -RT \ln \left(\frac{p_{eq}}{1 \text{ bar}}\right)$$

or
$$\Delta_r G_T^\circ = -RT \ln K_p^\circ$$

Illustration Calculate $\Delta_r H_T^\circ$, $\Delta_r S_T^\circ$, $\Delta_r G_T^\circ$, $(\Delta_r S_T^\circ)_{eq}$, p_{eq} and K_p° for the reaction $H_2O(l) \rightarrow H_2O(g)$ at different temperatures. Also plot p_{eq} versus T.

Given:

$\Delta_f H^\circ(H_2O, l) = -285.83 \text{ kJ mol}^{-1}$

$\Delta_f H^\circ(H_2O, g) = -241.82 \text{ kJ mol}^{-1}$

$S^\circ (H_2O, l, 298.15 \text{ K}) = 69.91 \text{ J K}^{-1} \text{ mol}^{-1}$

$S^\circ (H_2O, g, 298.15 \text{ K}) = 188.83 \text{ J K}^{-1} \text{ mol}^{-1}$

$C_p(H_2O, l) = 75.29 \text{ J K}^{-1} \text{ mol}^{-1}$

$C_p(H_2O, g) = 33.58 \text{ J K}^{-1} \text{ mol}^{-1}$

Program

```
CLS : DIM T(20), PEQ(20)
N = 15: R = 8.314: HL = -285830: HG = -241820
TI = 293.15: TS = 10: SL = 69.91
SG = 188.83: CPL = 75.29: CPG = 33.58
```

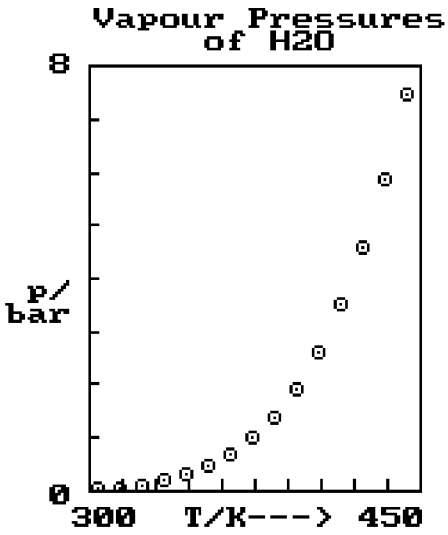
```
DH = HG - HL: DS = SG - SL: DCP = CPG - CPL
'PRINT "Vapour Pressures of H2O at Different Temperatures"
'PRINT STRING$(80, "_")
'PRINT "   T          DHT           DST         ";
'PRINT "TDS          DGT          DSTEQ        PEQ        Kp"
'PRINT " ---        -----       ---------    ";
'PRINT "-----        -----       ---------    ----       ----"
'PRINT "   K         J/mol        J/(K mol)    ";
'PRINT "J/mol        J/mol       J/(K mol)    bar        bar"
'PRINT STRING$(80, "_")
FOR I = 1 TO N
TI = TI + TS: T(I) = TI
DHT = DH + DCP * (T(I) - 298.15)
DST = DS + DCP * LOG(T(I) / 298.15)
DGT = DHT - T(I) * DST: DSTEQ = DHT / T(I)
PEQ(I) = EXP((DST - DSTEQ) / R)
KP = EXP(-DGT / (R * T(I))): PRINT USING "###.##"; T(I);
'PRINT USING "     #####.#"; DHT; DST; T(I) * DST; DGT; DSTEQ;
'PRINT USING "     ##.##"; PEQ(I); KP
NEXT I
'PRINT STRING$(80, "_")
'A$ = INPUT$(1)
SCREEN 1: COLOR 15, 0: VIEW (190, 20)-(310, 170)
WINDOW (300, 0)-(450, 8): LINE (300, 0)-(450, 8), , B
LOCATE 1, 25: PRINT "Vapour Pressures"
LOCATE 2, 30: PRINT "of H2O"
LOCATE 3, 5: PRINT STRING$(15, "-")
LOCATE 4, 6: PRINT "T/K      p/bar"
LOCATE 5, 5: PRINT STRING$(15, "-")
FOR I = 1 TO N
PSET (T(I), PEQ(I)): CIRCLE (T(I), PEQ(I)), 2
PRINT USING "    ####.##"; T(I);
PRINT USING " ###.###"; PEQ(I): A$ = INPUT$(1)
NEXT I
FOR I = 1 TO 9
LINE (300 + I * 15, 0)-(300 + I * 15, .2)
LINE (300, I)-(304, I)
NEXT I
LOCATE 3, 23: PRINT "8": LOCATE 13, 22: PRINT "p/"
LOCATE 14, 21: PRINT "bar"
```

```
LOCATE 21, 5: PRINT STRING$(15, "-")
LOCATE 22, 23: PRINT "0"
LOCATE 23, 1: PRINT STRING$(22, " "); " 300   T/K---> 450"
END
```

Output

Only *p*/bar at different temperatures along with its plot are displayed.

T/K	p/bar
303.15	0.042
313.15	0.074
323.15	0.123
333.15	0.199
343.15	0.310
353.15	0.471
363.15	0.695
373.15	1.000
383.15	1.408
393.15	1.942
403.15	2.628
413.15	3.494
423.15	4.569
433.15	5.886
443.15	7.476



2.63 SOLUTION OF DIFFERENTIAL EQUATION OF THE FIRST-ORDER

The differential equation of the first-order is of the form

$$y' = f(x, y)$$

with the intial condition $y_0 = y(x_0)$ to be satisfied by the solution of the differential equation. We consider here a few methods which are available for the solution of first-order differential equation.

In actual computations, the solutions are compared with the exact value of the function $y(x_i)$.

Euler and Improved Euler Methods

Consider the following differential equation

$$y' = f(x, y) \tag{1}$$

with the intial condition $y(x_0) = y_0$

If $x_1 = x_0 + h$, then by the Taylor series we write

$$y(x + h) = y(x) + h\, y'(x) + \frac{h^2}{2}\, y''(x) + \cdots$$

For small value of h, the higher powers h^2, h^3, …, may be ignored to give

$$y(x + h) = y(x) + h\, y'$$
$$= y(x) + h\, f \qquad (2)$$

The following iteration process may be adopted to determine the value of y at $x = x_0 + (n + 1)\, h$.

$$x_1 = x_0 + h; \qquad y_1 = y_0 + h\, f(x_0, y_0)$$
$$x_2 = x_1 + h = x_0 + 2h; \qquad y_2 = y_1 + h\, f(x_1, y_1)$$
$$\vdots$$
$$x_{n+1} = x_0 + (n + 1)\, h; \qquad y_{n+1} = y_n + h\, f(x_n, y_n)$$

The above method is know as Euler method or Euler-Cauchy method.

In the improved Euler method (also know as Heun's method), each step involves firstly the evaluation of the auxiliary value

$$y^*_{n+1} = y_n + h\, f\,(x_n, y_n)$$

and then the new improved value by the expression

$$y_{n+1} = y_n + \frac{1}{2}\, h\, [f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})]$$

The above method is a **predictor – collector method**, because in each step we predict the value y^*_{n+1} by the first equation and then correct it by the second equation.

Illustration Determine the solution of differential equation $y' = -k\, y$ for the first-order kinetics with $y(t = 0) = [\mathrm{A}]_0 = 0.25$ mol dm^{-3} at t/s = 200, 400, 600, 800, 1000 and 1200. Given: $k = 0.000\,622$ s^{-1}.

Program

```
REM Improved Euler Method(Heun's Method)
REM Function FNA is dy/dx & FNB is y
CLS : NMAX = 6
'X = 0: Y = 0: H = .2
'DEF FNA (X, Y) = X + Y: DEF FNB (X, Y) = EXP(X) - X - 1
REM DATA FOR FIRST ORDER KINETICS
X = 0: Y = .25: Y0 = Y: H = 200: K = .000622
'K = .01119: X = 0: Y = -34.8: Y0 = Y: H = 2
'K = .019368: X = 0: Y = 169.3: Y0 = Y: H = 1
DEF FNA (X, Y) = -K * Y: DEF FNB (X, Y) = Y0 * EXP(-K * X)
REM DATA FOR SECOND ORDER KINETICS
'K = .0000178: X = 0: Y = 632: Y0 = Y: H = 6
'DEF FNA (X, Y) = -K * Y ^ 2
'DEF FNB (X, Y) = Y0 / (1 + Y0 * X * K)
PRINT STRING$(50, "-")
```

```
PRINT " N        X              Y            YEXACT         ERROR"
PRINT STRING$(50, "-")
FOR N = 0 TO NMAX
Z = Y
TERM = FNA(X, Z): YST = Z + H * TERM
Y1 = Y + (H / 2) * (TERM + FNA(X + H, YST))
YE = FNB(X, Z): ER = YE - Y
PRINT USING "##"; N; : PRINT USING "    ####.##"; X;
PRINT USING "    ####.####"; Y; YE; ER
Y = Y1: X = X + H
NEXT N
PRINT STRING$(50, "-")
END
```

Output

N	X	Y	YEXACT	ERROR
0	0.00	0.2500	0.2500	0.0000
1	200.00	0.2208	0.2208	-0.0001
2	400.00	0.1951	0.1949	-0.0001
3	600.00	0.1723	0.1721	-0.0002
4	800.00	0.1522	0.1520	-0.0002
5	1000.00	0.1345	0.1342	-0.0002
6	1200.00	0.1188	0.1185	-0.0003

Exercise

Determine the solutions of differential equation

$$y' = -ky^2$$

for the second-order kinetics with y (t = 0) = 632 mol dm^{-3} at t/min = 3, 6, 9, 12, 15 and 18. Given: k = 0.000 017 8 mol^{-1} dm^3 min^{-1}

Runge – Kutta Method

Consider the differential equation

$$y' = f(x, y) \tag{1}$$

with the initial condition $y(x_0) = y_0$. Integrating Eq.(1), we get

$$y_1 = y_0 + \int_{x_0}^{x_1} f(x, y)\, dx$$

If the above integral is approximated by the trapezoidal rule, then

$$y_1 = y_0 + \frac{h}{2}\,[f(x_0, y_0) + f(x_1, y_1)]$$

where $x_1 = x_0 + h$. If $y_1 = y_0 + h\,f\,(x_0, y_0)$, then

$$y_1 = y_0 + \frac{h}{2}\,[f_0 + f\,(x_0 + h, y_0 + h\,f_0)]$$

where $f_0 = f(x_0, y_0)$. If the substitution

$$k_1 = h\,f_0 \quad \text{and} \quad k_2 = h\,f\,(x_0 + h, y_0 + k_1)$$

are made in the above expression, we get

$$y_1 = y_0 + \frac{1}{2}\,(k_1 + k_2) \tag{2}$$

Equation (2) is known as the second-order Runge-Kutta formula.

The most commonly used is the following fourth-order Runge-Kutta formula, for which we have

$$y_1 = y_0 + \frac{1}{6}\,(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = h\,f\,(x_0, y_0)$$

$$k_2 = h\,f\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1\right)$$

$$k_3 = h\,f\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2\right)$$

$$k_4 = h\,f\,(x_0 + h, y_0 + k_3)$$

Illustration Determine the solution of differential equation $y' = -k\,y$ for the first-order kinetics with $y(t = 0) = [\mathrm{A}]_0 = 0.25$ mol dm^{-3} at t/s = 200, 400, 600, 800, 1000 and 1200. Given: $k = 0.000\,622$ s^{-1}.

Program

```
REM Function FNA is dy/dx & FNB is y
NMAX = 6: PRINT STRING$(50, "-")
PRINT " N         X             Y             YE             ER"
PRINT STRING$(50, "-")
REM Consecutive reaction with [A]=1; Determination of [B]
'L1 = .0125: L2 = .00417: X = 0: Y = 0: H = 40
'DEF FNA (X, Y) = L1 * EXP(-L1 * X) - L2 * Y
'DEF FNB (X, Y) = L1 * (EXP(-L1 * X) / (L2 - L1) + EXP(-L2 * X)
/ (L1 - L2))
REM FIRST-ORDER KINETICS
X = 0: Y = .25: Y0 = Y: H = 200: K = .000622
'X = 0: Y = 169.3: Y0 = Y: H = 3: K = .019368
DEF FNA (X, Y) = -K * Y
DEF FNB (X, Y) = Y0 * EXP(-K * X)
```

```
REM SECOND-ORDER KINETICS
'X = 0: Y = 632: H = 6: K = .0000178: Y0 = Y
'DEF FNA (X, Y) = -K * Y ^ 2
'DEF FNB (X, Y) = Y0 / (1 + K * X * Y0)
FOR N = 0 TO NMAX
Z = Y
K1 = H * FNA(X, Z)
K2 = H * FNA(X + .5 * H, Z + .5 * K1)
K3 = H * FNA(X + .5 * H, Z + .5 * K2)
K4 = H * FNA(X + H, Z + K3)
Y1 = Y + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
YE = FNB(X, Z): ER = (YE - Y)
PRINT USING "##"; N; : PRINT USING "    ####.##"; X;
PRINT USING "    ####.####"; Y; YE; ER
X = X + H: Y = Y1
NEXT N
PRINT STRING$(50, "-")
END
```

Output

N	X	Y	YE	ER
0	0.00	0.2500	0.2500	0.0000
1	200.00	0.2208	0.2208	-0.0000
2	400.00	0.1949	0.1949	-0.0000
3	600.00	0.1721	0.1721	-0.0000
4	800.00	0.1520	0.1520	-0.0000
5	1000.00	0.1342	0.1342	-0.0000
6	1200.00	0.1185	0.1185	-0.0000

Solving Differential Equations Involved in Consecutive Reactions

To display the concentrations of A, B and C in a consecutive reactions $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ by solving the corresponding differential equation as given in the following.

(i) $$-\frac{d[A]}{dt} = k_1 [A]$$

Differential equation: $y' = -k_1 y$

(ii) $$\frac{d[B]}{dt} = k_1 [A] - k_2 [B]$$
$$= k_1 [A]_0 \exp(- k_1 t) - k_2 [B]$$

Differential equation: $y' = k_1 [A]_0 \exp(- k_1 x) - k_2 y$

(iii) $$\frac{d[C]}{dt} = k_2 [B] = k_2 \{[A]_0 - [A] - [C]\}$$
$$= k_2 [A]_0 \{1 - \exp(- k_1 t)\} - k_2 [C]$$

Differential equation: $y' = k_2 [A]_0 \{1 - \exp(- k_1 x)\} - k_2 y$

where $x \equiv t$ and $y \equiv$ concentration of either A or B or C.

Note: It is sufficient to solve two of the above three equations as $[A]_0 = [A] + [B] + [C]$.

Illustration To display or calculate [A], [B] and [C] during the progress of reaction

$$A \xrightarrow{k_1 = 0.0125 s^{-1}} B \xrightarrow{k_2 = .00417 s^{-1}} C$$

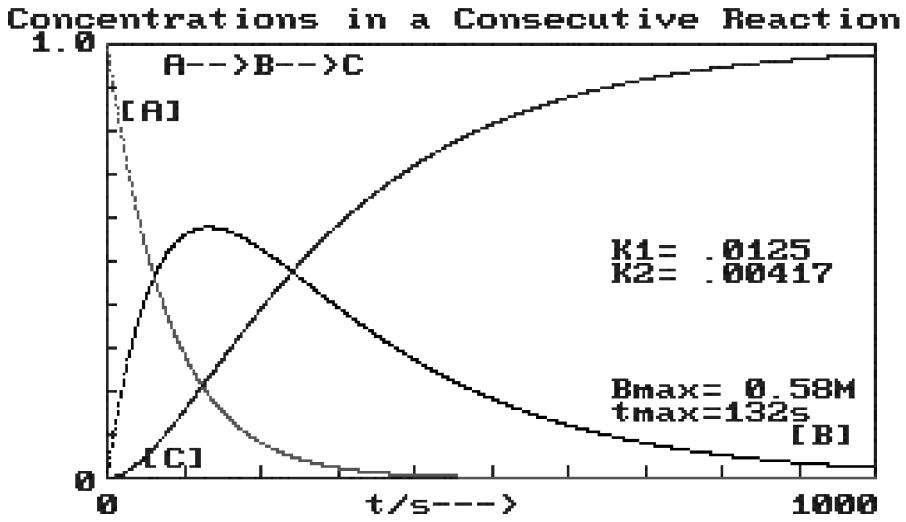
Also find the $[B]_{max}$ and the corresponding time by finding the maximum in [B] versus t plot.

Program

```
CLS
REM CONSECUTIVE REACTIONS  A--->B--->C
READ PLOT$, A0, TF: DATA Y,1,1000
READ L1, L2:  DATA .0125, .00417
LL = L2 - L1: YM = 0
REM Functions FNB, FND and FNF give analytical solutions of [A], [B] and [C]
DEF FNA (X, Y) = -L1 * Y: DEF FNB (X, Y) = A0 * EXP(-L1 * X)
DEF FNC (X, Y) = L1 * FNB(X, Y) - L2 * Y
DEF FND (X, Y) = A0 * (L1 / LL) * (EXP(-L1 * X) - EXP(-L2 * X))
DEF FNE (X, Y) = L2 * A0 * (1 - EXP(-L1 * X)) - L2 * Y
DEF FNF (X, Y) = A0 * (1 - (L2 * EXP(-L1 * X) - L1 * EXP(-L2 * X)) / LL)
IF PLOT$ = "Y" THEN
NMAX = TF: H = 1
SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(310, 170)
WINDOW (0, 0)-(TF, A0): LINE (0, 0)-(TF, A0), , B
LOCATE 2, 1: PRINT "Concentrations in a Consecutive Reaction"
LOCATE 3, 1: PRINT USING "##.#"; A0: LOCATE 4, 8: PRINT "A-->B-->C"
LOCATE 6, 6: PRINT "[A]": LOCATE 12, 28: PRINT "K1="; L1
LOCATE 13, 28: PRINT "K2="; L2: LOCATE 20, 36: PRINT "[B]"
LOCATE 21, 7: PRINT "[C]": LOCATE 22, 4: PRINT "0"
LOCATE 23, 5: PRINT "0           t/s--->           "; TF
FOR I = 1 TO 9
LINE (I * TF / 10, 0)-(I * TF / 10, A0 * .03)
LINE (0, A0 * I / 10)-(TF / 100, A0 * I / 10)
NEXT I
ELSE
```

```
NMAX = 18: H = TF / 20
PRINT STRING$(78, "-")
PRINT " N      t/s        [A]        [B]       [C]";
PRINT "         AE         BE         CE"
PRINT STRING$(78, "-")
END IF
X = 0: YA = A0: YB = 0: YC = 0
FOR N = 0 TO NMAX
Z = YA
K1 = H * FNA(X, Z)
K2 = H * FNA(X + .5 * H, Z + .5 * K1)
K3 = H * FNA(X + .5 * H, Z + .5 * K2)
K4 = H * FNA(X + H, Z + K3)
Y1 = YA + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
Z = YB
K1 = H * FNC(X, Z)
K2 = H * FNC(X + .5 * H, Z + .5 * K1)
K3 = H * FNC(X + .5 * H, Z + .5 * K2)
K4 = H * FNC(X + H, Z + K3)
Y2 = YB + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
Z = YC
K1 = H * FNE(X, Z)
K2 = H * FNE(X + .5 * H, Z + .5 * K1)
K3 = H * FNE(X + .5 * H, Z + .5 * K2)
K4 = H * FNE(X + H, Z + K3)
Y3 = YC + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
IF PLOT$ = "Y" THEN
FOR I = 1 TO 100 STEP .005: NEXT I
PSET (X, YA), 1: PSET (X, YB), 2: PSET (X, YC), 3
ELSE
PRINT USING "##"; N; : PRINT USING "   ####.##"; X;
PRINT USING "  ####.####"; YA; YB; YC; FNB(X, Y); FND(X, Y); FNF(X, Y)
END IF
X = X + H: YA = Y1: YB = Y2: YC = Y3
IF YB >= YM THEN YM = YB: BMAX = YM: TMAX = N
NEXT N
IF PLOT$ = "Y" THEN
LOCATE 18, 28: PRINT "Bmax=";
PRINT USING "##.##"; BMAX; : PRINT "M"
LOCATE 19, 28: PRINT "tmax=";
PRINT USING "###"; H * (TMAX + 1); : PRINT "s"
END IF
IF PLOT$ <> "Y" THEN PRINT STRING$(78, "-")
END
```

Output



Exercises

1. For the following parallel first- and second-order reactions

$$A \xrightarrow{k_1} D + E$$

$$A + B \xrightarrow{k_2} C + D$$

the differential rate law is

$$\frac{dx}{dt} = k_1 ([A]_0 - x) + k_2 ([A]_0 - x) ([B]_0 - x)$$

Solve the above differential equation for

$$[A]_0 = 1.0 \text{ M}, [B]_0 = 1.0 \text{ M}, k_1 = 0.005 \text{ s}^{-1}, k_2 = 0.005 \text{ mol}^{-1} \text{ dm}^3 \text{ s}^{-1}$$

Vary t from 0 to 1000 s in the step of 50 s. Compare the results with the exact expression

$$x = \frac{[A]_0 ([B]_0 + k_1/k_2)(Y - 1)}{([B]_0 + k_1/k_2)Y - [A]_0}$$

where $Y = \exp(k_2 t ([B]_0 - [A]_0 + k_1/k_2))$. Note that $x = 0$ at $t = 0$.

Hint Set up the functions as follows.

```
DEF    FNA (X, Y) = L1 * (A0 - Y) + L2 * (A0 - Y) * (B0 - Y)
DEF    FNB (X, Y)
       LP = L1 / L2
       YY = EXP(L2 * X * (B0 - A0 + LP))
       FNB = (A0 * (B0 + LP) * (YY - 1)) / ((B0 + LP) * YY - A0)
       END DEF
```

2. For the following parallel first- and second-order reactions

$$A \xrightarrow{k_1} \text{product}$$

$$A + A \xrightarrow{k_2} \text{product}$$

the differential rate law is

$$\frac{dx}{dt} = k_1 ([A]_0 - x) + 2k_2 ([A]_0 - x)^2$$

Solve the above differential equation for

$$[A]_0 = 0.5 \text{ M}, [B]_0 = 0.5 \text{ M}, k_1 = 0.002 \text{ s}^{-1}, k_2 = 0.004 \text{ mol}^{-1} \text{ dm}^3 \text{ s}^{-1}$$

Vary t from 0 to 1000 s in the step of 50 s. Compare the results with the exact expression

$$x = \frac{[A]_0 (k_1 + 2k_2[A]_0) e^{k_1 t} - k_1[A]_0 - 2k_2[A]_0^2}{(k_1 + 2k_2[A]_0) e^{k_1 t} - 2k_2[A]_0}$$

Note that $x = 0$ at $t = 0$.

Hint Set up the functions as follows

```
DEF    FNA (X, Y) = L1 * (A0 - Y) + 2 * L2 * (A0 - Y) ^ 2
DEF    FNB (X, Y)
       LP = (L1 + 2 * L2 * A0) * EXP(L1 * X)
       FNB = (A0 * LP - A0 * L1 - 2 * L2 * A0 ^ 2) / (LP - 2 * A0 * L2)
       END DEF
```

3. Determine the solutions of differential equation

$$y' = -ky^2$$

for the second-order kinetics with $y(t = 0) = 632$ mol dm^{-3} at t/min = 2, 4, 6, 8, 10 and 12. Given: $k = 0.000\ 017\ 8$ mol^{-1} dm^3 min^{-1}

Adams-Moulton Method

Consider the differential equation

$$y' = f(x, y) \tag{1}$$

with the intial condition $y(x_0) = y_0$. Integrating Eq. (1) from x_n to x_{n+1} (= $x_n + h$), we have

$$y_{n+1} = \int_{x_n}^{x_{n+1}} f(x, y)\, dx \tag{2}$$

In Adams-Moulton method, the function is replaced by a cubic interpolation polynomial, normally by the Newton backward difference formula:

$$p_3(x) = f_n + r \nabla f_n + \frac{1}{2} r (r + 1) \nabla^2 f_n + \frac{1}{6} r (r + 1) (r + 2) \nabla^3 f_n \tag{3}$$

where $r = (x - x_n)/h$. Thus, we have

$$\int_{x_n}^{x_{n+1}} f(x, y)\, \mathrm{d}x = h \int_0^1 \left[f_n + r\, \nabla f_n + \frac{1}{2} r(r+1)\, \nabla^2 f_n + \frac{1}{6} r(r+1)(r+2)\, \nabla^3 f_n \right] \mathrm{d}r$$

In the above expression, x is substitued in terms of r by the expression $r = (x - x_n)/h$. Thus $\mathrm{d}x = h\, \mathrm{d}r$ and the limit of integration is changed; x_n to 0 and x_{n+1} to 1. On carrying out the integration and its substitution in Eq. (2) gives

$$y^*_{n+1} = y_n + h \left[f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \frac{3}{8} \nabla^3 f_n \right] \tag{4}$$

Equation (4) is known as Adams-Bashforth formula and is used as a predictor formula to get more corrected value (known as collector value). The predicted value obtained from Eq. (1) is superscripted by asterisk, y^*_{n+1}. Equation (4) is normally expressed in terms of function f. Since

$$\nabla f_n = f_n - f_{n-1}$$

$$\nabla^2 f_n = (f_n - f_{n-1}) - (f_{n-1} - f_{n-2}) = f_n - 2 f_{n-1} + f_{n-2}$$

$$\nabla^3 f_n = f_n - 3 f_{n-1} + 3 f_{n-2} - f_{n-3}$$

we get $$y^*_{n+1} = y_n + \frac{h}{24} (55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}) \tag{5}$$

To calculate y^*_{n+1}, one needs the values of f_n, f_{n-1}, f_{n-2} and f_{n-3}. These may be computed by some other method, say, the Runge-Kutta method.

The collector formula to get more correct value is also the Newton's backward formula at f_{n+1}, i.e.

$$\tilde{p}_3(x) = f_{n+1} + r\, \nabla f_{n+1} + \frac{1}{2}\, r(r+1)\, \nabla^2 f_{n+1} + \frac{1}{6}\, r(r+1)\, (r+2)\, \nabla^3 f_{n+1} \tag{6}$$

where $r = (x - x_{n+1})/h$. Integrating this expression from x_n to x_{n+1}, we get

$$\int_{x_n}^{x_{n+1}} \tilde{p}_3(x)\, \mathrm{d}x = h \left[\int_{-1}^{0} \left(f_{n+1} + r\, \nabla f_{n+1} + \frac{1}{2} r(r+1)\, \nabla^2 f_{n+1} + \frac{1}{6} r(r+1)(r+2)\, \nabla^3 f_{n+1} \right) \mathrm{d}r \right]$$

$$= h \left(f_{n+1} - \frac{1}{2} \nabla f_{n+1} - \frac{1}{12} \nabla^2 f_{n+1} - \frac{1}{24} \nabla^3 f_{n+1} \right) \tag{7}$$

Expressing differences in terms of function f, and the substituting in Eq. (2), we get

$$y_{n+1} = y_n + \frac{h}{24} (9 f^*_{n+1} + 19 f_n - 5 f_{n-1} + f_{n-2}) \tag{8}$$

where $f^*_{n+1} = f(x_{n+1}, y_{n+1})$ and others are obtained by the usual y's, i.e.

$$f_n = f(x_n, y_n); \quad f_{n-1} = f(x_{n-1}, y_{n-1}) \quad \text{and} \quad f_{n-2} = f(x_{n-2}, y_{n-2}).$$

The predictor – collector method involving Eqs (5) and (8) is known as Adams – Moulton method. The collector formula (Eq. 8) may be used repeatedly until the relative difference between successive values of y_{n+1} is reduced to a small pre-assigned value.

Illustration Determine the solution of differential equation $y' = -k\, y$ for the first-order kinetics with $y(t = 0) = [\mathrm{A}]_0 = 0.25$ mol dm^{-3} at t/s = 200, 400, 600, 800, 1000 and 1200. Given: $k = 0.000\,622$ s^{-1}.

Program

```
CLS
DIM X1(20), Y1(20)
XMAX = 6: ITER = 1
REM FIRST-ORDER KINETICS
K = .000622: X = 0: Y = .25: Y0 = Y: H = 200
DEF FNA (X, Y) = -K * Y
DEF FNB (X, Y) = Y0 * EXP(-K * X)
REM SECOND-ORDER KINETICS
'X = 0: Y = 632: H = 6: K = .0000178: Y0 = Y
'DEF FNA (X, Y) = -K * Y ^ 2
'DEF FNB (X, Y) = Y0 / (1 + K * X * Y0)
PRINT STRING$(35, "-")
PRINT " N       X           Y          Yexact"
PRINT STRING$(35, "-")
FOR N = 0 TO 3
X1(N) = X: Y1(N) = Y
Z = Y
K1 = H * FNA(X, Z)
K2 = H * FNA(X + .5 * H, Z + .5 * K1)
K3 = H * FNA(X + .5 * H, Z + .5 * K2)
K4 = H * FNA(X + H, Z + K3)
5 PRINT USING "##"; N; : PRINT USING "   ###.##"; X1(N);
PRINT USING "     ###.####"; Y1(N); FNB(X, Z)
Y = Y + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
X = X + H
NEXT N
PRINT STRING$(35, "-"): PRINT STRING$(43, "-")
PRINT " N      X           YP          YC         Yexact"
PRINT STRING$(43, "-")
FOR M = 3 TO XMAX
TERM = 55 * FNA(X1(M), Y1(M))
TERM = TERM - 59 * FNA(X1(M - 1), Y1(M - 1))
TERM = TERM + 37 * FNA(X1(M - 2), Y1(M - 2))
TERM = TERM - 9 * FNA(X1(M - 3), Y1(M - 3))
Y1P = Y1(M) + (H / 24) * TERM
FOR I = 1 TO ITER
TERM = 9 * FNA(X, Y1P)
```

```
TERM = TERM + 19 * FNA(X1(M), Y1(M))
TERM = TERM - 5 * FNA(X1(M - 1), Y1(M - 1))
TERM = TERM + FNA(X1(M - 2), Y1(M - 2))
Y1C = Y1(M) + (H / 24) * TERM
PRINT USING "##"; M + 1; : PRINT USING " ####.##"; X;
PRINT USING "    ###.####"; Y1P; Y1C; FNB(X, Z)
IF ABS(Y1P - Y1C) < .0001 THEN 10
Y1P = Y1C
NEXT I
10 PRINT : A$ = INPUT$(1)
Y1(M + 1) = Y1C: X1(M + 1) = X: Z = Y1(M)
X = X + H
NEXT M
PRINT STRING$(43, "-")
END
```

Output

```
-----------------------------------
 N      X          Y         Yexact
-----------------------------------
 0    0.00       0.2500       0.2500
 1  200.00       0.2208       0.2208
 2  400.00       0.1949       0.1949
 3  600.00       0.1721       0.1721
-----------------------------------
-------------------------------------------
 N     X          YP         YC        Yexact
-------------------------------------------
 4  800.00     0.1520     0.1520     0.1520

 5 1000.00     0.1342     0.1342     0.1342

 6 1200.00     0.1185     0.1185     0.1185

 7 1400.00     0.1047     0.1047     0.1047

-------------------------------------------
```

Exercise Execute the program for the other data provided in the program

Milne's Method

The Milne's method makes use of Newton's forward difference formula

$$f(x, y) = f_0 + r\,\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 f_0 + \cdots \quad (1)$$

where $r = (x - x_0)/h$.

The predictor formula is based on the expression

$$y_4 = y_0 + \int_{x_0}^{x_4} f(x, y)\,\mathrm{d}x$$

Substituting $f(x, y)$ from Eq. (1) retaining terms upto $\Delta^3 f_0$, we get

$$y_4 = y_0 + \int_{x_0}^{x_4}\left(f_0 + r\,\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 f_0\right)\mathrm{d}x$$

since $r = (x - x_0)/h$, we get $\quad h\,\mathrm{d}r = \mathrm{d}x$

Limits of integration are changed from 0 to 4. Hence

$$y_4 = y_0 + h\int_0^4\left(f_0 + r\,\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 + \frac{r(r-1)(r-2)}{3!}\Delta^3 f_0\right)\mathrm{d}r \quad (2)$$

$$= y_0 + h\left(4f_0 + 8\Delta f_0 + \frac{20}{3}\Delta^2 f_0 + \frac{8}{3}\Delta^3 f_0\right)$$

Now $\quad \Delta f_0 = f_1 - f_0$

$$\Delta^2 f_0 = \Delta(\Delta f_0) = \Delta(f_1 - f_0) = (f_2 - f_1) - (f_1 - f_0) = f_2 - 2f_1 + f_0$$

$$\Delta^3 f_0 = f_3 - 3f_2 + 3f_1 + f_0$$

With these, Eq. (2) becomes

$$y_4 = y_0 + \frac{4h}{3}(2f_1 - f_2 + 2f_3) \quad (3)$$

The general form of predictor formula is

$$y^*_{n+1} = y_{n-3} + \frac{4h}{3}(2f_{n-2} - f_{n-1} + 2f_n) \quad (4)$$

The collector formula is based on the expression

$$y_2 = y_0 + \int_{x_0}^{x_2} f(x, y)\,dx$$

Substituting Eq. (1) retaining terms upto $\Delta^2 f_0$, we get

$$y_4 = y_0 + \int_{x_0}^{x_2} \left[f_0 + r\,\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 \right] \mathrm{d}x$$

$$= y_0 + h\int_0^2 \left[f_0 + r\,\Delta f_0 + \frac{r(r-1)}{2!}\Delta^2 f_0 \right] \mathrm{d}r$$

$$= y_0 + h\left(2f_0 + 2\Delta f_0 + \frac{1}{3}\Delta^2 f_0 \right)$$

$$= y_0 + h\left[2f_0 + 2(f_1 - f_0) + \frac{1}{3}(f_2 - 2f_1 + f_0) \right]$$

$$= y_0 + \frac{h}{3}\,(f_0 + 4f_1 + f_2)$$

The general form of collector formula is

$$y_{n+1} = y_{n-1} + \frac{h}{3}\,(f_{n-1} + 4f_n + f^*_{n+1}) \tag{5}$$

where $f^*_{n+1} = y^*_{n+1}$ (from Eq. 4)

Illustration Determine the solution of $y' = \exp(x)$ (with $y = 1$ at $x = 0$) at $x = 0.8$, 1.0, 1.2 and 1.4 and compare the results $y = \exp(x)$.

Program

```
CLS
DIM X1(20), Y1(20)
XMAX = 6: ITER = 1
X = 0: Y = 1: H = .2
DEF FNA (X, Y) = EXP(X): DEF FNB (X, Y) = EXP(X)
'X = 0: Y = 0: H = .2
'DEF FNA (X, Y) = X + Y: DEF FNB (X,Y) = EXP(X) - X - 1
'DEF FNA (X, Y) = 1 + Y ^ 2: DEF FNB (X,Y) = TAN(X)
''X = 1: Y = 0: H = .2
''DEF FNA (X, Y) = 1 / X: DEF FNB (X,Y) = LOG(X)
REM FIRST-ORDER KINETICS
'X = 0: Y = .25: Y0 = Y: K = .000622: H = 200
'DEF FNA (X, Y) = -K * Y: DEF FNB (X, Y) = Y0 * EXP(-K * X)
REM SECOND-ORDER KINETICS
'X = 0: Y = 632: H = 6: K = .0000178: Y0 = Y
'DEF FNA (X, Y) = -K * Y ^ 2: DEF FNB (X, Y) = Y0 / (1 + K *
X * Y0)
PRINT STRING$(33, "-")
```

```
PRINT " N      X            Y          Yexact"
PRINT STRING$(33, "-")
FOR N = 0 TO 3
X1(N) = X: Y1(N) = Y
K1 = H * FNA(X, Y)
K2 = H * FNA(X + .5 * H, Y + .5 * K1)
K3 = H * FNA(X + .5 * H, Y + .5 * K2)
K4 = H * FNA(X + H, Y + K3)
5 PRINT USING "##"; N; : PRINT USING "####.##"; X1(N);
PRINT USING "      ###.###"; Y1(N); FNB(X, Y)
Y = Y + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
X = X + H
NEXT N
PRINT STRING$(33, "-"): PRINT STRING$(45, "-")
PRINT " N       X           YP           YC         Yexact"
PRINT STRING$(45, "-")
FOR M = 3 TO XMAX
TERM = 2 * FNA(X1(M - 2), Y1(M - 2))
TERM = TERM - FNA(X1(M - 1), Y1(M - 1))
TERM = TERM + 2 * FNA(X1(M), Y1(M))
Y1P = Y1(M - 3) + (4 * H / 3) * TERM
FOR I = 1 TO ITER
TERM = FNA(X1(M - 1), Y1(M - 1))
TERM = TERM + 4 * FNA(X1(M), Y1(M))
TERM = TERM + FNA(X, Y1P)
Y1C = Y1(M - 1) + (H / 3) * TERM
PRINT USING "##"; M + 1; : PRINT USING "   ####.##"; X;
PRINT USING "     ###.###"; Y1P; Y1C; FNB(X, Y)
IF ABS(Y1P - Y1C) < .00001 THEN 10
Y1P = Y1C
NEXT I
10 PRINT : A$ = INPUT$(1)
Y1(M + 1) = Y1C: X1(M + 1) = X
X = X + H
NEXT M
PRINT STRING$(45, "-")
END
```

Output

```
--------------------------------
 N      X           Y          Yexact
--------------------------------
 0    0.00        1.000        1.000
 1    0.20        1.221        1.221
 2    0.40        1.492        1.492
 3    0.60        1.822        1.822
--------------------------------
-----------------------------------------------
 N       X          YP          YC         Yexact
-----------------------------------------------
 4      0.80       2.225       2.226       2.226

 5      1.00       2.718       2.718       2.718

 6      1.20       3.320       3.320       3.320

 7      1.40       4.055       4.055       4.055

-----------------------------------------------
```

Exercise Run the program for the other data provided in the program

2.64 SOLUTION OF DIFFERENTIAL EQUATION OF THE SECOND-ORDER

The solution of the second-order differential equation

$$y'' = f(x, y, y'), \quad \text{with two initial conditions} \quad y(x_0) = y_0 \quad \text{and} \quad y'(x_0) = y'_0$$

may be carried out by using Runge-Kutta-Nyström method. The computational procedure for the fourth-order method (which include the term up to h^4 (where h is increment in the value of x) in the Taylor formulas for y and y') is as follows.

Compute first the following four terms.

$$k_1 = \frac{1}{2}\ h\ f(x_n, y_n, y'_n)$$

$$k_2 = \frac{1}{2} h\, f\left(x_n + \frac{1}{2}h,\ y_n + K,\ y'_n + k_1\right); \quad \text{where } k = \frac{1}{2} h\left(y'_n + \frac{1}{2}k_1\right)$$

$$k_3 = \frac{1}{2} h\, f\left(x_n + \frac{1}{2}h,\ y_n + K,\ y'_n + k_2\right)$$

$$k_4 = \frac{1}{2}\ hf(x_n + h,\ y_n + L,\ y'_n + 2\ k_3); \qquad \text{where } L = h(y'_n + k_3)$$

The values of x_{n+1}, y_{n+1} and y'_{n+1} are then computed by using the expressions

$$x_{n+1} = x_n + h$$

$$y_{n+1} = y_n + h\left(y'_n + \frac{1}{3}(k_1 + k_2 + k_3)\right)$$

$$y'_{n+1} = y'_n + \frac{1}{3}\ (k_1 + 2\ k_2 + 2\ k_3 + k_4)$$

The above computations are carried out for $n = 0, 1, \ldots, N - 1$

Illustration To display the wave functions of a particle in a one-dimensional box of length unity by solving the second-order differential equation $y'' = -\alpha^2\ y$ where $\alpha = n\pi$. Given $y(0) = 0$ and $y'(0) = n\pi$.

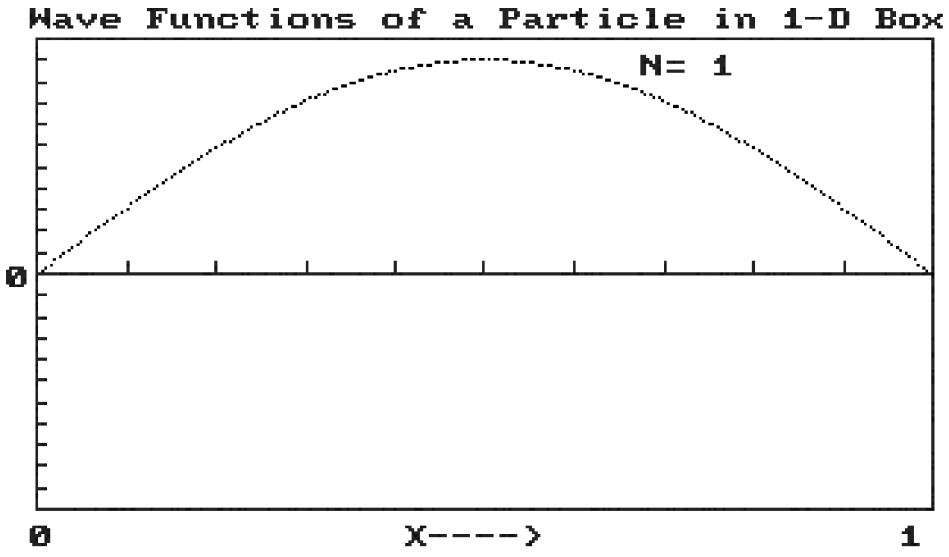
For a particle in one – dimensional box we know that $y = \sin(n\pi\ x)$.

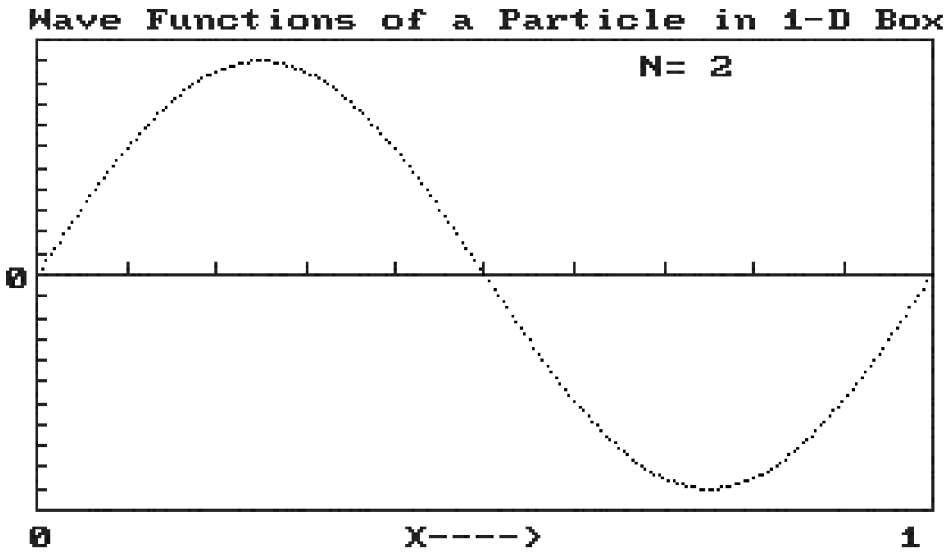
Program

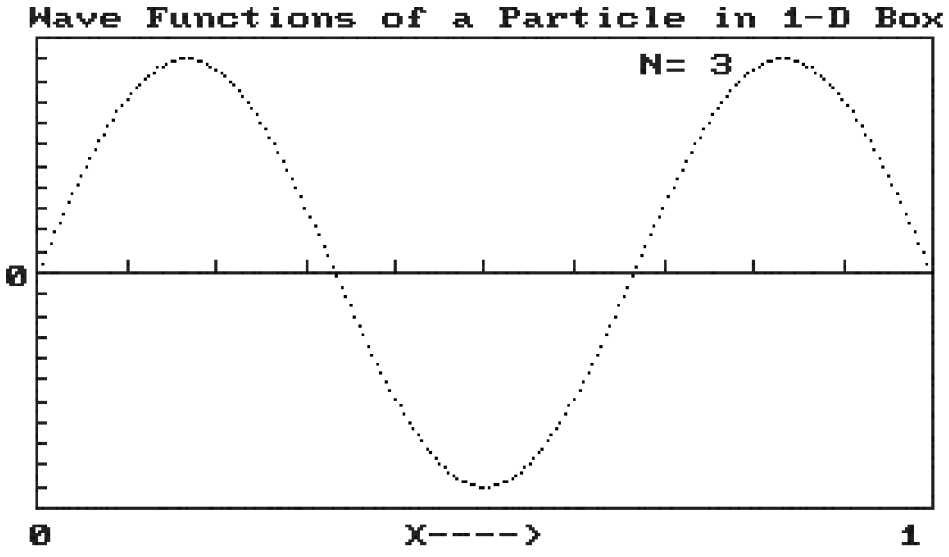
```
REM Solution of Second Order Differential Equation
REM Runge-Kutta-Nystrom (Fourth-Order) Method
REM BOX$="BOX" if One-Dimensional Box
REM PLOT$="Y" if plot is required
REM NMAX=Number of wave functions to be displayed
CLS
READ BOX$, PLOT$, NMAX: DATA BOX,Y,5
'DEF FNA (X, Y, YP) = .5 * (X + Y + YP + 2)
'DEF FNB (X, Y, YP) = EXP(X) - X - 1
'DEF FNC (X, Y, YP) = EXP(X) - 1
REM Application to Particle in a One-Dimensional Box
REM Function FNA is the given differential equation.
REM Function FNB gives the exact value of y
REM Function FNC gives the exact value of y'.
DEF FNA (X, Y, YP) = -NP ^ 2 * Y
DEF FNB (X, Y, YP) = SIN(NP * X)
DEF FNC (X, Y, YP) = -COS(NP * X) * NP
FOR NN = 1 TO NMAX
X = 0: Y = 0: YP = 0: NP = NN * 3.14159
IF BOX$ = "BOX" THEN : YP = NP
IF PLOT$ = "Y" THEN
CLS : N = 200: H = 1 / N
SCREEN 1: COLOR 14, 0: VIEW (10, 10)-(315, 170)
WINDOW (0, -1.1)-(1, 1.1)
LINE (0, -1.1)-(1, 1.1), , B
LOCATE 1, 2: PRINT "Wave Functions of a Particle in 1-D Box"
LINE (0, 0)-(1, 0): LOCATE 23, 2
PRINT "0               X---->               1"
LOCATE 12, 1: PRINT "0"
FOR I = .1 TO 1.1 STEP .1: LINE (I, 0)-(I, .05)
LINE (0, I)-(.01, I): LINE (0, -I)-(.01, -I)
NEXT I
ELSE N = 10: H = 1 / N: PRINT STRING$(50, "-")
PRINT "  N    X       Y       Yexact      YP     YPexact"
PRINT STRING$(50, "-")
```

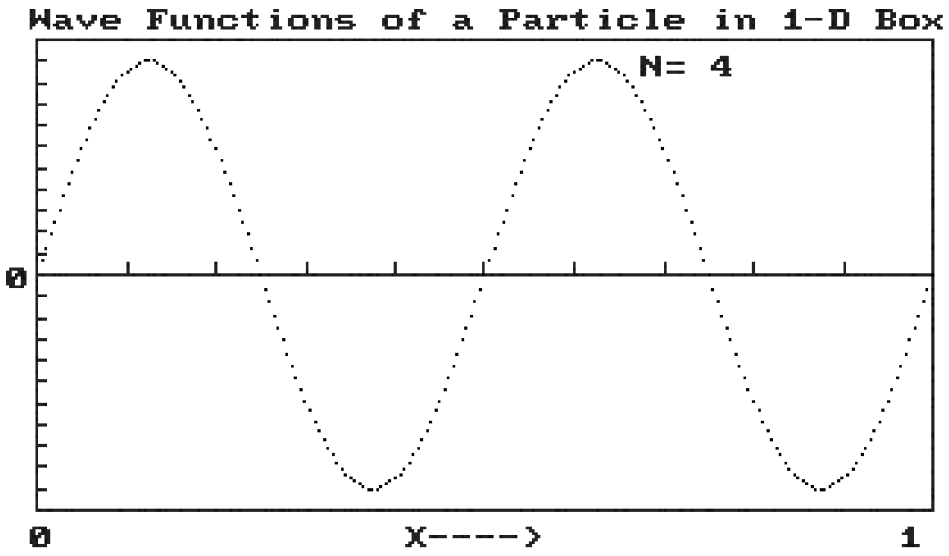
```
PRINT "  0"; : PRINT USING "  #.##"; X;
PRINT USING "  ##.#####"; Y; FNB(X, Y, YP); YP; FNC(X, Y, YP)
END IF
FOR I = 0 TO N - 1
K1 = .5 * H * FNA(X, Y, YP)
KK = .5 * H * (YP + .5 * K1)
K2 = .5 * H * FNA(X + .5 * H, Y + KK, YP + K1)
K3 = .5 * H * FNA(X + .5 * H, Y + KK, YP + K2)
LL = H * (YP + K3)
K4 = .5 * H * FNA(X + H, Y + LL, YP + 2 * K3)
X = X + H
Y = Y + H * (YP + (1 / 3) * (K1 + K2 + K3))
YP = YP + (1 / 3) * (K1 + 2 * K2 + 2 * K3 + K4)
IF PLOT$ = "Y" THEN
LOCATE 3, 28: PRINT "N="; : PRINT USING "##"; NN
PSET (X, Y), 2
ELSE PRINT USING "###"; I + 1; : PRINT USING "  #.##"; X;
PRINT USING " ###.#####"; Y; FNB(X, Y, YP); YP; FNC(X, Y, YP)
A$ = INPUT$(1)
END IF
NEXT I
A$ = INPUT$(1)
IF PLOT$ <> "Y" THEN PRINT STRING$(50, "-")
NEXT NN
END
```

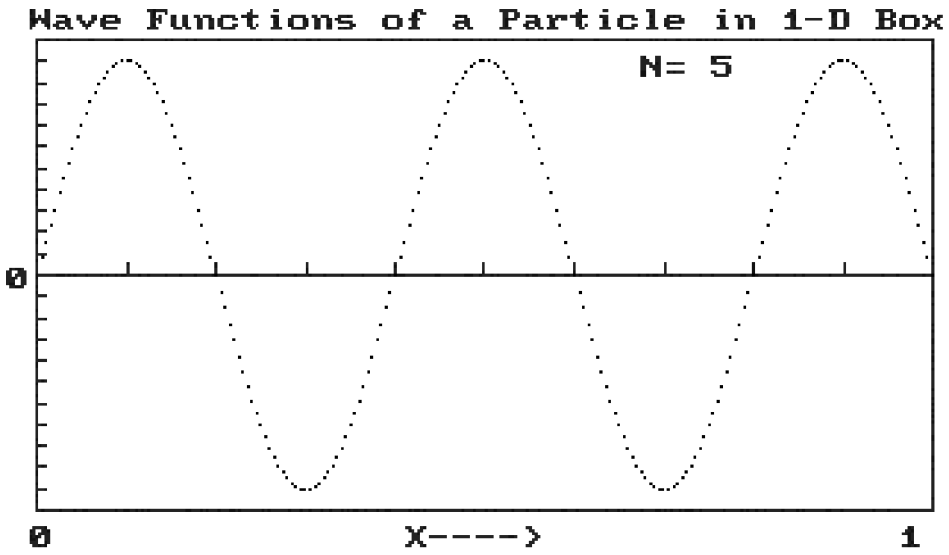
Output




Wave Functions of a Particle in 1-D Box
N= 2
0
0
X---->
1


Wave Functions of a Particle in 1-D Box
N= 3
0
0
X---->
1





Note: Only wave function for $n = 5$ is shown here.

Exercise

Determine the values of function $y(x)$ at $x = 0.2, 0.4, 0.6, 0.8$ and 1.0 which satisfies the differential equation $y'' - xy' + 2y = 0$. Given: $y(0) = 1$ and $y'(0) = 0$ (**Ans:** 0.96, 0.84, 0.64, 0.36, 0)

2.65 AUTOCATALYSIS

For an autocatalytic reaction

$$A + C \longrightarrow 2C$$

the proposed mechanism is

$$A + C \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} D \qquad \text{(fast)}$$

$$D \longrightarrow 2C \qquad \text{(slow)}$$

The rate expression is

$$\frac{x}{[C]_0} = \frac{e^{\beta t - 1}}{1 + Be^{\beta t}}$$

where x is the extent of reaction, $B = [C]_0/[A]_0$ and $b = ([A]_0 + [C]_0)/(2k_2k_1/k_{-1})$.

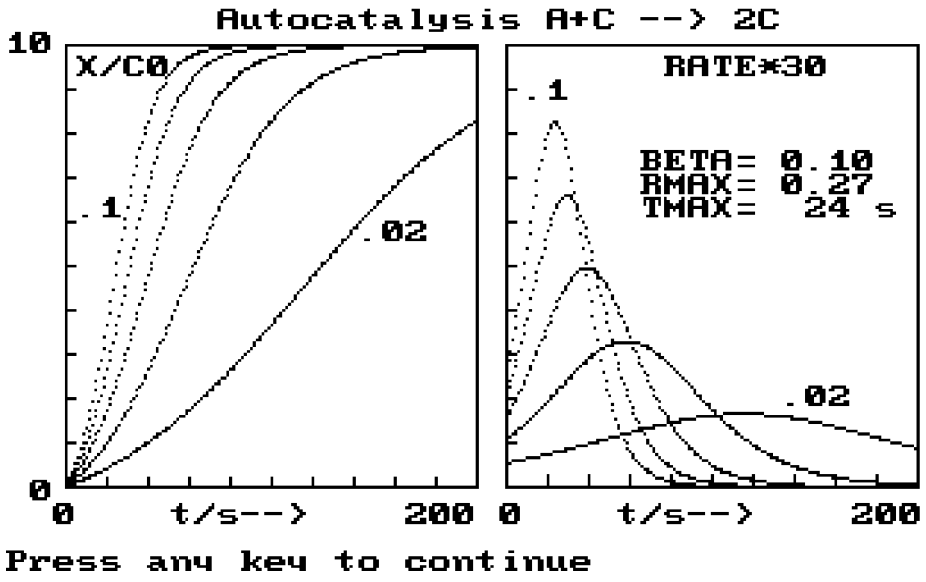
Illustration Plot $x/[C]_0$ versus t for $\beta = 0.02$, 0.04, 0.06, 0.08 and 0.1. Take $B = 0.1$. Also determine the value of maximum rate and the corresponding time for each plot.

Program

```
REM Autocatalysis
CLS : READ TF, XF: DATA 200,10
BETAF = .1: B = .1: TS = 1: Y = 0: RMAX = 0: TMAX = 0
SCREEN 1: COLOR 15, 0: RF = XF / 30
LOCATE 2, 10: PRINT "Autocatalysis A+C --> 2C"
LOCATE 4, 4: PRINT "X/C0": LOCATE 4, 29: PRINT "RATE*30"
LOCATE 3, 1: PRINT "10": LOCATE 11, 16: PRINT ".02"
LOCATE 18, 34: PRINT ".02": LOCATE 22, 2: PRINT "0"
LOCATE 23, 3: PRINT "0     t/s-->    "; TF; "0     t/s-->    "; TF
FOR BETA = .02 TO BETAF STEP .02
FOR T = 0 TO 200 STEP TS
BT = BETA * T
X = (EXP(BT) - 1) / (1 + B * EXP(BT))
VIEW (20, 20)-(160, 170): WINDOW (0, 0)-(TF, XF)
LINE (0, 0)-(TF, XF), , B: PSET (T, X)
FOR I = 1 TO 9
LINE (TF * .1 * I, 0)-(TF * .1 * I, .2)
LINE (0, XF * .1 * I)-(TF / 50, XF * .1 * I)
NEXT I
RATE = (X - Y) / TS
VIEW (170, 20)-(310, 170): WINDOW (0, 0)-(TF, RF)
LINE (0, 0)-(TF, RF), , B: PSET (T, RATE)
```

```
FOR I = 1 TO 9
LINE (TF * .1 * I, 0)-(TF * .1 * I, .01)
LINE (0, RF * .1 * I)-(TF / 50, RF * .1 * I)
NEXT I
Y = X
IF RATE > RMAX THEN RMAX = RATE: TMAX = T
NEXT T
LOCATE 8, 28: PRINT "BETA="; : PRINT USING " #.##"; BETA
LOCATE 9, 28: PRINT "RMAX="; : PRINT USING " #.##"; RMAX
LOCATE 10, 28: PRINT "TMAX="; : PRINT USING " ###"; TMAX;
PRINT " s": A$ = INPUT$(1)
NEXT BETA
LOCATE 5, 23: PRINT ".1": LOCATE 10, 4: PRINT ".1"
END
```

Output



3 Projects

3.1 DISPLAY OF VAN DER WAALS ISOTHERMS ALONG WITH ITS SURFACE OF DISCONTINUITY

To draw the variation of pressure with volume of a real gas at temperatures lower than its critical temperature. Also draw its surface of discontinuity.

The van der Waals equation for one mole of gas is

$$\left(p + \frac{a}{V^2}\right)(V - b) = RT$$

which on opening gives

$$V^3 - V^2\left(\frac{pb + RT}{p}\right) + V\frac{a}{p} - \frac{ab}{p} = 0 \tag{1}$$

This equation gives three real roots V_a, V_b and V_c for a given value of p at a temperature lower than the critical temperature of the gas. Of these, the smallest value (say, V_a) and the largest value (say, V_c) lie on the surface of discontinuity provided the area between the line $V_a V_b$ and the isotherm (labelled as Area-1) is equal and of opposite sign of the area between the line $V_b V_c$ and the isotherm (labelled as Area-2).

There roots of Eq. (1) can be obtained by Lin-Bairstow's method in which Eq. (1) is resolved into a linear and a quadratic factor as described in the following.

Let Eq. (1). be written as

$$f(x) = A_3x^3 + A_2x^2 + A_1x + A_0 = 0 \tag{2}$$

Let $x^2 + Rx + S$ be its quadratic factor and let $x^2 + rx + s$ be its approximate factor.

The first approximation of r and s is obtained by writing

$$r = \frac{A_1}{A_2} \quad \text{and} \quad s = \frac{A_0}{A_2} \tag{3}$$

Equation (2) may be written as

$$f(x) = (x^2 + rx + s)(B_2x + B_1) + Cx + D \tag{4}$$

$$= B_2x^3 + (B_2r + B_1)x^2 + (C + B_1r + sB_2)x + (B_1s + D) \tag{5}$$

Equating the corresponding coefficients in Eqs (2) and (5), we get

$$\begin{aligned} B_2 &= A_3 \\ B_2r + B_1 &= A_2 \\ C + B_1r + sB_2 &= A_1 \\ B_1s + D &= A_0 \end{aligned} \tag{6}$$

These give

$$B_2 = A_3$$
$$B_1 = A_2 - rB_2$$
$$C = A_1 - B_1 r - B_2 s$$
$$D = A_0 - B_1 s \tag{7}$$

We find that the coefficients B_1, C and D are functions of r and s.

Since $x^2 + Rx + S$ is a factor of Eq. (2), we may write

$$f(x) = (x^2 + Rx + S)\,(B_2\, x + B_1) \tag{8}$$

Comparing Eqs (4) and (7), we conclude that $r \to R$, $s \to S$ provided $C(R, S) = 0$ and $D(R, S) = 0$ where we wirte

$$R = r + \Delta r \quad \text{and} \quad S = s + \Delta s. \tag{9}$$

Expanding $C(R, S)$ and $D(R, S) = 0$ as Taylor's series, we get

$$C(R, S) = C(r, s) + \Delta r\, \frac{\partial C}{\partial r} + \Delta s \frac{\partial C}{\partial s} = 0 \tag{10}$$

$$D(R, S) = D(r, s) + \Delta r\, \frac{\partial D}{\partial r} + \Delta s \frac{\partial D}{\partial s} = 0 \tag{11}$$

where the derivatives are evaluated at r and s. Solving Eqs (10) and (11) for Δr and Δs, we get

$$\Delta r = \frac{-C(\partial D/\partial s) + D(\partial C/\partial s)}{(\partial C/\partial r)\,(\partial D/\partial s) - (\partial D/\partial r)\,(\partial C/\partial s)} \tag{12}$$

$$\Delta s = \frac{-C(\partial D/\partial r) + D(\partial C/\partial r)}{(\partial C/\partial s)\,(\partial D/\partial r) - (\partial D/\partial s)\,(\partial C/\partial r)} \tag{13}$$

For the values of r and s provided by Eq. (3), the above two expressions are used to provide the values of Δr and Δs. With these, Eq. (9) provides the next approximation to R and S, respectively.

The process is repeated with the new values of R and S to provide new refined values of Δr and Δs. This iterative process is continued until the values of Δr and Δs tend toward some minimum value near to zero.

Once the factored equations $x^2 + Rx + S = 0$ and $B_2 x + B_1 = 0$ are obtained, these provide the three roots of the given cubic expression.

For the van der waals equation of state (Eq. 1), the various expressions to be used are

$$A_3 = 1 \quad ; \quad A_2 = -\,(pb + RT)/p$$
$$A_1 = a/p \quad ; \quad A_0 = -\,ab/p$$
$$r = \frac{A_1}{A_2} = -\frac{a}{pb + RT} \quad ; \quad s = \frac{A_0}{A_2} = \frac{a}{pb + RT}$$
$$B_2 = 1$$
$$B_1 = A_2 - rB_2 = -\frac{pb + RT}{p} - r$$
$$C = A_1 - rB_1 - sB_2 = \frac{a}{p} + r\left(\frac{pb + RT}{p} + r\right) - s$$

$$D = A_0 - sB_1 = -\frac{ab}{p} + s\left(\frac{pb + RT}{p} + r\right)$$

$$\frac{\partial C}{\partial r} = \frac{pb + RT}{p} + 2R \quad ; \quad \frac{\partial C}{\partial s} = -1$$

$$\frac{\partial D}{\partial r} = s \quad ; \quad \frac{\partial D}{\partial \mathrm{s}} = \frac{pb + RT}{p} + r$$

Once the three roots are obtained, the areas 'Area-1' and 'Area-2' as mentioned in the beginning are evaluated. The process is repeated for different values of p and the value of p is selected for which Area-1 = – Area-2. For this value of p, the roots V_a and V_c lie on the surface of discontinuity.

Illustration Van der Waals isotherms of O_2, vary pressure form 3800 kPa to 5100 kPa and temperature from 144 K to 154 K.

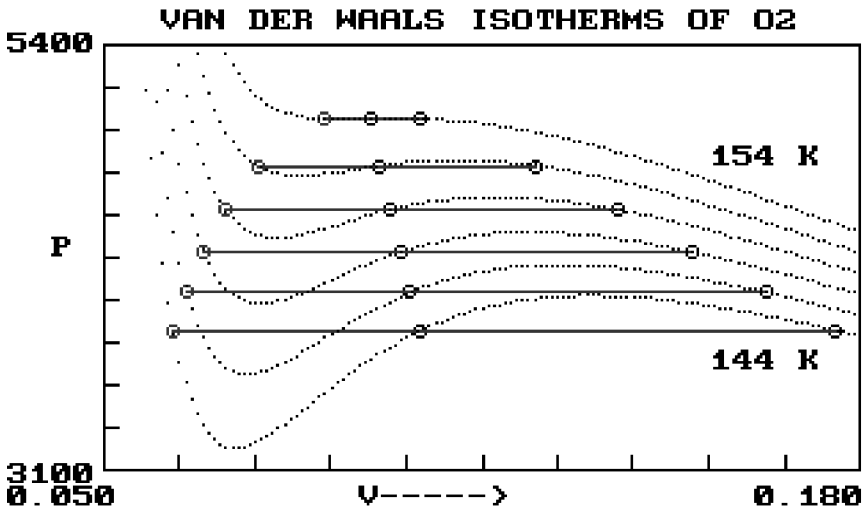
Program

```
REM PROGRAM ROOTVAN1;VAN DER WAALS ISOTHERMS
REM OPT=1 FOR DISPLAY OF VOLUMES
REM OPT=2 FOR DISPLAY OF PLOTS
CLS : READ OPT: DATA 2
READ NAME$, a, b, PI, PF, PS, TI, TF, TS, VI, VF, WP1, WP2
DATA O2,137.802,.03183,3800,5100,.5,144,154,2,.05,.18,700,300
'DATA CO2,363.96,.04267,5700,7500,1,286,304,1,.075,.25,700,300
'DATA C3H8,877.88,.08445,3900,4800,1,359,369,1,.15,.4,100,0
R1 = 8.314: A3 = 1: B2 = A3: PIP = PI
FOR T = TI TO TF STEP TS
FOR P = PIP TO PF STEP PS
R = -a / (P * b + R1 * T): S = a * b / (P * b + R1 * T)
TERM = b + R1 * T / P
FOR I = 1 TO 30
C = a / P + R * (TERM + R) - S
D = -a * b / P + S * (TERM + R): B1 = -TERM - R
DCDR = b + R1 * T / P + 2 * R: DCDS = -1
DDDR = S: DDDS = b + R1 * T / P + R
DR = (-C * DDDS + D * DCDS) / (DCDR * DDDS - DDDR * DCDS)
DS = (-C * DDDR + D * DCDR) / (DCDS * DDDR - DDDS * DCDR)
6 R = R + DR: S = S + DS
NEXT I
V1 = -B1 / B2: DIS = R * R - 4 * S
IF DIS < 0 THEN 20
V2 = (-R + DIS ^ .5) / 2: V3 = (-R - DIS ^ .5) / 2
IF V1 > V2 THEN SWAP V1, V2
IF V1 > V3 THEN SWAP V1, V3
IF V2 > V3 THEN SWAP V2, V3
```

```
VMIN = V1: VINT = V2: VMAX = V3
ON OPT GOTO 8, 11
8 PRINT "V1="; V1; "     V2="; V2; "         V3="; V3
11 AREA1 = 0: AREA2 = 0
FOR V = VMIN TO VINT STEP .00001
PP = R1 * T / (V - b) - a / V ^ 2
AREA1 = AREA1 + V * (PP - P)
NEXT V
FOR V = VINT TO VMAX STEP .00001
PP = R1 * T / (V - b) - a / V ^ 2
AREA2 = AREA2 + V * (PP - P)
NEXT V
ON OPT GOTO 14, 16
14 PRINT "T="; T; "   P="; P; "    AREA1="; AREA1; "    AREA2"; AREA2
a$ = INPUT$(1)
16 IF ABS(AREA1) > AREA2 THEN 35
GOTO 30
20 ON OPT GOTO 22, 30
22 PRINT "T="; T; "   P="; P;
PRINT "  The other two roots are imaginary": a$ = INPUT$(1)
30 NEXT P
35 ON OPT GOTO 36, 38
36 PRINT : PRINT : PRINT "V1="; V1, "V2="; V2, , "V3="; V3
PRINT "T="; T, "P="; P, "AREA1="; AREA1, "AREA2="; AREA2: PRINT
38 PIP = P
ON OPT GOTO 45, 40
40 SCREEN 1: COLOR 15, 0
VIEW (35, 20)-(310, 170)
WPS1 = PI - WP1: WPS2 = PF + WP2
WINDOW (VI, WPS1)-(VF, WPS2): LINE (VI, WPS1)-(VF, WPS2), , B
PSET (V1, P), 1: CIRCLE (V1, P), .001, 1
PSET (V2, P), 2: CIRCLE (V2, P), .001, 2
PSET (V3, P): CIRCLE (V3, P), .001
LINE (V1, P)-(V3, P), 1
FOR VV = VI TO VF STEP .001
PP = R1 * T / (VV - b) - a / VV ^ 2: PSET (VV, PP)
FOR KK = 1 TO 100 STEP .1: NEXT KK: NEXT VV
45 NEXT T
ON OPT GOTO 48, 46
'46 FOR T = TI TO TF STEP TS
'FOR V = VI TO VF STEP .001
'p = R1 * T / (V - B) - A / V ^ 2: PSET (V, p)
```

```
'NEXT V: NEXT T
46 LOCATE 2, 8: PRINT "VAN DER WAALS ISOTHERMS OF "; NAME$
LOCATE 3, 1: PRINT USING "####"; PF + WP2
LOCATE 12, 3: PRINT "P"
LOCATE 22, 1: PRINT USING "####"; PI - WP1
LOCATE 23, 1: PRINT USING "#.###"; VI;
PRINT "            V-----> "; : PRINT USING "#.###"; VF
FOR I = 1 TO 9
VS = (VF - VI) * .1: PS = (WPS2 - WPS1) * .1
LINE (VI + VS * I, WPS1)-(VI + VS * I, WPS1 + PS * .3)
LINE (VI, WPS1 + PS * I)-(VI + VS * .2, WPS1 + PS * I)
NEXT I
LOCATE 8, 32: PRINT TF; "K"
LOCATE 17, 32: PRINT TI; "K"
48 END
```

Output



Exercise Draw van der waals of CO_2 and C_3H_8 (data given in the program.)

3.2 DISPLAY OF GIBBS VALLEY OF A REACTION

To draw Gibbs valley of the reaction $N_2O_4(g) \rightleftharpoons 2NO_2(g)$ and to determine its standard equilibrium constant from the extent of reaction at equilibrium.

The plot of G_{total} versus ξ (extent of reaction) of a reaction is known as the Gibbs valley. To determine

G_{total} in terms of ξ, the following expressions are used.

$$G_{\text{total}} = G_{\text{pure}} + \Delta_{\text{mix}}G$$

where $$G_{\text{pure}} = \sum_{\text{B}} \{(n_{\text{B}})_0 + \nu_{\text{B}}\xi\}\mu^{\circ}_{\text{B}} + RT\, n_{\text{total}} \ln\left(\frac{p}{p^{\circ}}\right)$$

$$\Delta_{\text{mix}}G = RT \sum_{\text{B(g)}} n_{\text{B}} \ln \frac{n_{\text{B}}}{n_{\text{total}}}$$

Here, $$n_{\text{B}} = (n_{\text{B}})_0 + \nu_{\text{B}}\xi$$

$$n_{\text{total}} = \sum_{\text{B(g)}} (n_{\text{B}})_0 + \nu_{\text{B}}\xi$$

To determine extent of reaction at equilibrium, the expression of $\Delta_{\text{r}}\tilde{G} = (\partial G/\partial\xi)_{T,\,p}$ may be used. It has a zero value at equilibrium.

The expression of $\Delta_{\text{r}}\tilde{G}$ is

$$\Delta_{\text{r}}\tilde{G} = \sum_{\text{B}} \nu_{\text{B}}\mu^{\circ}_{\text{B}} + \left(\sum_{\text{B}} \nu_{\text{B}}\right) RT \ln\left(\frac{p}{p^{\circ}}\right) + RT \sum_{\text{B(g)}} \nu_{\text{B}} \ln\left(\frac{n_{\text{B}}}{n_{\text{total}}}\right)$$

The above expressions required the following input of data at 298 K.

Temperature (298 K), pressure (= 1 atm), number of constituents (other than elements in their standard states) in the reaction along with their phases (gas or liquid or solid), initial amounts and stoichiometric numbers (negative for reactants and positive for products).

If the temperature is other than 298 K, the following expression may be used to determine the value of $\Delta_{\text{f}}G^{\circ}_{T}$.

$$\Delta_{\text{f}}G^{\circ}_{T} = -(\Delta_{\text{r}}a)T\, \ln\left(\frac{T}{\text{K}}\right) - \left(\frac{\Delta_{\text{r}}b}{2}\right)T^2 - \left(\frac{\Delta_{\text{r}}c}{6}\right)T^3 + K_1 + K_2T$$

where $$K_1 = \Delta_{\text{r}}H^{\circ}_{T_0} - \left[(\Delta_{\text{r}}a)T_0 + (\Delta_{\text{r}}b)\frac{T_0^2}{2} + (\Delta_{\text{r}}c)\frac{T_0^3}{3}\right]$$

$$K_2 = \frac{\Delta_{\text{r}}G^{\circ}_{T_0}}{T_0} + (\Delta_{\text{r}}a)\ln\left(\frac{T_0}{\text{K}}\right) + \left(\frac{\Delta_{\text{r}}b}{2}\right)T_0 + \left(\frac{\Delta_{\text{r}}c}{6}\right)T_0^2 - \frac{K_1}{T_0}$$

$$T_0 = 298 \text{ K}.$$

Make a program to study the reaction $N_2O_4(g) \rightleftharpoons 2NO_2(g)$

Given: $\mu^{\circ}_{NO_2(g)} = 51.31 \text{ kJ mol}^{-1}$ and $\mu^{\circ}_{N_2O_4(g)} = 97.89 \text{ kJ mol}^{-1}$

Constituent	$\frac{C_p/\text{J K}^{-1}\text{ mol}^{-1} = a + b(T/\text{K}) + c(T/\text{K})^2}{a \quad b\times10^3 \quad c\times10^6}$			$\frac{\Delta_f H^\circ_{298\text{K}}}{\text{kJ mol}^{-1}}$	$\frac{\Delta_f G^\circ_{298\text{K}}}{\text{kJ mol}^{-1}}$
$N_2(g)$	29.295	– 2.2634	5.6649	0	0
$O_2(g)$	26.648	9.55	–1.328	0	0
$NO_2(g)$	25.036	45.550	–18.420	33.18	51.31
$N_2O_4(g)$	36.208	169.98	–77.984	9.16	97.89

Program

```
2 CLS
3 DEF FNA$ (J)
4 IF J = 1 THEN FNA$ = "FIRST"
5 IF J = 2 THEN FNA$ = "SECOND"
6 IF J = 3 THEN FNA$ = "THIRD"
7 IF J = 4 THEN FNA$ = "FOURTH"
8 IF J = 5 THEN FNA$ = "FIFTH"
9 IF J = 6 THEN FNA$ = "SIXTH"
10 END DEF
11 DIM NUB(6), MUB(6), PHASE$(6), NAME$(6), NAM$(6), BG(6)
12 DIM GPURE(9, 910), GTOTAL(9, 910), DGDXI(9, 910), XIEQ(9)
   DIM KP(9), GEQ(9), NUCF(9), DMIXG(9, 910), A(6), B(6)
   DIM C(6), K1(6), K2(6), H0(6), G0(6), IN(6, 6), NUCR(6, 6)
14 DIM NCR(6), TT(9), PP(9), NB(6)
16 READ TI: PRINT "Initial temperature in kelvin="; TI
              DATA 298
18 READ TF: PRINT "Final temperature in kelvin="; TF
              DATA 358
              'DATA 298
              'DATA 328
20 READ TD: PRINT "Temperature inerval in kelvin="; TD
              DATA 30
25 READ PI: PRINT "Initial pressure in atm="; PI
              DATA 1
30 READ PF: PRINT "Final pressure in atm="; PF
              DATA 1
32 READ PD: PRINT "Pressure interval in atm="; PD
              DATA .15
34 IF TF = 298 THEN GT = 0 ELSE GT = 1
35 READ NC
   PRINT "Number of constituents in the main chemical reaction,NC="; NC
              DATA 2
37 READ PLOT: PRINT "PLOT=1 if plot required else 0;PLOT="; PLOT
```

```
              DATA 1
    PRINT : A$ = INPUT$(1)
38  R = .008314: XIMAX = 10
39  FOR I = 1 TO NC
40  READ NAME$(I)
    PRINT "Name of the "; FNA$(I); " spieces="; NAME$(I)
41  READ PHASE$(I): PRINT "Phase of "; NAME$(I); "="; PHASE$(I)
43  READ NUB(I): PRINT "Stoichometric number of "; NAME$(I); "="; NUB(I)
45  READ NB(I): PRINT "Amount of "; NAME$(I); " in mole="; NB(I)
    PRINT : A$ = INPUT$(1)
47  IF NUB(I) >= 0 THEN 50
48  NBDNUB = NB(I) / ABS(NUB(I))
49  IF XIMAX > NBDNUB THEN XIMAX = NBDNUB
50  NEXT I
             DATA N2O4,G,-1,1,NO2,G,2,0
51  FOR I = 1 TO NC
52  READ MUB(I)
    PRINT "Chemical potential of "; NAME$(I); " in kJ/mol="; MUB(I)
    PRINT : A$ = INPUT$(1)
53  NEXT I
                DATA 97.89,51.31
54  JJ = 0
55  IF GT = 0 THEN 112
56  READ TNC
    PRINT "Total index number of the constituents including elements whose ";
    PRINT " FREE ENERGY is to be computed at different temperatures,TNC="; TNC
    PRINT : A$ = INPUT$(1)
                 DATA 4
    T0 = 298
70  FOR J = 1 TO TNC
    READ NAM$(J): PRINT "Index number "; J; " for the constituent "; NAM$(J)
    PRINT : A$ = INPUT$(1)
    PRINT "Molar heat capacity data in J/(K mol) of ";
    PRINT NAM$(J); " without exponent part"
75  READ A(J): PRINT "a="; A(J), : A(J) = A(J) * 10 ^ (-3)
80  READ B(J): PRINT "b="; B(J), : B(J) = B(J) * 10 ^ (-6)
85  READ C(J): PRINT "c="; C(J):  C(J) = C(J) * 10 ^ (-9)
    PRINT "Molar values of H0 and G0 in kJ/mol for "; NAM$(J)
90  READ H0(J): PRINT "H0="; H0(J),
91  READ G0(J): PRINT "G0="; G0(J)
    PRINT : A$ = INPUT$(1)
92  K1(J) = H0(J) - (A(J) * T0 + (B(J) / 2) * T0 ^ 2 + (C(J) / 3) * T0 ^ 3)
94  K2(J) = G0(J) / T0 + A(J) * LOG(T0) + (B(J) / 2) * T0
```

```
   K2(J) = K2(J) + (C(J) / 6) * T0 ^ 2 - K1(J) / T0
96 NEXT J
               DATA N2,29.295,-2.2634,5.6649,0,0
               DATA O2,26.648,9.55,-1.328,0,0
               DATA N2O4,36.208,160.98,-77.984,9.16,97.89
               DATA NO2,25.036,45.55,-18.42,33.18,51.31
100 FOR I = 1 TO NC
102 READ NCR(I): PRINT "Number of speices in the formation reaction of ";
PRINT NAME$(I); " ="; NCR(I)
104 FOR J = 1 TO NCR(I)
106 READ IN(I, J)
PRINT "Index number of the "; FNA$(J); " species in this";
PRINT " formation reaction "; IN(I, J)
108 READ NUCR(I, J)
PRINT "NU of this species="; NUCR(I, J)
PRINT : A$ = INPUT$(1)
110 NEXT J: NEXT I
               DATA 3,1,-1,2,-2,3,1
               DATA 3,1,-.5,2,-1,4,1
112 CLS
113 FOR T = TI TO TF STEP TD
114 NUBMUB = 0: NBMUB = 0: SUMNB = 0: SUMNUB = 0
115 IF GT = 0 THEN 130
116 FOR J = 1 TO TNC
117 BG(J) = -A(J) * T * LOG(T) - (B(J) / 2) * T ^ 2
    BG(J) = BG(J) - (C(J) / 6) * T ^ 3 + K1(J) + K2(J) * T
118 A$ = INPUT$(1)
119 PRINT T; "K", NAM$(J), "G="; BG(J); "kJ/mol"
120 NEXT J
130 FOR I = 1 TO NC
132 IF GT = 0 THEN 210
134 TEMP = 0
136 FOR J = 1 TO NCR(I): I1 = IN(I, J)
142 TEMP = TEMP + NUCR(I, J) * BG(I1)
144 NEXT J
146 PRINT "    O=";
170 FOR J = 1 TO NCR(I): I1 = IN(I, J)
172 IF NUCR(I, J) < 0 THEN PRINT "-";  ELSE PRINT "+";
174 PRINT ABS(NUCR(I, J)); "("; NAM$(I1); ")";
175 NEXT J
180 MUB(I) = TEMP
     PRINT "      "; "MUB("; I; ")="; TEMP; "kJ/mol"
182 A$ = INPUT$(1)
```

```
210 NBMUB = NBMUB + NB(I) * MUB(I)
    NUBMUB = NUBMUB + NUB(I) * MUB(I)
218 IF PHASE$(I) = "S" OR PHASE$(I) = "L" THEN 224
220 SUMNB = SUMNB + NB(I)
    SUMNUB = SUMNUB + NUB(I)
224 NEXT I
225 FOR P = PI TO PF STEP PD
226 KK = 0: JJ = JJ + 1: JJJ = 0
229 PRINT "    "; STRING$(50, "-")
230 PRINT "       XI      GPURE        DMIXG     GTOTAL      DGDXI"
232 PRINT "    "; STRING$(50, "-")
240 FOR J = 0 TO 100: XI = (J / 100) * XIMAX
245 TERM1 = 0: TERM2 = 0
250 FOR I = 1 TO NC
255 IF PHASE$(I) = "S" OR PHASE$(I) = "L" THEN 275
260 TERM = (NB(I) + NUB(I) * XI) / (SUMNB + SUMNUB * XI)
261 IF TERM <= 0 THEN 275
262 TERM = LOG(TERM)
265 TERM1 = TERM1 + (NB(I) + NUB(I) * XI) * TERM
270 TERM2 = TERM2 + NUB(I) * TERM
275 NEXT I
276 DMIXG(JJ, J) = R * T * TERM1
277 IF XI = 0 OR XI = XIMAX THEN DGDXI(JJ, J) = 0: GOTO 280
278 DGDXI(JJ, J) = NUBMUB + R * T * (SUMNUB * LOG(P) + TERM2)
280 GPURE(JJ, J) = NBMUB + XI * NUBMUB
282 GPURE(JJ, J) = GPURE(JJ, J) + R * T * (SUMNB + SUMNUB * XI) * LOG(P)
290 GTOTAL(JJ, J) = GPURE(JJ, J) + DMIXG(JJ, J)
291 IF JJJ = 1 THEN 300
292 IF DGDXI(JJ, J) > 0 THEN XIEQ(JJ) = XI: GEQ(JJ) = GTOTAL(JJ,
J): JJJ = 1
300 IF J <> 10 * KK THEN 315
301 KK = KK + 1
306 PRINT USING "    ##.##"; XI;
    PRINT USING "    ####.##"; GPURE(JJ, J); DMIXG(JJ, J);
    PRINT USING "    ####.##"; GTOTAL(JJ, J); DGDXI(JJ, J)
315 NEXT J
316 PRINT "    "; STRING$(50, "-")
351 PRINT "T="; T; "K,"; "P="; P; "atm"; "  XI at equilibrium =";
352 PRINT USING "##.##"; XIEQ(JJ);
353 KP(JJ) = EXP(-NUBMUB / (R * T))
354 PRINT "  Equilibrium constant=";
    PRINT USING "##.###^^^^"; KP(JJ)
355 PRINT "At equilibrium";
356 FOR I = 1 TO NC
```

```
357 AMOUNT = NB(I) + NUB(I) * XIEQ(JJ)
358 PRINT "      AMOUNT OF "; NAME$(I); "=";
359 PRINT USING "##.####"; AMOUNT; : PRINT " mol";
361 NEXT I: PRINT
363 A$ = INPUT$(1) ': CLS
    PP(JJ) = P: TT(JJ) = T
364 NEXT P: NEXT T: A$ = INPUT$(1)
371 IF PLOT = 0 THEN 690
372 XMIN = 0: XMAX = XIMAX
373 YMIN = GTOTAL(1, 1): YMAX = GTOTAL(1, 1)
374 FOR M1 = 1 TO JJ: FOR M2 = 1 TO 100
376 IF YMIN > GTOTAL(M1, M2) THEN YMIN = GTOTAL(M1, M2)
377 IF YMAX < GTOTAL(M1, M2) THEN YMAX = GTOTAL(M1, M2)
378 NEXT M2: NEXT M1
382 FOR M1 = 1 TO JJ: FOR M2 = 1 TO 100
384 IF YMIN > GPURE(M1, M2) THEN YMIN = GPURE(M1, M2)
386 IF YMAX < GPURE(M1, M2) THEN YMAX = GPURE(M1, M2)
388 NEXT M2: NEXT M1
392 YMIN = YMIN - 1: YMAX = YMAX + 1
435 SCREEN 1: COLOR 15, 0: LOCATE 2, 10: PRINT "0=";
441 FOR M1 = 1 TO NC
442 IF NUB(M1) < 0 THEN PRINT "-";  ELSE PRINT "+";
443 PRINT ABS(NUB(M1)); : PRINT "("; NAME$(M1); ")";
445 NEXT M1
447 LOCATE 3, 1: PRINT USING "###"; YMAX: LOCATE 13, 2: PRINT "G"
449 LOCATE 22, 1: PRINT USING "###"; YMIN
450 LOCATE 23, 5: PRINT USING "#"; XMIN;
451 PRINT "     XI--->     "; : PRINT USING "#"; XMAX
460 VIEW (30, 17)-(155, 170)
462 WINDOW (XMIN, YMIN)-(XMAX, YMAX)
    LINE (XMIN, YMIN)-(XMAX, YMAX), 3, B
469 XS = (XMAX - XMIN) / 10: YS = (YMAX - YMIN) / 30
472 XS1 = XS / 4: YS1 = YS * 3
    FOR IX = 1 TO 9
    X1 = XMIN + IX * XS: Y1 = YMIN + IX * YS * 3
    LINE (X1, YMIN)-(X1, YMIN + YS)
    LINE (XMIN, Y1)-(XMIN + XS1, Y1)
475 NEXT IX: JP = 0
483 FOR J = 1 TO JJ: FOR I = 1 TO 100
486 XI = (I / 100) * XIMAX: YI = GPURE(J, I): ZI = GTOTAL(J, I)
488 PSET (XI, YI), 1: PSET (XI, ZI), 2
492 NEXT I
493 XX = XIEQ(J): YY = GEQ(J): PSET (XX, YY): CIRCLE (XX, YY), .02
498 LINE (XIEQ(J), YMIN)-(XIEQ(J), YMIN + 4 * YS)
```

```
499 JP = JP + 1: LOCATE 16 - 4 * (JP - 1), 22
    PRINT TT(J); "K ;"; : PRINT USING "#.##"; PP(J); : PRINT " atm"
     LOCATE 17 - 4 * (JP - 1), 23
     PRINT "XIEQ="; : PRINT USING "#.##"; XIEQ(J)
     LOCATE 18 - 4 * (JP - 1), 23
     PRINT "Kp="; : PRINT USING "##.###^^^^"; KP(J)
     IF JP = 6 THEN JP = 1
     A$ = INPUT$(1)
500 NEXT J
501 A$ = INPUT$(1): CLS : YMIN = DGDXI(1, 2): YMAX = DGDXI(1, 2)
505 FOR M1 = 1 TO JJ: FOR M2 = 2 TO 99
508 IF YMIN > DGDXI(M1, M2) THEN YMIN = DGDXI(M1, M2)
510 IF YMAX < DGDXI(M1, M2) THEN YMAX = DGDXI(M1, M2)
512 NEXT M2: NEXT M1
516 CLS : YMIN = INT(YMIN): YMAX = INT(YMAX)
530 LOCATE 3, 1: PRINT USING "###."; YMAX
    LOCATE 13, 2: PRINT " 0"
534 LOCATE 10, 1: PRINT " dG": LOCATE 11, 1: PRINT "---"
538 LOCATE 12, 1: PRINT "dXI": LOCATE 22, 1: PRINT USING "###."; YMIN
545 VIEW (30, 17)-(155, 170)
547 WINDOW (XMIN, YMIN)-(XMAX, YMAX)
    LINE (XMIN, YMIN)-(XMAX, YMAX), , B
550 YS = (YMAX - YMIN) / 40
554 FOR IX = 1 TO 9
    X1 = XMIN + IX * XS: Y1 = YMIN + IX * 4 * YS
556 LINE (X1, YMIN)-(X1, YMIN + YS): LINE (XMIN, Y1)-(XMIN + XS1, Y1)
558 NEXT IX
572 FOR J = 1 TO JJ: FOR I = 1 TO 100
576 XI = (I / 100) * XIMAX: YI = DGDXI(J, I)
577 PSET (XI, YI), 2
578 NEXT I
580 LINE (0, 0)-(XIEQ(J), 0), 2
    LINE (XIEQ(J), YMIN)-(XIEQ(J), 0), 2
586 PSET (XIEQ(J), 0): CIRCLE (XIEQ(J), 0), .02
593 NEXT J
595 A$ = INPUT$(1)
602 CLS : XMIN = 0: XMAX = 1
604 YMIN = DMIXG(1, 2): YMAX = DMIXG(1, 2)
606 FOR M1 = 1 TO JJ: FOR M2 = 2 TO 100
610 IF YMIN > DMIXG(M1, M2) THEN YMIN = DMIXG(M1, M2)
612 IF YMAX < DMIXG(M1, M2) THEN YMAX = DMIXG(M1, M2)
```

```
614 NEXT M2: NEXT M1
618 CLS : YMIN = INT(YMIN - 1): YMAX = INT(YMAX + 1)
624 LOCATE 3, 1: PRINT USING "#.#"; YMAX
    LOCATE 5, 10: PRINT "DmixG"
FOR II = 0 TO 3: LOCATE 11 + II, 1: PRINT "     ": NEXT II
652 LOCATE 15, 1: PRINT "    "
    LOCATE 22, 1: PRINT USING "##."; YMIN
654 VIEW (30, 17)-(155, 170)
656 WINDOW (XMIN, YMIN)-(XMAX, YMAX)
    LINE (XMIN, YMIN)-(XMAX, YMAX), , B
660 YS = (YMAX - YMIN) / 40
662 FOR IX = 1 TO 9
    X1 = XMIN + IX * XS: Y1 = YMIN + IX * YS * 4
    LINE (X1, YMIN)-(X1, YMIN + YS)
    LINE (XMIN, Y1)-(XMIN + XS1, Y1)
666 NEXT IX
676 FOR J = 1 TO JJ: FOR I = 1 TO 100
678 XI = (I / 100) * XIMAX: YI = DMIXG(J, I)
682 PSET (XI, YI), J
684 NEXT I: NEXT J
690 END
```

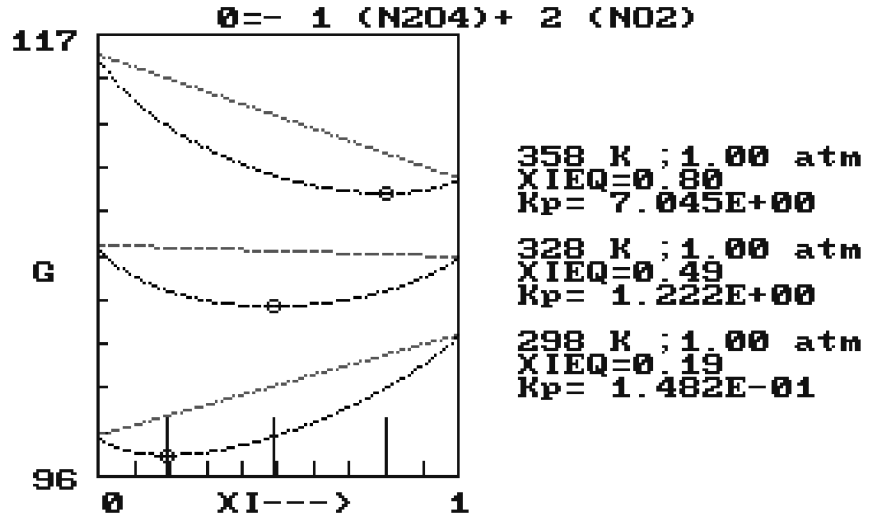
Output

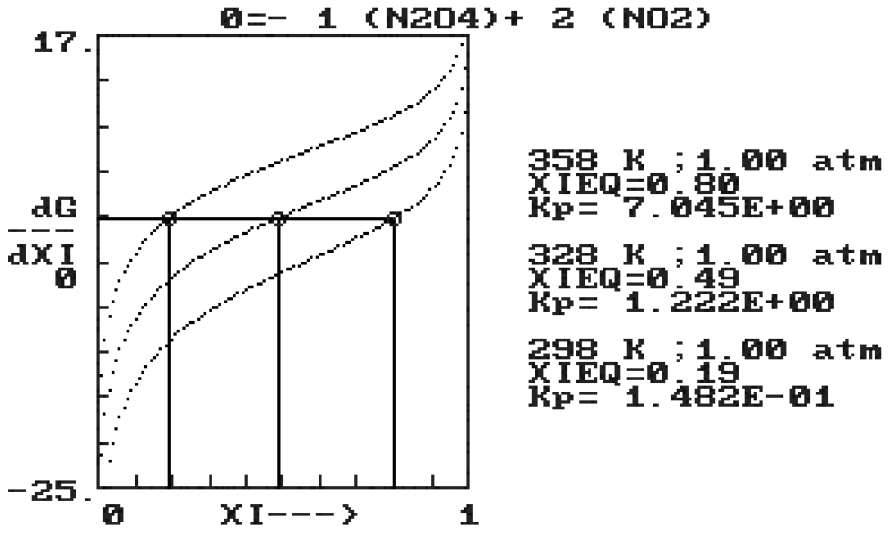
```
Initial temperature in kelvin= 298
Final temperature in kelvin= 358
Temperature inerval in kelvin= 30
Initial pressure in atm= 1
Final pressure in atm= 1
Pressure interval in atm= .15
Number of constituents in the main chemical reaction,NC= 2
PLOT=1 if plot required else 0;PLOT= 1

Name of the FIRST spieces=N2O4
Phase of N2O4=G
Stoichometric number of N2O4=-1
Amount of N2O4 in mole= 1

Name of the SECOND spieces=NO2
Phase of NO2=G
Stoichometric number of NO2= 2
Amount of NO2 in mole= 0

Chemical potential of N2O4 in kJ/mol= 97.89

Chemical potential of NO2 in kJ/mol= 51.31
```

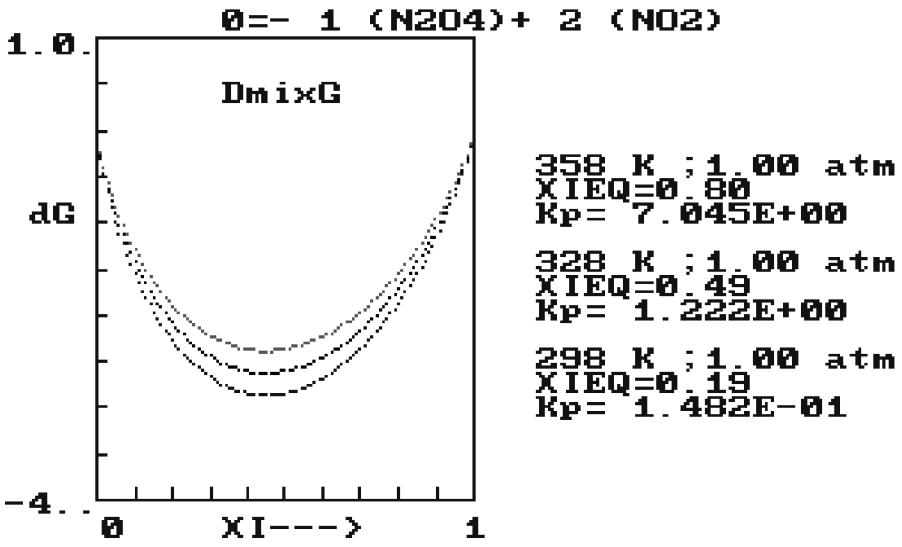
```
Total index number of the constituents including elements whose
 FREE ENERGY is to be computed at different temperatures,TNC= 4

Index number  1  for the constituent N2

Molar heat capacity data in J/(K mol) of N2 without exponent part
a= 29.295     b=-2.2634     c= 5.6649
Molar values of H0 and G0 in kJ/mol for N2
H0= 0         G0= 0

Index number  2  for the constituent O2

Molar heat capacity data in J/(K mol) of O2 without exponent part
a= 26.648     b= 9.55       c=-1.328
Molar values of H0 and G0 in kJ/mol for O2
H0= 0         G0= 0

Index number  2  for the constituent O2

Molar heat capacity data in J/(K mol) of O2 without exponent part
a= 26.648     b= 9.55       c=-1.328
Molar values of H0 and G0 in kJ/mol for O2
H0= 0         G0= 0

Index number  3  for the constituent N2O4

Molar heat capacity data in J/(K mol) of N2O4 without exponent part
a= 36.208     b= 160.98     c=-77.984
Molar values of H0 and G0 in kJ/mol for N2O4
H0= 9.16      G0= 97.89

Index number  4  for the constituent NO2

Molar heat capacity data in J/(K mol) of NO2 without exponent part
a= 25.036     b= 45.55      c=-18.42
Molar values of H0 and G0 in kJ/mol for NO2
H0= 33.18     G0= 51.31

Number of speices in the formation reaction of N2O4 = 3
Index number of the FIRST species in this formation reaction  1
NU of this species=-1

Index number of the SECOND species in this formation reaction  2
NU of this species=-2

Index number of the THIRD species in this formation reaction  3
NU of this species= 1
```

```
      Number of speices in the formation reaction of NO2 = 3
      Index number of the FIRST species in this formation reaction  1
      NU of this species=-.5

      Index number of the SECOND species in this formation reaction  2
      NU of this species=-1

      Index number of the THIRD species in this formation reaction  4
      NU of this species= 1
 298 K        N2            G=-9.411248E-07 kJ/mol
 298 K        O2            G= 1.058777E-06 kJ/mol
 298 K        N2O4          G= 97.88998 kJ/mol
 298 K        NO2           G= 51.31 kJ/mol
   0=- 1 (N2)- 2 (O2)+ 1 (N2O4)     MUB( 1 )= 97.88998 kJ/mol
   0=- .5 (N2)- 1 (O2)+ 1 (NO2)     MUB( 2 )= 51.31 kJ/mol
   ------------------------------------------------
     XI      GPURE      DMIXG    GTOTAL     DGDXI
   ------------------------------------------------
    0.00     97.89       0.00     97.89      0.00
    0.10     98.36      -1.29     97.07     -3.22
    0.20     98.84      -1.89     96.94      0.29
    0.30     99.31      -2.22     97.09      2.43
    0.40     99.78      -2.37     97.41      4.06
    0.50    100.25      -2.37     97.89      5.44
    0.60    100.73      -2.23     98.50      6.74
    0.70    101.20      -1.96     99.24      8.07
    0.80    101.67      -1.56    100.12      9.59
    0.90    102.15      -0.97    101.18     11.76
    1.00    102.62       0.00    102.62      0.00
   ------------------------------------------------
T= 298 K,P= 1 atm  XI at equilibrium = 0.19  Equilibrium constant= 1.482E-01
At equilibrium     AMOUNT OF N2O4= 0.8100 mol     AMOUNT OF NO2= 0.3800 mol

 328 K        N2            G=-4.259098E-02 kJ/mol
 328 K        O2            G=-4.306893E-02 kJ/mol
 328 K        N2O4          G= 106.708 kJ/mol
 328 K        NO2           G= 53.08063 kJ/mol
   0=- 1 (N2)- 2 (O2)+ 1 (N2O4)     MUB( 1 )= 106.8367 kJ/mol
   0=- .5 (N2)- 1 (O2)+ 1 (NO2)     MUB( 2 )= 53.14499 kJ/mol
```

```
 ------------------------------------------------
   XI      GPURE       DMIXG     GTOTAL      DGDXI
 ------------------------------------------------
  0.00     106.84       0.00     106.84       0.00
  0.10     106.78      -1.42     105.36      -9.30
  0.20     106.73      -2.08     104.64      -5.43
  0.30     106.67      -2.45     104.23      -3.08
  0.40     106.62      -2.61     104.01      -1.29
  0.50     106.56      -2.60     103.96       0.24
  0.60     106.51      -2.45     104.06       1.66
  0.70     106.45      -2.16     104.29       3.12
  0.80     106.40      -1.71     104.69       4.80
  0.90     106.34      -1.07     105.28       7.19
  1.00     106.29       0.00     106.29       0.00
 ------------------------------------------------
T= 328 K,P= 1 atm  XI at equilibrium = 0.49  Equilibrium constant= 1.222E+00
At equilibrium     AMOUNT OF N2O4= 0.5100 mol     AMOUNT OF NO2= 0.9800 mol

 358 K         N2             G=-.1653102 kJ/mol
 358 K         O2             G=-.1675607 kJ/mol
 358 K         N2O4           G= 115.3046 kJ/mol
 358 K         NO2            G= 54.7469 kJ/mol
   O=- 1 (N2)- 2 (O2)+ 1 (N2O4)     MUB( 1 )= 115.8051 kJ/mol
   O=- .5 (N2)- 1 (O2)+ 1 (NO2)     MUB( 2 )= 54.99712 kJ/mol
 ------------------------------------------------
   XI      GPURE       DMIXG     GTOTAL      DGDXI
 ------------------------------------------------
  0.00     115.81       0.00     115.81       0.00
  0.10     115.22      -1.55     113.67     -15.36
  0.20     114.64      -2.27     112.37     -11.14
  0.30     114.06      -2.67     111.39      -8.57
  0.40     113.48      -2.85     110.64      -6.62
  0.50     112.90      -2.84     110.06      -4.95
  0.60     112.32      -2.68     109.64      -3.40
  0.70     111.74      -2.36     109.38      -1.80
  0.80     111.16      -1.87     109.29       0.03
  0.90     110.58      -1.17     109.41       2.63
  1.00     109.99       0.00     109.99       0.00
 ------------------------------------------------
T= 358 K,P= 1 atm  XI at equilibrium = 0.80  Equilibrium constant= 7.045E+00
At equilibrium     AMOUNT OF N2O4= 0.2000 mol     AMOUNT OF NO2= 1.6000 mol
```


0=- 1 (N204)+ 2 (NO2)
117
G
96
0
XI--->
1
358 K ;1.00 atm
XIEQ=0.80
Kp= 7.045E+00
328 K ;1.00 atm
XIEQ=0.49
Kp= 1.222E+00
298 K ;1.00 atm
XIEQ=0.19
Kp= 1.482E-01


0=- 1 (N204)+ 2 (NO2)
17.
dG
---
dXI
0
-25.
0
XI--->
1
358 K ;1.00 atm
XIEQ=0.80
Kp= 7.045E+00
328 K ;1.00 atm
XIEQ=0.49
Kp= 1.222E+00
298 K ;1.00 atm
XIEQ=0.19
Kp= 1.482E-01



3.3 CALCULATION OF π-ELECTRON HÜCKEL MOLECULAR ORBITALS

To construct π-electron molecular orbitals for a conjugated molecule by using Hückel approximations. Also calculate its atomic charges and bond orders.

For a conjugated hydrocarbon, the secular determinant as per Hückel approximations includes

(i) α–E as the diagonal element for each carbon atom, where α is the Coulomb integral.

(ii) β as the off-diagonal element for each pair of connected atoms, where β is the exchange integral

(iii) zero as the off-diagonal element for each pair of non-connected atoms.

For example, for butadiene, $\left(\overset{(1)}{\mathrm{CH_2}} = \overset{(2)}{\mathrm{CH}} - \overset{(3)}{\mathrm{CH}} = \overset{(4)}{\mathrm{CH_2}} \right)$ the secular determinant is

$$\begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \end{array} \begin{array}{c} \begin{array}{cccc} (1) & (2) & (3) & (4) \end{array} \\ \begin{vmatrix} \alpha - E & \beta & 0 & 0 \\ \beta & \alpha - E & \beta & 0 \\ 0 & \beta & \alpha - E & \beta \\ 0 & 0 & \beta & \alpha - E \end{vmatrix} \end{array} = 0$$

If each element is divided by β, we get

$$\begin{vmatrix} x & 1 & 0 & 0 \\ 1 & x & 1 & 0 \\ 0 & 1 & x & 1 \\ 0 & 0 & 1 & x \end{vmatrix} = 0 \quad \text{where} \quad x = \frac{\alpha - E}{\beta}$$

Since

$$\underset{\boldsymbol{M}}{\begin{bmatrix} x & 1 & 0 & 0 \\ 1 & x & 1 & 0 \\ 0 & 1 & x & 1 \\ 0 & 0 & 1 & x \end{bmatrix}} = \underset{\boldsymbol{B}}{\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{bmatrix}} + \underset{\boldsymbol{A}}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \tag{1}$$

The matrix $\boldsymbol{A}$ is subjected to orthogonal transformations

$$\boldsymbol{D} = \tilde{\boldsymbol{S}}_n \tilde{\boldsymbol{S}}_{n-1} \ldots \tilde{\boldsymbol{S}}_2 \tilde{\boldsymbol{S}}_1 \boldsymbol{A}\, \boldsymbol{S}_1 \boldsymbol{S}_2 \ldots \boldsymbol{S}_{n-1}\, \boldsymbol{S}_n$$

so as to give the matrix $\boldsymbol{D}$ which includes only diagonal elements with all off diagonal elements equal to zero. The $\tilde{\boldsymbol{S}}$ matrix is the transpose of $\boldsymbol{S}$ and is equal to $\boldsymbol{S}^{-1}$, since $\boldsymbol{S}$ is an orthogonal matrix.

Each orthogonal transformation makes the largest off-diagonal element in the matrix $\boldsymbol{A}$ and in the subsequent matrices equal to zero by selecting proper angle θ in the matrix $\boldsymbol{S}$ given by

$$S = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix}$$

The above procedure of carrying out the diagonalization is known as Jacobi's method. **You are provided with a program which carry out the diagonalization along with the matrix**

$$\boldsymbol{C} = \boldsymbol{S}_1 \boldsymbol{S}_2 \ldots \boldsymbol{S}_{n-1} \boldsymbol{S}$$

which represents the eigen vectors (columnwise) corresponding to each diagonal element. These vectors represent the respective coefficients of atomic orbitals in the various molecular orbitals.

With the diagnonalized matrix, Eq. (1) is written as

$$\boldsymbol{M} = \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{bmatrix} + \begin{bmatrix} y_1 & 0 & 0 & 0 \\ 0 & y_2 & 0 & 0 \\ 0 & 0 & y_3 & 0 \\ 0 & 0 & 0 & y_4 \end{bmatrix} = \begin{bmatrix} x+y_1 & 0 & 0 & 0 \\ 0 & x+y_2 & 0 & 0 \\ 0 & 0 & x+y_3 & 0 \\ 0 & 0 & 0 & x+y_4 \end{bmatrix}$$

The corresponding secular determinant is

$$\begin{vmatrix} x+y_1 & 0 & 0 & 0 \\ 0 & x+y_2 & 0 & 0 \\ 0 & 0 & x+y_3 & 0 \\ 0 & 0 & 0 & x+y_4 \end{vmatrix} = 0$$

This gives

$$x + y_1 = 0 \quad ; \quad x + y_2 = 0$$

$$x + y_3 = 0 \quad \text{and} \quad x + y_4 = 0$$

Since $x = (\alpha - E)/\beta$, the various energies of molecular orbitals are

$$E_1 = x + y_1\beta \quad ; \quad E_2 = x + y_2\beta$$

$$E_3 = x_3 + y_3\beta \quad \text{and} \quad E_4 = x + y_4\beta.$$

Once the eigenvectors (which are columns of the matrix $\boldsymbol{C}(= \boldsymbol{S_1 S_2 \ldots S_{n-1} S_n})$ matrix is known, the atomic charges (q_r) and bond orders (p_{rs}) are evaluated by the expressions

$$q_r = \sum_{i,\text{occ}} n_i C_{ir}^2 \quad \text{and} \quad p_{rs} = \sum_{i,\text{occ}} n_i C_{ir} C_{is}$$

where the summation i is over occupied molecular orbitals, n_i is the occupancy number (of electrons) in the ith molecular orbitals.

For heteroatom, the inteqral α and β are written as

$$\alpha_B = \alpha_C + h_B\beta_{CC} \quad \text{and} \quad \beta_{BC} = k_{BC}\,\beta_{CC}$$

where the recommended values of h_B and k_{BC} are as follows.

Atom B	h_B
$\dot{\text{N}}$	0.5
$\ddot{\text{N}}$	1.5
$\overset{+}{\text{N}}$	2.0
$\dot{\text{O}}$	1.0
$\ddot{\text{O}}$	2.0

Bond, B–C	k_{BC}
C — $\dot{\text{N}}$	0.8
C — $\ddot{\text{N}}$	1.0
N — O	0.7
C — O	0.8
C = O	1.0

For example, for pyridine the secular determinant is

$$\begin{vmatrix} (\alpha+0.5\beta)+E & 0.8\beta & 0 & 0 & 0 & 0.8\beta \\ 0.8\beta & \alpha-E & \beta & 0 & 0 & 0 \\ 0 & \beta & \alpha-E & \beta & 0 & 0 \\ 0 & 0 & \beta & \alpha-E & \beta & 0 \\ 0 & 0 & 0 & \beta & \alpha-E & \beta \\ 0.8\beta & 0 & 0 & 0 & \beta & \alpha-E \end{vmatrix} = 0$$

On dividing by β and letting $x = (\alpha - E)/\beta$, we get

$$\begin{vmatrix} x+0.5 & 0.8 & 0 & 0 & 0 & 0.8 \\ 0.8 & x & 1 & 0 & 0 & 0 \\ 0 & 1 & x & 1 & 0 & 0 \\ 0 & 0 & 1 & x & 1 & 0 \\ 0 & 0 & 0 & 1 & x & 1 \\ 0.8 & 0 & 0 & 0 & 1 & x \end{vmatrix} = 0$$

The matrix to be diagonalized is $\begin{pmatrix} 0.5 & 0.8 & 0 & 0 & 0 & 0.8 \\ 0.8 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0.8 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

For the hyperconjugation effect of methyl group, the effective nuclear charge of carbon to which methyl group is attached is decreased by a value 0.2. Thus, the diagonal element of this carbon is given a value of –0.2.

Illustration Eigenvalues, eigenvectors, bond order and charge densities of butadiene.

Program

```
REM PROGRAM HUCKEL
CLS : DIM A(10, 10), B(10, 10), V(10, 10)
READ NAME$, N, NE: PRINT NAME$: PRINT
FOR I = 1 TO N: FOR J = I TO N
READ A(I, J): A(J, I) = A(I, J): B(I, J) = A(I, J)
NEXT J: NEXT I
PRINT "SECULAR DETERMINANT": PRINT
FOR I = 1 TO N: FOR J = 1 TO N: PRINT USING "    ##.#"; A(I, J);
NEXT J: PRINT : NEXT I
A$ = INPUT$(1)
REM
REM INITIALIZE EIGENVECTOR MATRIX TO UNIT MATRIX
REM
FOR I = 1 TO N: FOR J = 1 TO N
IF I = J THEN V(I, J) = 1 ELSE V(I, J) = 0
NEXT J: NEXT I
REM
REM SCAN FOR LARGEST OFF-DIAGONAL ELEMENT
REM
15 XMAX = 0
FOR I = 1 TO N - 1: FOR J = I + 1 TO N
IF XMAX < ABS(A(I, J)) THEN XMAX = ABS(A(I, J)): IP = I: JP = J
NEXT J: NEXT
REM
```

```
REM TEST FOR TOLERANCE
REM
EPS = 1E-08
IF XMAX <= EPS THEN 75
REM
REM COMPUTE TAN,COS,SIN,A(I,I) and A(J,J)
REM
DENOM = ABS(A(IP, IP) - A(JP, JP))
TERM = SQR((A(IP, IP) - A(JP, JP)) ^ 2 + 4 * A(IP, JP) ^ 2)
DENOM = DENOM + TERM
IF A(IP, IP) >= A(JP, JP) THEN 27
TANG = -2 * A(IP, JP) / DENOM
GOTO 30
27 TANG = 2 * A(IP, JP) / DENOM
30 COSN = 1 / SQR(1 + TANG ^ 2)
SINE = TANG * COSN: AII = A(IP, IP)
TERM = AII + TANG * (2 * A(IP, JP) + TANG * A(JP, JP))
A(IP, IP) = COSN ^ 2 * TERM
TERM = A(JP, JP) - TANG * (2 * A(IP, JP) - TANG * AII)
A(JP, JP) = COSN ^ 2 * TERM
A(IP, JP) = 0
IF A(IP, IP) >= A(JP, JP) THEN 36
SWAP A(IP, IP), A(JP, JP)
REM
REM ADJUST SIN,COS FOR COMPUTATION OF A(I,K) AND V(I,K)
REM
IF SINE >= 0 THEN 34
TEMP = COSN
GOTO 35
34 TEMP = -COSN
35 COSN = ABS(SINE)
SINE = TEMP
36 FOR I = 1 TO N
IF I = IP THEN 50
IF I > IP THEN 42
TEMP = A(I, IP)
A(I, IP) = COSN * TEMP + SINE * A(I, JP)
A(I, JP) = -SINE * TEMP + COSN * A(I, JP)
GOTO 50
42 IF I = JP THEN 50
IF I > JP THEN 48
TEMP = A(IP, I)
A(IP, I) = COSN * TEMP + SINE * A(I, JP)
```

```
A(I, JP) = -SINE * TEMP + COSN * A(I, JP)
GOTO 50
48 TEMP = A(IP, I)
A(IP, I) = COSN * TEMP + SINE * A(JP, I)
A(JP, I) = -SINE * TEMP + COSN * A(JP, I)
50 NEXT I
REM
REM COMPUTE EIGENVECTORS
REM
FOR I = 1 TO N
TEMP = V(I, IP)
V(I, IP) = COSN * TEMP + SINE * V(I, JP)
V(I, JP) = -SINE * TEMP + COSN * V(I, JP)
NEXT I
GOTO 15
REM
REM ARRANGE EIGENVALUES IN ASCENDING ORDER
REM
75 FOR I = 1 TO N: X = A(I, I): K = I
FOR M = I TO N
IF A(M, M) <= X THEN 85
X = A(M, M): K = M
85 NEXT M
A(K, K) = A(I, I): A(I, I) = X
FOR L = 1 TO N
SWAP V(L, K), V(L, I)
NEXT L: NEXT I
REM
REM CHAGE SIGN IF FIRST EIGENVECTOR IS NEGATIVE
REM
FOR I = 1 TO N
K = 1
IF V(K, I) >= O THEN 92
FOR J = 1 TO N
V(J, I) = -V(J, I)
92 NEXT J
NEXT I
PRINT : PRINT "EIGEN VALUES": PRINT
FOR I = 1 TO N: FOR J = 1 TO N: A(J, I) = A(I, J)
PRINT USING "   ###.###"; A(I, J);
NEXT J: PRINT : NEXT I: A$ = INPUT$(1)
PRINT : PRINT "EIGEN VECTORS": PRINT
FOR I = 1 TO N: FOR J = 1 TO N
```

```
PRINT USING "    ##.####"; V(I, J);
NEXT J: PRINT : NEXT I: PRINT : A$ = INPUT$(1)
PRINT "ELECTRON DENSITIES": PRINT
NOC = INT(NE / 2)
IF INT(NE / 2) * 2 = NE THEN KK = 1
FOR I = 1 TO N
ED = 0
FOR J = 1 TO NOC
ED = ED + 2 * V(I, J) ^ 2
NEXT J
IF KK = 1 THEN 100
ED = ED + V(I, J) ^ 2
100 PRINT "    ATOM("; I; ")="; : PRINT USING "##.###"; ED
A$ = INPUT$(1)
NEXT I: PRINT
PRINT "BOND ORDERS": PRINT
FOR I = 1 TO N - 1
FOR J = I + 1 TO N
BO = 0
IF B(I, J) = 0 THEN 160
FOR K = 1 TO NOC
BO = BO + 2 * V(I, K) * V(J, K)
NEXT K
IF KK = 1 THEN 150
BO = BO + V(I, K) * V(J, K)
150 PRINT "    ATOMS("; I; ","; J; ")=";
PRINT USING "##.###"; BO
A$ = INPUT$(1)
160 NEXT J
NEXT I
'DATA BUTADIENE,4,4
'DATA 0,1,0,0,0,1,0,0,1,0
DATA BENZENE,6,6
DATA 0,1,0,0,0,1,0,1,0,0,0,0,1,0,0,0,1,0,0,1,0
'DATA ALLYL RADICAL,3,3
'DATA 0,1,0,0,1,0
'DATA PYRIDINE,6,6
'DATA .5,.8,0,0,0,.8,0,1,0,0,0,0,1,0,0,0,1,0,0,1,0
'DATA ANILINE,7,8
'DATA 0,1,0,0,0,1,1,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,1.5
'DATA ANILINIUM ION,7,7
'DATA 0,1,0,0,0,1,.8,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,2
END
```

Output

```
BENZENE

SECULAR DETERMINANT

    0.0    1.0    0.0    0.0    0.0    1.0
    1.0    0.0    1.0    0.0    0.0    0.0
    0.0    1.0    0.0    1.0    0.0    0.0
    0.0    0.0    1.0    0.0    1.0    0.0
    0.0    0.0    0.0    1.0    0.0    1.0
    1.0    0.0    0.0    0.0    1.0    0.0

EIGEN VALUES

    2.000    0.000   -0.000   -0.000    0.000   -0.000
    0.000    1.000    0.000   -0.000   -0.000   -0.000
   -0.000    0.000    1.000    0.000   -0.000   -0.000
   -0.000   -0.000    0.000   -1.000   -0.000   -0.000
    0.000   -0.000   -0.000   -0.000   -1.000   -0.000
   -0.000   -0.000   -0.000   -0.000   -0.000   -2.000

EIGEN VECTORS

    0.4082    0.0556    0.5747    0.5774    0.0000    0.4082
    0.4082   -0.4699    0.3355   -0.2886   -0.5000   -0.4082
    0.4082   -0.5255   -0.2391   -0.2887    0.5000    0.4082
    0.4082   -0.0556   -0.5747    0.5774    0.0000   -0.4082
    0.4082    0.4699   -0.3355   -0.2886   -0.5000    0.4082
    0.4082    0.5255    0.2391   -0.2887    0.5000   -0.4082

ELECTRON DENSITIES

   ATOM( 1 )= 1.000
   ATOM( 2 )= 1.000
   ATOM( 3 )= 1.000
   ATOM( 4 )= 1.000
   ATOM( 5 )= 1.000
   ATOM( 6 )= 1.000

BOND ORDERS

   ATOMS( 1 , 2 )= 0.667
   ATOMS( 1 , 6 )= 0.667
   ATOMS( 2 , 3 )= 0.667
   ATOMS( 3 , 4 )= 0.667
   ATOMS( 4 , 5 )= 0.667
   ATOMS( 5 , 6 )= 0.667
```

3.4 EUTECTIC PHASE DIAGRAM VIA FREE ENERGY CONSIDERATIONS

In this project, the basic principle in the formation of eutectic phase diagram is highlighted.

For a binary system involving separation of pure components from ideal liquid solution, the criterion of equilibrium gives

$$\mu^*_{A(s)} = \mu_{A(l)} \quad \text{and} \quad \mu^*_{B(s)} = \mu_{B(l)} \tag{1}$$

These equations may be solved separately for the equilibrium compositions $x_{A(l)}$ and $x_{B(l)}$ at which solid components A and B exist in equilibrium with the corresponding components in the liquid phase. The compositions $x_{A(l)}$ and $x_{B(l)}$ may be evaluated either analytically or graphically as described in the following.

The free energy difference between pure solid phase and pure liquid phase of component i(where i = A or B) is given by

$$\begin{aligned}\Delta_{fus}G_i &= \Delta_{fus}H_i - T\,\Delta_{fus}S_i = \Delta_{fus}\,H_i - T(\Delta_{fus}H_i/T_i^*) \\ &= \Delta_{fus}H_i(1 - T/T_i^*)\end{aligned} \tag{2}$$

While replacing $\Delta_{fus}S_i$ by $\Delta_{fus}H_i/T_i^*$, it is assumed that $C_{p,i(s)} = C_{p,i(l)}$ so that $\Delta_{fus}H_i$ may be treated as independent of temperature.

For 1 mol of each of the two components, we can write

$$\Delta_{fus}G_i \equiv \Delta_{fus}\,\mu_i = \mu_{i(l)} - \mu_{i(s)} \tag{3}$$

On a graph between G_m versus x_B, the values of $\mu_{i(s)}$ and $\mu_{i(l)}$ may be depicted relative to each other by setting

$$\mu_{i(l)} = 0 \tag{4a}$$

and

$$\mu_{i(s)} = -\Delta_{fus}H(1 - T/T_i^*) \tag{4b}$$

In view of Eq. (4a), the free energy of liquid phase will be equal to the free energy of mixing of the two components. Hence

$$G_l = \Delta_{mix}G = RT\,[x_{A(l)} \ln x_{A(l)} + x_{B(l)} \ln x_{B(l)}] \tag{5}$$

The free energy of solid mixture will be given by

$$G_s = x_{A(s)}G_{A(s)} + x_{B(s)}G_{B(s)} \tag{6}$$

In Eqs (5) and (6), the symbol x represents amount fraction satisfying the expressions

$$x_{A(l)} + x_{B(l)} = 1 \quad \text{and} \quad x_{A(s)} + x_{B(s)} = 1 \tag{7}$$

The plots of G_l versus $x_{B(l)}$ and G_s with $x_{B(s)}$ will be like those in shown Fig. 1.

Increase in temperature causes $\mu_{i(s)}$ to become less negative (i.e. its value increases, Eq. 4b) and G_l to become more negative (i.e. its value decreases, Eq. 5). Consequently, the graps of G_l versus $x_{B(l)}$ and G_s versus $x_{B(s)}$ move in the opposite directions.

If a tangent line is drawn from a point on the G_l versus $x_{B(l)}$ curve, its intercepts on the ordinate axes give the partial molar free energies of the two components in the liquid phase at that point. Conversely, if a tangent line is drawn from the $G_{A(s)}$ on the ordinate axis of A to the G_l versus $x_{B(l)}$ curve, the point P (see, Fig. 2) on the liquid curve is obtained where the condition

$$G_{A(s)} = \mu_{A(l)} \tag{8}$$

holds good and thus at this point, solid A will exist in equilibrium with the liquid phase.

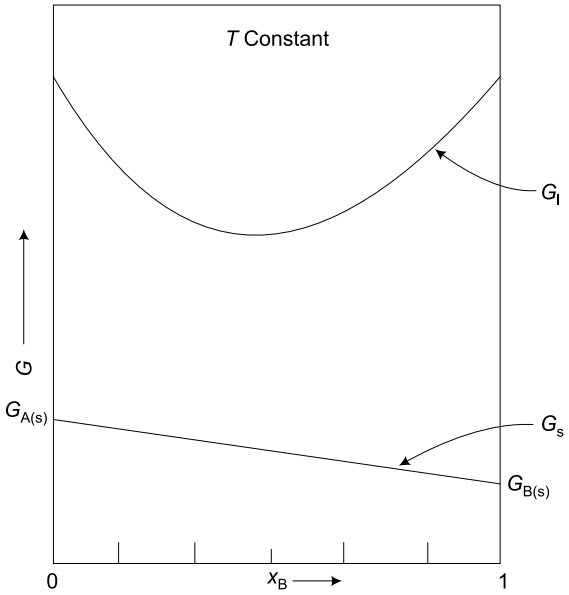


Fig. 1 The plots of G_l versus $x_{B(l)}$ and G_s versus $x_{B(s)}$

The composition of the point P in Fig. 2 may also be worked out by substituting the expressions of $G_{A(s)}$ and $\mu_{A(l)}$ in Eq. (8), which gives

$$-\Delta_{fus}H_A\left(1-\frac{T}{T_A^*}\right) = RT\ln x_{A(l)} \quad \text{or} \quad x_{A(l)} = \exp\left[-\frac{\Delta_{fus}H_A}{RT}\left(1-\frac{T}{T_A^*}\right)\right] \tag{9}$$

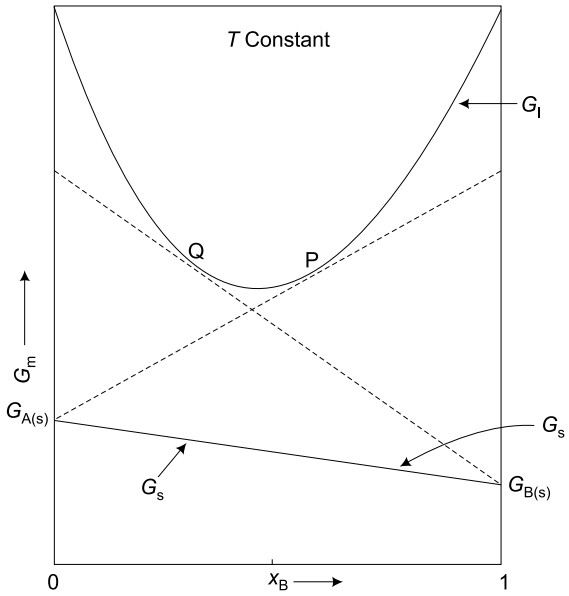


Fig. 2 Tangent lines from $G_{A(s)}$ and $G_{B(s)}$ on the ordinate axes to G_l versus $x_{B(l)}$ curve

Similarly, if a tangent line is drawn from the $G_{B(s)}$ on the ordinate axis of B to the G_l versus $x_{B(l)}$ curve, the point Q on the liquid curve is obtained where the condition.

$$G_{B(s)} = \mu_{B(l)} \tag{10}$$

holds good and thus at this point, solid B will exist in equilibrium with the liquid phase. The composition of the point Q may also be worked out by substituting the expressions of $G_{B(s)}$ and $\mu_{B(l)}$ in Eq. (8), which gives

$$-\Delta_{fus}H_B\left(1-\frac{T}{T_B^*}\right) = RT \ln x_{B(l)} \quad \text{or} \quad x_{B(l)} = \exp\left[-\frac{\Delta_{fus}H_B}{RT}\left(1-\frac{T}{T_B^*}\right)\right] \tag{11}$$

The tangent line drawn from $G_{A(s)}$ (or $G_{B(s)}$) on the ordinate axis A (or B) to the G_l versus $x_{B(l)}$ curve may meet the ordinate axis B(or A) at a point higher than or equal to or lower than the value of $G_{B(s)}$ (or $G_{A(s)}$).

A brief discussion of these three alternatives for a tangent line drawn from $G_{A(s)}$ is in order.

Case 1 When the point lies above $G_{B(s)}$

In this case, the chemical potential of B in liquid phase at P (see, Fig. 3) is greater than its molar Gibbs free energy in the solid phase.

Since $\mu_{B(l)} > G_{B(s)}$, the component B in liquid phase at the point P does not represent a stable system. In order to attain stability, solid B starts separating from the liquid phase. This results into the shifting of the liquid composition towards the ordinate axis A along the liquid curve. At the new composition point of the liquid phase, if a tangent line is drawn it results into $\mu_{B(l)}$ more near to $G_{B(s)}$ but at the same time $\mu_{A(l)}$ becomes greater than $G_{A(s)}$. This, in turn, results into the separation of solid phase A until the composition of liquid phase returns back to the point P. Thus, the separation of solid B is accompanied with the separation of solid A and the separated solid mixture has a composition identical to that of the liquid phase. This separation of solid mixture is continued with the free energy of the system moving along PR (i.e. a vertical line corresponding to the constant composition of the liquid phase at P (see, Fig. 4).

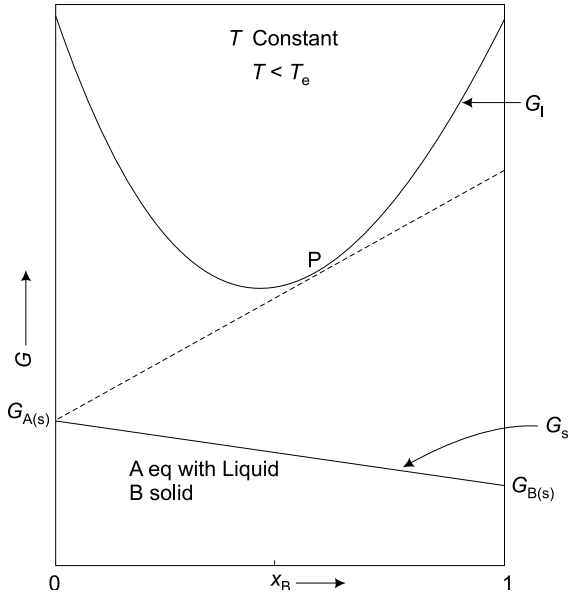


Fig. 3 A case where $m^*_{B(l)} > G_{B(s)}$

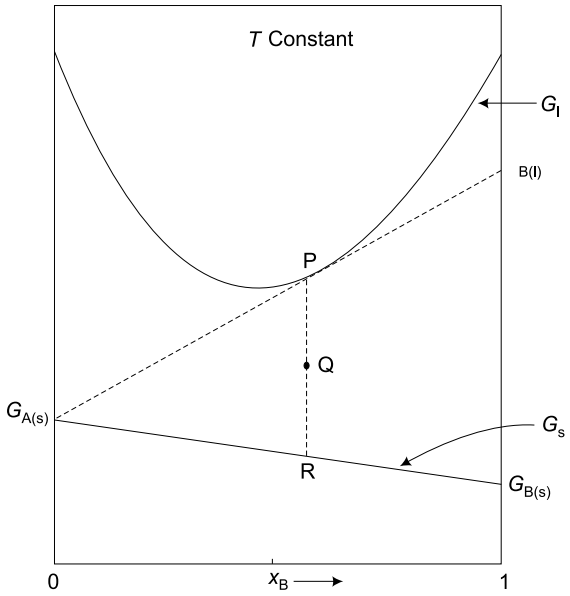


Fig. 4 Separation of solid mixture from the liquid solution at P

The fraction of solid mixture separated at any instant may be determined as follows.

Let y be the fraction of solid mixture separate at any instant. And let G_Q be the free erergy of the system at this instant. It follows that

$$G_Q = yG_R + (1 - y)G_p$$

This gives $\quad y = \dfrac{G_Q - G_P}{G_R - G_P} \quad$ and $\quad 1 - y = \dfrac{G_R - G_Q}{G_R - G_P}$

Thus $$\frac{\text{Fraction of solid separated}}{\text{Fraction of liquid remaining}} = \frac{y}{1-y} = \frac{G_Q - G_P}{G_R - G_Q} = \frac{PQ}{QR} \tag{12}$$

Equation (12) is the *lever rule*,

The above analysis of separation of both the solid components from the liquid mixture at the point P until the point R is reached also follows from the fact that G_{mix}(solid) at the point R is lesser than G_{mix}(liquid) at the point P. The temperature of the system corresponds to a temperature lower than the eutectic temperature of the system.

Case 2 When the point lies at $G_{B(s)}$

In this case, the chemical potential of B in the liquid phase is equal to its molar Gibbs free energy in the solid phase (see, Fig. 5)

Since at the point P

$$\mu_{A(l)} = G_{A(s)} \quad \text{and} \quad \mu_{B(l)} = G_{B(s)} \tag{13}$$

both the components in the solid phase are in equilibrium with the corresponding components in the liquid phase. The temperature of the system corresponds to the eutectic temperature of the system.

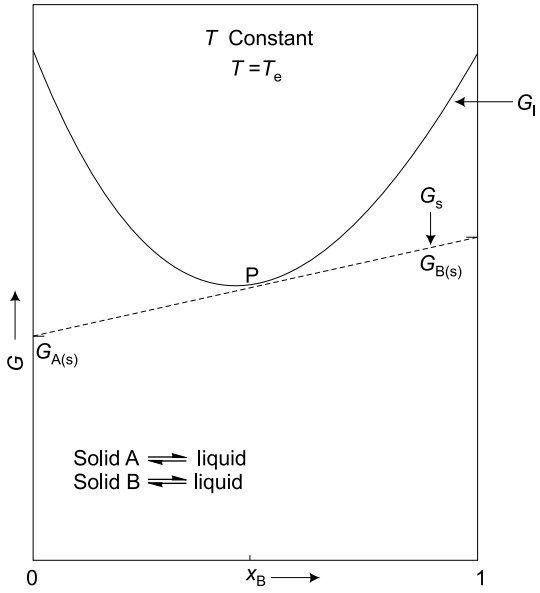


Fig. 5 A case where $\mu_{B(l)} = G_{B(s)}$

Case 3 When the point lies below $G_{B(s)}$

In this case, the chemical potential of B in the liquid phase at the point P is lower than its Gibbs free energy in the solid phase (see, Fig. 6)

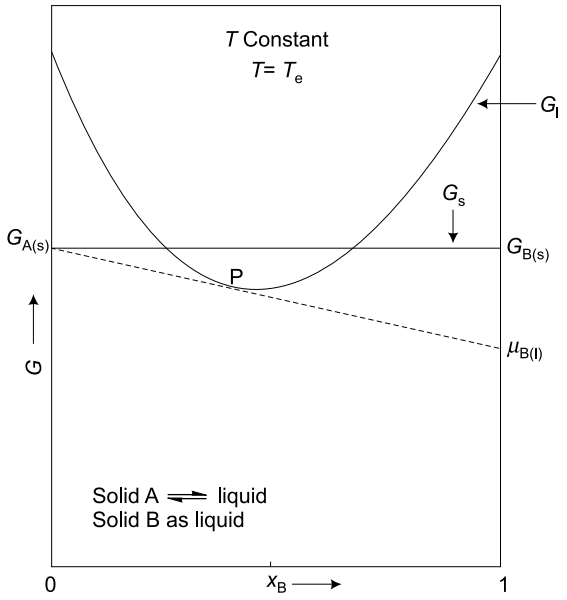


Fig. 6 A case where $\mu_{B(l)} < G_{B(s)}$

The component B remains in the liquid phase. In fact, at the point P, the system exists as the liquid phase with solid A just in equilibrium with the liquid phase. This also follows from the fact that G_{mix}(liquid) at the point P is lesser than G_{mix}(solid) which lies on the line joining $G_{A(s)}$ and $G_{B(s)}$ in Fig. 6. The temperature of the system corresponds to a temperature higher than the eutectic temperature of the system.

Separation of a Component From a Liquid of Composition Other than Equilibrium Composition

Consider a system shown by the point Q in Fig. 7 This system contains larger amount of A as compared that present in the equilibrium composition of A (represented by the point P).

The tangent line at Q gives $\mu_{A(l)} > G_{A(s)}$ $(= (\mu_{A(l)})_{eq})$. Thus, the component A will exhibit solidification from the solution Q. The solidification of A causes the shift in composition of liquid towards P along the liquid curve. This also causes the shift of $\mu_{A(l)}$ towards $G_{A(s)}$. The solidification of A is continued until the composition of liquid reaches to the point P.

Let n_A and n_B be the amounts of A and B, respectively, at the point Q. If a is the amount of solid A separated on reaching the point P, we will have

$$x_p = \frac{n_A - a}{(n_A - a) + n_B} \tag{14}$$

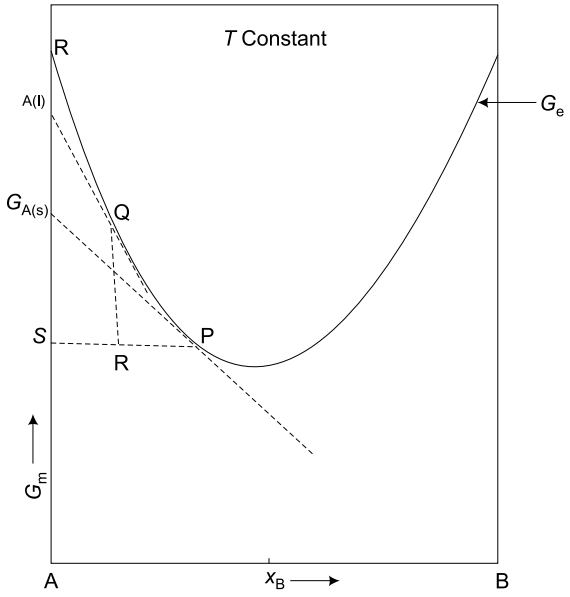


Fig. 7 Separation of solid A from liquid solution of composition Q

Knowing n_A, n_B and x_P, the value of a may be computed from the above expression. Alternatively, one may use the lever rule along the line SP. We have

The amount fractions of B at the points Q and P are

$$x_Q = \frac{n_B}{n_A + n_B} \tag{15}$$

and
$$x_p = \frac{n_B}{(n_A - a) + n_B} \tag{16}$$

where n_A and n_B are the amounts of A and B at Q and a is the amount of A separated on reaching from Q to P. From Eq. (16), we get

$$a = (n_A + n_B) - \frac{n_B}{x_P}$$

or
$$\frac{a}{n_A + n_B} = 1 - \frac{n_B}{n_A + n_B}\frac{1}{x_P}$$

Hence
$$\frac{a}{n_A + n_B} = 1 - \frac{x_Q}{x_P} = \frac{x_P - x_Q}{x_P} = \frac{PR}{PS} \tag{17}$$

Thus, the amount of A precipitated is

$$a = \left(\frac{PR}{PS}\right)(n_A + n_B) \tag{18}$$

Eutectic Phase Diagram

From the plots of the type shown in Fig. 2, the compositions P and Q of the liquid phase at which the solid A and B, respectively, are in equilibrium with the liquid phase at different temperatures are determined. If a plot is made between T and $x_{A(l)}$, and between T and $x_{B(l)}$, one gets the eutectic phase diagram as shown in Fig. 8. The three cases discussed above are also marked in Fig. 8.

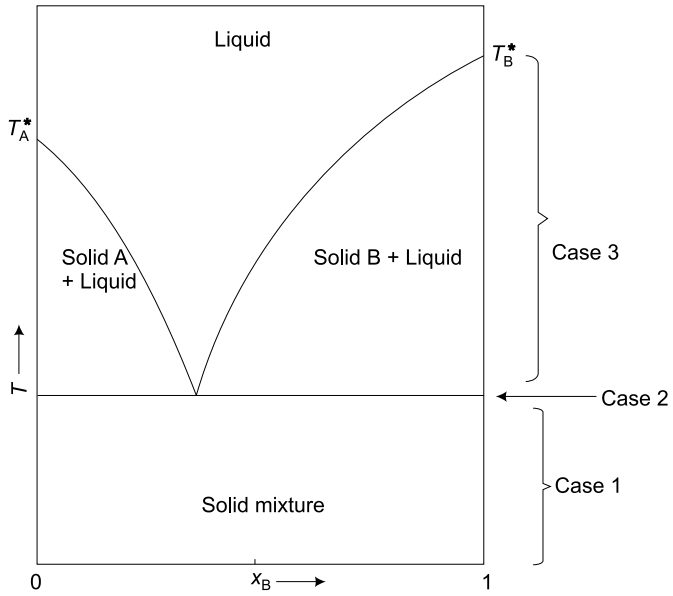


Fig. 8 Eutectic phase diagram

Illustration

Table 1 shows the computed values of various parameters for a system having the following data.

$H_A = \Delta_{fus}H_A = 17\,304$ J mol^{-1} $\quad T_A^* = 395$ K

$H_B = \Delta_{fus}H_B = 22\,602$ J mol^{-1} $\quad T_B^* = 406$ K

The various expressions to be used are as follows.

$G_{A(s)} = -H_A(1 - T/T_A^*)$; $G_{B(s)} = -H_B(1 - T/T_B^*)$

x_B (= $1 - x_A$) at which $G_{A(s)} = \mu_{A(l)}$; $x_A = \exp(G_A/RT)$

x_B at which $G_{B(s)} = \mu_{B(l)}$; $x_B = \exp(G_B/RT)$

$\mu_{A(l)} = RT \ln x_A$; $\mu_{B(l)} = RT \ln x_B$

$G_{mix(l)} = RT\,(x_A \ln x_A + x_B \ln x_B)$; $G_{mix(s)} = x_A G_{A(s)} + x_B G_{B(s)}$

Table 1 Computed values of the given system at different typical temperatures

$T = 341$ K	$\frac{G_{A(s)}}{\text{J mol}^{-1}}$	x_B (= $1 - x_A$) at which $G_{A(s)} = \mu_{A(l)}$	$\frac{\mu_{B(1)}}{\text{J mol}^{-1}}$	$\frac{G_{mix(1)}}{\text{J mol}^{-1}}$	$\frac{G_{mix(s)}}{\text{J mol}^{-1}}$	comments
	–2366	0.57	–1614	–1937	–3080	$G_{A(s)} < \mu_{A(l)}$ $G_{B(s)} < \mu_{B(l)}$ $G_{mix(s)} < G_{mix(l)}$
Case 1	$\frac{G_{B(s)}}{\text{J mol}^{-1}}$	x_B at which $G_{B(s)} = \mu_{B(l)}$	$\frac{\mu_{A(1)}}{\text{J mol}^{-1}}$	$\frac{G_{mix(1)}}{\text{J mol}^{-1}}$	$\frac{G_{mix(s)}}{\text{J mol}^{-1}}$	System exists as solid mixture
	–3619	0.28	–931	–1681	–2717	
$T = 349$ K	–2015	0.50	–2007	–2011	–2594	Same as above
Case 1	–3173	0.34	–1184	–1860	–2409	
T=358.69 K	–1590.7	0.413	–2633.7	–2021.7	–2021.3	$G_{A(s)} = \mu_{A(l)}$ $G_{B(s)} = \mu_{B(l)}$
Case 2	–2633.7	0.413	–1590.7	–2021.7	–2021.3	$G_{mix(s)} = G_{mix(l)}$ Both solids are in equilibrium with liquid mixture
$T = 365$ K	–1314	0.35	–3173	–1965	–1653	$\mu_{A(l)} < G_{A(s)}$ $\mu_{B(l)} < G_{B(s)}$
Case 3	–2282	0.47	–1934	–2098	–1769	$G_{mix(l)} < G_{mix(s)}$ System exists as liquid mixture.
$T = 373$ K	–964	0.27	–4094	–1809	–1200	
Case 3	–1837	0.55	–2497	–2134	–1444	Same as above

Determination of Eutectic Temperature and Composition

At the eutectic point, both the solids are in equilibrium with the liquid phase. We will have

$$G_A = RT \ln x_A \quad \Rightarrow \quad x_A = \exp(G_A/RT) \tag{19}$$

$$G_B = RT \ln x_B \quad \Rightarrow \quad x_B = \exp(G_B/RT) \tag{20}$$

More over, $x_A + x_B = 1$. Hence

$$\exp(G_A/RT) = 1 - \exp(G_B/RT) \tag{21}$$

where $G_A = -H_A(1 - T/T_A^*)$ and $G_B = -H_B(1 - T/T_B^*)$

We evaluate the left-hand and right-hand sides of Eq. (21) for the decreasing value of T starting from the minimum value of T_A^* and T_B^*. The value of T giving identical values of left-hand and right-hand sides is the eutectic temperature (see, Table 2). Knowing the value of eutectic temperature, the value of x_B(or x_A) may be computed by using Eqs (19) or (Eq. 20).

Table 2 Evaluation of Left-hand Side (LHS) and Right-hand Side (RHS) of Eq.(21) at Different Temperatures

T/K	G_B/J mol^{-1}	G_A/J mol^{-1}	*RHS*	*LHS*
385	–1 169	–438	0.694	0.128
375	–1 726	–876	0.575	0.245
365	–2 282	–1 314	0.471	0.351
355	–2 839	–1 752	0.382	0.448
363	–2 393	–1 402	0.453	0.389
361	–2 505	–1 489	0.434	0.391
359	–2 616	–1 577	0.416	0.410
357	–2 728	–1 665	0.399	0.429
358	–2 672	–1 621	0.407	0.420
358.8	–2 628	–1 586	0.414	0.412
358.6	–2 639	–1 595	0.413	0.414
358.7	–2 633	–1 590	0.414	0.413

Taking $T_e = 358.6$ K, we get

$$x_B = \exp(G_B/RT_e) = \exp\{-2639/(8.314 \times 358.6)\} = 0.413.$$

Alternatively, we may determine firstly the eutectic composition followed by eutectic temperature as shown in the following.

We have

$$G_{A(s)} = \mu_{A(l)}$$

i.e.
$$-\mu_A\left(1 - \frac{T}{T_A^*}\right) = RT \ln x_A \quad \text{or} \quad \frac{1}{T} - \frac{1}{T_A^*} = -\frac{R}{H_A} \ln x_A \tag{22}$$

Similarly, from $G_{B(s)} = \mu_{B(l)}$, we get

$$\frac{1}{T} - \frac{1}{T_B^*} = -\frac{R}{H_A} \ln x_B \tag{23}$$

Subtracting Eq. (22) from Eq. (23), we get

$$\frac{1}{T_A^*} - \frac{1}{T_B^*} = \frac{R}{H_A} \ln x_A - \frac{R}{H_B} \ln x_B$$

or, $$\frac{H_B(T_B^* - T_A^*)}{R\,T_A^* T_B^*} = \frac{H_B}{H_A} \ln x_A - \ln x_B \quad \text{or} \quad x_A^{(H_B/H_A)} = x_B \exp\left[\frac{H_B(T_B^* - T_A^*)}{R\,T_A^* T_B^*}\right]$$

or $$(1 - x_B)^{(H_B/H_A)} = x_B \exp\left[\frac{H_B(T_B^* - T_A^*)}{R\,T_A^* T_B^*}\right] \tag{24}$$

The left-hand side and right-hand sides of the above expression are evaluated for increasing value of x_B. The value of x_B giving the identical values gives the eutectic composition. Substitution of this value in Eq. (22) gives the value of eutectic temperature.

Table 3

x_B	*LHS*	*RHS*
0.1	0.8714	0.1205
0.2	0.7472	0.2410
0.3	0.6276	0.3615
0.4	0.513	0.4820
0.5	0.404	0.6025
0.41	0.502	0.494
0.42	0.491	0.5061
0.411	0.5009	0.4952
0.412	0.4998	0.4965
0.413	0.4986	0.4977
0.414	0.4975	0.4989

Let $x_e = 0.413$. Substituting this value in Eq. (23), we get

$T_e = 358.6$ K

Illustration Eutectic phase diagram for a system of A ($\Delta_{fus} H_A^* = 17304$ J mol^{-1}, $T_A^* = 395$ K) and B($\Delta_{fus} H_B^* = 22602$ J mol^{-1} and $T_B^* = 406$ K).

Program

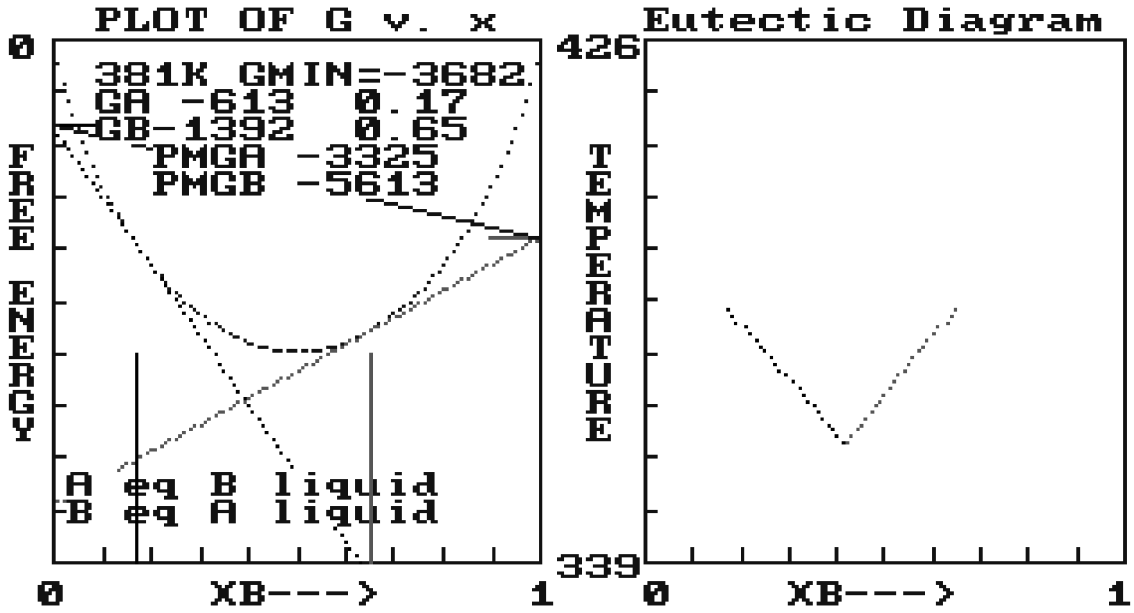
```
CLS
REM PROGRAM EUTECTCC;EUTECTIC DIAGRAM VIA FREE ENERGY
CLS : DEF FNA (X) = (1 - X) * LOG(1 - X) + X * LOG(X)
R = 8.314: HA = 17304: HB = 22602: TA = 395: TB = 406: TS = 1
```

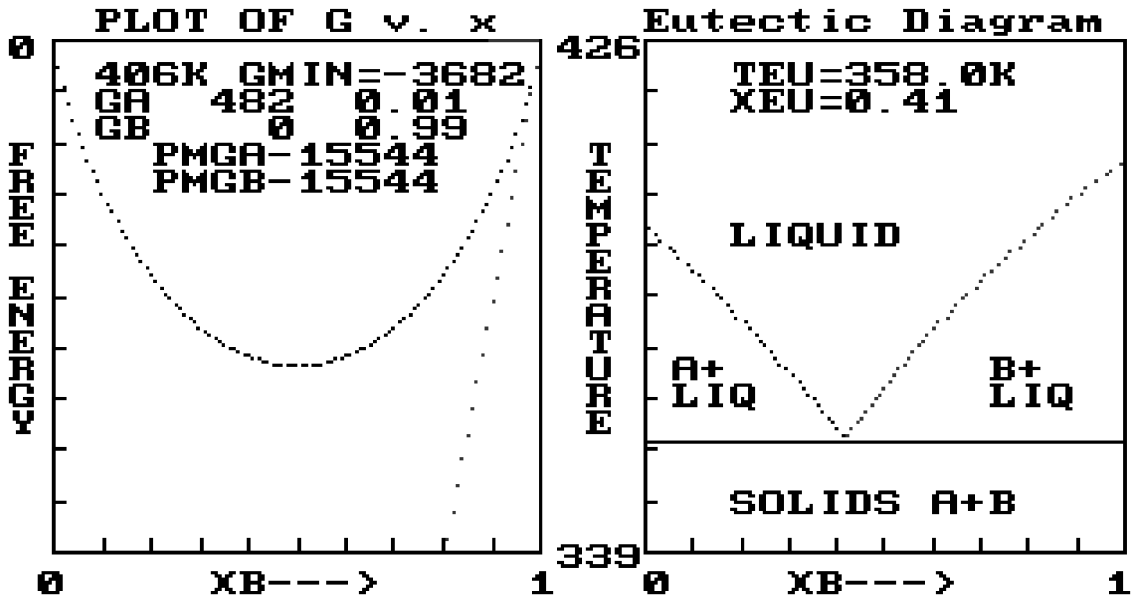
```
IF TA > TB THEN TF = TA + 20 ELSE TF = TB + 20
JJ = 0
FOR T = 200 TO TF STEP TS
LS = EXP((-HA / R) * (1 / T - 1 / TA))
RS = 1 - EXP((-HB / R) * (1 / T - 1 / TB))
IF JJ = 0 THEN MIN = LS: MAX = RS
IF MIN > MAX THEN 1
IF LS > RS THEN 4
GOTO 2
1 IF LS < RS THEN 4
2 JJ = 1
NEXT T
4 TI = T - 20
GGI = 1.5 * R * TF * LOG(.5)
SCREEN 1:  COLOR 15, 0
LOCATE 2, 5: PRINT "PLOT OF G v. x"
LOCATE 2, 24: PRINT "Eutectic Diagram"
LOCATE 3, 21: PRINT USING "###"; TF
LOCATE 3, 2: PRINT "0": A$ = "FREE ENERGY": NN = LEN(A$)
LOCATE 22, 21: PRINT USING "###"; TI
FOR I = 1 TO NN: LOCATE 6 + I, 2: PRINT MID$(A$, I, 1): NEXT I
A$ = "TEMPERATURE": NN = LEN(A$)
FOR I = 1 TO NN: LOCATE 6 + I, 22: PRINT MID$(A$, I, 1): NEXT I
LOCATE 23, 2: PRINT " 0      XB--->      1"
FOR T = TI TO TF STEP TS
VIEW (20, 17)-(155, 170)
IF T > TA AND T > TB THEN 19
CLS : KKK = 0: KKKK = 0
WINDOW (0, GGI)-(1, 0): LINE (0, GGI)-(1, 0), , B
FOR I = 1 TO 9
LINE (I * .1, GGI)-(I * .1, GGI + 100)
LINE (0, GGI * I * .1)-(.02, GGI * I * .1)
NEXT I
IF T > TA AND T > TB THEN 19
GA = -HA * (1 - T / TA): GB = -HB * (1 - T / TB)
FOR X2 = .01 TO .99 STEP .01
PMG = R * T * LOG(X2): MIXGL = FNA(X2) * R * T
PSET (X2, MIXGL)
IF KKK = 1 THEN 6
IF PMG <= GA THEN 6
```

```
XAP = 1 - X2: MIXGA = MIXGL: KKK = 1
6 IF KKKK = 1 THEN 7
IF PMG <= GB THEN 7
XBP = X2: MIXGB = MIXGL: KKKK = 1
7 NEXT X2
SLOPA = (MIXGA - GA) / XAP: SLOPB = (MIXGB - GB) / (XBP - 1)
YYY = R * T * LOG(XAP): XXX = R * T * LOG(1 - XBP)
LINE (0, GA)-(.1, GA), 2:  LINE (.9, GB)-(1, GB), 1
LINE (0, GA)-(1, GB)
FOR X2 = .01 TO .99 STEP .01
GAP = GA + SLOPA * X2: GBP = GB - SLOPB * X2
IF T > TA THEN 8
PSET (X2, GAP), 2
8 IF T > TB THEN 9
PSET (1 - X2, GBP), 1
FOR LL = 1 TO 1000 STEP .1: NEXT LL
9 NEXT X2
LOCATE 19, 4
IF XXX >= GA THEN
PRINT "A eq B solid"
ELSEIF T < TA THEN
PRINT "A eq B liquid"
ELSEIF T > TA AND T < TB THEN
PRINT "                "
END IF
LOCATE 20, 4
IF YYY >= GB THEN
PRINT "B eq A solid"
ELSEIF T < TB THEN
PRINT "B eq A liquid"
ELSEIF T > TB AND T < TA THEN
PRINT "                "
END IF
LOCATE 21, 4
IF XXX >= GA AND YYY >= GB THEN PRINT "A & B solids"
IF T >= TA THEN 17
LINE (XAP, GGI)-(XAP, GGI - GGI * .4), 2
17 IF T >= TB THEN 18
LINE (XBP, GGI)-(XBP, GGI - GGI * .4), 1
```

```
18 LOCATE 4, 5: PRINT USING "###"; T; : PRINT "K"
LOCATE 4, 10:  PRINT "GMIN="; : PRINT USING "#####"; GGI
LOCATE 5, 5: PRINT "GA"; : PRINT USING "#####"; GA
LOCATE 5, 14: PRINT USING "#.##"; XAP
LOCATE 6, 5: PRINT "GB"; : PRINT USING "#####"; GB
LOCATE 6, 14: PRINT USING "#.##"; XBP
LOCATE 7, 7: PRINT "PMGA"; : PRINT USING "######"; XXX
LOCATE 8, 7: PRINT "PMGB"; : PRINT USING "######"; YYY
'A$ = INPUT$(1)
11 VIEW (184, 17)-(317, 170)
WINDOW (0, TI)-(1, TF): LINE (0, TI)-(1, TF), , B
IF XAP > XBP THEN 14
IF T >= TA THEN 12
PSET (XAP, T), 2
12 IF T >= TB THEN 15
PSET (XBP, T), 1
GOTO 15
14 TEU = T: XEU = XAP
15 FOR I = 1 TO 9
LINE (I * .1, TI)-(I * .1, TI + (TF - TI) / 40)
TX = TI + (TF - TI) * I * .1: LINE (0, TX)-(.02, TX)
NEXT I
LOCATE 23, 22: PRINT "   0     XB--->      1"
A$ = INPUT$(1)
19 NEXT T
VIEW (184, 17)-(317, 170)
WINDOW (0, TI)-(1, TF): LINE (0, TI)-(1, TF), , B
LINE (0, TEU)-(1, TEU), 3
LOCATE 15, 25: PRINT "A+": LOCATE 16, 25: PRINT "LIQ"
LOCATE 15, 36: PRINT "B+": LOCATE 16, 36: PRINT "LIQ"
LOCATE 4, 27: PRINT "TEU=";
PRINT USING "###.#"; TEU; : PRINT "K"
LOCATE 5, 27: PRINT "XEU="; : PRINT USING "#.##"; XEU
LOCATE 10, 27: PRINT "LIQUID"
LOCATE 20, 27: PRINT "SOLIDS A+B"
29 END
```

Output


PLOT OF G v. x
0
381K GMIN=-3682
GA -613 0.17
GB-1392 0.65
PMGA -3325
PMGB -5613
FREE ENERGY
A eq B liquid
B eq A liquid
0
XB--->
1
Eutectic Diagram
426
TEMPERATURE
339
0
XB--->
1


PLOT OF G v. x
0
406K GMIN=-3682
GA 482 0.01
GB 0 0.99
PMGA-15544
PMGB-15544
FREE ENERGY
0
XB--->
1
Eutectic Diagram
426
TEU=358.0K
XEU=0.41
LIQUID
A+
LIQ
B+
LIQ
SOLIDS A+B
TEMPERATURE
339
0
XB--->
1

3.5 PHASE DIAGRAM OF IDEAL SOLID – IDEAL LIQUID SOLUTIONS VIA FREE ENERGY CONSIDERATIONS

For a binary system involving two solutions α and β in equilibrium with each other, the criterion of equilibrium gives

$$\mu_{A(\alpha)} = \mu_{A(\beta)} \quad \text{and} \quad \mu_{B(\alpha)} = \mu_{B(\beta)} \tag{1}$$

These two equations may be solved for equilibrium compositions $x_{B(\alpha)}$ and $x_{B(\beta)}$ of the two solutions, respectively. The solutions may be carried out graphically or analytically.

Graphic solution involves drawing a common tangent to the graphs of $G_m(\alpha)$ versus $x_{B(\alpha)}$ and $G_{m(\beta)}$ versus $x_{B(\beta)}$ as shown in Fig. 1.

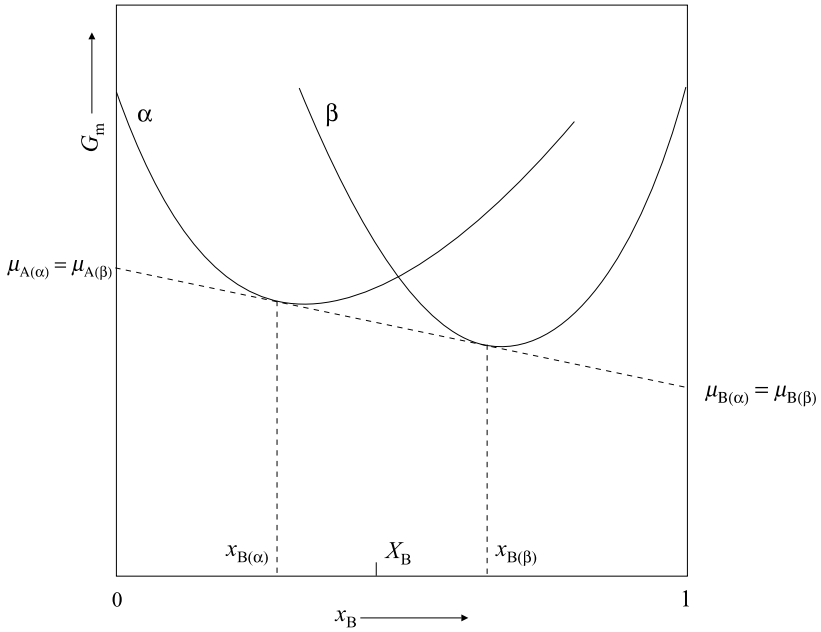


Fig. 1 Graphs of $G_{m(\beta)}$ versus $x_{B(\alpha)}$ and $G_{m(\beta)}$ versus $x_{B(\beta)}$ with a common tangent line

From Fig. 1, it may be concluded that

- The system exists as a single phase α in the composition range $x_B = 0$ to $x_B = x_{B(\alpha)}$, since $G_m(\alpha) < G_m(\beta)$.
- The system exists as a single phase β in the composition range $x_B = x_{B(\beta)}$ to $x_B = 1$, since $G_m(\beta) < G_m(\alpha)$.
- Within the range $x_B = x_{B(\alpha)}$ to $x_B = x_{B(\beta)}$, the system exists as two phases α and β in equilibrium with each other.

At any point in between $x_{B(\alpha)}$ and $x_{B(\beta)}$, the two phases have compositions $x_{B(\alpha)}$ and $x_{B(\beta)}$, respectively, with their relative amounts given by the expression

$$\text{Proportion of } \alpha\text{-phase} = \frac{x_{B(\beta)} - X_B}{x_{B(\beta)} - X_{B(\alpha)}} \tag{2a}$$

$$\text{Proportion of } \beta\text{-phase} = \frac{X_B - x_{B(\alpha)}}{x_{B(\beta)} - x_{B(\alpha)}} \tag{2b}$$

where X_B is the composition of the given point. Equations (2a) and (2b) are known as the lever rule.

The characteristics of phase diagram is determined by the relative placement of free energy versus composition curves of the two phases in Fig. 1. A simple translational of the two curves parallel to the free energy axis has no effect on the characteristics of the phase diagram. Even if the chemical potential of component A or B in both phases α and β are arbitrarily changed by the same value, it has no effect on the characteristics of the phase diagram, since the difference $\mu_{A(\alpha)} - \mu_{A(\beta)}$ remains unchanged as shown in the following.

$$[\mu_{A(\alpha)} + \delta\mu_A] - [\mu_{A(\alpha)} + \delta\mu_A] \equiv \mu_{A(\alpha)} - \mu_{A(\beta)} \tag{3}$$

The appearance of the free energy versus composition curves is affected by the above changes, but the points with a common tangent line is not affected and thus the phase diagram of the system remains unchanged.

For a binary system exhibiting ideal solid and liquid solutions, the free energy versus composition curves may be drawn with the following procedure.

- The relative placement of $\mu_{i(l)}$ and $\mu_{i(s)}$ on the axes may be depicted by setting.

  $$\mu_{i(l)} = 0 \tag{4}$$

  With this assumption, $\mu_{i(s)}$ is given by

  $$\begin{aligned}\mu_{i(s)} &= -\Delta_{fus}G_i = -[\Delta_{fus}H_i - T\,\Delta_{fus}S_i] = -[\Delta_{fus}H_i - T(\Delta_{fus}H_i/T_i^*)] \\ &= -\Delta_{fus}H_i(1 - T/T_i^*)\end{aligned} \tag{5}$$

  Note While replacing $\Delta_{fus}S_i$ by $\Delta_{fus}H_i/T_i^*$, it is assumed that $C_{p,i(s)} = C_{p,i(l)}$ so that $\Delta_{fus}H_i$ becomes independent of temperature.

  For $T > T_i^*$, $\mu_{i(s)}$ is positive. Hence, the point representing free energy of the solid phase will lie above that of the liquid phase for which $\mu_i = 0$ has been assumed. For $T_i < T_i^*$, the reverse is observed.
- The free energy of liquid solution will be equal to the free energy of mixing of the two components, since $\mu_{i(l)} = 0$ has been assumed. Hence,

  $$G_l = \Delta_{mix}G = RT\,[x_{A(l)} \ln x_{A(l)} + x_{B(l)} \ln x_{B(l)}] \tag{6}$$
- The free energy of solid solution will be equal to the total free energy before mixing plus the free energy of mixing of the two components. Hence

  $$G_s = [x_{A(s)}\,G_{A(s)} + x_{B(s)}\,G_{B(s)}] + RT\,[x_{A(s)} \ln x_{A(s)} + x_{B(s)} \ln x_{B(s)}] \tag{7}$$

DESCRIPTION OF FREE ENERGY CURVES AT DIFFERENT TEMPERATURES

- For $T < T_A^*$ and $T < T_B^*$, the free energy curve for liquid solution lies above that of solid solution in the entire range of composition (i.e. from $x_B = 0$ to $x_B = 1$) as shown in Fig. 2.
- On increasing temperature, the free energy curves of both solid and liquid solutions move upwardly (i.e. the values of G_l and G_s increase). The movement of liquid phase is comparatively larger due to the higher entropies of pure liquid components as compared to pure solid components.

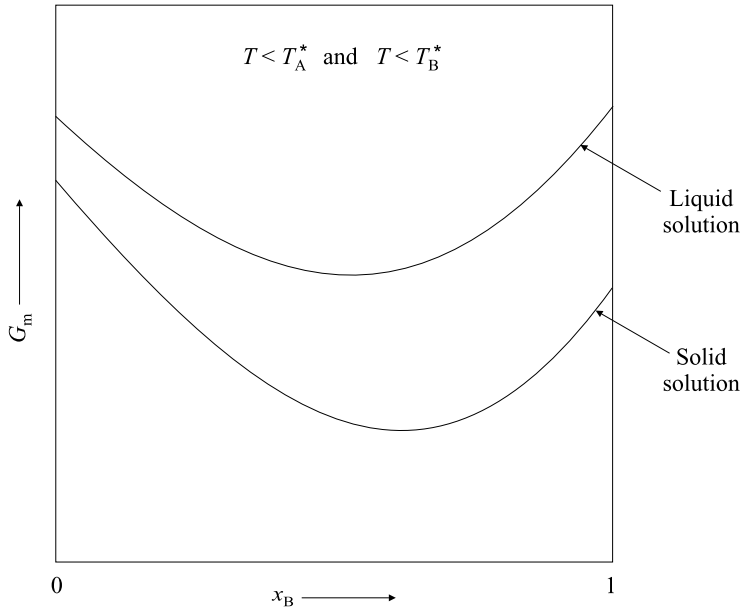


Fig. 2 Free energy curves for $T < T_A^*$ and $T < T_B^*$

For $T = T_A^*$, we will have

$$\Delta_{fus}G_A = 0$$

indicating that $G_{A(s)} = G_{A(l)}$. Hence, the free energy versus composition curves of solid and liquid solutions meet each other at $x_A = 1$. The curve of liquid solution still lies above that of the solid solution in the entire range of composition as shown in Fig. 3. Hence, the system is present as solid solution in the entire range of composition except at $x_A = 1$ where solid A exists in equilibrium with liquid A.

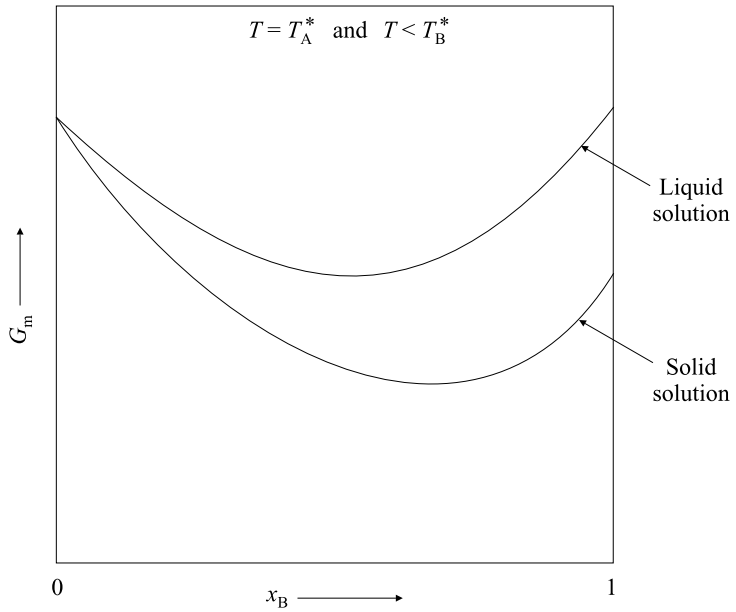


Fig. 3 Free energy curves for $T = T_A^*$ and $T < T_B^*$

- For $T_A^* < T < T_B^*$, the free energy curves of solid and liquid solutions intersect each other at some composition x'_B depending upon the temperature (Fig. 4).

  The composition x'_B of point of intersection of solid and liquid solutions may be determined by setting $G_s = G_l$. This gives

$$x_{A(s)}G_{A(s)} + x_{B(s)}G_{B(s)} + RT\,[x_{A(s)} \ln x_{A(s)} + x_{B(s)} \ln x_{B(s)}] = RT \ln [x_{A(l)} \ln x_{A(l)} + x_{B(l)} \ln x_{B(l)}] \quad (8)$$

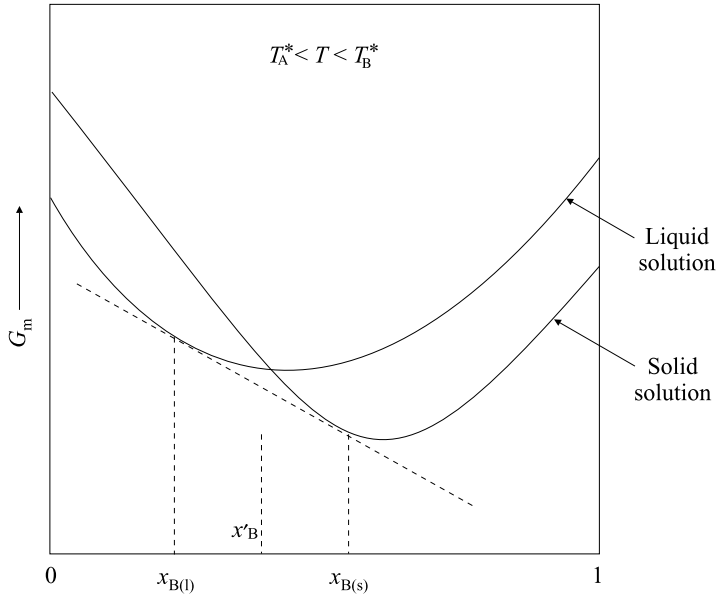


Fig. 4 Free energy curves for $T_A^* < T < T_B^*$

At the intersection point,

$$x_{A(s)} = x_{A(l)} \quad \text{and} \quad x_{B(s)} = x_{B(l)}.$$

with the expressions

$$x_{A(s)} + x_{B(s)} = 1 \quad \text{and} \quad x_{A(l)} + x_{B(l)} = 1 \quad (10)$$

Let $x'_A = x_{A(s)}$ and $x'_B = x_{B(s)}$, such that $x'_A + x'_B = 1$. Hence, Eq. (8) is reduced to

$$(1 - x'_B)\, G_{A(s)} + x'_B\, G_{B(s)} = 0 \quad (11)$$

which gives

$$x'_B = \frac{G_{A(s)}}{G_{A(s)} - G_{B(s)}} \quad (12)$$

For $x_B < x'_B$, the free energy of liquid phase is lesser than that of solid phase while the reverse is true for $x_B > x'_B$. The compositions of solid and liquid phases in equilibrium with each other may be determined by drawing a common tangent line to the free energy versus composition curves (Fig. 4). As mentioned earlier, the compositions $x_{B(l)}$ and $x_{B(s)}$ at which the common tangent line touches the liquid and solid curves, respectively, are the compositions of liquid and solid phases in equilibrium in the composition range $x_{B(l)}$ to $x_{B(s)}$.

- Analytically, the composition $x_{B(l)}$ and $x_{B(s)}$ may be determined by comparing the slopes of tangent lines drawn for decreasing value of x_B from x'_B for the liquid phase and for increasing value of x_B from x'_B for the solid phase. The compositions at which the two slopes become identical give the compositions $x_{B(l)}$ and $x_{B(s)}$, respectively.[†] The expressions of slopes of tangent lines are as follows.

$$\frac{dG_l}{dx_{B(l)}} = RT \frac{d}{dx_{B(l)}} [\{1 - x_{B(l)}\} \ln(1 - x_{B(l)}) + x_{B(l)} \ln x_{B(l)}]$$

$$= RT [-\ln (1 - x_{B(l)}) + \ln x_{B(l)}]$$

$$= RT [-\ln x_{A(l)} + \ln x_{B(l)}] \tag{13}$$

$$\frac{dG_s}{dx_{B(s)}} = \frac{d}{dx_{B(s)}} [(1 - x_{B(s)}) G_{A(s)} + x_{B(s)} G_{B(s)} + RT [(1 - x_{B(s)} \ln (1 - x_{B(s)}) + x_{B(s)} \ln x_{B(s)}]$$

$$= -G_{A(s)} + G_{B(s)} + RT [-\ln (1 - x_{B(s)}) + \ln x_{B(s)}]$$

$$= -G_{A(s)} + G_{B(s)} + RT [-\ln x_{A(s)} + \ln x_{B(s)}] \tag{14}$$

At $x_{B(l)}$ and $x_{B(s)}$, not only slopes to free energy versus composition curves have identical values but also the partial molar free energies of components A and B in liquid and solid phases separately have identical values. This fact may be used to determine the compositions $x_{B(l)}$ and $x_{B(s)}$. The expressions of partial molar free energies are derived in the following.

Solid Solution For this solution, we have

$$G_s = x_{A(s)}G_{A(s)} + x_{B(s)}G_{B(s)} + RT(x_{A(s)} \ln x_{A(s)} + x_{B(s)} \ln x_{B(s)})$$

i.e. $$[n_{A(s)} + n_{B(s)}]G_s = n_{A(s)}G_{A(s)} + n_{B(s)}G_{B(s)}$$

$$+ RT \left[n_{A(s)} \ln \left\{ \frac{n_{A(s)}}{n_{A(s)} + n_{B(s)}} \right\} + n_{B(s)} \ln \left\{ \frac{n_{B(s)}}{n_{A(s)} + n_{B(s)}} \right\} \right]$$

Since $$G_{A(s),pm} = \left[\frac{\partial\{(n_{A(s)} + n_{B(s)})G_s}{\partial n_{A(s)}} \right]_{T,p,n_{B(s)}}$$

we find that $$G_{A(s),pm} = G_{A(s)} + RT \ln x_{A(s)} \tag{15}$$

Working similarly, we find that $$G_{B(s),pm} = G_{B(s)} + RT \ln x_{B(s)} \tag{16}$$

Liquid Solution For this solution, we have

$$G_l = RT [x_{A(l)} \ln x_{A(l)} + x_{B(l)} \ln x_{B(l)}]$$

[†]It is assumed that $x_{B(l)}$ and $x_{B(s)}$ lie symmetrically with reference to the point of intersection of solid and liquid phases.

i.e. $$[n_{A(l)} + n_{B(l)}]G_l = RT\left[n_{A(1)} \ln\left\{\frac{n_{A(1)}}{n_{A(1)} + n_{B(1)}}\right\}\right] + n_{B(1)} \ln\left\{\frac{n_{B(1)}}{n_{A(1)} + n_{B(1)}}\right\}$$

Since $$G_{A(l),\,pm} = \left[\frac{\partial\{(n_{A(1)} + n_{B(1)})G_l}{\partial n_{A(1)}}\right]_{T,p,n_{B(1)}}$$

We find that $$G_{A(l),\,pm} = RT \ln x_{A(l)} \quad (17)$$

Working similarly, we get $$G_{B(l),\,pm} = RT \ln x_{B(l)} \quad (18)$$

Equating the corresponding partial molar free energies, we get

$$G_{A(s),\,pm} = G_{A(l),\,pm}$$

i.e. $$G_{A(s)} + RT \ln x_{A(s)} = RT \ln x_{A(l)} \quad \text{or} \quad \frac{x_{A(1)}}{x_{A(s)}} = \exp\left(\frac{G_{A(s)}}{RT}\right) \equiv a \quad (19)$$

Similarly

$$G_{B(s),\,pm} = G_{B(l),\,pm}$$

i.e. $$G_{B(s)} + RT \ln x_{B(s)} = RT \ln x_{B(l)} \quad \text{or} \quad \frac{x_{B(1)}}{x_{B(s)}} = \exp\left(\frac{G_{B(s)}}{RT}\right) \equiv b \quad (20)$$

Since $x_{A(l)} = 1 - x_{B(l)}$ and $x_{A(s)} = 1 - x_{B(s)}$, we can write Eq. (19) as

$$\frac{1 - x_{B(1)}}{1 - x_{B(s)}} = a \quad (21)$$

which on using Eq. (20) becomes

$$\frac{1 - bx_{B(s)}}{1 - x_{B(s)}} = a \quad (22)$$

which gives $$x_{B(s)} = \frac{a - 1}{a - b} \quad (23)$$

Using Eq. (23) in Eq. (20), we get

$$x_{B(l)} = b\left(\frac{a - 1}{a - b}\right) \quad (24)$$

- For $T = T_B^*$, the two free energy versus composition curves meet at $x_B = 1$ and the curve of liquid phase lies lower than that of solid phase in the entire range of composition. The system exists as the liquid phase in the entire range of composition with solid B at equilibrium at $x_B = 1$ (Fig. 5).

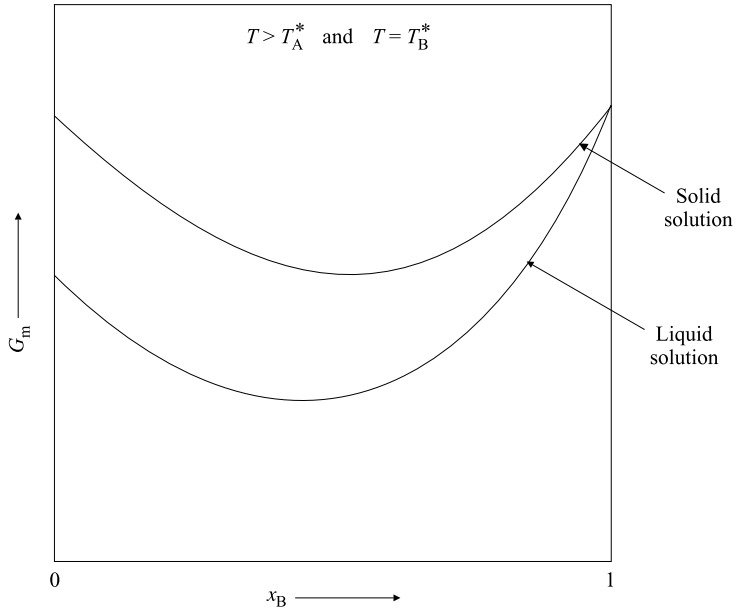


Fig. 5 Free energy curves for $T > T_A^*$ and $T = T_B^*$

- For $T > T_B^*$, the curve of liquid phase lies lower than that of solid phase in the entire range of composition (Fig. 6). The system exists as liquid phase throughout the entire composition.

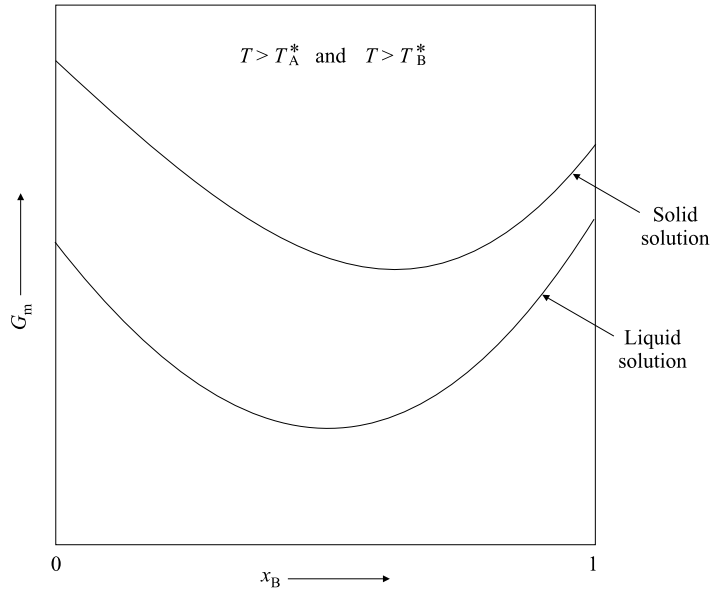


Fig. 6 Free energy curves for $T > T_A^*$ and $T > T_B^*$

Phase Diagram From the free energy versus compositions plots at different temperatures, the values of $x_{B(s)}$ and $x_{B(l)}$ at which solid and liquid solutions exist at equilibrium may be determined. Alternatively, these may be obtained by using Eqs (23) and (24), resepectively. The phase diagram of the system is obtained by plotting $x_{B(s)}$ versus T and $x_{B(l)}$ versus T. The general feature of the phase diagram is shown in Fig. 7.

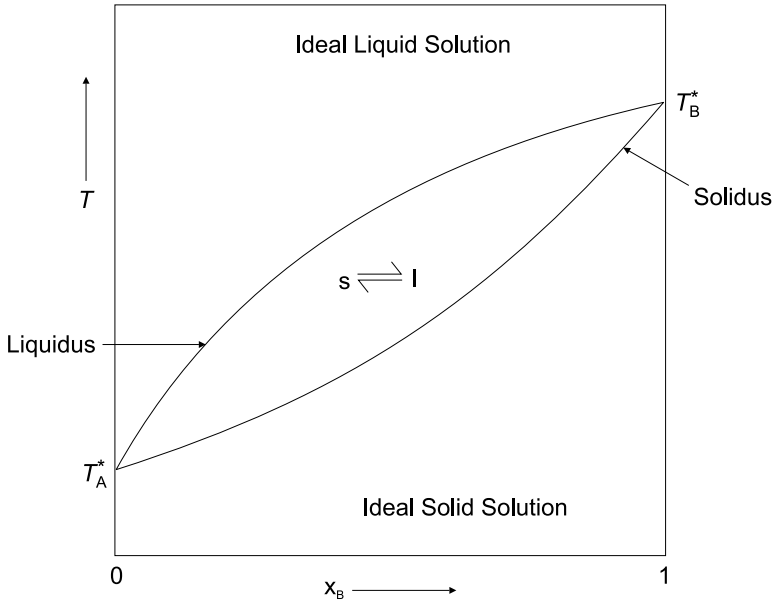


Fig. 7 Phase diagram of a binary system exhibiting a complete series of solid solution

Comment: In the above analysis, it is assumed that the heat capacities of components A and B in solid and liquid phases have the same value. If this is not correct, the following expressions may be used to compute $\Delta_{fus}\,\mu_i$ values.

$$\Delta_{fus}\,\mu_i = [H_{i(l)} - H_{i(s)}] - T\,[S_{i(l)} - S_{i(s)}]$$

$$= \left[\Delta_{fus} H_i + \int_{T_i^*}^{T} \Delta C_{p,i}\,dT\right] - T\left[\Delta_{fus} S_i + \int_{T_i^*}^{T} \frac{\Delta C_{p,i}}{T}\,dT\right] \tag{25}$$

where $\Delta C_{p,1i} = C_{p,i(l)} - C_{p,i(s)}$. If the heat capacity variations is represented as

$$C_p = a + bT + cT^{-2} \tag{26}$$

then

$$\frac{\Delta_{fus}\mu_i}{RT} = \frac{\Delta_{fus}H_i}{R}\left(\frac{1}{T} - \frac{1}{T_i^*}\right) + \frac{\Delta a_i}{R}\left(1 - \frac{T_i^*}{T} + \ln\frac{T_i^*}{T}\right) - \frac{\Delta b_i}{2RT}\left(T - T_i^*\right)^2$$

$$- \frac{\Delta c_i}{2R}\left(\frac{1}{T} - \frac{1}{T_i^*}\right)^2 \tag{27}$$

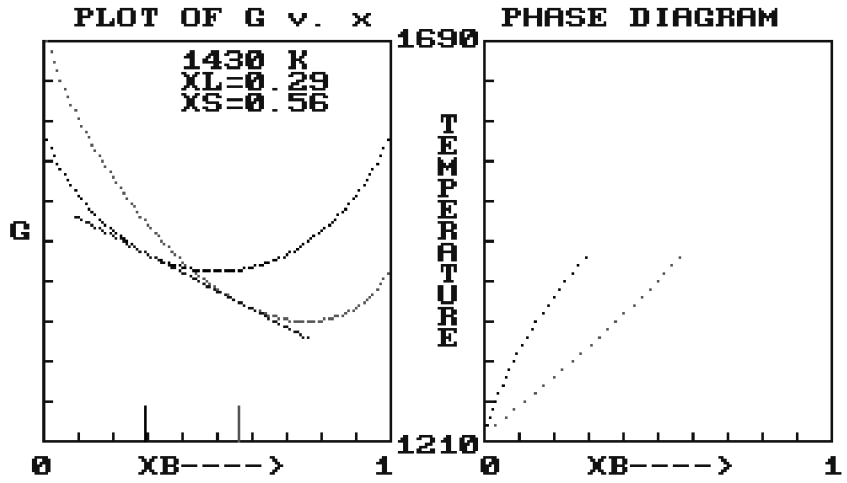
Illustration Phase diagram of system of A($T_A^* = 1\,210$ K, $\Delta_{fus}H_A = (26.28\,T_A)$ kJ mol^{-1}) and B($T_B^* = 1\,690$ K, $\Delta_{fus}H_B^* = (30.04\ T_B)$ kJ mol^{-1}).

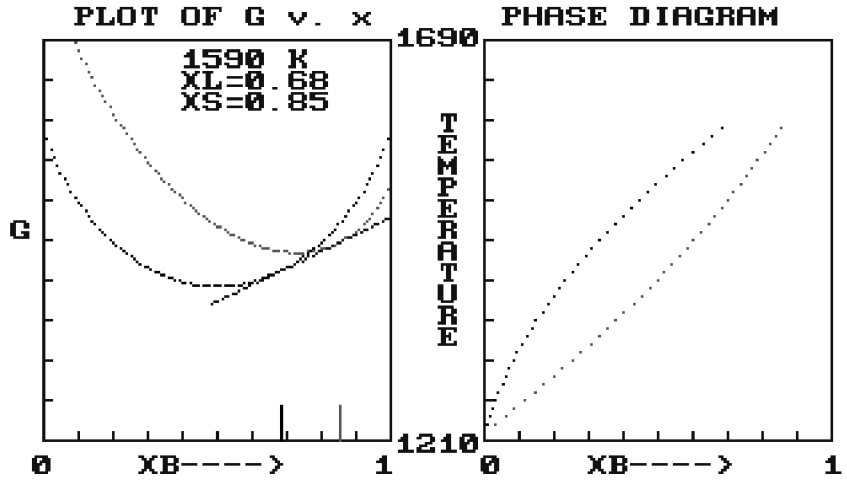
Program

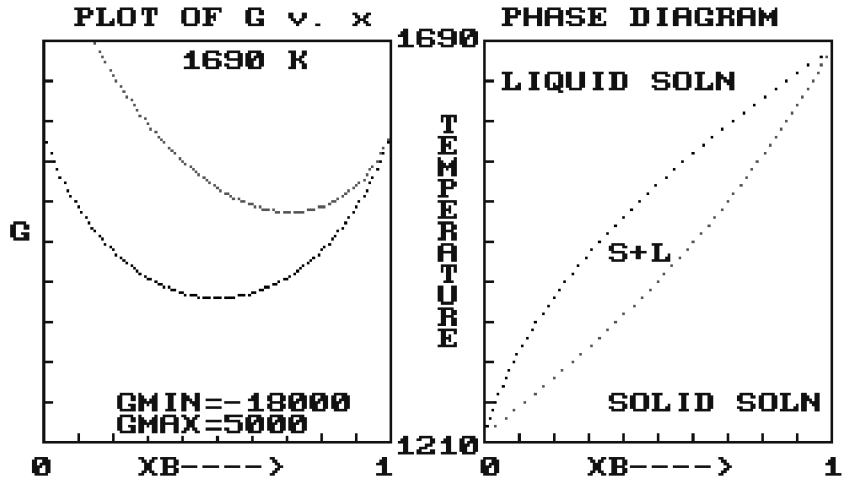
```
CLS
REM PROGRAM SOLSOL1;SOLID-LIQUID SOLUTIONS VIA FREE ENERGY
DEF FNA (X) = R * T * ((1 - X) * LOG(1 - X) + X * LOG(X))
CLS : R = 8.314: TA = 1210: TB = 1690
HA = 26.28 * TA: HB = 30.04 * TB
SCREEN 1: COLOR 15, 0
LOCATE 2, 5: PRINT "PLOT OF G v. x"
LOCATE 2, 25: PRINT "PHASE DIAGRAM"
LOCATE 3, 20: PRINT USING "####"; TB
LOCATE 12, 2: PRINT "G"
A$ = "TEMPERATURE"
FOR I = 1 TO 11: LOCATE 6 + I, 22: PRINT MID$(A$, I, 1): NEXT I
LOCATE 22, 20: PRINT USING "####"; TA
LOCATE 23, 2: PRINT " 0     XB---->     1"
LOCATE 23, 23: PRINT " 0     XB---->     1"
FOR T = TA TO TB STEP 10
GA = -HA * (1 - T / TA): GB = -HB * (1 - T / TB)
VIEW (20, 20)-(150, 170): CLS
WINDOW (0, -18000)-(1, 5000): LINE (0, -18000)-(1, 5000), , B
LOCATE 4, 9: PRINT T; "K"
FOR I = 1 TO 9
LINE (I * .1, -18000)-(I * .1, -17500)
YS = -18000 + 2300 * I: LINE (0, YS)-(.02, YS)
NEXT I
FOR XB = .01 TO .99 STEP .01
DG = (1 - XB) * GA + XB * GB
GMIX = FNA(XB): GBS = DG + GMIX: GBL = GMIX
PSET (XB, GBS), 1: PSET (XB, GBL), 2
NEXT XB
IF T = TA OR T = TB THEN 20
XBP = GA / (GA - GB): XXL = XBP: XXS = XBP
FOR II = 1 TO 500
XXL = XXL - .001: XXS = XXS + .001
SLOPL = R * T * (-LOG(1 - XXL) + LOG(XXL))
```

```
SLOPS = -GA + GB + R * T * (-LOG(1 - XXS) + LOG(XXS))
IF SLOPS > SLOPL THEN 6
NEXT II
6 LOCATE 5, 10: PRINT "XL="; : PRINT USING "#.##"; XXL
  LOCATE 6, 10: PRINT "XS="; : PRINT USING "#.##"; XXS
  GL1 = FNA(XXL): GS1 = (1 - XXS) * GA + XXS * GB + FNA(XXS)
IF XXS < 0 OR XXS >= 1 THEN 20
IF XXL < XXS THEN
XMN = XXL - .2: XMX = XXS + .2
ELSE XMN = XXS - .2: XMX = XXL + .2
END IF
IF XMN < 0 THEN XMN = 0
IF XMX > 1 THEN XMX = 1
FOR XX = XMN TO XMX STEP .01
YY = GS1 + ((GS1 - GL1) / (XXS - XXL)) * (XX - XXS)
PSET (XX, YY), 3
FOR LL = 1 TO 10 STEP .01: NEXT LL
NEXT XX
LINE (XXL, -18000)-(XXL, -16000), 2
LINE (XXS, -18000)-(XXS, -16000), 1
VIEW (185, 20)-(315, 170): WINDOW (0, TA - 10)-(1, TB + 10)
LINE (0, TA - 10)-(1, TB + 10), , B
PSET (XXL, T), 2: PSET (XXS, T), 1
FOR I = 1 TO 9
LINE (I * .1, TA - 10)-(I * .1, TA)
TX = TA - 10 + (TB - TA + 20) * I * .1
LINE (0, TX)-(.02, TX)
NEXT I
A$ = INPUT$(1)
20 NEXT T
LOCATE 20, 7: PRINT "GMIN=-18000"
LOCATE 21, 7: PRINT "GMAX=5000"
LOCATE 5, 25: PRINT "LIQUID SOLN"
LOCATE 13, 30: PRINT "S+L"
LOCATE 20, 30: PRINT "SOLID SOLN"
END
```

Output


PLOT OF G v. x
1430 K
XL=0.29
XS=0.56
G
0
XB---->
1
PHASE DIAGRAM
1690
TEMPERATURE
1210
0
XB---->
1


PLOT OF G v. x
1590 K
XL=0.68
XS=0.85
G
0
XB---->
1
PHASE DIAGRAM
1690
TEMPERATURE
1210
0
XB---->
1


PLOT OF G v. x
1690 K
G
GMIN=-18000
GMAX=5000
0
XB---->
1
PHASE DIAGRAM
1690
LIQUID SOLN
S+L
TEMPERATURE
SOLID SOLN
1210
0
XB---->
1

3.6 PHASE DIAGRAM OF NON-IDEAL BINARY SYSTEM† VIA FREE ENERGY CONSIDERATIONS

Phase diagram involving (i) regular solid solution-ideal liquid solution, (ii) regular liquid solution-ideal solid solution, and (iii) regular solid solution-regular liquid solution may be drawn by following the procedure described for the system involving ideal solid solution-ideal liquid solution.

Expressions of Free Energies of Regular Solid and Liquid Solutions

We have

$$G_s = x_{A(s)}\, G_{A(s)} + x_{B(s)} G_{B(s)} + RT\,[x_{A(s)} \ln x_{A(s)} + x_{B(s)} \ln x_{B(s)}] + x_{A(s)}\, x_{B(s)}\, \Omega_s \quad (1)$$

$$G_l = RT\,[x_{A(l)} \ln x_{A(l)} + x_{B(l)} \ln x_{B(l)}] + x_{A(l)}\, x_{B(l)}\, \Omega_l \quad (2)$$

Intersection Points of Free Energy versus Composition Curves of Solid and Liquid Solutions

We have

$$G_s = G_l$$

along with $x_{A(s)} = x_{A(l)}$ and $x_{B(s)} = x_{B(l)}$. With these facts, we have

$$x_{A(s)}\, G_{A(s)} + x_{B(s)}\, G_{B(s)} + x_{A(s)}\, x_{B(s)}\, \Omega_s - x_{A(l)}\, x_{B(l)}\, \Omega_l = 0$$

or

$$x_{A(s)}\, G_{A(s)} + x_{B(s)}\, G_{B(s)} + x_{A(s)}\, x_{B(s)}\, (\Omega_s - \Omega_l) = 0$$

or

$$[1 - x_{B(s)}]\, G_{A(s)} + x_{B(s)}\, G_{B(s)} + [1 - x_{B(s)}]\, x_{B(s)}\, \Omega' = 0 \quad (3)$$

---

†A system exhibits nonideality if the interaction energy between unlike components is different from those existing in the like components. Additional parameters are needed in the Gibbs energy expression for the solid or liquid solution in a nonideal binary system. The solid or liquid solution may be commonly classified in different categories based on the interaction energies amongst the constituents of the system.

Regular Solution In this solution, excess Gibbs free energy is given by $G^E = G^{(real)} - G^{(ideal)} = x_A x_B \Omega$ where the parameter Ω has a simple atomistic interpretation. It compares the energy u_{AB} of the bond between unlike atoms A and B with the arithmetic average of the bonds between like atoms.

$$\Omega = Z\left[u_{AB} - \frac{1}{2}(u_{AA} + u_{BB})\right]$$

where Z is the number of nearest neighbours to an atom.

Subregular Solution If a regular solution does not represent the properties of a real solution, additional parameter are needed in the expression of G^E. For a subregular solution, G^E is given by

$$G^E = x_A x_B (A_{BA} x_A + A_{AB} x_B + A_{BB} x_A x_B)$$

where A's are constants.

Quasi-Regular Solution For a quasi-regular solution the expression of excess Gibbs free energy is given by

$$G^E = x_A x_B \Omega\,[1 - T/\tau]$$

with $H^E = x_A x_B \Omega = x_A x_B B_{AA}$ and $S^E = x_A x_B C_{AA}$

where $\tau = B_{AA}/C_{AA} = H^E/S^E$

For most solution, τ is the order of (2500 ± 1000) K.

where $\Omega' = \Omega_s - \Omega_l$. Rearranging the above expression, we get

$$x_{B(s)}^2\,\Omega' + (G_{A(s)} - G_{B(s)} - \Omega')x_{B(s)} - G_{A(s)} = 0 \tag{4}$$

Note Equation (3) is reduced to Eq. (11) of project 5 for $\Omega' = 0$

The roots of Eq. (4) are

$$x_{B(s)} = \left[\frac{-B \pm \sqrt{D}}{2\Omega'}\right] \tag{5}$$

where

$$B = G_{A(s)} - G_{B(s)} - \Omega' \tag{6}$$

$$D = (G_{A(s)} - G_{B(s)} - \Omega')^2 + 4G_{A(s)}\Omega' \tag{7}$$

Of the two roots, the realistic value(s) lying in the range $0 < x_{B(s)} < 1$ is/are accepted and the unrealistic value(s) is/are ignored.

Figure 1 illustrates the two points of intersection of free energy verses composition curves.

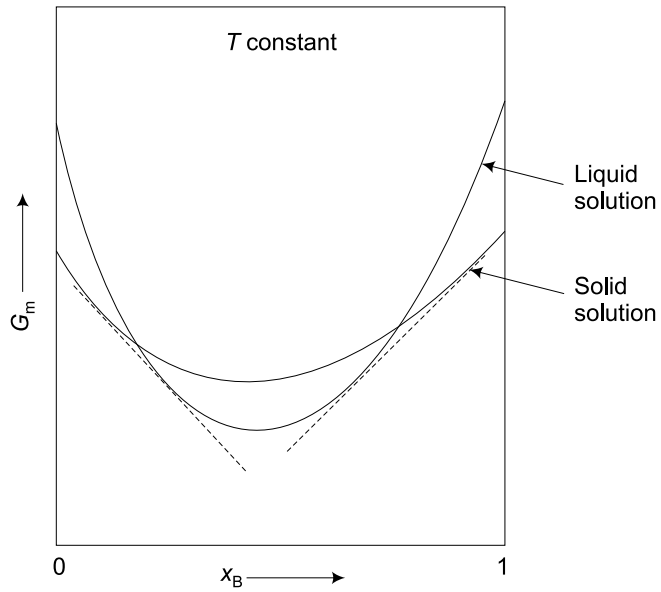


Fig. 1 Intersection at two points of G_m versus x_B plots

Slopes of Free Energy Versus Composition Plots

The expression of slopes are as follows.

$$\frac{dG_s}{dx_{B(s)}} = -G_{A(s)} + G_{B(s)} + RT\,[-\ln x_{A(s)} + \ln x_{B(s)}] + [1 - 2x_{B(s)}]\,\Omega_s \tag{8}$$

$$\frac{dG_l}{dx_{B(l)}} = RT[-\ln x_{A(l)} + \ln x_{B(l)}] + [1 - 2x_{B(l)}]\Omega_l \tag{9}$$

Note Equations (8) and (9) reduce to Eq. (14) and (13) of Project 5, respectively for Ω's = 0

Expressions of Partial Molar Free Energies

We have

$$G_{A(s),\,pm} = G_{A(s)} + RT \ln x_{A(s)} + x^2_{B(s)}\,\Omega_s \tag{10}$$

$$G_{B(s),\,pm} = G_{B(s)} + RT \ln x_{B(s)} + x^2_{A(s)}\,\Omega_s \tag{11}$$

$$G_{A(l),\,pm} = RT \ln x_{A(l)} + x^2_{B(l)}\,\Omega_l \tag{12}$$

$$G_{B(l),\,pm} = RT \ln x_{B(l)} + x^2_{A(l)}\,\Omega_l \tag{13}$$

Note Equations (10)-(13) reduce to (15)-(18) of Project 5, respectively, if Ω's = 0.

Free Energy Versus Compositon Plots at Azeotropic Temperature

Figure 2 displays the free energy versus composition plots at the azeotropic temperature where the two points of interseciton have identical values. At this temperature, not only slopes but also the partial molar free energies of both A and B in the solid as well as in the liquid phase separately have identical values.

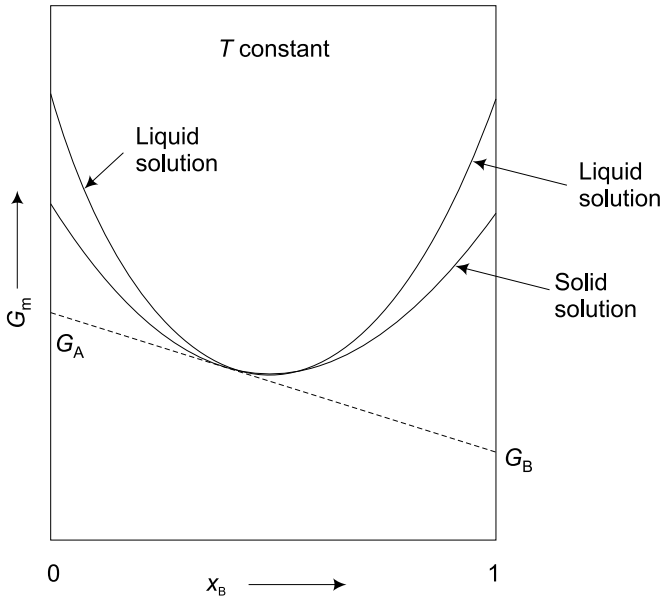


Fig. 2 Plots of G_m versus x_B for solid and liquid solution at the azeotropic temperature

Construction of Phase Diagram

Around each of the two intersection points, the compositions $x_{B(s)}$ and $x_{B(l)}$ of solid and liquid solutions, respectively, in equilibrium may be determined by drawing a common tangent line to the free energy versus composition curves of solid and liquid solutions (see, Fig. 1). Assuming that $x_{B(s)}$ and $x_{B(l)}$ lie symmetrically on either sides of the intersection point, these may be determined by comparing slopes (Eqs 8 and 9) for decreasing and increasing value of x_B from the intersection composition. A direct solution of $x_{B(s)}$ and $x_{B(l)}$ from the expressions $G_{A(s),pm} = G_{A(l),pm}$ and $G_{B(s),pm} = G_{B(l),pm}$ is not straight forward as the expressions are difficult to solve for x_B. These may be seen from the following resultant expressions.

$$G_{A(s)} + RT \ln (x_{A(s)}/x_{A(l)}) + x^2_{B(s)}\,\Omega_s - x^2_{B(l)}\,\Omega_l = 0 \quad (14)$$

and

$$G_{B(s)} + RT \ln (x_{B(s)}/x_{B(l)}) + x^2_{A(s)}\,\Omega_s - x^2_{A(l)}\,\Omega_l = 0 \quad (15)$$

Figure 3 displays the expected phase diagram for a binary system exhibiting regular solid and ideal solution in equilibrium with each others. For this type of system $\Omega_l = 0$ in Eqs (14) and (15).

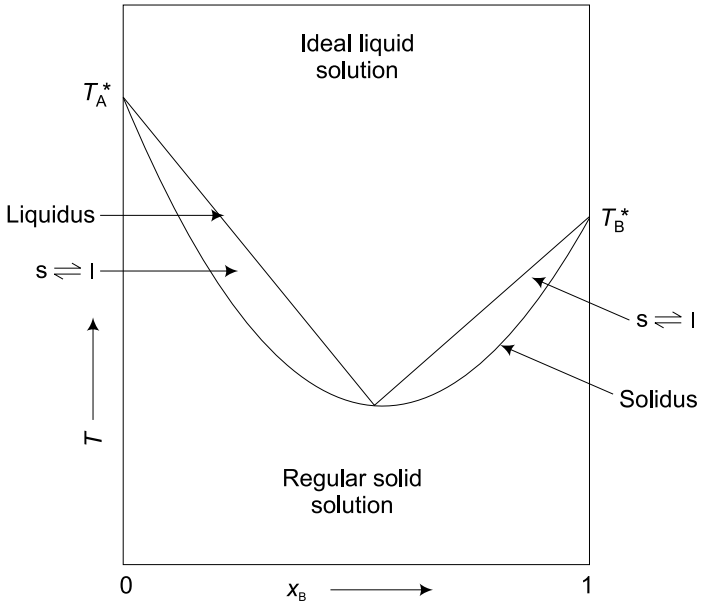


Fig. 3 Phase diagram of a system exhibiting regular solid and ideal liquid solutions

Figure 4 displays the expected phase diagram for a binary system exhibiting ideal solid and regular liquid solutions in equilibrium with each other. For this system, $\Omega_s = 0$ in Eqs (41) and (42).

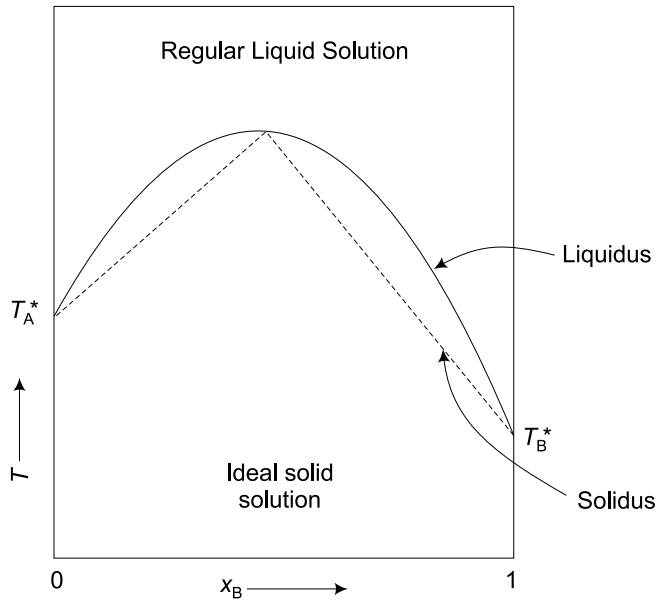


Fig. 4 Phase diagram of a system exhibiting regular liquid and ideal solid solutions

Computation of Azeotropic Composition and Temperature

The solid and liquid solutions at the azeotropic point (i.e. minimum/maximum temperature in the phase diagram) have identical compositions. The azeotropic temperature and the corresponding composition may be determined by setting $D = 0$ in Eq. (5). This gives

$$x_{B(s)} = -\frac{B}{2\Omega'} = -\frac{G_{A(s)} - G_{B(s)} - \Omega'}{2\Omega'} \tag{16}$$

The experssion of D gives

$$D = [G_{A(s)} - G_{B(s)} - \Omega']^2 + 4G_{A(s)}\Omega' = 0 \tag{17}$$

The expression of $G_{A(s)}$ and $G_{B(s)}$ are

$$G_{A(s)} = -\Delta_{fus}H_A(1 - T/T_A^*) \tag{18}$$

$$G_{B(s)} = -\Delta_{fus}H_B(1 - T/T_B^*) \tag{19}$$

Writing $\Delta_{fus}H_A$ and $\Delta_{fus}H_B$ simply as H_A and H_B, respectively, we can write Eq. (17) as

$$\left[-H_A\left(1-\frac{T_{az}}{T_A^*}\right) + H_B\left(1-\frac{T_{az}}{T_B^*}\right) - \Omega'\right]^2 - 4\,H_A\Omega'\left(1-\frac{T_{az}}{T_A^*}\right) = 0 \tag{20}$$

Opening this expression, we may write

$$a\,T_{az}^2 + b\,T_{az} + c = 0 \tag{21}$$

where

$$a = \left(\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*}\right)^2 \tag{22}$$

$$b = 2(-H_A + H_B - \Omega')\left(\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*}\right) + \frac{4\,H_A\Omega'}{T_A^*} \tag{23}$$

$$c = (-H_A + H_B - \Omega')^2 - 4H_A\Omega' \tag{24}$$

The acceptable value of T_{az} is

$$T_{az} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{25}$$

Note Of the two values of T_{az}, a realistic value represents azeotropic temperature.

The expression of x_B in terms of H_A and H_B is

$$(x_B)_{az} = -\frac{G_{A(s)} - G_{B(s)} - \Omega'}{2\Omega'} = \frac{G_{B(s)} - G_{A(s)}}{2\Omega'} + \frac{1}{2}$$

$$= \frac{1}{2\Omega'}\left[-H_B\left(1-\frac{T_{az}}{T_B^*}\right) + H_A\left(1-\frac{T_{az}}{T_A^*}\right)\right] + \frac{1}{2}$$

$$= \frac{1}{2\Omega'}\left[(H_A - H_B) + \left(\frac{H_B}{T_B^*} - \frac{H_A}{T_A^*}\right)T_{az}\right] + \frac{1}{2}$$

Note The expression of x_B may also be derived by equating the slops of free energy versus composition plots (Eqs. 8 and 9).

Illustration

The computation of azeotropic temperature and the corresponding composition may be illustrated for systems having the following data.

$H_A = 5\,200$ J mol^{-1} $\qquad T_A^* = 500$ K

$H_B = 4\,000$ J mol^{-1} $\qquad T_B^* = 600$ K

(i) $\Omega' = -5\,400$ J mol^{-1} and (ii) $\Omega' = 5\,400$ J mol^{-1}

(i) We have

$$\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*} = \left(\frac{5200}{500} - \frac{4000}{600}\right) \text{ J K}^{-1} \text{ mol}^{-1} = 3.733\,3 \text{ J K}^{-1} \text{ mol}^{-1}$$

$$-H_A + H_B - \Omega' = (-5200 + 4\,000 + 5\,400) \text{ J mol}^{-1} = 4\,200 \text{ J mol}^{-1}$$

$$4\,H_A\Omega' = 4(5\,200 \text{ J mol}^{-1})\,(-5\,400 \text{ J mol}^{-1}) = -1.123\,2 \times 10^8 \text{ J}^2 \text{ mol}^{-2}$$

Now $$a = \left(\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*}\right)^2 = (3.733\,3 \text{ J K}^{-1} \text{ mol}^{-1})^2 = 13.94 \text{ J}^2 \text{ K}^{-2} \text{ mol}^{-2}$$

$$b = 2(-H_A + H_B - \Omega')\left(\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*}\right) + \frac{4\,H_A\Omega'}{T_A^*}$$

$$= 2\,(4\,200 \text{ J mol}^{-1})\,(3.733\,3 \text{ J K}^{-1} \text{ mol}^{-1}) - \frac{(1.1232 \times 10^8 \text{ J}^2 \text{ mol}^{-2}}{(500 \text{ K})}$$

$$= (31\,360 - 224\,640) \text{ J}^2 \text{ K}^{-1} \text{ mol}^{-2}$$

$$= -1.932\,8 \times 10^5 \text{ J}^2 \text{ K}^{-1} \text{ mol}^{-2}$$

$$c = (-H_A + H_B - \Omega')^2 - 4\,H_A\Omega'$$

$$= (4\,200 \text{ J mol}^{-1})^2 + (1.123\,2 \times 10^8 \text{ J}^2 \text{ mol}^{-2})$$

$$= 1.299\,6 \times 10^8 \text{ J}^2 \text{ mol}^{-2}$$

Since $$T_{az} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$, we have

$$T_{az} = \frac{1.9328\times10^5 \pm \sqrt{(1.9328\times10^5)^2 - 4(13.94)(1.2996\times10^8)}}{2(13.94)}$$

$$= \frac{1.9328\times10^5 \pm 1.7352\times10^5}{27.88}\ \text{K}$$

$$= 1.32\times10^4\ \text{K} \quad \text{and} \quad 708.8\ \text{K}$$

The acceptable value of azeotropic temperature is 708.8 K.

The azeotropic composition is given by

$$(x_B)_{az} = \frac{1}{2\Omega'}\left[(H_A - H_B) + \left(\frac{H_B}{T_B^*} - \frac{H_A}{T_A^*}\right)T_{az}\right] + \frac{1}{2}$$

$$= \frac{1}{2(-5400)}\ [(5\,200 - 4\,000) + (-3.733\,3)\ (708.8)] + \frac{1}{2}$$

$$= \frac{1}{2(-5400)}\ [1\,200 - 2\,646] + \frac{1}{2}$$

$$= 0.134 + 0.5 = 0.634$$

(ii) We have

$$\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*} = 3.733\,3\ \text{J K}^{-1}\ \text{mol}^{-1}$$

$$-H_A + H_B - \Omega' = (-5\,200 + 4\,000 - 5\,400)\ \text{J mol}^{-1} = -6\,600\ \text{J mol}^{-1}$$

$$4H_A\,\Omega' = 4(5\,200\ \text{J mol}^{-1})\ (5\,400\ \text{J mol}^{-1}) = 1.123\,2\times10^8\ \text{J}^2\ \text{mol}^{-2}$$

Hence $$a = \left(\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*}\right)^2 = (3.733\,3\ \text{J K}^{-1}\ \text{mol}^{-1})^2 = 13.94\ \text{J}^2\ \text{K}^{-2}\ \text{mol}^{-2}$$

$$b = 2(-H_A + H_B - \Omega')\left(\frac{H_A}{T_A^*} - \frac{H_B}{T_B^*}\right) + \frac{4\,H_A\Omega'}{T_A^*}$$

$$= 2(-6\,600\ \text{J mol}^{-1})\ (3.733\,3\ \text{J K}^{-1}\ \text{mol}^{-1}) + \frac{1.123\,2\times10^8\ \text{J}^2\ \text{mol}^{-2}}{500\ \text{K}}$$

$$= (-49\,279.56 + 224\,640)\text{J}^2\ \text{K}^{-1}\ \text{mol}^{-2} = 1.753\,6\times10^5\ \text{J}^2\ \text{K}^{-1}\ \text{mol}^{-2}$$

$$c = (-H_A + H_B - \Omega')^2 - (1.123\,2\times10^8\ \text{J}^2\ \text{mol}^{-1})$$

$$= (-6\,600\ \text{J mol}^{-1})^2 - (1.123\,2\times10^8\ \text{J}^2\ \text{mol}^{-1})$$

$$= -6.876\times10^7\ \text{J}^2\ \text{mol}^{-1}$$

$$T_{az} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= \frac{-1.7536 \times 10^5 \pm \sqrt{(1.7536 \times 10^5)^2 + 4(13.94)\,(6.876 \times 10^7)}}{2(13.94)} \text{ K}$$

$$= \frac{-1.7536 \times 10^5 \pm 1.860 \times 10^5}{27.88} \text{ K} = -12\,961 \text{ K and } 380.6 \text{ K}$$

The acceptable value is 380.6 K.

The azeotropic composition is

$$(x_B)_{az} = \frac{1}{2\Omega'}\left[(H_A - H_B) + \left(\frac{H_B}{T_B^*} - \frac{H_A}{T_A^*}\right)T_{az}\right] + \frac{1}{2}$$

$$= \frac{1}{2(5\,400)}\ [(5\,200 - 4\,000) + (-3.733\,3)\ (380.6) + \frac{1}{2}$$

$$= \frac{1}{2(5\,400)}\ [1\,200 - 1\,420.90] + 0.5 = (-0.02 + 0.50) = 0.48$$

Illustration Phase diagram of a non-ideal system of A ($\Delta_{fus}H_A$ = 5 200 J mol^{-1}, T_A^* 500 K) and B($\Delta_{fus}H_B$ = 4 000 J mol^{-1}, T_B^* = 600 K). Given: Ω_s = 5 400 J mol^{-1} and Ω_l = 2 200 J mol^{-1}.

Program

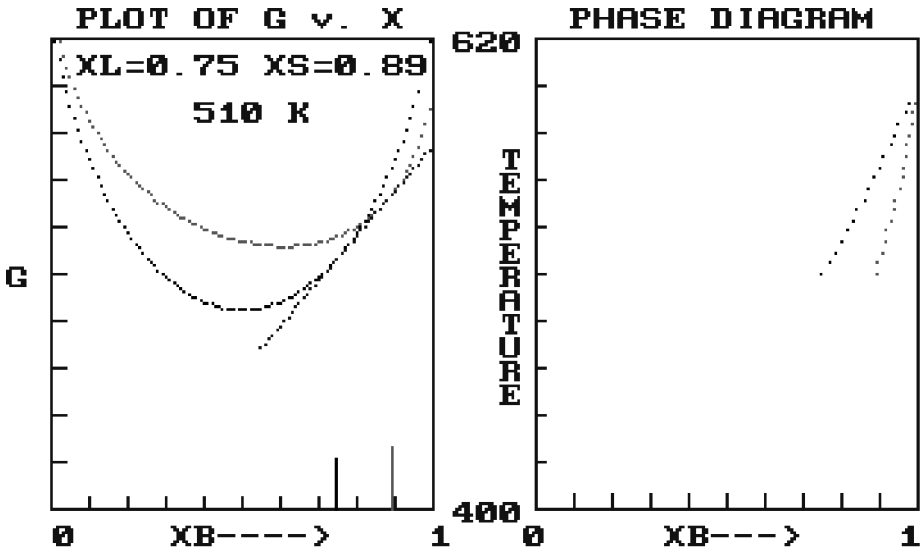
```
CLS
HA = 5200: HB = 4000: TA = 500: TB = 600: TSS = 20
RHOS = 5400: RHOL = 2200: RP = RHOS - RHOL: R = 8.314
Z = HA / TA - HB / TB: Y = -HA + HB - RP: A = Z * Z
B = 2 * Y * Z + 4 * HA * RP / TA: C = Y * Y - 4 * HA * RP
D = (B * B - 4 * A * C) ^ .5: TERM = 1 / (2 * RP)
TAZ1 = (-B + D) / (2 * A): TAZ2 = (-B - D) / (2 * A)
XAZ1 = TERM * (HA - HB - Z * TAZ1) + .5
XAZ2 = TERM * (HA - HB - Z * TAZ2) + .5
IF RP > 0 AND TA < TB THEN
TWIN1 = TB + TSS
ELSE TWIN1 = TA - TSS
END IF
IF RP > 0 AND ABS(TAZ1) < ABS(TAZ2) THEN
TWIN2 = INT(TAZ1 - TSS)
ELSE TWIN2 = INT(TAZ2 + TSS)
END IF
IF RP > 0 THEN
TWIN2 = TWIN2 - TWIN2 MOD 100
ELSE TWIN2 = TWIN2 + 2 * TSS - TWIN2 MOD 100
```

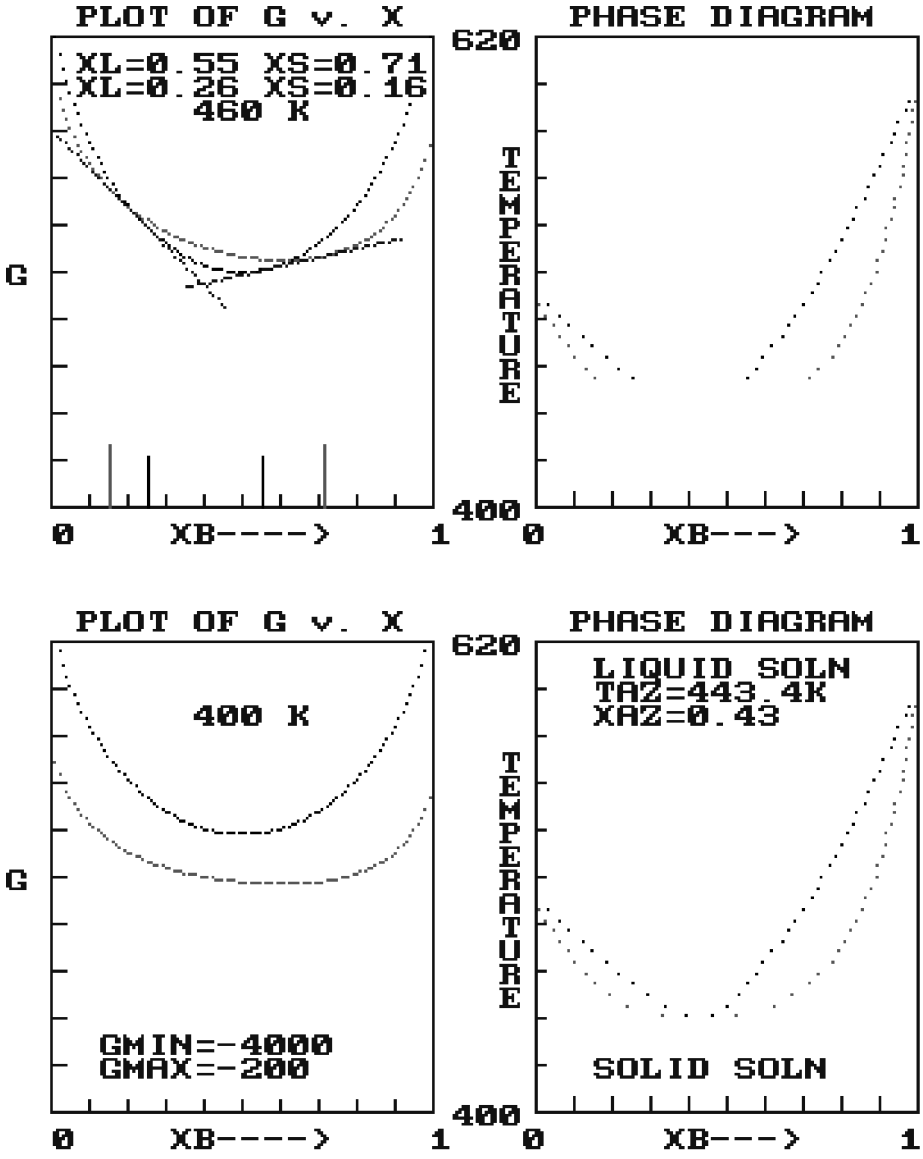
```
END IF
DEF FNM (X, T) = R * T * ((1 - X) * LOG(1 - X) + X * LOG(X))
DEF FNA (X, GA, GB) = (1 - X) * GA + X * GB
DEF FND (X, RHOW) = (1 - X) * X * RHOW
DEF FNS (X, T, RHOS) = FNA(X, GA, GB) + FNM(X, T) + FND(X, RHOS)
DEF FNL (X, T, RHOL) = FNM(X, T) + FND(X, RHOL)
DEF FNP (X) = R * T * (-LOG(1 - X) + LOG(X))
SCREEN 1: COLOR 15, 0
IF RP > 0 THEN TSTEP = -5 ELSE TSTEP = 5
GMIN = -4000: GMAX = -200
FOR T = TWIN1 TO TWIN2 STEP TSTEP
GA = -HA * (1 - T / TA): GB = -HB * (1 - T / TB)
VIEW (15, 10)-(145, 170): CLS
WINDOW (0, GMIN)-(1, GMAX): LINE (0, GMIN)-(1, GMAX), , B
LOCATE 1, 4: PRINT "PLOT OF G v. X"
LOCATE 1, 25: PRINT "PHASE DIAGRAM"
LOCATE 5, 8: PRINT USING "####"; T; : PRINT " K"
FOR XB = .01 TO .99 STEP .01
GS = FNS(XB, T, RHOS): GL = FNL(XB, T, RHOL)
PSET (XB, GS), 1: PSET (XB, GL), 2
NEXT XB
LOCATE 12, 1: PRINT "G"
LOCATE 23, 2: PRINT " 0     XB---->     1"
FOR I = 1 TO 9: LINE (I * .1, GMIN)-(I * .1, GMIN + 100)
GMSC = GMIN + 380 * I: LINE (0, GMSC)-(.04, GMSC)
'LINE (0, GMSC)-(1, GMSC): LINE (I * .1, GMIN)-(I * .1, GMAX)
NEXT I
B = GA - GB - RP: D = B ^ 2 + 4 * GA * RP
IF D < 0 THEN 104
XP1 = (-B + D ^ .5) / (2 * RP): XP2 = (-B - D ^ .5) / (2 * RP)
LL = 1
IF XP1 >= 1 OR XP1 < 0 THEN 5
XL = XP1: XS = XP1
GOTO 6
5 LL = 2
IF XP2 >= 1 OR XP2 < 0 THEN 100
XL = XP2: XS = XP2
6 XT = XL + .01
IF XT >= 1 OR XT < 0 THEN 104
GS = FNS(XT, T, RHOS): GL = FNL(XT, T, RHOL)
INC = .01
IF GS > GL THEN INC = -.01
FOR I = 1 TO 50
XL = XL - INC: XS = XS + INC
IF XL >= 1 OR XL < 0 THEN 104
```

```
IF XS >= 1 OR XS < 0 THEN 104
SLOPS = -GA + GB + FNP(XS) + (1 - 2 * XS) * RHOS
SLOPL = FNP(XL) + (1 - 2 * XL) * RHOL
IF LL = 2 THEN 15
IF SLOPL < SLOPS THEN 20
GOTO 17
15 IF SLOPL > SLOPS THEN 25
17 NEXT I
20 XLL = XL: XSS = XS
LOCATE 3, 4: PRINT "XL="; : PRINT USING "#.##"; XL
LOCATE 3, 12: PRINT "XS="; : PRINT USING "#.##"; XS
GOTO 30
25 XLL1 = XL: XSS1 = XS
LOCATE 4, 4: PRINT "XL="; : PRINT USING "#.##"; XL
LOCATE 4, 12: PRINT "XS="; : PRINT USING "#.##"; XS
30 GS1 = FNS(XS, T, RHOS): GL1 = FNL(XL, T, RHOL)
IF XL < XS THEN
XMN = XL - .2: XMX = XS + .2
ELSE XMN = XS - .2: XMX = XL + .2
END IF
IF XMX < 0 THEN XMN = 0: IF XMX > 1 THEN XMX = 1
FOR XX = XMN TO XMX STEP .01
YY = GS1 + ((GS1 - GL1) / (XS - XL)) * (XX - XS)
PSET (XX, YY)
FOR D = 1 TO 10 STEP .01: NEXT D
NEXT XX
LINE (XS, GMIN)-(XS, GMIN + 500), 1
LINE (XL, GMIN)-(XL, GMIN + 400), 2
LL = LL + 1': A$ = INPUT$(1)
IF LL = 2 THEN 5
100 VIEW (180, 10)-(310, 170)
WINDOW (0, TWIN1)-(1, TWIN2): LINE (0, TWIN1)-(1, TWIN2), , B
PSET (XLL, T), 2: PSET (XSS, T), 1
PSET (XLL1, T), 2: PSET (XSS1, T), 1
FOR K = 1 TO 9
LINE (K * .1, TW2)-(K * .1, TW2 + (TW1 - TW2) / 30)
TTT = TW2 + ((TW1 - TW2) / 10) * K
LINE (0, TTT)-(.02, TTT)
'LINE (K * .1, TW2)-(K * .1, TW1): LINE (0, TTT)-(1, TTT)
NEXT K
104 LOCATE 23, 21: PRINT "   0      XB--->     1"
LOCATE 2, 20
IF RP > 0 THEN
PRINT USING "###"; TWIN1
ELSE PRINT USING "###"; TWIN2
```

```
END IF
A$ = "TEMPERATURE"
FOR I = 1 TO 11: LOCATE 6 + I, 22: PRINT MID$(A$, I, 1): NEXT I
LOCATE 22, 20
IF RP > 0 THEN
PRINT USING "###"; TWIN2
ELSE PRINT USING "###"; TWIN1
END IF
IF RP > 0 THEN TW1 = TWIN1: TW2 = TWIN2 ELSE TW1 = TWIN2: TW2 = TWIN1
A$ = INPUT$(1)
NEXT T
LOCATE 4, 26: PRINT "TAZ=";
IF RP > 0 AND ABS(TAZ1) < ABS(TAZ2) THEN
PRINT USING "###.#"; TAZ1;
ELSE PRINT USING "###.#"; TAZ2;
END IF
PRINT "K": LOCATE 5, 26: PRINT "XAZ=";
IF ABS(TAZ1) < ABS(TAZ2) THEN
PRINT USING "#.##"; XAZ1
ELSE PRINT USING "#.##"; XAZ2
END IF
LOCATE 3, 26: PRINT "LIQUID SOLN"
LOCATE 20, 26: PRINT "SOLID SOLN"
LOCATE 19, 5: PRINT "GMIN="; GMIN
LOCATE 20, 5: PRINT "GMAX="; GMAX
END
```

Output





3.7 DISPLAY OF SHAPES OF ORBITALS

To draw equi-probability contour diagrams enclosing increasing probability of finding an electron in the given orbital of hydrogen atom.

The wave function of an orbital of hydrogen atom ($Z = 1$) is $\Psi_{n,\,l,\,m} = R_{n,\,l}\,\Theta_{l,\,|m|}\,\Phi_m$

The probability distribution is given by $\Psi^2_{n,\,l,\,m} = R^2_{n,\,l}\,\Theta^2_{l,\,|m|}\,\Phi^2_m$

For enclosing increasing probability, we integrate the above expression to give

$$P = \int \Psi^2_{n,l,m}\, \mathrm{d}\tau = \int_0^r R^2_{n,l}\, r^2 \mathrm{d}r \int_0^\pi \Theta^2 \sin\theta\, \mathrm{d}\theta \int_0^{2\pi} \Phi^2\, \mathrm{d}\phi$$

For the normalized Θ and Φ functions, we get

$$P = \int_0^r R^2_{n,l}\, r^2\, \mathrm{d}r$$

For example, for 1s orbital we get

$$P = \int_0^r \left[2\left(\frac{1}{a_0}\right)^{3/2} \mathrm{e}^{-r/a_0} \right]^2 r^2\, \mathrm{d}r = 4\int_0^{r/a_0} \mathrm{e}^{-r/a_0}\ (r/a_0)^2\, \mathrm{d}(r/a_0)$$

The integral on the right-hand side may be evaluated numerically (which is area under the curve) with increasing value of r/a_0 until the integral value becomes equal to the given value of the probability. A sphere of radius r/a_0 with nucleus at the centre gives the region where there exist the given probability of finding the electron in the atomic orbital. In a two-dimensional plots, the probability distribution will be represented by concentric circles.

Illustration
Program

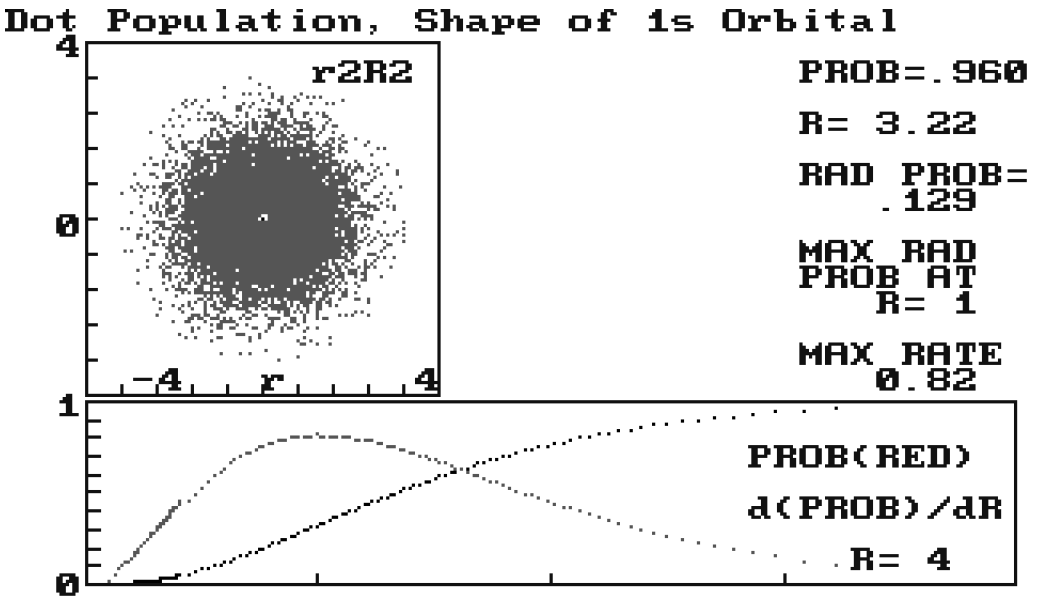
Probability distributions of 1s, 2s, 2p, 3s, 3p, 3d, 4s, 4p and 4d orbitals

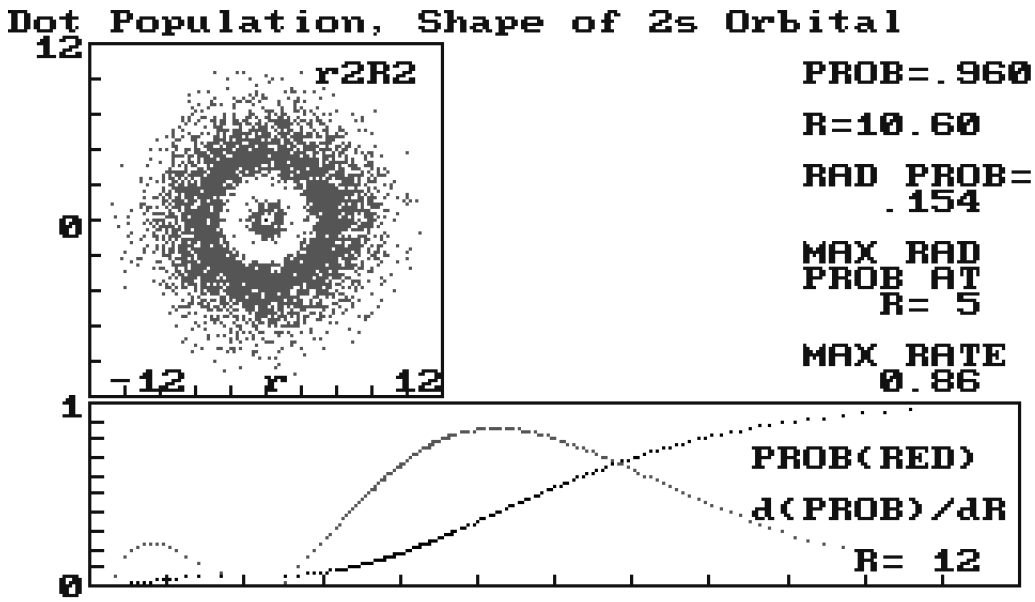
```
REM PROGRAM DOT POPULATION OR SHAPE OF ORBITALS
REM OPT=1 For 1s: 2 For 2s: 3 For 2p: 4 For 3s: 5 For 3p: 6 For 3d
REM OPT=7 for 4s:8 for 4p:9 for 4d
REM OPT1=1 For px: 2 For py: 3 For pz
REM OPT1=1 For dxy: 2 For dxz: 3 For dyz: 4 For dx2-y2: 5 For dz2
REM PINT; CHANGE IT FOR GETTING BETTER NODES
REM PS1 & PS2;PROBABILITY VARIATIONS: ANS;RANDOM No. OF ANGLES
REM FD is rate of variation of probability
REM FDS is multiplication factor for plotting FD
 ' RANDOMIZE
  READ OPT: DATA 9
  PS1 = .002: PS2 = .01: PINT = .3
  ON OPT GOTO 2, 2, 1, 2, 1, 1, 2, 1, 1
1 READ OPT1: DATA 1
2 CLS : SCREEN 1: COLOR 15, 0
  ON OPT GOTO 3, 4, 5, 6, 7, 8, 9, 60, 61
3 N$ = "1s": SC = 4: PS1 = .001: PINT = .05: FDS = 1.5: ANS = 300: GOTO 10
4 N$ = "2s": SC = 12: PS1 = .004: FDS = 4.5: ANS = 300: GOTO 10
5 N$ = "2p": SC = 10: PINT = .1: FDS = 4.5: ANS = 300: GOTO 10
6 N$ = "3s": SC = 22: FDS = 8: ANS = 500: GOTO 10
7 N$ = "3p": SC = 24: FDS = 8: ANS = 500: GOTO 10
8 N$ = "3d": SC = 24: FDS = 8: ANS = 500: GOTO 10
9 N$ = "4s": SC = 36: FDS = 8: ANS = 1000: GOTO 10
60 N$ = "4p": SC = 36: FDS = 8: ANS = 1000: GOTO 10
61 N$ = "4d": SC = 36: FDS = 8: ANS = 1000
10 LOCATE 1, 1: PRINT "Dot Population, Shape of "; N$;
```

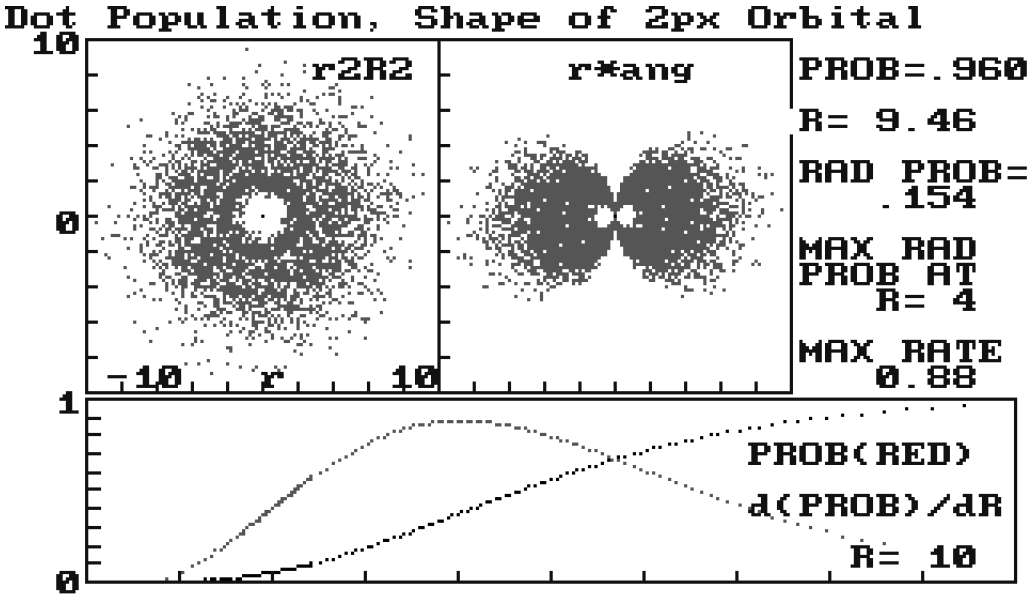
```
   ON OPT GOTO 22, 22, 11, 22, 11, 12, 22, 11, 12
11 ON OPT1 GOTO 13, 14, 15
13 PRINT "x"; : GOTO 22
14 PRINT "y"; : GOTO 22
15 PRINT "z"; : GOTO 22
12 ON OPT1 GOTO 17, 18, 19, 20, 21
17 PRINT "xy"; : GOTO 22
18 PRINT "xz"; : GOTO 22
19 PRINT "yz"; : GOTO 22
20 PRINT "x2-y2"; : GOTO 22
21 PRINT "z2"; : GOTO 22
22 PRINT " Orbital": LOCATE 2, 1: PRINT USING "###"; SC:
    LOCATE 9, 3: PRINT "0"': LOCATE 15, 1: PRINT USING "##"; -SC
    LOCATE 15, 5: PRINT USING "###"; -SC: LOCATE 15, 11: PRINT "r"
    LOCATE 15, 16: PRINT USING "##"; SC
    LOCATE 18, 30: PRINT "PROB(RED)"
    LOCATE 20, 30: PRINT "d(PROB)/dR"
    RI = 0: AREA = 0: KK% = 0: JJ% = 1: K1 = 1: FDMAX = 0
    PROBI = 0: PROBF = PINT: PROBS = PS1: GOSUB 23
    PROBI = PINT: PROBF = .96: PROBS = PS2: GOSUB 23
    LOCATE 16, 3: PRINT "1"
    LOCATE 22, 34: PRINT "R="; SC: LOCATE 23, 3: PRINT "0"
    GOTO 90
23 FOR PROB = PROBI TO PROBF STEP PROBS
    DR = .0001
    FOR R = RI TO 100 STEP DR
    ON OPT GOTO 25, 26, 27, 28, 29, 30, 31, 72, 73
25 RR = 2 * EXP(-R): GOTO 32
26 RR = (1 / 2) ^ 1.5 * (2 - R) * EXP(-R / 2): GOTO 32
27 RR = (1 / 3 ^ .5) * (1 / 2) ^ 1.5 * R * EXP(-R / 2): GOTO 32
28 RR = (2 / 3) * (1 / 3) ^ 1.5 * (3 - 2 * R + 2 * R ^ 2 / 9)
    RR = RR * EXP(-R / 3): GOTO 32
29 RR = ((2 * 2 ^ .5) / 9) * (1 / 3) ^ 1.5 * (2 * R - R ^ 2 / 3)
    RR = RR * EXP(-R / 3): GOTO 32
30 RR = (4 / (27 * 10 ^ .5)) * (1 / 3) ^ 1.5 * R ^ 2 * EXP(-R / 3): GOTO 32
31 RR = (1 / 768) * (192 - 144 * R + 24 * R ^ 2 - R ^ 3) * EXP(-R / 4): GOTO 32
72 RR = (3 / (768 * 15 ^ .5)) * (80 - 20 * R + R ^ 2) * R * EXP(-R / 4): GOTO 32
73 RR = (1 / (768 * 5 ^ .5)) * (12 - R) * R ^ 2 * EXP(-R / 4)
32 AREA = AREA + RR ^ 2 * R ^ 2 * DR
    IF AREA > PROB THEN 35
    NEXT R
35 RI = R: PSET (0, 0), 2: KK = INT(FD * ANS)
    FOR K = 0 TO KK
    AN = INT(RND * 360): TH = 3.1415 * AN / 180
    RC = R * COS(TH): RS = R * SIN(TH)
    VIEW (25, 10)-(135, 120): WINDOW (-SC, -SC)-(SC, SC)
    LINE (-SC, -SC)-(SC, SC), , B: GOSUB 80
    PSET (RS, RC), 1: PSET (0, 0), 2
```

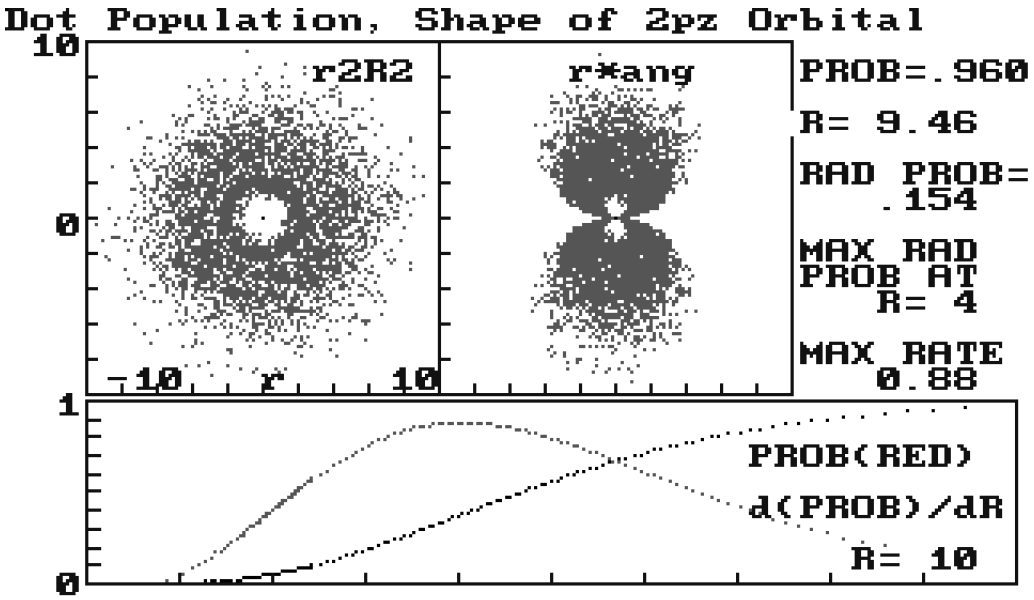
```
    ON OPT GOTO 55, 55, 39, 55, 39, 44, 55, 39, 44
39 ON OPT1 GOTO 40, 40, 42
40 ORTH = SIN(TH): GOTO 52
42 ORTH = COS(TH): GOTO 52
44 ON OPT1 GOTO 46, 46, 46, 48, 50
46 ORTH = SIN(2 * TH): GOTO 52
48 ORTH = COS(2 * TH): GOTO 52
50 ORTH = (3 * COS(TH) ^ 2 - 1) / 1.7
52 RC1 = RC * ORTH: RS1 = RS * ORTH
    LOCATE 3, 23: PRINT "r*ang"
    VIEW (135, 10)-(245, 120): WINDOW (-SC, -SC)-(SC, SC)
    LINE (-SC, -SC)-(SC, SC), , B: GOSUB 80
    PSET (RS1, RC1), 1: PSET (-RS1, -RC1), 1: PSET (0, 0), 2
55 LOCATE 3, 13: PRINT "r2R2"
    NEXT K
    PSET (0, 0), 2: GOSUB 80
    LOCATE 5, 32: PRINT "PROB="; : PRINT USING ".###"; PROB
    LOCATE 7, 32: PRINT " R="; : PRINT USING "##.##"; R
    LOCATE 9, 32: PRINT "RAD PROB=": LOCATE 10, 37
    PRINT USING ".###"; FD
   ' B$ = INPUT$(1)
   VIEW (25, 122)-(315, 179): WINDOW (0, 0)-(SC, 1)
    LINE (0, 0)-(SC, 1), , B: PSET (R, PROB), 2
    FOR I = 1 TO SC - 1: LINE (I, 0)-(I, .05): NEXT I
   FOR I = 1 TO 9: II = .1 * I: LINE (0, II)-(SC / 80, II): NEXT I
    IF R - R1 = 0 THEN 75
    FD = (PROB - PROB1) / (R - R1): PSET (R, FD * FDS), 1
    ON K1 GOTO 53, 54
53 IF FDMAX > FD THEN 56
    FDMAX = FD: GOTO 75
54 IF FDMAX > FD THEN 75
    FDMAX = FD: K1 = 1
    GOTO 53
56 LOCATE 13, 32: PRINT "MAX RAD"
    LOCATE 14, 32: PRINT "PROB": LOCATE 15, 32
    PRINT "R="; : PRINT USING "##"; R;
    PRINT USING " #.##"; FDMAX * FDS
    K1 = 2: 'A$ = INPUT$(1)
75 PROB1 = PROB: R1 = R
    KK% = KK% + 1
    NEXT PROB
    RETURN
80 FOR I = 1 TO 9: S1 = -SC + I * .2 * SC: S2 = -SC + SC / 20
    LINE (S1, -SC)-(S1, S2): LINE (-SC, S1)-(S2, S1): NEXT I
    RETURN
90 END
```

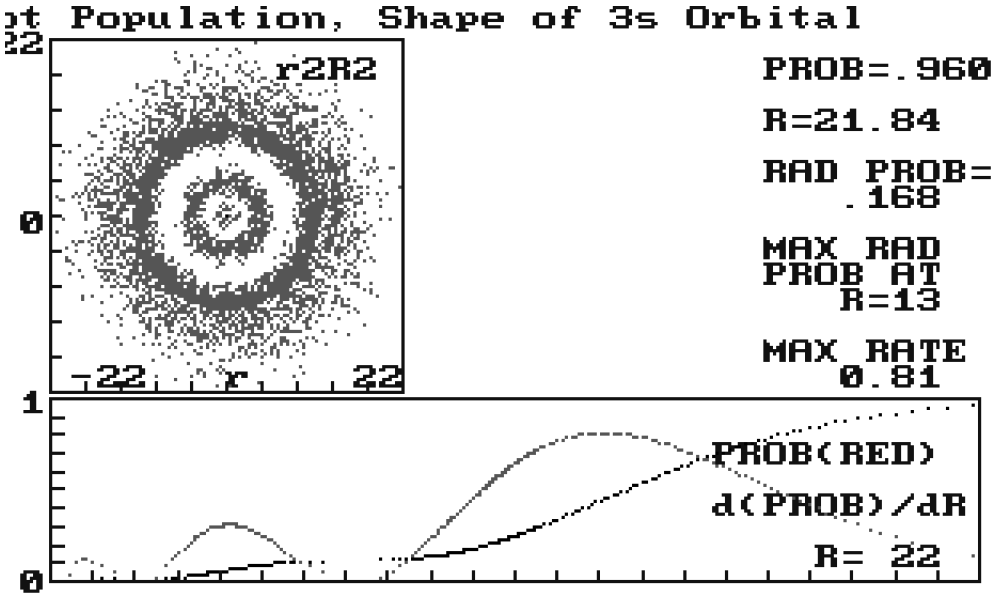
Output

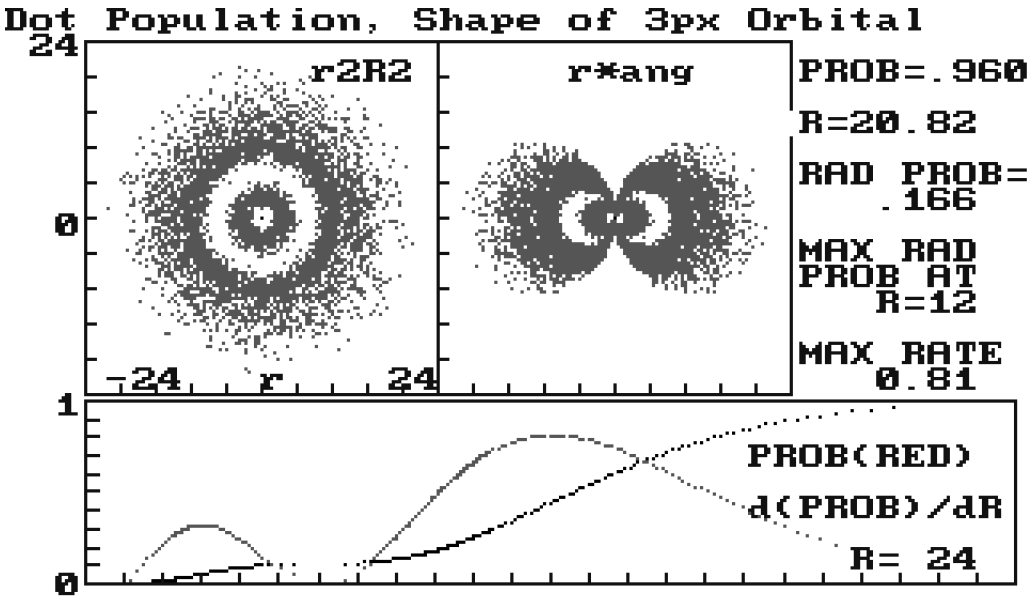

Dot Population, Shape of 1s Orbital
4
0
r2R2
-4
r
4
PROB=.960
R= 3.22
RAD PROB=
.129
MAX RAD
PROB AT
R= 1
MAX RATE
0.82
1
0
PROB(RED)
d(PROB)/dR
R= 4

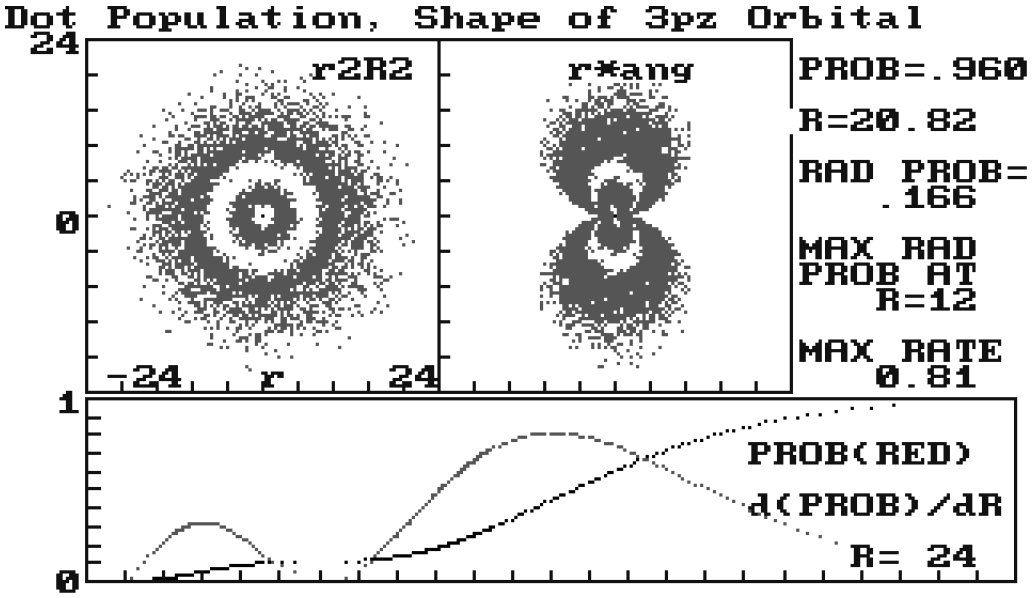

Dot Population, Shape of 2s Orbital
12
0
r2R2
-12
r
12
PROB=.960
R=10.60
RAD PROB=
.154
MAX RAD
PROB AT
R= 5
MAX RATE
0.86
1
0
PROB(RED)
d(PROB)/dR
R= 12

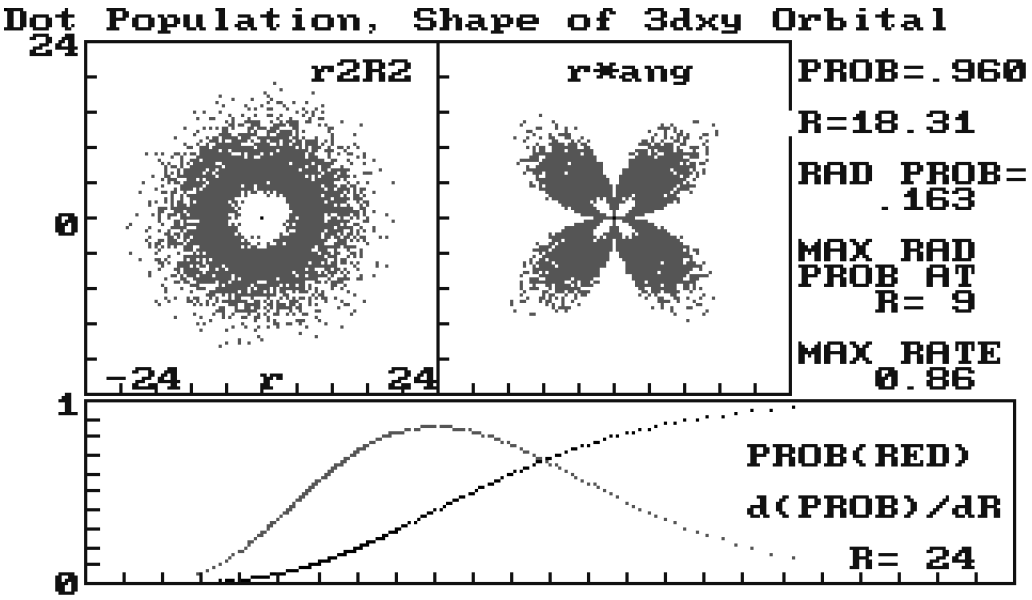

Dot Population, Shape of 2px Orbital
10
0
1
0
r2R2
r*ang
−10
r
10
PROB=.960
R= 9.46
RAD PROB=
.154
MAX RAD
PROB AT
R= 4
MAX RATE
0.88
PROB(RED)
d(PROB)/dR
R= 10

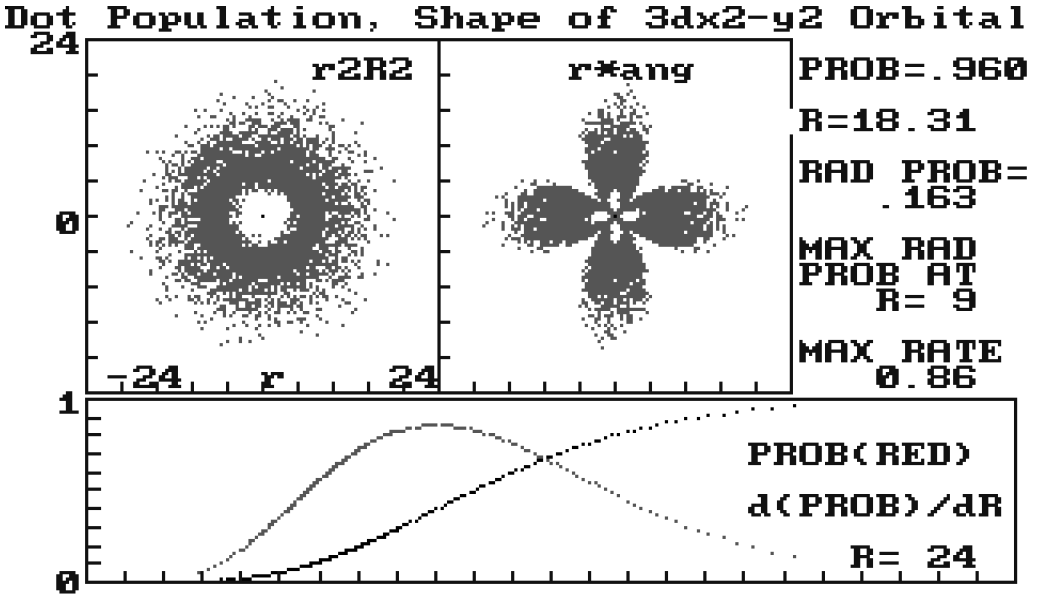

Dot Population, Shape of 2pz Orbital
10
0
1
0
r2R2
r*ang
−10
r
10
PROB=.960
R= 9.46
RAD PROB=
.154
MAX RAD
PROB AT
R= 4
MAX RATE
0.88
PROB(RED)
d(PROB)/dR
R= 10

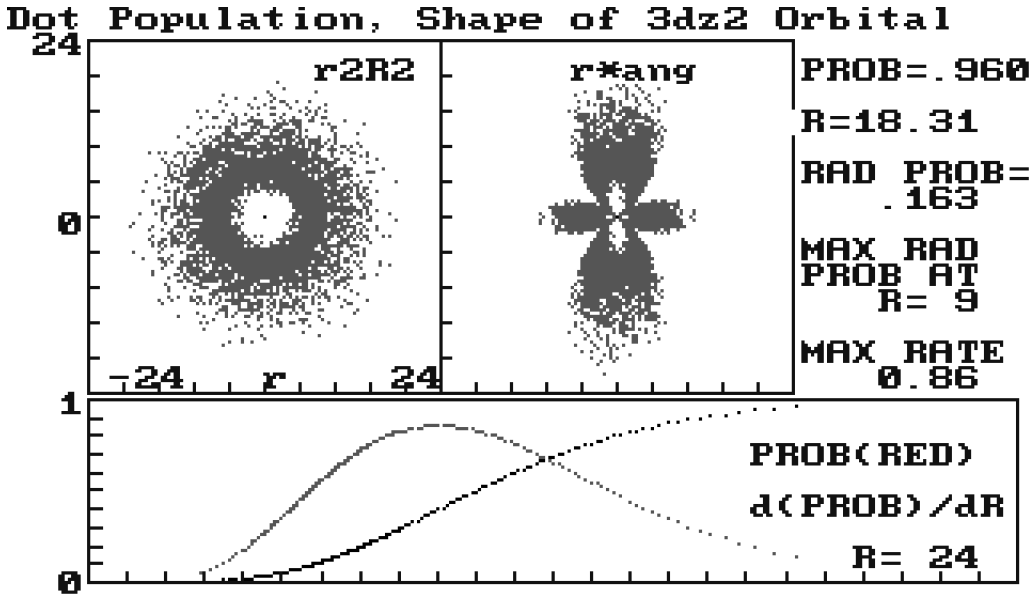

ot Population, Shape of 3s Orbital
22
r2R2
0
-22
r
22
1
0
PROB=.960
R=21.84
RAD PROB=
.168
MAX RAD
PROB AT
R=13
MAX RATE
0.81
PROB(RED)
d(PROB)/dR
R= 22

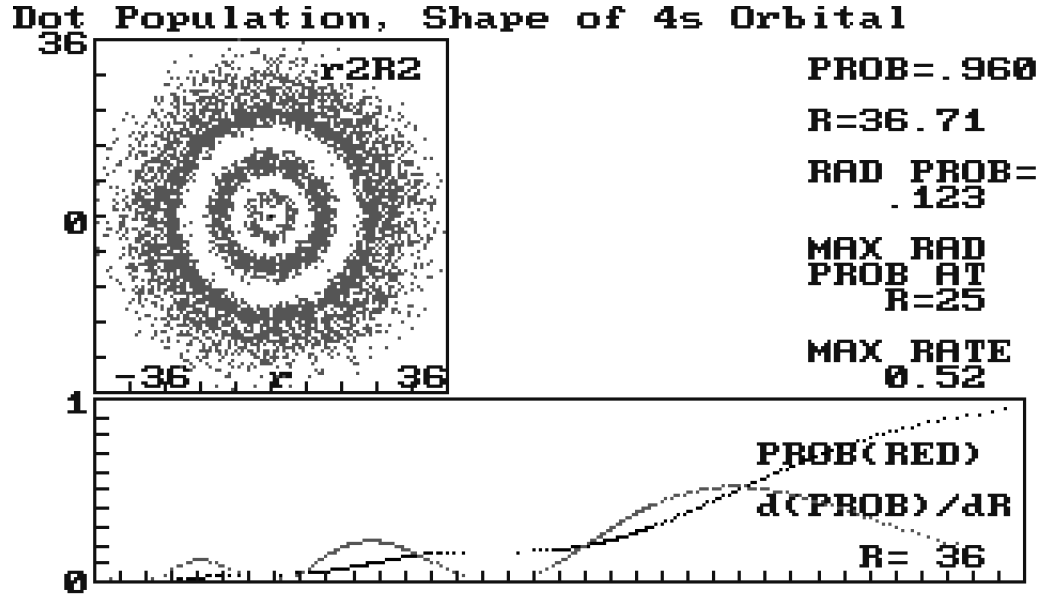

Dot Population, Shape of 3px Orbital
24
r2R2
r*ang
0
-24
r
24
1
0
PROB=.960
R=20.82
RAD PROB=
.166
MAX RAD
PROB AT
R=12
MAX RATE
0.81
PROB(RED)
d(PROB)/dR
R= 24

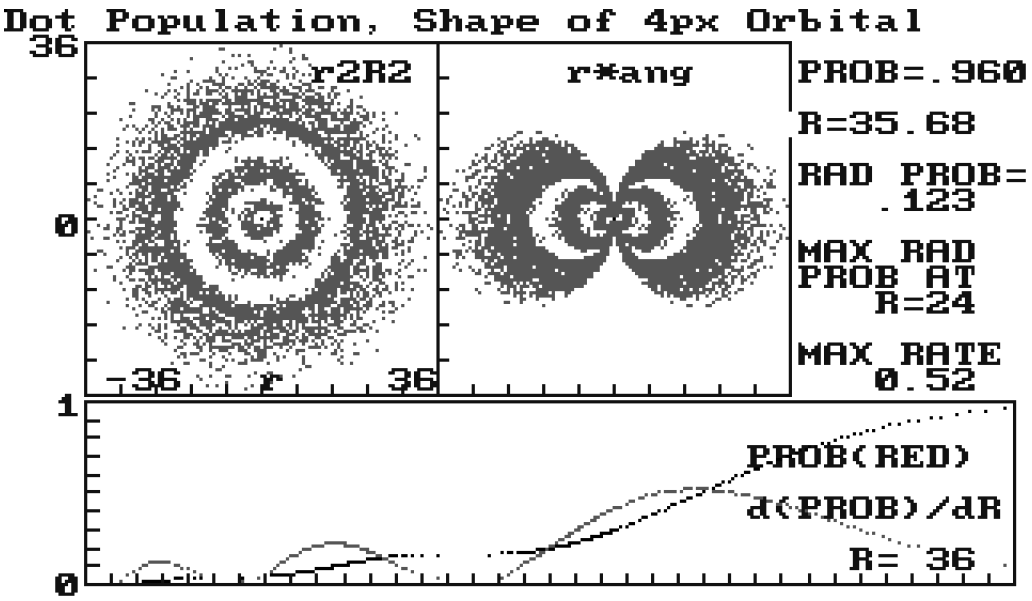

Dot Population, Shape of 3pz Orbital
24
0
1
0
r2R2
r*ang
-24
r
24
PROB=.960
R=20.82
RAD PROB=
.166
MAX RAD
PROB AT
R=12
MAX RATE
0.81
PROB(RED)
d(PROB)/dR
R= 24

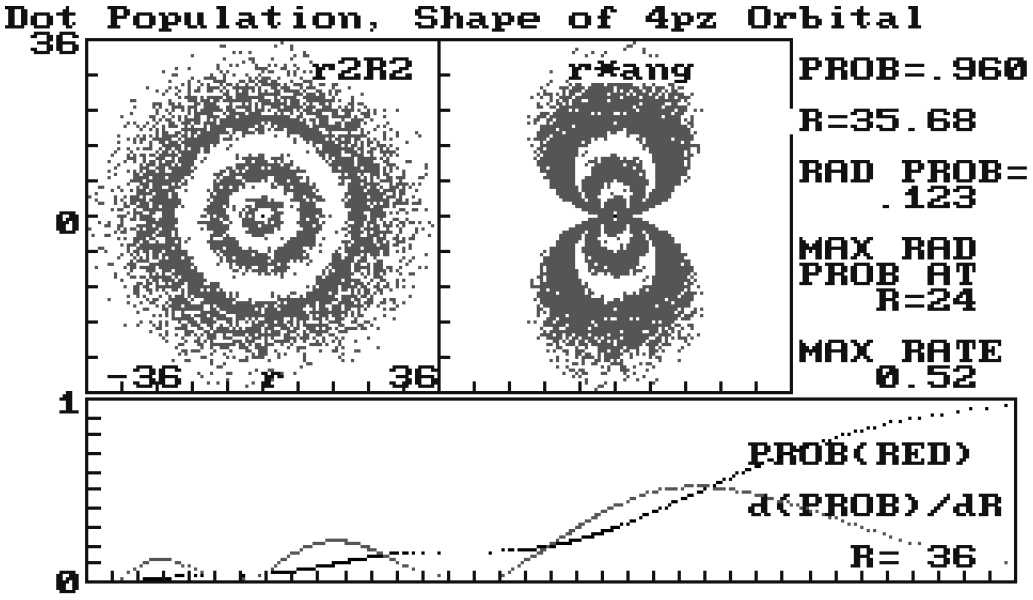

Dot Population, Shape of 3dxy Orbital
24
0
1
0
r2R2
r*ang
-24
r
24
PROB=.960
R=18.31
RAD PROB=
.163
MAX RAD
PROB AT
R= 9
MAX RATE
0.86
PROB(RED)
d(PROB)/dR
R= 24

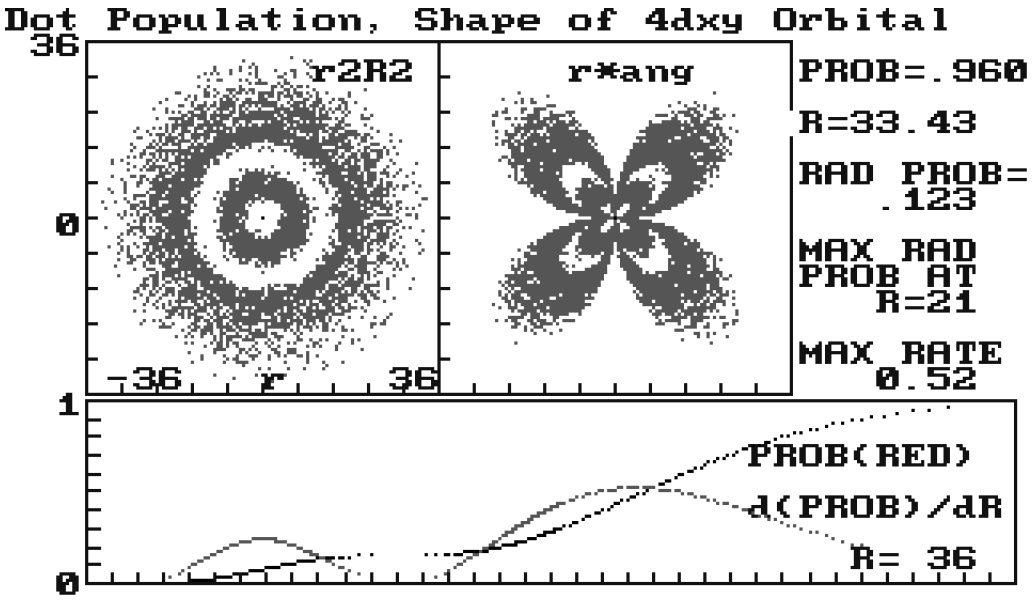

Dot Population, Shape of 3dx2-y2 Orbital
24
0
r2R2
r*ang
-24
r
24
PROB=.960
R=18.31
RAD PROB=
.163
MAX RAD
PROB AT
R= 9
MAX RATE
0.86
1
0
PROB(RED)
d(PROB)/dR
R= 24

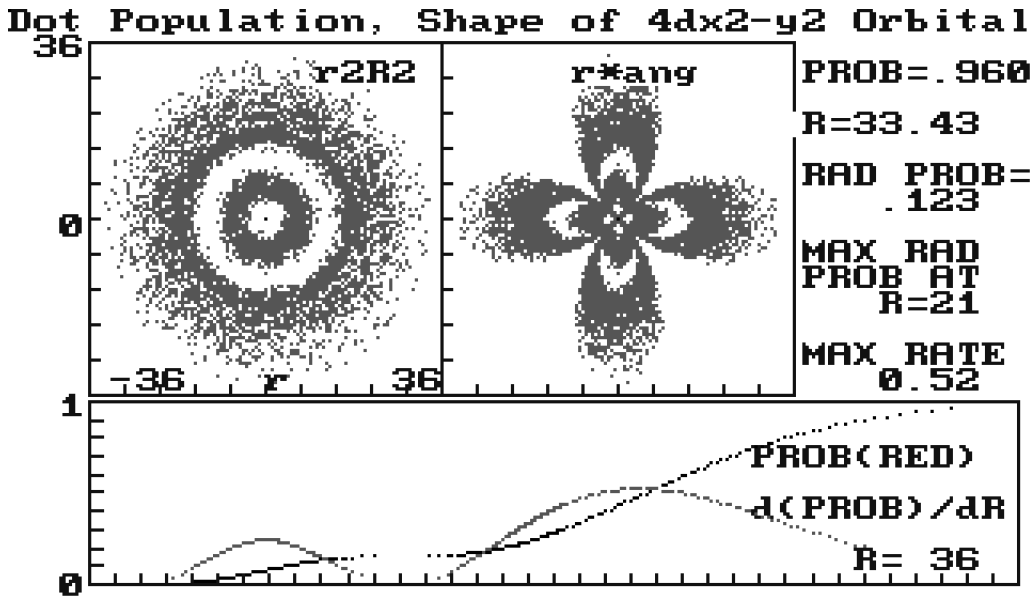

Dot Population, Shape of 3dz2 Orbital
24
0
r2R2
r*ang
-24
r
24
PROB=.960
R=18.31
RAD PROB=
.163
MAX RAD
PROB AT
R= 9
MAX RATE
0.86
1
0
PROB(RED)
d(PROB)/dR
R= 24

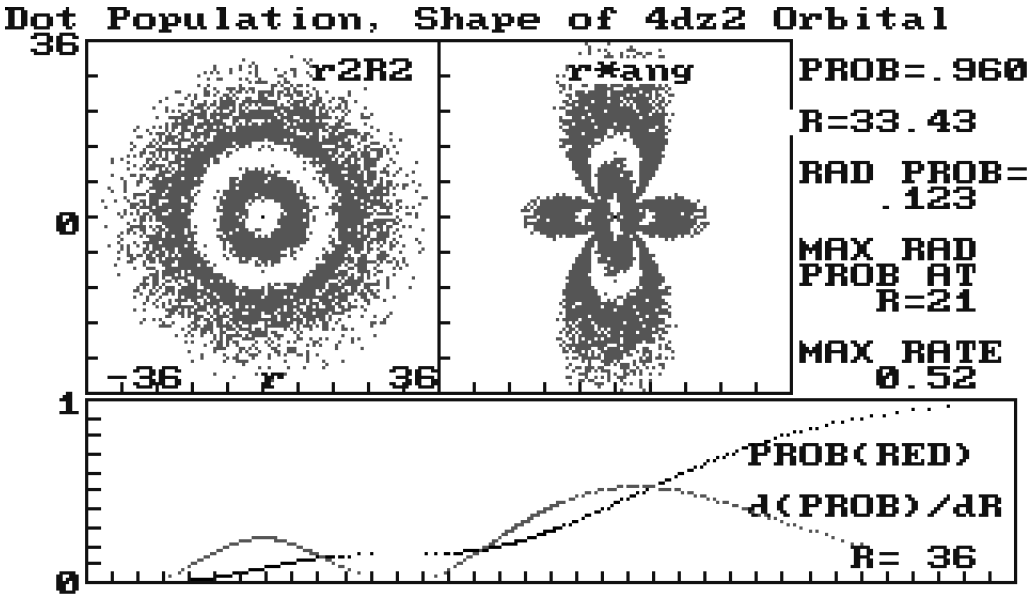

Dot Population, Shape of 4s Orbital
36
r2R2
0
-36
r
36
PROB=.960
R=36.71
RAD PROB=
.123
MAX RAD
PROB AT
R=25
MAX RATE
0.52
1
PROB(RED)
d(PROB)/dR
R= 36
0


Dot Population, Shape of 4px Orbital
36
r2R2
r*ang
0
-36
r
36
PROB=.960
R=35.68
RAD PROB=
.123
MAX RAD
PROB AT
R=24
MAX RATE
0.52
1
PROB(RED)
d(PROB)/dR
R= 36
0


Dot Population, Shape of 4pz Orbital
36
0
1
0
r2R2
r*ang
-36
r
36
PROB=.960
R=35.68
RAD PROB=
.123
MAX RAD
PROB AT
R=24
MAX RATE
0.52
PROB(RED)
d(PROB)/dR
R= 36


Dot Population, Shape of 4dxy Orbital
36
0
1
0
r2R2
r*ang
-36
r
36
PROB=.960
R=33.43
RAD PROB=
.123
MAX RAD
PROB AT
R=21
MAX RATE
0.52
PROB(RED)
d(PROB)/dR
R= 36


Dot Population, Shape of 4dx2-y2 Orbital
36
0
1
0
r2R2
r*ang
-36
r
36
PROB=.960
R=33.43
RAD PROB=
.123
MAX RAD
PROB AT
R=21
MAX RATE
0.52
PROB(RED)
d(PROB)/dR
R= 36


Dot Population, Shape of 4dz2 Orbital
36
0
1
0
r2R2
r*ang
-36
r
36
PROB=.960
R=33.43
RAD PROB=
.123
MAX RAD
PROB AT
R=21
MAX RATE
0.52
PROB(RED)
d(PROB)/dR
R= 36

3.8 EIGENVALUE PROBLEM

The eigenvalue of a matrix $\boldsymbol{A}$ are the solutions of the characteristic equations

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = \begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0$$

This determinant provides a characteristic polynomial of $\boldsymbol{A}$, which is of degree n in λ. For a Hermitian matrix*, all the n values of λ are real. This matrix has a basis of eigenvectors in C^n. A matrix $\boldsymbol{X}$ formed from these vectors as column vector can diagonlize the Hermitian matrix $\boldsymbol{A}$ through the transformation

$$\boldsymbol{D} = \boldsymbol{X}^{-1}\, \boldsymbol{A}\, \boldsymbol{X}$$

In the diagonalized matrix, the eigenvalues of $\boldsymbol{A}$ occupy the main diagonal positions. We describe a method which is widely used in finding the eigenvalues of a symmetric matrix. This method involves two stages as described in the following.

Stage I Householder's Method

In the Householder's method, the given symmetric matrix is transformed into a tridiagonal matrix which include nonzero entries on the main diagonal and in positions immediately adjacent to the main diagonal. This transformation of a real symmetric $n \times n$ matrix $\boldsymbol{A} = [a_{jk}]$ into tridiagonal matrix is achieved by $n - 2$ successive similarity transformations.† The transformations go as follows.

$$\boldsymbol{A}^{(1)} = \boldsymbol{P}^{(1)} \boldsymbol{A}^{(0)}\, \boldsymbol{P}^{(1)}$$

$$\boldsymbol{A}^{(2)} = \boldsymbol{P}^{(2)}\, \boldsymbol{A}^{(1)}\, \boldsymbol{P}^{(2)}$$

$$\boldsymbol{B} = \boldsymbol{A}^{(n-2)} = \boldsymbol{P}^{(n-2)}\, \boldsymbol{A}^{(n-3)}\, \boldsymbol{P}^{(n-2)}$$

where the matrices $\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \cdots, \boldsymbol{P}^{(n-1)}$ are orthogonal symmetric matrices‡. The first transformattion creates the necessary zeros in row 1 and column 1, the second transformation creates zeros in row 2 and column 2, and so on.

The matrices $\boldsymbol{P}^{(r)}$'s are of the form

$$\boldsymbol{P}^{(r)} = \boldsymbol{I} - 2\, \boldsymbol{v}^{(r)}\, [\boldsymbol{v}^{(r)}]^{\mathrm{T}} \qquad r = 1, 2, \ldots, n-2$$

where $\boldsymbol{v}^{(r)}$'s are vectors whose elements are computed as follows.

$$v_j^{(r)} = 0 \qquad j = 1 \text{ to } r$$

---

*A matrix is Hermitian if $\overline{\boldsymbol{A}}^{\mathrm{T}} = \boldsymbol{A}$, that is $\overline{a}_{kj} = a_{jk}$. If the matrix is real then $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$. Thus, it is a symmetric matrix.

†The superscripts within the brackets (i.e. (1), (2),....) to the matrices represent first, second, . . ., $(n-2)$th similiarity transformations

‡Similarity transformation involves the operation $\boldsymbol{B} = \boldsymbol{T}^{-1}\, \boldsymbol{A}\, \boldsymbol{T}$

The matrix $\boldsymbol{B}$ is called similar to $\boldsymbol{A}$, i.e. the matrix $\boldsymbol{B}$ has the same eigenvalues as those of the matrix $\boldsymbol{A}$.

$$v_{r+1}^{(r)} = \sqrt{\frac{1}{2}\left(1 + \frac{\left|a_{r+1,r}^{(r-1)}\right|}{S_r}\right)}$$

$$v_j^{(r)} = \frac{a_{j,r}^{(r-1)} \operatorname{sgn}\left(a_{r+1,r}^{(r-1)}\right)}{2 v_{r+1}^{(r)} S_r}; \qquad j = r + 2, \ldots, n$$

$$S_r = \sqrt{\sum_{j=r+1}^{n} \left[a_{j,r}^{(r-1)}\right]^2}$$

where $\operatorname{sgn}(a_{r+1,r}^{(r-1)}) = 1$ if $a_{r+1,r}^{(r-1)} \geq 0$ and is -1 if $a_{r+1,r}^{(r-1)} < 0$.

Stage II QR-Factorization

In the QR-fractorization, the off diagonal of the tridiagonal matrix are made progressively smaller and smaller until these are nearly equal to zero. The procedure involves the following steps.

Let $\boldsymbol{B}^{(0)}$ be the tridiagonal matrix. This matrix has $n - 1$ generally nonzero elements below the main diagonal. These are b_{21}, b_{32}, ..., $b_{n,\, n-1}$. In the first step, these elements are made equal to zero one by one.

Making $b_{21}^{(0)}$ equal to zero A matrix $\boldsymbol{C}_2$ is constructed in which the submatrix

$$\begin{bmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{bmatrix}$$

is inserted in the first and second rows and the first and second columns. The remaining main diagonal elements are set equal to unity. All the rest of elements are set equal to zero. Thus, the matrix $\boldsymbol{C}_2$ is

$$\boldsymbol{C}_2 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 & \ldots & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

The matrix $\boldsymbol{B}^{(0)}$ is left multiplied by $\boldsymbol{C}_2$ so as to give $\boldsymbol{B}^{(1)} = \boldsymbol{C}_2 \boldsymbol{B}^{(0)}$. In $\boldsymbol{C}_2$, θ_2 is chosen so as to give $b_{21}^{(1)}$ equal to zero. This element is obtained by multiplying the second row of $\boldsymbol{C}_2$ with the first column of $\boldsymbol{B}^{(0)}$. Thus, we have

$$(-\sin\theta_2)\, b_{11}^{(0)} + (\cos\theta_2)\, b_{21}^{(0)} = 0$$

This gives $\tan \theta_2 = b_{21}^{(0)}/b_{11}^{(0)}$.

From this, $\cos \theta_2$ and $\sin \theta_2$ are evaluated by the expressions

$$\cos \theta_2 = \frac{1}{\sqrt{1 + \tan^2 \theta_2}} \quad \text{and} \quad \sin \theta_2 = \frac{\tan \theta_2}{\sqrt{1 + \tan^2 \theta_2}}$$

After evaluating $\cos \theta_2$ and $\sin \theta_2$, the matrix $\boldsymbol{C}_2$ is formed and then right multiplied by $\boldsymbol{B}^{(0)}$.

Making $b_{32}^{(1)}$ equal to zero A matrix $\boldsymbol{C}_3$ is constructed in which the submatrix

$$\begin{bmatrix} \cos \theta_3 & \sin \theta_3 \\ -\sin \theta_3 & \cos \theta_3 \end{bmatrix}$$

is inserted in the second and third rows and the second and third columns. The remaining main diagonal are set equal to unity and all other elements are set equal to zero. Thus, the matrix is

$$\boldsymbol{C}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \cos \theta_3 & \sin \theta_3 & 0 & \dots & 0 \\ 0 & -\sin \theta_3 & \cos \theta_3 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

The matrix $\boldsymbol{B}^{(1)}$ is left multiplied by $\boldsymbol{C}_3$ so as to give $\boldsymbol{B}^{(2)} = \boldsymbol{C}_3 \boldsymbol{B}^{(1)}$. In $\boldsymbol{C}_3$, θ_2 is chosen so as to give $b_{32}^{(1)}$ equals to zero. This element is obtained by multiplying the third row $\boldsymbol{C}_3$ with the second column of $\boldsymbol{B}^{(1)}$. Thus, we have

$$(-\sin \theta_3)\, b_{22}^{(1)} + (\cos \theta_3)\, b_{23}^{(1)} = 0$$

This gives $\tan \theta_3 = b_{23}^{(1)}/b_{22}^{(1)}$

From this, $\cos \theta_3$ and $\sin \theta_3$ are evaluated by using the expressions given above.

After evaluating $\cos \theta_3$ and $\sin \theta_3$, the matrix C_3 is formed is then right multiplied by $\boldsymbol{B}^{(1)}$.

The above procedure is continued to make each of the element $b_{43}^{(2)}$, $b_{54}^{(3)}$, ..., of the resultant matrices $\boldsymbol{B}^{(2)}$, $\boldsymbol{B}^{(3)}$, ..., respectively, equal to zero.

Finally, we get the matrix

$$\boldsymbol{R}_0 = \boldsymbol{C}_n \boldsymbol{C}_{n-1} \dots \boldsymbol{C}_3 \boldsymbol{C}_2 \boldsymbol{B}^{(0)}$$

From this, it follows that

$$\begin{aligned} \boldsymbol{B}^{(0)} &= (\boldsymbol{C}_n \boldsymbol{C}_{n-1} \dots \boldsymbol{C}_3 \boldsymbol{C}_2)^{-1} \boldsymbol{R}_0 \\ &= \boldsymbol{Q}_0 \boldsymbol{R}_0 \end{aligned}$$

All $\boldsymbol{C}$'s matrices are orthogonal. Hence, their products is also orthogonal and so is the inverse of the product. Thus, we will have

$$\boldsymbol{Q}_0 = (\boldsymbol{C}_n \boldsymbol{C}_{n-1} \ldots \boldsymbol{C}_3 \boldsymbol{C}_2)^{-1} = \boldsymbol{C}_2^{\mathrm{T}} \boldsymbol{C}_3^{\mathrm{T}} \ldots \boldsymbol{C}_{n-1}^{\mathrm{T}} \boldsymbol{C}_n^{\mathrm{T}}$$

where the superscript T stand for the transpose of the indicated matrix.

In the second step, the matrix $\boldsymbol{B}_1^{(0)}$ is computed from the expression

$$\boldsymbol{B}_1^{(0)} = \boldsymbol{R}_0 \boldsymbol{Q}_0 = \boldsymbol{R}_0 \boldsymbol{C}_2^{\mathrm{T}} \boldsymbol{C}_3^{\mathrm{T}} \ldots \boldsymbol{C}_{n-1}^{\mathrm{T}} \boldsymbol{C}_n^{\mathrm{T}}$$

It is not necessary to form $\boldsymbol{Q}_0$ and then left multiply it by $\boldsymbol{R}_0$ but the operations $(\boldsymbol{R}_0\boldsymbol{C}_2^{\mathrm{T}})$, $(\boldsymbol{R}_0\boldsymbol{C}_2^{\mathrm{T}})\,\boldsymbol{C}_3^{\mathrm{T}}$, ..., may be carried out one after the other.

The entire above procedure is now repeated with the matrix $\boldsymbol{B}_1^{(0)}$ to give finally $\boldsymbol{B}_2^{(0)}$. The off-diagonal element of $\boldsymbol{B}_2^{(0)}$ will be smaller than those of $\boldsymbol{B}_1^{(0)}$. This procedure of forming $\boldsymbol{B}_3^{(0)}$, $\boldsymbol{B}_4^{(0)}$, ..., is continued until all the off-diagonal elements become smaller than the pre-selected value near to zero. Eventually, the main diagonal elements represent eigenvalues of the given matrix.

If the tridiagonal matrix contains zeros in the main diagonal, then the determination of tan θ_2 would involve division by zero. In order to avoid such a difficulty, the eigenvalues can still be obtained based on the following theorem.

If the matrix $\boldsymbol{A}$ has the eigenvalues $\lambda_1, \ldots, \lambda_n$, then the matrix $\boldsymbol{A} + k\boldsymbol{I}$ has the eigenvalues $\lambda_1 + k, \ldots, \lambda_n + k$, from these $\lambda_1, \ldots, \lambda_n$ may be evaluated.

This procedure finds applications in the Hückel molecular orbital calculations. The main diagonal elements are made nonzero by adding a constant (say, $k = 1$) in the main diagonal. The eigenvalues of Hückel matrix will be obtained by subtracting k from the resultant eigenvalues.

Illustration Determine the eigenvalues of $\begin{bmatrix} 6 & 4 & 1 & 1 \\ 4 & 6 & 1 & 1 \\ 1 & 1 & 5 & 2 \\ 1 & 1 & 2 & 5 \end{bmatrix}$

Program

```
CLS
REM Only data for lower triangular matrix is to be given
REM N=order of the matrix:  MI=number of iterations in QR
READ N, MI: DATA 4,20
FOR I = 1 TO N: FOR J = 1 TO I
READ A(I, J): A(J, I) = A(I, J)
NEXT J: NEXT I
PRINT "Given Matrix"
FOR I = 1 TO N: FOR J = 1 TO N
PRINT A(I, J); : AD(I, J) = A(I, J)
NEXT J: PRINT : NEXT I
A$ = INPUT$(1)
DATA 6,4,6,1,1,5,1,1,2,5
'DATA 11.4,2,14.4,.6,1.2,14.6
```

```
'DATA 1,1,1,0,1,1,0,0,1,1
'DATA 1,1,1
'DATA 1,1,1,0,1,1,0,0,1,1,0,0,0,1,1,1,0,0,0,1,1
FOR L = 1 TO N - 2
S1 = 0
FOR I = L + 1 TO N: S1 = S1 + A(I, L) ^ 2: NEXT I
S1 = SQR(S1)
PRINT : PRINT "V"; L; " vector"
FOR I = 1 TO L: V(I, I) = 0: PRINT V(I, I): NEXT I
V(L + 1, L) = SQR(.5 * (1 + ABS(A(L + 1, L)) / S1))
PRINT V(L + 1, L)
FOR J = L + 2 TO N
V(J, L) = A(J, L) * SGN(A(L + 1, L)) / (2 * V(L + 1, L) * S1)
PRINT V(J, L)
NEXT J
FOR I = 1 TO N: FOR J = 1 TO N
VVT(I, J) = V(I, L) * V(J, L)
NEXT J: NEXT I
FOR I = 1 TO N: FOR J = 1 TO N
IF I = J THEN K = 1 ELSE K = 0
P1(I, J) = K - 2 * VVT(I, J)
NEXT J
NEXT I
A$ = INPUT$(1): PRINT
FOR I = 1 TO N: FOR J = 1 TO N
SUM = 0
FOR K = 1 TO N: SUM = SUM + AD(I, K) * P1(K, J): NEXT K
AI(I, J) = SUM
NEXT J: NEXT I
PRINT "A"; L; " Matrix"
FOR I = 1 TO N: FOR J = 1 TO N
SUM = 0
FOR K = 1 TO N: SUM = SUM + P1(I, K) * AI(K, J): NEXT K
A1(I, J) = SUM: AD(I, J) = A1(I, J)
NEXT J
FOR J = 1 TO N: PRINT USING "    ###.####"; A1(I, J); : NEXT J
PRINT : NEXT I: A$ = INPUT$(1)
IF L = 1 THEN 5
FOR I = 1 TO N: FOR J = 1 TO N: SUM = 0
FOR K = 1 TO N
SUM = SUM + VEC(I, K) * P1(K, J)
NEXT K
VECT(I, J) = SUM
NEXT J: NEXT I
```

```
FOR I = 1 TO N: FOR J = 1 TO N
VEC(I, J) = VECT(I, J)
NEXT J: NEXT I
GOTO 10
5  FOR I = 1 TO N: FOR J = 1 TO N
   VEC(I, J) = P1(I, J)
   NEXT J: NEXT I
10 NEXT L
REM QR Factorization begins
FOR I = 1 TO N: FOR J = 1 TO N: B(I, J) = A1(I, J): NEXT J: NEXT I
PRINT : PRINT "B Matrix"
FOR I = 1 TO N: FOR J = 1 TO N: PRINT USING " ###.####"; B(I, J);
NEXT J: PRINT : NEXT I: A$ = INPUT$(1)
FOR M = 1 TO MI
FOR L = 1 TO N - 1
IF B(L, L) = 0 THEN COSTH = 0: SINTH = 1: GOTO 9
TERM = B(L + 1, L) / B(L, L): TERM1 = SQR(1 + TERM ^ 2)
COSTH = 1 / TERM1: SINTH = TERM / TERM1
9 FOR I = 1 TO N: FOR J = 1 TO N
IF I = J THEN C(I, J) = 1 ELSE C(I, J) = 0
NEXT J: NEXT I
C(L, L) = COSTH: C(L, L + 1) = SINTH
C(L + 1, L) = -SINTH: C(L + 1, L + 1) = COSTH
FOR I = 1 TO N: FOR J = 1 TO N
SUM = 0
FOR K = 1 TO N: SUM = SUM + C(I, K) * B(K, J): NEXT K
R(I, J) = SUM
NEXT J: NEXT I
FOR I = 1 TO N: FOR J = 1 TO N
B(I, J) = R(I, J)
NEXT J: NEXT I
   FOR I = 1 TO N: FOR J = 1 TO N
   SUM = 0
   FOR K = 1 TO N: SUM = SUM + VEC(I, K) * C(K, J): NEXT K
   VECT(I, J) = SUM
   NEXT J: NEXT I
   FOR I = 1 TO N: FOR J = 1 TO N
   VEC(I, J) = VECT(I, J)
   NEXT J: NEXT I
FOR I = 1 TO N: FOR J = 1 TO N
ON L GOTO 21, 26, 31, 36, 41
21 C2(I, J) = C(I, J): GOTO 50
26 C3(I, J) = C(I, J): GOTO 50
```

```
31 C4(I, J) = C(I, J): GOTO 50
36 C5(I, J) = C(I, J): GOTO 50
41 C6(I, J) = C(I, J)
50 NEXT J: NEXT I
NEXT L
FOR L = 1 TO N - 1
FOR I = 1 TO N: FOR J = 1 TO N
ON L GOTO 61, 66, 71, 76, 81
61 C(I, J) = C2(I, J): GOTO 90
66 C(I, J) = C3(I, J): GOTO 90
71 C(I, J) = C4(I, J): GOTO 90
76 C(I, J) = C5(I, J): GOTO 90
81 C(I, J) = C6(I, J)
90 NEXT J: NEXT I
FOR I = 1 TO N: FOR J = 1 TO N
SUM = 0
FOR K = 1 TO N: SUM = SUM + R(I, K) * C(J, K): NEXT K
D(I, J) = SUM
NEXT J: NEXT I
FOR I = 1 TO N: FOR J = 1 TO N
R(I, J) = D(I, J)
NEXT J: NEXT I
NEXT L
PRINT : PRINT "B"; M; "Matrix"
FOR I = 1 TO N: FOR J = 1 TO N
B(I, J) = D(I, J): PRINT USING " ###.####"; D(I, J);
NEXT J: PRINT : NEXT I: A$ = INPUT$(1)
NEXT M
REM Arrange eigenvalues in incresing order
FOR I = 1 TO N: FOR J = 1 TO N
IF B(I, I) < B(J, J) THEN
SWAP B(I, I), B(J, J)
FOR K = 1 TO N: SWAP VEC(K, I), VEC(K, J): NEXT K
END IF
NEXT J: NEXT I
REM Printing eigenvalues and eigenvectors
PRINT : PRINT "Eigenvalues"
FOR J = 1 TO N: PRINT USING "  ###.####"; B(J, J); : NEXT J: PRINT
PRINT "Eigenvectors"
FOR I = 1 TO N: FOR J = 1 TO N
PRINT USING "  ###.####"; VEC(I, J);
NEXT J: PRINT : NEXT I
END
```

Output

```
Given Matrix
 6  4  1  1
 4  6  1  1
 1  1  5  2
 1  1  2  5

V 1  vector
 0
 .9855986
 .1195732
 .1195732

A 1  Matrix
     6.0000    -4.2426    -0.0000    -0.0000
    -4.2426     7.0000    -1.0000    -1.0000
    -0.0000    -1.0000     4.5000     1.5000
    -0.0000    -1.0000     1.5000     4.5000

V 2  vector
 0
 0
 .9238796
 .3826834
A 2  Matrix
     6.0000    -4.2426     0.0000    -0.0000
    -4.2426     7.0000     1.4142    -0.0000
     0.0000     1.4142     6.0000    -0.0000
    -0.0000    -0.0000     0.0000     3.0000
B Matrix
   6.0000  -4.2426   0.0000  -0.0000
  -4.2426   7.0000   1.4142  -0.0000
   0.0000   1.4142   6.0000  -0.0000
  -0.0000  -0.0000   0.0000   3.0000

B 1 Matrix
  10.3333  -2.0548  -0.0000   0.0000
  -2.0548   4.0351   2.0055  -0.0000
   0.0000   2.0055   4.6316   0.0000
   0.0000  -0.0000   0.0000   3.0000
```

```
B 2 Matrix
  10.8799  -0.7964  -0.0000   0.0000
  -0.7964   5.4474   1.5070  -0.0000
   0.0000   1.5070   2.6727   0.0000
   0.0000  -0.0000   0.0000   3.0000

B 3 Matrix
  10.9669  -0.4075  -0.0000   0.0000
  -0.4075   5.9459   0.5852  -0.0000
   0.0000   0.5852   2.0872   0.0000
   0.0000  -0.0000   0.0000   3.0000

.
.
.

B 19 Matrix
  11.0000  -0.0000  -0.0000   0.0000
  -0.0000   6.0000   0.0000  -0.0000
   0.0000   0.0000   2.0000   0.0001
   0.0000  -0.0000   0.0001   3.0000

B 20 Matrix
  11.0000  -0.0000  -0.0000   0.0000
  -0.0000   6.0000   0.0000  -0.0000
   0.0000   0.0000   2.0000   0.0001
   0.0000  -0.0000   0.0001   3.0000

Eigenvalues
    2.0000    3.0000    6.0000   11.0000
Eigenvectors
   -0.7071   -0.0001   -0.3162    0.6325
   -0.5500   -0.0001   -0.3162   -0.7730
   -0.3142   -0.7072    0.6325   -0.0351
   -0.3144    0.7071    0.6325   -0.0351
```

Exercise

Diagonalize the secular determinant of butadiene and obtain its eigenvalues and eigenvectors.

3.9 FORMATION OF PARTIALLY MISCIBLE LIQUIDS

The thermodynamic properties of a binary nonideal liquid solutions are expressed in terms of those of an ideal solution by the expression

$$Y^{E} = Y^{(real)} - Y^{(ideal)} \tag{1}$$

where Y^{E} is known as excess thermodynamic function, and $Y^{(\mathrm{real})}$ and $Y^{(\mathrm{ideal})}$ are the corresponding functions of the real and ideal solutions, respectively. The function Y may be V, U, H, C_p, S, A or G of the solution. For one mole of the solution, Eq. (1) is written as

$$Y^{E}_{\mathrm{m}} = Y^{(\mathrm{real})}_{\mathrm{m}} - Y^{(\mathrm{ideal})}_{\mathrm{m}} \tag{2}$$

The change in the thermodynamic function on forming a molar solution from its constituents is

$$\Delta_{\mathrm{mix}} Y_{\mathrm{m}} = Y_{\mathrm{m}} - \Sigma_i\, x_i\, Y^{*}_{i,\,\mathrm{m}} \tag{3}$$

where $Y^{*}_{i,\,\mathrm{m}}$ is the molar thermodynamic function of the ith pure component and x_i is its amount fraction in the solution. For a real and ideal solutions, we have

$$\Delta_{\mathrm{mix}}\, Y^{(\mathrm{real})}_{\mathrm{m}} = Y^{(\mathrm{real})}_{\mathrm{m}} - \Sigma_i\, x_i\, Y^{*}_{i,\,\mathrm{m}} \tag{4}$$

$$\Delta_{\mathrm{mix}}\, Y^{(\mathrm{ideal})}_{\mathrm{m}} = Y^{(\mathrm{ideal})}_{\mathrm{m}} - \Sigma_i\, x_i\, Y^{*}_{i,\,\mathrm{m}} \tag{5}$$

Hence, $$\Delta_{\mathrm{mix}}\, Y^{(\mathrm{real})}_{\mathrm{m}} - \Delta_{\mathrm{mix}}\, Y^{(\mathrm{ideal})}_{\mathrm{m}} = Y^{(\mathrm{real})}_{\mathrm{m}} - Y^{(\mathrm{ideal})}_{\mathrm{m}} = Y^{\mathrm{E}}_{\mathrm{m}} \tag{6}$$

that is, the molar excess function is also equal to the difference of molar function of mixing of an ideal solution from the corresponding molar function of mixing of the real solution, both are at the same T and p.

For the properties V, U, H and C_p, $\Delta_{\mathrm{mix}}\, Y^{(\mathrm{ideal})}_{\mathrm{m}} = 0$ and hence

$$\Delta_{\mathrm{mix}}\, Y^{(\mathrm{real})}_{\mathrm{m}} = Y^{\mathrm{E}}_{\mathrm{m}} \tag{7}$$

For the properties S and G, we have

$$\Delta_{\mathrm{mix}}\, S^{(\mathrm{ideal})}_{\mathrm{m}} = -\,R\,\Sigma_i\, x_i \ln x_i \tag{8}$$

$$\Delta_{\mathrm{mix}}\, G^{(\mathrm{ideal})}_{\mathrm{m}} = RT\,\Sigma_i\, x_i \ln x_i \tag{9}$$

Hence, $$S^{\mathrm{E}}_{\mathrm{m}} = \Delta_{\mathrm{mix}} S^{(\mathrm{real})}_{\mathrm{m}} + R\,\Sigma_i\, x_i \ln x_i \tag{10}$$

$$G^{\mathrm{E}}_{\mathrm{m}} = \Delta_{\mathrm{mix}}\, G^{(\mathrm{real})}_{\mathrm{m}} - RT\,\Sigma_i\, x_i \ln x_i \tag{11}$$

The chemical potential of the ith constituent in a solution is expressed as

Ideal solution $$\mu_i = \mu^{\circ}_i + RT \ln x_i \tag{12}$$

Real solution $$\mu_i = \mu^{\circ}_i + RT \ln a_i \tag{13}$$

where x_i and a_i are the amount fraction and activity of the ith constituent in the solution.

With these, the expression of $G^{\mathrm{E}}_{\mathrm{m}}$ becomes

$$\begin{aligned} G^{\mathrm{E}}_{\mathrm{m}} &= \Delta_{\mathrm{mix}}\, G^{(\mathrm{real})}_{\mathrm{m}} - \Delta_{\mathrm{mix}}\, G^{(\mathrm{ideal})}_{\mathrm{m}} \\ &= RT\,\Sigma_i\, x_i \ln a_i - RT\,\Sigma_i\, x_i \ln x_i \\ &= RT\,\Sigma_i\, x_i \ln (a_i/x_i) = RT\,\Sigma_i\, x_i \ln \gamma_i \\ &= \Sigma_i\, x_i\, (RT \ln \gamma_i) \end{aligned} \tag{14}$$

Thus, $RT \ln \gamma_i$ may be considered as the partial molar excess free energy of the ith constituent in the solution. Since the molar excess free energy is related to the activity coefficients of the constituents in a real solution, it is a useful quantity in describing the nature of real solution.

The molar excess functions are related to each other by the equations similar to those applicable for the ordinary thermodynamic functions. For example,

$$G_{\mathrm{m}}^{\mathrm{E}} = H_{\mathrm{m}}^{\mathrm{E}} - TS_{\mathrm{m}]}^{\mathrm{E}} \tag{15}$$

$$V_{\mathrm{m}}^{\mathrm{E}} = \left(\frac{\partial G_{\mathrm{m}}^{\mathrm{E}}}{\partial p}\right)_{T,x} \tag{16}$$

$$S_{\mathrm{m}}^{\mathrm{E}} = -\left(\frac{\partial G_{\mathrm{m}}^{\mathrm{E}}}{\partial T}\right)_{p,x} \tag{17}$$

$$H_{\mathrm{m}}^{\mathrm{E}} = -RT^2\left(\frac{\partial\left(G_{\mathrm{m}}^{\mathrm{E}} / RT\right)}{\partial T}\right)_{p,x} \tag{18}$$

The excess function contains information regarding the nature and the extent of deviations of a real solution from the corresponding ideal solution. One of the expressions of excess free energy of a regular binary liquid solution as described in Project 6 is

$$G_{\mathrm{m}}^{\mathrm{E}} = bx_1x_2 \tag{19}$$

where b is a constant, independent of T and p of the solution. Since $x_1 + x_2 = 1$, Eq. (19) may be written as

$$\begin{aligned} G_{\mathrm{m}}^{\mathrm{E}} &= bx_1x_2(x_1 + x_2) \\ &= bx_1x_2^2 + bx_1^2x_2 \end{aligned} \tag{20}$$

Equating Eqs. (14) and (20), we get

$$x_1\, RT \ln \gamma_1 + x_2\, RT \ln \gamma_2 = bx_1x_2^2 + bx_1^2x_2$$

From this equation, we conclude that

$$x_1\, RT \ln \gamma_1 = b\, x_1\, x_2^2 \qquad \text{i.e.} \qquad RT \ln \gamma_1 = bx_2^2 \tag{21}$$

$$x_2\, RT \ln \gamma_2 = b\, x_1^2\, x_2 \qquad \text{i.e.} \qquad RT \ln \gamma_2 = bx_1^2 \tag{22}$$

(**Note:** The second alternative equality leads to the same expression of γ_1 and γ_2, and thus may be ignored.)

Equations (21) and (22) are in agreement with the fact that $\gamma_1 = 1$ as $x_2 \to 0$

The other excess functions may be derived by using Eqs (16) – (18). Thus

$$V_{\mathrm{m}}^{\mathrm{E}} = \left(\frac{\partial G_{\mathrm{m}}^{\mathrm{E}}}{\partial p}\right)_{T,x} = \left(\frac{\partial (bx_1x_2)}{\partial p}\right)_{T,x} = 0 \tag{23}$$

$$S_{\mathrm{m}}^{\mathrm{E}} = -\left(\frac{\partial G_{\mathrm{m}}^{\mathrm{E}}}{\partial T}\right)_{p,x} = \left(\frac{\partial (bx_1x_2)}{\partial T}\right)_{p,x} = 0 \tag{24}$$

$$H_{\mathrm{m}}^{\mathrm{E}} = -RT^2\left(\frac{\partial\left(G_{\mathrm{m}}^{\mathrm{E}} / RT\right)}{\partial T}\right)_{p,x} = -RT^2\left(\frac{\partial (bx_1x_2 / RT)}{\partial T}\right)_{p,x}$$

$$= bx_1x_2 \tag{25}$$

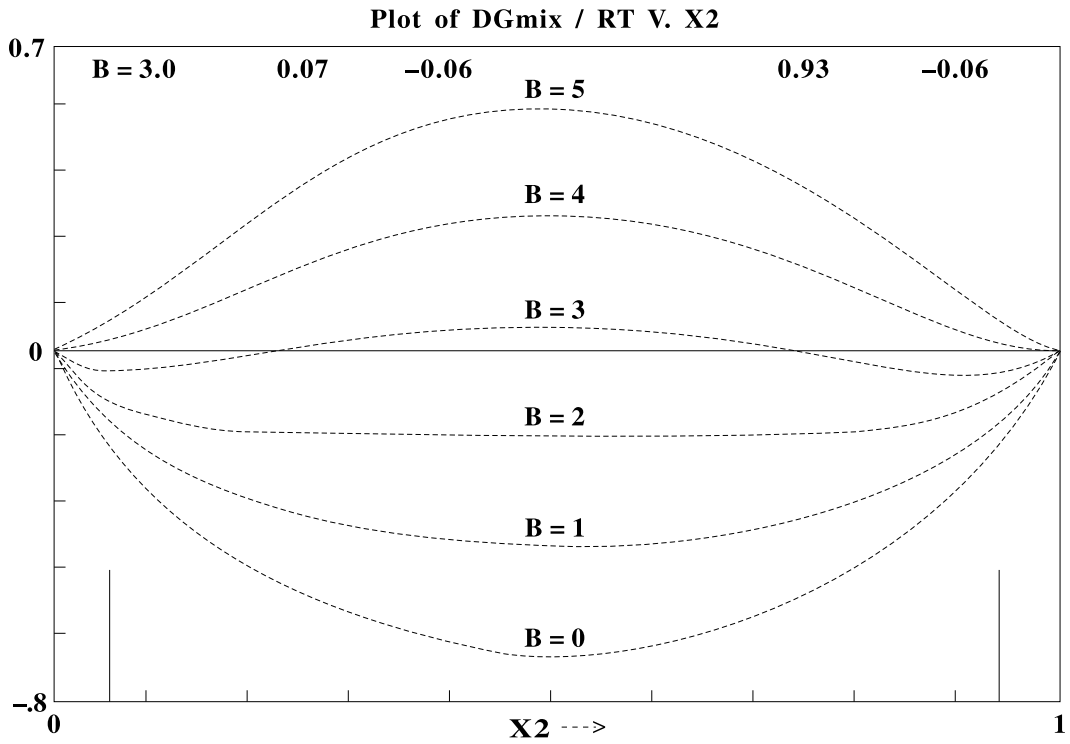
The expression of $\Delta_{mix} G_m^{(real)}$ is

$$\Delta_{mix} G_m^{(real)} = \Delta_{mix} G_m^{(ideal)} + G_m^E$$

$$= RT (x_1 \ln x_1 + x_2 \ln x_2) + bx_1x_2$$

or
$$\frac{\Delta_{mix} G_m^{(real)}}{RT} = (1 - x_2) \ln (1 - x_2) + x_2 \ln x_2 + \frac{b}{RT} (1 - x_2)x_2 \tag{26}$$

The plots of $\Delta_{mix} G_m^{(real)}/RT$ versus x_2 for various values of b/RT are shown below. This is obtainded from a computer program not listed here. These plots may be interpreted for increasing value of b at constant T or for decreasing value of T at constant b. The lowest curve corresponds to $\Delta_{mix} G^{(ideal)}/RT$ as the value of $b/RT = 0$. On increasing the value of b(or decreasing the value of T), the curve moves up with lesser and lesser dip at $x_2 = 0.5$. At larger value of b/RT, the curve shows two minima with a hump in between. This shows the formation of two conjugate solutions whose compositions lie at the two minima. With increasing value of b/RT, these compositions move towards $x_2 = 0$ and $x_2 = 1$ axes, respectively. At still larger value b/RT, the curve shows $\Delta_{mix} G^{(real)}/RT$ positive throughout indicating immiscibility.



Comment

Plots shown in the above figure are symmetrical with respect to $x_2 = 0.5$. This may be attributed to the symmetric nature of the function given by Eq. (26). The real system may not be symmetrical. This indicates that the function as given by Eq. (26) requires modification so as to include assymmetric terms in it.

The critical point of the phase separation is obtained when the second derivative

$$d^2(\Delta_{mix} G^{(real)}/RT)/dx_2^2 = 0 \text{ at } x_2 = 0.5.$$

Thus, we have $$\left[\frac{d^2\{(1-x_2)\ln(1-x_2) + x_2 \ln x_2 + (b/RT)(1-x_2)x_2\}}{dx_2^2}\right]_{x_2=0.5} = 0$$

This gives $$\left[\frac{1}{1-x_2} + \frac{1}{x_2} + \frac{b}{RT_c}(-2)\right]_{x_2=0.5} = 0 \quad \Rightarrow \quad T_c = b/2R$$

Program

```
CLS
DEF FNA (B, X) = X * LOG(X) + (1 - X) * LOG(1 - X) + B * X * (1 - X)
READ BI, BF, BS
DATA 0,5,.02
Y1 = .2: Y2 = .6: YI = -.8: YF = .7
SCREEN 1: COLOR 14, 0
FOR B = BI TO BF STEP BS
IF B = 0 THEN YI = -.8: YF = 0
IF B > 0 AND B < .5 THEN YI = -.75: YF = .15
IF B >= .5 AND B <= 1 THEN YI = -.65: YF = .15
IF B > 1 AND B <= 1.5 THEN YI = -.45: YF = .05
IF B > 1.5 AND B < 2 THEN YI = -.35: YF = .05
IF B >= 2 AND B <= 2.5 THEN YI = -.2: YF = .02
IF B > 2.5 AND B < 3 THEN YI = -.15: YF = .07
IF B >= 3 AND B <= 3.5 THEN YI = -.1: YF = .25
IF B > 3.5 AND B <= 4.1 THEN YI = -.05: YF = .35
IF B > 4.1 AND B <= 5 THEN YI = -.1: YF = .6
IF B > 5 AND B <= 6 THEN YI = -.1: YF = .9
VIEW (35, 20)-(160, 170): WINDOW (0, YI)-(1, YF)
LINE (0, YI)-(1, YF), , B: LINE (0, 0)-(1, 0)
FOR I = 1 TO 9
YD = YF - YI: YY = YI + YD * I * .1
LINE (I * .1, YI)-(I * .1, YI + YD * .03)
LINE (0, YY)-(.04, YY)
NEXT I
```

```
LOCATE 1, 30: PRINT "Partial"
LOCATE 2, 5: PRINT "  DGmix/RT V. X2      Misicibility"
LOCATE 3, 1: PRINT USING "#.##"; YF
LOCATE 3, 22: PRINT USING ".#"; Y2
LOCATE 4, 6: PRINT "B="; : PRINT USING "#.##"; B
LOCATE 12, 23: PRINT "1": LOCATE 13, 23: PRINT "-"
LOCATE 14, 23: PRINT "B"
LOCATE 22, 1: PRINT USING "#.##"; YI
LOCATE 22, 22: PRINT USING ".#"; Y1
LOCATE 23, 5: PRINT "0     X2-->    1   0     X2-->     1"
K = 1: GM(1) = 0: GM(2) = FNA(B, .5)
FOR X = .0001 TO 1 STEP .01
G = FNA(B, X)
IF G < GM(K) THEN GM(K) = G: XM(K) = X: GOTO 50
K = 2
50 PSET (X, G)
NEXT X
FOR K = 1 TO 2
IF XM(1) > .45 THEN 52
LINE (XM(K), YI)-(XM(K), GM(K)), 2
52 NEXT K
IF B > BF - BS THEN 60
A$ = INPUT$(1): CLS
FOR K = 1 TO 2
IF XM(1) > .45 THEN 55
LOCATE 4, 23 + 10 * (K - 1): PRINT USING " ##.##"; XM(K)
VIEW (185, 20)-(315, 170)
WINDOW (0, Y1)-(1, Y2): LINE (0, Y1)-(1, Y2), , B
FOR I = 1 TO 9: YY = Y1 + (Y2 - Y1) * I * .1
LINE (I * .1, Y1)-(I * .1, Y1 + .01): LINE (0, YY)-(.03, YY)
NEXT I
PSET (XM(K), 1 / B)
55 NEXT K
NEXT B
60 END
```

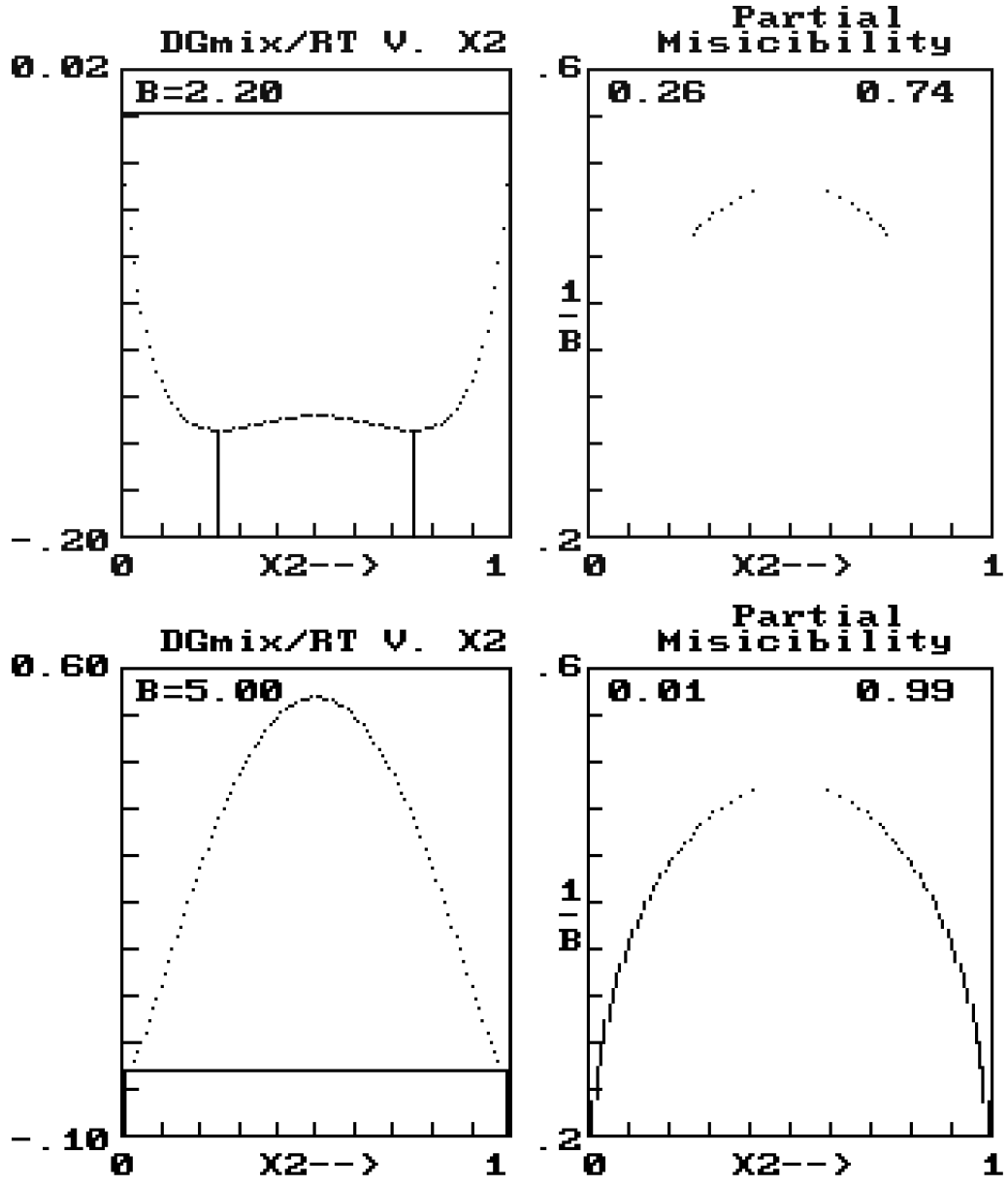
Output

Note: The plot of $\Delta G_{mix}/RT$ versus x_2 in the output is shown only for $B = 2.8$.

The following output shows the formation of two conjugate solutions (only for $B = 2.20$ is shown) which also includes the plot of $1/B$ versus x_2 (minima) observed in the plot of $\Delta_{mix} G_m$ versus x_2.

Since $B = b/RT$, this plot is a representation of the composition of two conjugate solutions at various temperatures.

Output



3.10 OSCILLATORY REACTIONS

The simultaneous differential equations involved in oscillatory reactions may be solved by the fourth-order Runge – Kutta method. The computational process of the method is described in the following.

Let the differential equations may be represented as

$$\frac{dx}{dt} = f_1(x, y, z, \ldots)$$

$$\frac{dy}{dt} = f_2(x, y, z, \ldots)$$

$$\frac{dz}{dt} = f_3(x, y, z, \ldots)$$

.....

For the given initial concentrations of x, y, and z, the values of Δx (= k), Δy (=l) and Δz (= m) for the chosen value of Δt (= h) are determined by the following expressions.

$$k_1 = hf_1\ (x_0, y_0, z_0, \ldots)$$

$$l_1 = hf_2\ (x_0, y_0, z_0, \ldots)$$

$$m_1 = hf_3\ (x_0, y_0, z_0, \ldots)$$

.....

$$k_2 = h\ f_1(x_0 + \tfrac{1}{2}\ k_1, y_0 + \tfrac{1}{2}\ l_1, z_0 + \tfrac{1}{2}\ m_1, \ldots)$$

$$l_2 = h\ f_2(x_0 + \tfrac{1}{2}\ k_1, y_0 + \tfrac{1}{2}\ l_1, z_0 + \tfrac{1}{2}\ m_1, \ldots)$$

$$m_2 = h\ f_3(x_0 + \tfrac{1}{2}\ k_1, y_0 + \tfrac{1}{2}\ l_1, z_0 + \tfrac{1}{2}\ m_1, \ldots)$$

.....

$$k_3 = h\ f_1(x_0 + \tfrac{1}{2}\ k_2, y_0 + \tfrac{1}{2}\ l_2, z_0 + \tfrac{1}{2}\ m_2, \ldots)$$

$$l_3 = h\ f_2(x_0 + \tfrac{1}{2}\ k_2, y_0 + \tfrac{1}{2}\ l_2, z_0 + \tfrac{1}{2}\ m_2, \ldots)$$

$$m_3 = h\ f_3(x_0 + \tfrac{1}{2}\ k_2, y_0 + \tfrac{1}{2}\ l_2, z_0 + \tfrac{1}{2}\ m_2, \ldots)$$

.....

$$k_4 = h\ f_1(x_0 + k_3, y_0 + l_3, z_0 + m_3, \ldots)$$

$$l_4 = h\ f_2(x_0 + k_3, y_0 + l_3, z_0 + m_3, \ldots)$$

$$m_4 = h\ f_3(x_0 + k_3, y_0 + l_3, z_0 + m_3, \ldots)$$

.....

Finally, the values of x, y, and z for $t + \Delta t$ are determined from the following expression

$$x_1 = x_0 + (1/6)\ (k_1 + 2k_2 + 2k_3 + k_4)$$

$$y_1 = y_0 + (1/6)\ (l_1 + 2l_2 + 2l_3 + l_4)$$

$$z_1 = x_0 + (1/6)\ (m_1 + 2m_2 + 2m_3 + m_4)$$

The above scheme is executed in an iterative manner until the time t becomes equal to the chosen final value.

LOTKA – VOLTERRA MECHANISM

The scheme is

$$A + X \xrightarrow{k_1} 2X$$

$$X + Y \xrightarrow{k_2} 2Y$$

$$Y \xrightarrow{k_3} B$$

The rate expressions for X and Y are

$$\frac{d[X]}{dt} = k_1[A][X] - k_2\,[X]\,[Y]$$

$$\frac{d[Y]}{dt} = k_2[X][Y] - k_3\,[Y]$$

Use the following data for the computation of [X], [Y], d[X]/dt and d[Y]/dt.

$k_1 = 10$ mol^{-1} L min^{-1}, $k_2 = 5$ mol^{-1} L min^{-1}, $k_3 = 2$ min^{-1},

$[A]_0 = 0.1$ mol L^{-1}, $[X]_0 = 0.1$ mol L^{-1}, and $[Y]_0 = 0.1$ mol L^{-1}

Make a program to display (i) [X] and [Y] with time, (ii) [Y] versus [X] and (iii) the rates of variations of X and Y. Also determine the times at which [X] and [Y] have maximum and minimum values.

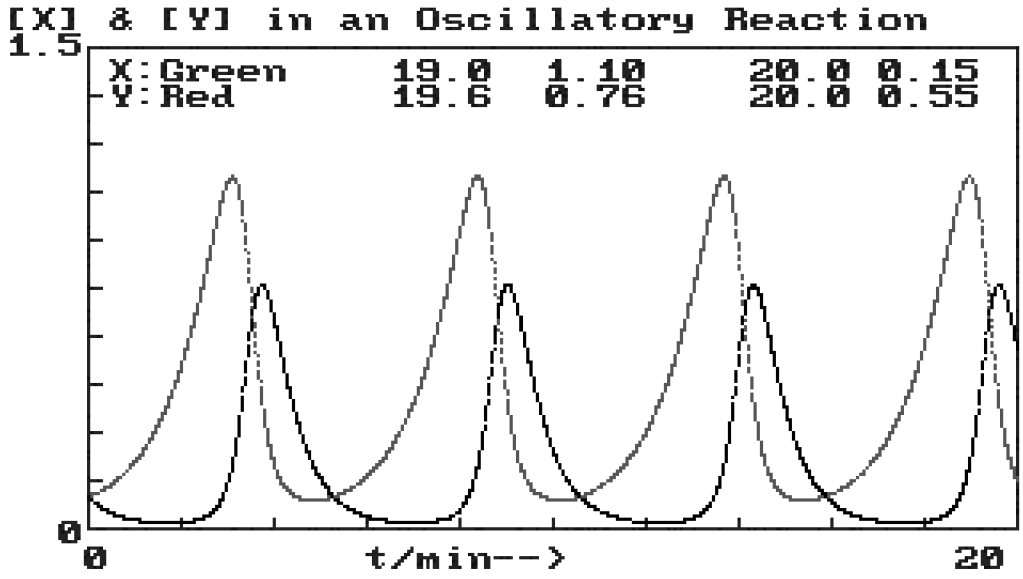
Program

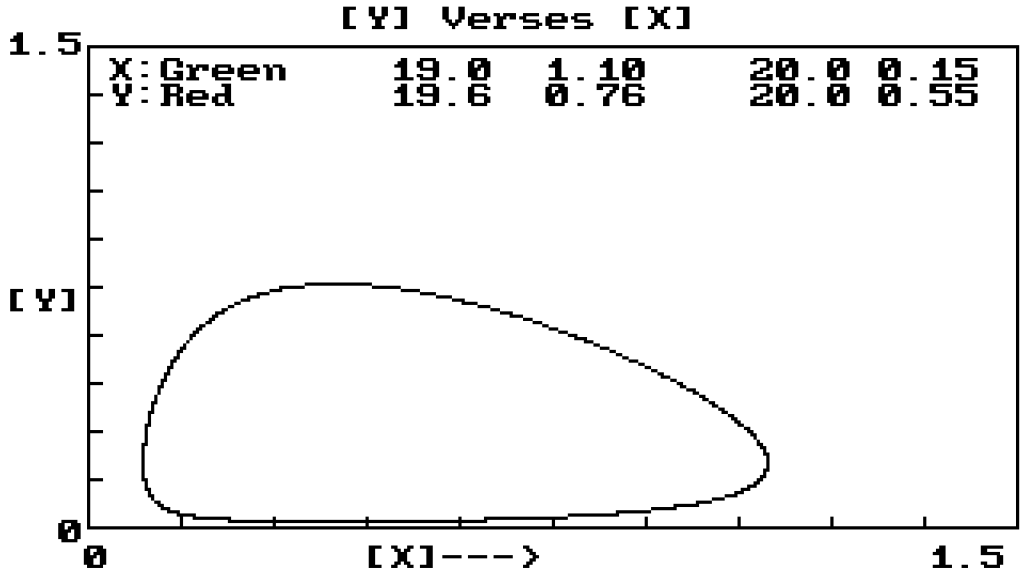
```
CLS : REM Oscillatory Raction(Lotka-Volterra)
REM TYP=1:[X] & [Y] verses time;TYP=2:[Y] verse [X]
REM TYP=3:Rates of change of concentrations
READ PLOT$, TYP, A0, TF: DATA Y,1,1.5,20
R1 = 10: R2 = 5: R3 = 2: CA = .1: X0 = .1: Y0 = .1
H = .01: NMAX = TF / H: NN = 0: XMAX = 0: XMIN = A0
YMAX = 0: YMIN = A0: M = 1: M1 = 1: C$ = "t/min--->"
REM Identifing maxima and minima in the plots of [X] &[Y] verses
time
DEF FNM (Q, P1, PMAX, PMIN, S)
ON Q GOTO 15, 20
15 IF P1 > PMAX THEN
```

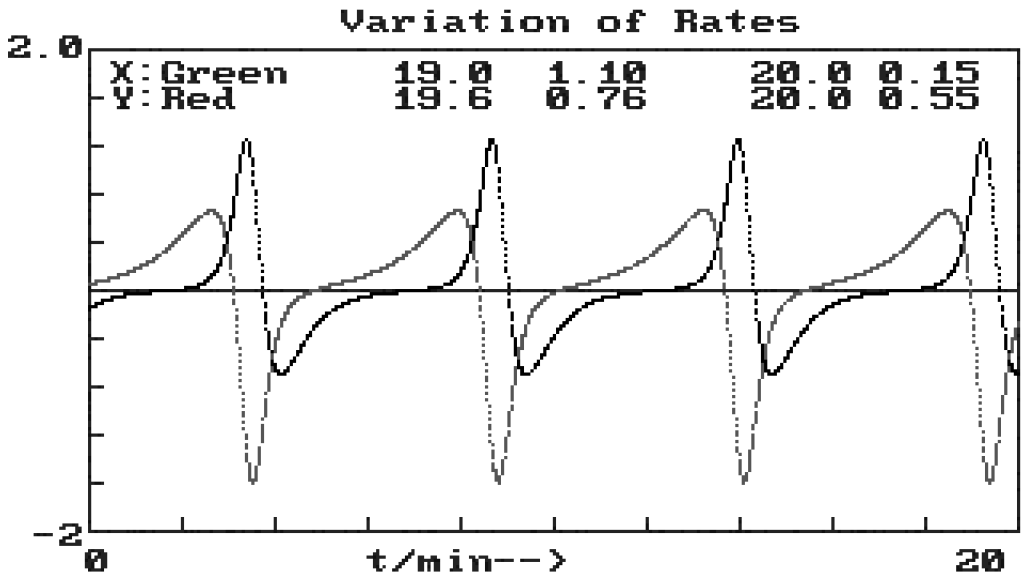
```
PMAX = P1: TPMAX = T: LOCATE S, 15
PRINT USING " ##.#"; TPMAX; : PRINT USING " ##.##"; PMAX
ELSEIF P1 < PMAX THEN
LOCATE S, 13: PRINT "Max": A$ = INPUT$(1)
LOCATE S, 13: PRINT "    ": PMIN = A0: Q = 2
END IF
GOTO 25
20 IF P1 < PMIN THEN
PMIN = P1: TPMIN = T: LOCATE S, 28
PRINT USING "  ##.#"; TPMIN; : PRINT USING " #.##"; PMIN
ELSEIF P1 > PMIN THEN
LOCATE S, 27: PRINT "Min": A$ = INPUT$(1)
LOCATE S, 27: PRINT "    ": PMAX = 0: Q = 1
END IF
25 IF S = 4 THEN M = Q: XMAX = PMAX: XMIN = PMIN: GOTO 26
M1 = Q: YMAX = PMAX: YMIN = PMIN
26 END DEF
DEF FNA (X, Y) = R1 * CA * X - R2 * X * Y
DEF FNB (X, Y) = R2 * X * Y - R3 * Y
IF PLOT$ = "Y" THEN
SCREEN 1: COLOR 15, 0: VIEW (25, 20)-(315, 170)
ON TYP GOTO 1, 2, 3
1 XI = 0: XF = TF: YI = 0: YF = A0: GOTO 4
2 XI = 0: XF = A0: YI = 0: YF = A0: C$ = "[X]--->"
LOCATE 13, 1: PRINT "[Y]": GOTO 4
3 XI = 0: XF = TF: YI = -2: YF = 2
4 WINDOW (XI, YI)-(XF, YF): LINE (XI, YI)-(XF, YF), , B
LOCATE 2, 1
ON TYP GOTO 31, 32, 33
31 PRINT "[X] & [Y] in an Oscillatory Reaction": GOTO 34
32 PRINT "                [Y] Verses [X]": GOTO 34
33 PRINT "                Variation of Rates"
34 IF TYP = 3 THEN LINE (XI, 0)-(XF, 0)
LOCATE 3, 1: PRINT USING "#.#"; YF
LOCATE 4, 5: PRINT "X:Green": LOCATE 5, 5: PRINT "Y:Red"
LOCATE 22, 2: PRINT USING "##"; YI
LOCATE 23, 4: PRINT "0            "; C$; "             "; XF
FOR I = 1 TO 9
XS = XF / 10: YS = YI + (YF - YI) * I / 10
IF TYP = 3 THEN
LINE (I * XS, YI)-(I * XS, YI + .1)
LINE (0, YS)-(.3, YS)
```

```
ELSE
LINE (I * XS, YI)-(I * XS, .03)
LINE (0, YF * I / 10)-(XF / 80, YF * I / 10)
END IF
NEXT I
ELSE
PRINT STRING$(30, "-")
PRINT "    t/min        [X]     [Y]"
PRINT STRING$(30, "-")
END IF
T = 0: XA = X0: YB = Y0
FOR N = 0 TO NMAX
K1 = H * FNA(XA, YB)
L1 = H * FNB(XA, YB)
K2 = H * FNA(XA + .5 * K1, YB + .5 * L1)
L2 = H * FNB(XA + .5 * K1, YB + .5 * L1)
K3 = H * FNA(XA + .5 * K2, YB + .5 * L2)
L3 = H * FNB(XA + .5 * K2, YB + .5 * L2)
K4 = H * FNA(XA + K3, YB + L3)
L4 = H * FNB(XA + K3, YB + L3)
X1 = XA + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
Y1 = YB + (1 / 6) * (L1 + 2 * L2 + 2 * L3 + L4)
IF PLOT$ = "Y" THEN
'FOR I = 1 TO 100 STEP .001: NEXT I
ON TYP GOTO 6, 7, 8
6 PSET (T, XA), 1: PSET (T, YB), 2: GOTO 9
7 PSET (X1, Y1): GOTO 9
8 PSET (T, K1 / H), 1: PSET (T, L1 / H), 2
9 ELSE
IF N = NN * 20 THEN A$ = INPUT$(1): NN = NN + 1
ON TYP GOTO 11, 11, 12
11 PRINT USING "   ##.##"; T; : PRINT USING "    #.####"; XA; YB
GOTO 13
12 PRINT USING "   ##.#"; T; : PRINT USING "   ##.######"; K1; L1
13 END IF
T = T + H: XA = X1: YB = Y1
IF PLOT$ = "Y" THEN
DUM = FNM(M, X1, XMAX, XMIN, 4)
DUM = FNM(M1, Y1, YMAX, YMIN, 5)
END IF
40 NEXT N
IF PLOT$ <> "Y" THEN PRINT STRING$(30, "-")
END
```

Output


[X] & [Y] in an Oscillatory Reaction
1.5
X:Green 19.0 1.10 20.0 0.15
Y:Red 19.6 0.76 20.0 0.55
0
0
t/min-->
20


[Y] Verses [X]
1.5
X:Green 19.0 1.10 20.0 0.15
Y:Red 19.6 0.76 20.0 0.55
[Y]
0
0
[X]--->
1.5



Comment The maxima and minima in the plots of [X] and [Y] with time are as follows:

Max. [X]	t/min →	3.1	8.4	13.7	19.0
Min. [X]	t/min →	4.8	10.1	15.4	20.7
Max. [X]	t/min →	3.7	9.0	14.3	19.6
Min. [X]	t/min →	1.7	7.0	12.3	17.7

Brusselator Mechanism

The scheme is

$$\mathrm{A} \xrightarrow{k_1} \mathrm{X}$$

$$\mathrm{B} + \mathrm{X} \xrightarrow{k_2} \mathrm{Y} + \mathrm{C}$$

$$2\mathrm{X} + \mathrm{Y} \xrightarrow{k_3} 3\mathrm{X}$$

$$\mathrm{X} \xrightarrow{k_4} \mathrm{D}.$$

The rate expressions for X and Y are

$$\frac{\mathrm{d[X]}}{\mathrm{d}t} = k_1[\mathrm{A}] - k_2[\mathrm{B}]\,[\mathrm{X}] + k_3[\mathrm{X}]^2\,[\mathrm{Y}] - k_4[\mathrm{X}]$$

$$\frac{\mathrm{d[Y]}}{\mathrm{d}t} = k_2[\mathrm{B}]\,[\mathrm{X}] - k_3[\mathrm{X}]^2\,[\mathrm{Y}]$$

Use the following data for the computation of [X], [Y], d[X]/dt and d[Y]/dt.

$k_1 = 1\ min^{-1}$; $k_2 = 1\ mol^{-1}\ L\ min^{-1}$; $k_3 = 1\ mol^{-2}\ L^2\ min^{-1}$; $k_4 = 1\ min^{-1}$;

$[A]_0 = 1.0\ mol\ L^{-1}$, $[B]_0 = 3.0\ mol\ L^{-1}$, $[X]_0 = 1.0\ mol\ L^{-1}$ and $[Y]_0 = 1.0\ mol\ L^{-1}$

Make a program to display (i) [X] and [Y] with time, (ii) [Y] versus [X], and (iii) the rates of variations of [X] and [Y] with time. Also determine the times at which [X] and [Y] have maximum and minimum values.

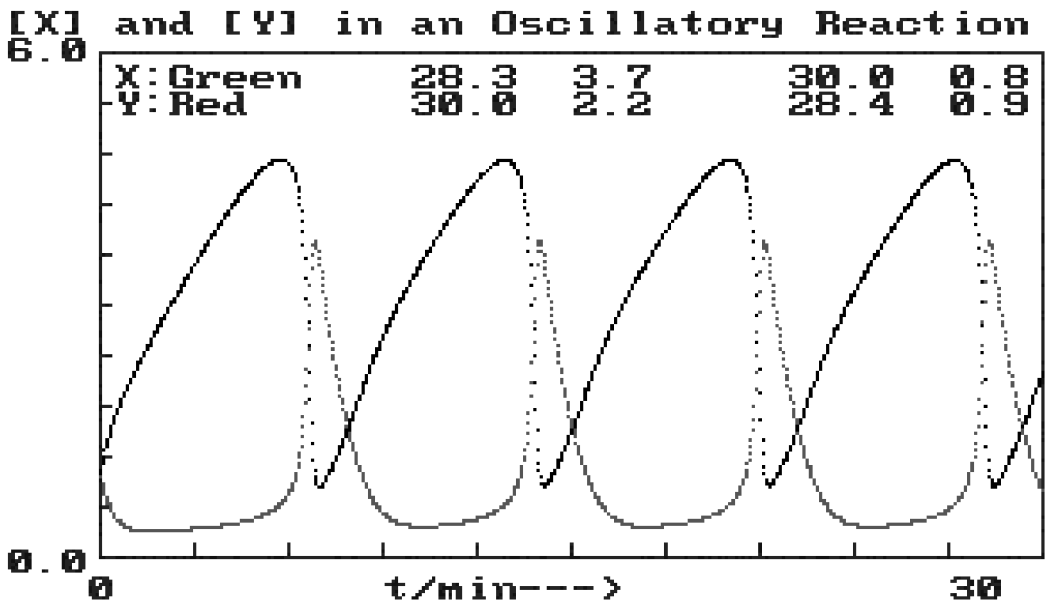
Program

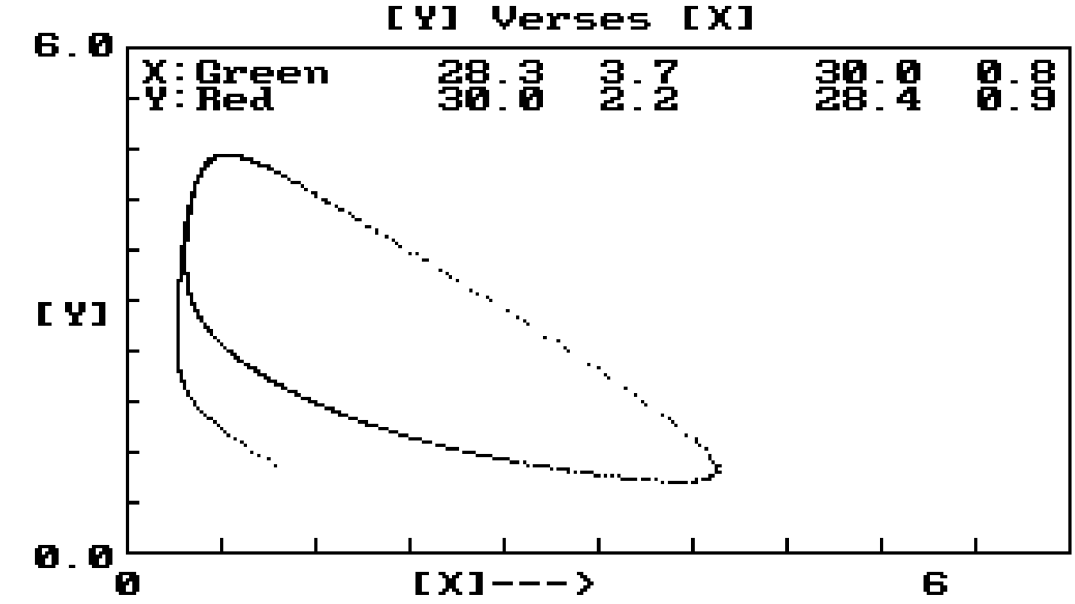
```
REM Oscillatory Reaction(Brusselator)
REM TYP=1:[X] & [Y] verses time;TYP=2;[Y] verses [X]
REM typ=3:Rates verses time
CLS : READ PLOT$, TYP, A0, TF: DATA Y,3,6,30
READ R1, R2, R3, R4, CA, CB: DATA 1,1,1,1,1,3
X0 = 1: Y0 = 1: NN = 0: XMAX = 0: XMIN = A0
YMAX = 0: YMIN = A0: M = 1: M1 = 1: C$ = "t/min"
IF TYP = 3 THEN H = .02 ELSE H = .03
REM Identifing maxima and minima in the plots of [x] & [y]
DEF FNM (Q, P1, PMAX, PMIN, S)
ON Q GOTO 5, 10
5 IF P1 > PMAX THEN
PMAX = P1: TPMAX = T: LOCATE S, 16: PRINT USING " ##.#"; TPMAX; PMAX
ELSEIF P1 < PMAX THEN
LOCATE S, 14: PRINT "Max": A$ = INPUT$(1)
LOCATE S, 13: PRINT "     ": PMIN = A0: Q = 2
END IF
GOTO 15
10 IF P1 < PMIN THEN
PMIN = P1: TPMIN = T: LOCATE S, 30: PRINT USING " ##.#"; TPMIN; PMIN
ELSEIF P1 > PMIN THEN
LOCATE S, 28: PRINT "Min": A$ = INPUT$(1): LOCATE S, 28
PRINT "    "
PMAX = 0: Q = 1
END IF
15 IF S = 4 THEN M = Q: XMAX = PMAX: XMIN = PMIN: GOTO 16
M1 = Q: YMAX = PMAX: YMIN = PMIN
16 END DEF
DEF FNA (X, Y) = R1 * CA - R2 * CB * X + R3 * X ^ 2 * Y - R4 * X
DEF FNB (X, Y) = R2 * CB * X - R3 * X ^ 2 * Y
IF PLOT$ = "Y" THEN
SCREEN 1: COLOR 15, 0: VIEW (35, 20)-(315, 170)
ON TYP GOTO 1, 2, 3
1 XI = O: XF = TF: YI = 0: YF = A0: GOTO 4
2 XI = 0: XF = A0: YI = 0: YF = A0: C$ = "[X]"
LOCATE 13, 2: PRINT "[Y]": GOTO 4
```

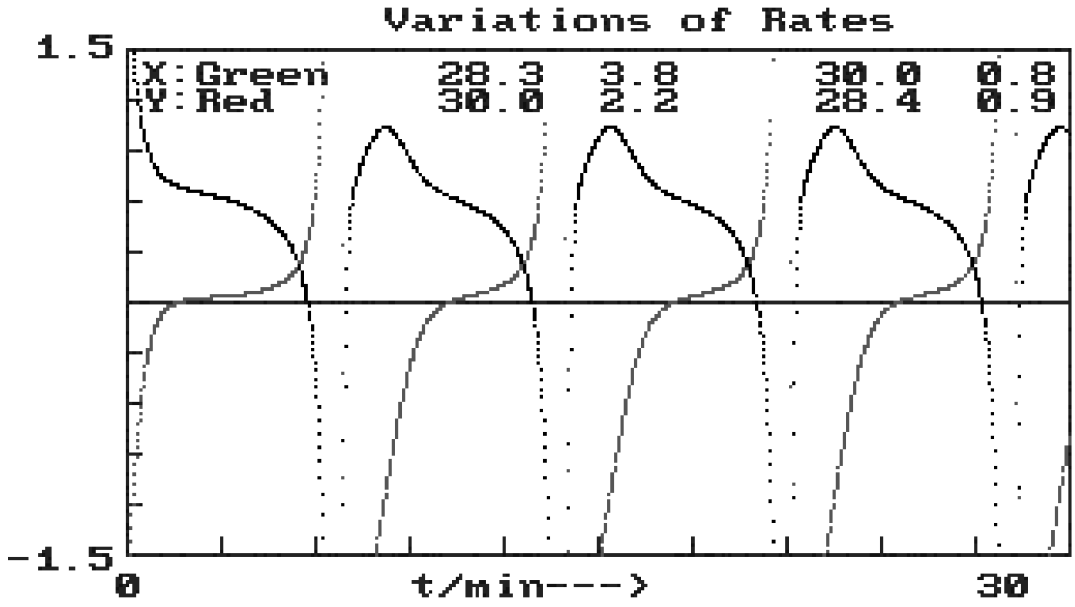
```
3 XI = 0: XF = TF: YI = -1.5: YF = 1.5
4 WINDOW (XI, YI)-(XF, YF): LINE (XI, YI)-(XF, YF), , B
IF TYP = 3 THEN LINE (0, 0)-(TF, 0)
LOCATE 2, 1
ON TYP GOTO 31, 32, 33
31 PRINT " [X] and [Y] in an Oscillatory Reaction": GOTO 34
32 PRINT "                 [Y] Verses [X]": GOTO 34
33 PRINT "                 Variations of Rates"
34 LOCATE 3, 1: PRINT USING "##.#"; YF
LOCATE 4, 6: PRINT "X:Green": LOCATE 5, 6: PRINT "Y:Red"
LOCATE 22, 1: PRINT USING "##.#"; YI
LOCATE 23, 5: PRINT "0            "; C$; "--->             "; XF
FOR I = 1 TO 9
IF TYP = 3 THEN
XS = XF / 10: YS = YI + (YF - YI) * I / 10
LINE (I * XS, YI)-(I * XS, YI + .1)
LINE (XI, YS)-(XI + .4, YS)
ELSE
LINE (I * XF / 10, YI)-(I * XF / 10, YF * .03)
LINE (0, YF * I / 10)-(XF * .01, YF * I / 10)
END IF
NEXT I
ELSE
PRINT STRING$(30, "-"): PRINT "    t/s         [X]        [Y]"
PRINT STRING$(30, "-")
END IF
T = 0: XA = X0: YB = Y0: NMAX = TF / H
FOR N = 0 TO NMAX
K1 = H * FNA(XA, YB)
L1 = H * FNB(XA, YB)
K2 = H * FNA(XA + .5 * K1, YB + .5 * L1)
L2 = H * FNB(XA + .5 * K1, YB + .5 * L1)
K3 = H * FNA(XA + .5 * K2, YB + .5 * L2)
L3 = H * FNB(XA + .5 * K2, YB + .5 * L2)
K4 = H * FNA(XA + K3, YB + L3)
L4 = H * FNB(XA + K3, YB + L3)
X1 = XA + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
Y1 = YB + (1 / 6) * (L1 + 2 * L2 + 2 * L3 + L4)
IF PLOT$ = "Y" THEN
'FOR I = 1 TO 100 STEP .001: NEXT I
ON TYP GOTO 6, 7, 8
```

```
6 PSET (T, XA), 1: PSET (T, YB), 2: GOTO 9
7 PSET (X1, Y1): GOTO 9
8 PSET (T, K1 / H), 1: PSET (T, L1 / H), 2
9 ELSE
IF N = 20 * NN THEN A$ = INPUT$(1): NN = NN + 1
ON TYP GOTO 11, 11, 12
11 PRINT USING "  ##.#"; T; : PRINT USING "  ###.####"; XA; YB
GOTO 13
12 PRINT USING "  ##.#"; T;
PRINT USING "  ###.####"; (K1 / H) * 10 / 1.5; (L1 / H) * 10 / 1.5
13 END IF
T = T + H: XA = X1: YB = Y1
IF PLOT$ = "Y" THEN
DUM = FNM(M, X1, XMAX, XMIN, 4)
DUM = FNM(M1, Y1, YMAX, YMIN, 5)
END IF
30 NEXT N
IF PLOT$ <> "Y" THEN PRINT STRING$(30, "-")
END
```

Output




[Y] Verses [X]
6.0
X:Green 28.3 3.7 30.0 0.8
Y:Red 30.0 2.2 28.4 0.9
[Y]
0.0
0
[X]--->
6


Variations of Rates
1.5
X:Green 28.3 3.8 30.0 0.8
Y:Red 30.0 2.2 28.4 0.9
-1.5
0
t/min--->
30

Comments The maxima and minima in the plots of [X] and [Y] with time are as follows.

Max. [X]	t/min →	6.8	14.0	21.2	28.3
Min. [Y]	t/min →	1.6	10.2	17.4	24.6
Max. [X]	t/min →	5.7	12.9	20.0	27.2
Min. [Y]	t/min →	7.0	14.1	21.8	28.5

Belousov – Zhabotinsky Mechanism

The scheme is

$$A + Y \xrightarrow{k_1} X + P$$

$$A + X \xrightarrow{k_2} 2X + 2Z$$

$$X + Y \xrightarrow{k_3} 2P$$

$$2X \xrightarrow{k_4} A + P$$

$$B + Z \longrightarrow (f/2)\, Y$$

The rates of change of [X], [Y] and [*Z*] as given by the FKN (Field - Körös - Noyes) model are as follows.

$$\frac{d[X]}{dt} = k_1 [A] [Y] + k_2[A] [X] - k_3 [X] [Y] - 2k_4 [X]^2$$

$$\frac{d[Y]}{dt} = - k_1 [A] [Y] - k_3 [X] [Y] + \frac{f}{2} k_5 [B] [Z]$$

$$\frac{d[Z]}{dt} = 2k_2 [A] [X] - k_5 [B] [Z]$$

Use the following data for the computation of [X], [Y], [Z], d[X]/d*t*, d[Y]/d*t* and d[Z]/d*t*

$k_1 = 1.28$ mol^{-1} L s^{-1}; $k_2 = 8.0$ mol^{-1} L s^{-1}, $k_3 = 8.0 \times 10^4$ mol^{-1} L s^{-1};

$k_4 = 2.0 \times 10^3$ mol^{-1} L s^{-1}, $k_5 = 1.0$ mol^{-1} L s^{-1}, [A] = 0.6 M

[B] = 0.02 M, $[X]_0 = 0$, $[Y]_0 = 0$, $[Z]_0 = 0.0002$, and $f = 1.5$

Make a program to display (i) [X], [Y] and [Z] with time, (ii) [Y] versus [Z], and (iii) the rates of variations of [Y] and [Z] with time. Also determine the times at which [Y] and [Z] have maximum and minimum values.

Program

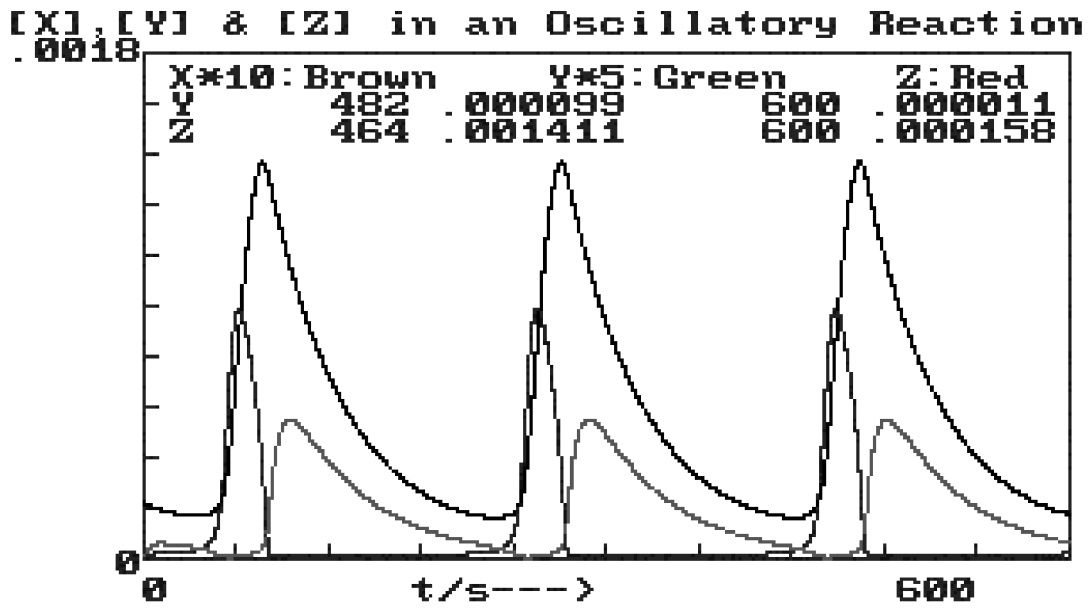
```
CLS : REM Oscillatory Raction(Belousov-Zhabotinsky)
REM TYP=1:[X] & [Y] verses time;TYP=2:[Y] verse [X]
REM TYP=3:Rates of change of concentrations
REM PLOT$ is "Y" if plotting required.For any other symbol
REM numerical values are displayed
CLS : READ PLOT$, TYP, A0, TF: DATA Y,1,.0018,600
```

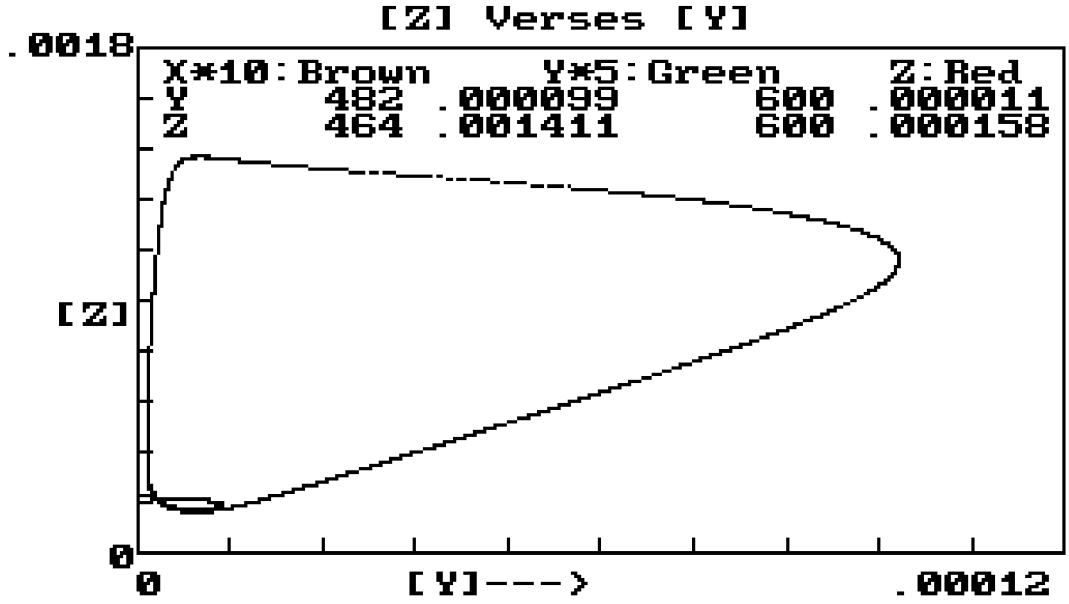
```
R1 = 1.28: R2 = 8: R3 = 80000: R4 = 2000: R5 = 1
CA = .06: CB = .02: F = 1.5: ZMAX = 0: ZMIN = A0: N1 = 1
H = .4: NMAX = TF / H: X0 = 0: Y0 = 0: Z0 = .0002: C$ = "t/s"
REM Identifing maxima and minima in the plots of [X] &[Y]
DEF FNM (Q, P1, PMAX, PMIN, S)
ON Q GOTO 35, 40
35 IF P1 > PMAX THEN
PMAX = P1: TPMAX = T: LOCATE S, 12: PRINT USING "####"; TPMAX;
PRINT USING " .######"; PMAX
ELSEIF P1 < PMAX THEN
LOCATE S, 9: PRINT "Max": A$ = INPUT$(1)
LOCATE S, 9: PRINT "    ": PMIN = A0: Q = 2
END IF
GOTO 45
40 IF P1 < PMIN THEN
PMIN = P1: TPMIN = T: LOCATE S, 28: PRINT USING "####"; TPMIN;
PRINT USING " .######"; PMIN
ELSEIF P1 > PMIN THEN
LOCATE S, 25: PRINT "MIN": A$ = INPUT$(1)
LOCATE S, 25: PRINT "    ": PMAX = 0: Q = 1
END IF
45 IF S = 5 THEN N1 = Q: YMAX = PMAX: YMIN = PMIN: GOTO 46
N2 = Q: ZMAX = PMAX: ZMIN = PMIN
46 END DEF
DEF FNA (X, Y, Z) = R1 * CA * Y + R2 * CA * X - R3 * X * Y - 2 * R4 * X ^ 2
DEF FNB (X, Y, Z) = -R1 * CA * Y - R3 * X * Y + (F / 2) * R5 * CB * Z
DEF FNC (X, Y, Z) = 2 * R2 * CA * X - R5 * CB * Z
IF PLOT$ = "Y" THEN
SCREEN 1: COLOR 15, 0: VIEW (40, 20)-(315, 170)
ON TYP GOTO 1, 2, 3
1 XI = 0: XF = TF: YI = 0: YF = A0: GOTO 4
2 XI = 0: XF = A0 / 15: YI = 0: YF = A0: C$ = "[Y]"
LOCATE 13, 3: PRINT "[Z]": GOTO 4
3 XI = 0: XF = TF: YI = -A0: YF = A0
LOCATE 8, 8: PRINT "Y": LOCATE 8, 11: PRINT "Z"
4 WINDOW (XI, YI)-(XF, YF): LINE (XI, YI)-(XF, YF), , B
LOCATE 2, 1
ON TYP GOTO 31, 32, 33
31 PRINT "[X],[Y] & [Z] in an Oscillatory Reaction": GOTO 34
32 PRINT "                [Z] Verses [Y]": GOTO 34
33 PRINT "                Variations of Rates"
34 LOCATE 3, 1: PRINT USING ".####"; A0
LOCATE 4, 7: PRINT "X*10:Brown": LOCATE 4, 21: PRINT "Y*5:Green"
```

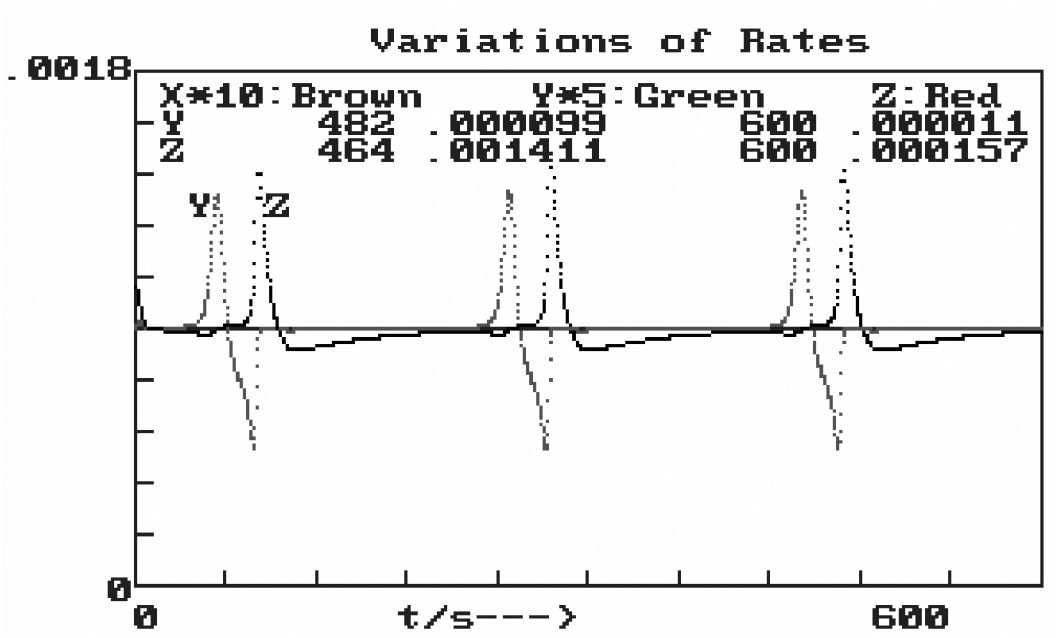
```
LOCATE 4, 34: PRINT "Z:Red"
LOCATE 5, 7: PRINT "Y": LOCATE 6, 7: PRINT "Z"
39 LOCATE 22, 4: PRINT " 0"
LOCATE 23, 3: PRINT "    0           "; C$; "--->            "; XF
FOR I = 1 TO 9
IF TYP = 3 THEN
LINE (I * XF / 10, YI)-(I * XF / 10, YI + .0001)
LINE (XI, YI + (YF - YI) * I / 10)-(XI + 10, YI + (YF - YI) * I / 10)
ELSE
LINE (I * XF / 10, 0)-(I * XF / 10, YF * .03)
LINE (XI, YF * I / 10)-(XF / 70, YF * I / 10)
END IF
NEXT I
IF TYP = 3 THEN LINE (XI, 0)-(XF, 0)
ELSE
PRINT STRING$(50, "-")
PRINT "    t/s        [X]*1E7     [Y]*1E7     [Z]*1E7"
PRINT STRING$(50, "-")
END IF
T = 0: XA = X0: YB = Y0: ZC = Z0: NN = 0
FOR N = 0 TO NMAX
K1 = H * FNA(XA, YB, ZC)
L1 = H * FNB(XA, YB, ZC)
M1 = H * FNC(XA, YB, ZC)
K2 = H * FNA(XA + .5 * K1, YB + .5 * L1, ZC + .5 * M1)
L2 = H * FNB(XA + .5 * K1, YB + .5 * L1, ZC + .5 * M1)
M2 = H * FNC(XA + .5 * K1, YB + .5 * L1, ZC + .5 * M1)
K3 = H * FNA(XA + .5 * K2, YB + .5 * L2, ZC + .5 * M2)
L3 = H * FNB(XA + .5 * K2, YB + .5 * L2, ZC + .5 * M2)
M3 = H * FNC(XA + .5 * K2, YB + .5 * L2, ZC + .5 * M2)
K4 = H * FNA(XA + K3, YB + L3, ZC + M3)
L4 = H * FNB(XA + K3, YB + L3, ZC + M3)
M4 = H * FNC(XA + K3, YB + L3, ZC + M3)
X1 = XA + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
Y1 = YB + (1 / 6) * (L1 + 2 * L2 + 2 * L3 + L4)
Z1 = ZC + (1 / 6) * (M1 + 2 * M2 + 2 * M3 + M4)
IF PLOT$ = "Y" THEN
'FOR I = 1 TO 100 STEP .01: NEXT I
ON TYP GOTO 6, 7, 8
6 PSET (T, XA * 10): PSET (T, YB * 5), 1: PSET (T, ZC), 2
GOTO 9
7 PSET (Y1, Z1): GOTO 9
```

```
8 PSET (T, 100 * K1 / H), 1: PSET (T, 100 * L1 / H), 2
'PSET (T, 50 * M1 / H), 3
9 ELSE
IF N = NN * 20 THEN A$ = INPUT$(1): NN = NN + 1
PRINT USING " ####.#"; T;
ON TYP GOTO 11, 11, 12
11 PRINT USING "      ###.##"; XA * 1000000; YB * 500000; ZC * 100000
GOTO 13
12 PRINT USING "      ####.##"; K1 * 1E+07; L1 * 1E+07; M1 * 1E+07
13 END IF
T = T + H: XA = X1: YB = Y1: ZC = Z1
IF PLOT$ = "Y" THEN
DUM = FNM(N1, Y1, YMAX, YMIN, 5)
DUM = FNM(N2, Z1, ZMAX, ZMIN, 6)
END IF
60 NEXT N
IF PLOT$ <> "Y" THEN PRINT STRING$(50, "-")
END
```

Output




[Z] Verses [Y]
.0018
X*10:Brown
Y*5:Green
Z:Red
Y 482 .000099 600 .000011
Z 464 .001411 600 .000158
[Z]
0
0
[Y]--->
.00012


Variations of Rates
.0018
X*10:Brown
Y*5:Green
Z:Red
Y 482 .000099 600 .000011
Z 464 .001411 600 .000157
Y
Z
0
0
t/s--->
600

Comments The maxima and minima in the plots of [Y] and [Z] with time are as follows.

Max. [Y]	t/s →	94	288	482
Min. [Y]	t/s →	56	249	442
Max. [Z]	t/s →	77	270	464
Min. [Z]	t/s →	34	228	422

Iodate – Iodide Reaction

The scheme is

$$NaIO_3 \xrightarrow{k_1} Na^+ + \underset{(X)}{IO_3^-}$$

$$\underset{(X)}{IO_3^-} \xrightarrow[k_2]{AsO_3^-} \underset{(Y)}{I^-} + \text{products}$$

$$\underset{(X)}{IO_3^-} + \underset{(Y)}{2I^-} \xrightarrow[k_3]{AsO_3^-} \underset{(Y)}{3I^-}$$

$$\underset{(Y)}{I^-} \xrightarrow{k_4} \underset{B}{\frac{1}{2}\,I_2}$$

The involved equations are

$$\frac{d[X]}{dt} = k_1\,[A] - k_2\,[X] - k_3\,[X]\,[Y]^2$$

$$\frac{d[Y]}{dt} = k_2\,[X] + k_3\,[X]\,[Y]^2 - k_4\,[Y]$$

Use the following data for the computation of [X], [Y], d[X]/dt and d[Y]/dt

$k_1 = 0.001\ \text{min}^{-1}$, $k_2 = 0.01\ \text{min}^{-1}$, $k_3 = 2.5 \times 10^9\ \text{dm}^6\ \text{mol}^{-2}\ \text{min}^{-1}$, $k_4 = 1\ \text{min}^{-1}$,

$[A]_0 = 0.01\ \text{mol dm}^{-3}$, $[X]_0 = 2 \times 10^{-5}\ \text{mol dm}^{-3}$, $[Y]_0 = 1.0 \times 10^{-5}\ \text{mol dm}^{-3}$, and $[B]_0 = 0$

Make a program to display (i) $[IO_3^-]$ and $[I^-]$ with time, (ii) $[IO_3^-]$ versus $[I^-]$, and (iii) the variations of rates of $[IO_3^-]$ and $[I^-]$ with time. Also determine the periodic time varations for the maximum and minimum in $[IO_3^-]$ and $[I^-]$.

Program

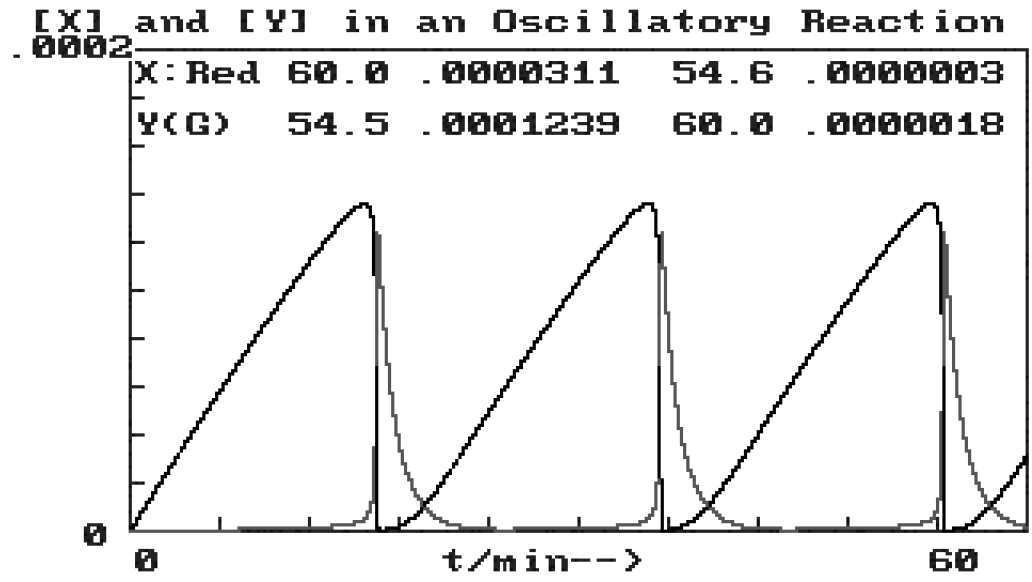
```
REM Oscillatory Reaction(Iodate-iodide)
REM TYP=1 for CONC verses TIME;TYP=2 for X verses Y
REM TYP=3 Rates verses time
REM PLOT& is "Y" if plotting is required. If not,
REM numerical values are displayed
CLS : READ PLOT$, TYP, A0, TF: DATA Y,3,.0002,60
READ R1, R2, R3, R4, CA: DATA .001,.01,2.5E9,1,.01
X0 = 0: Y0 = 0: H = .0005: NMAX = TF / H: NN = 0: A$ = "t/min-->"
XMAX = 0: XMIN = A0: YMAX = 0: YMIN = A0: M = 1: M1 = 1
```

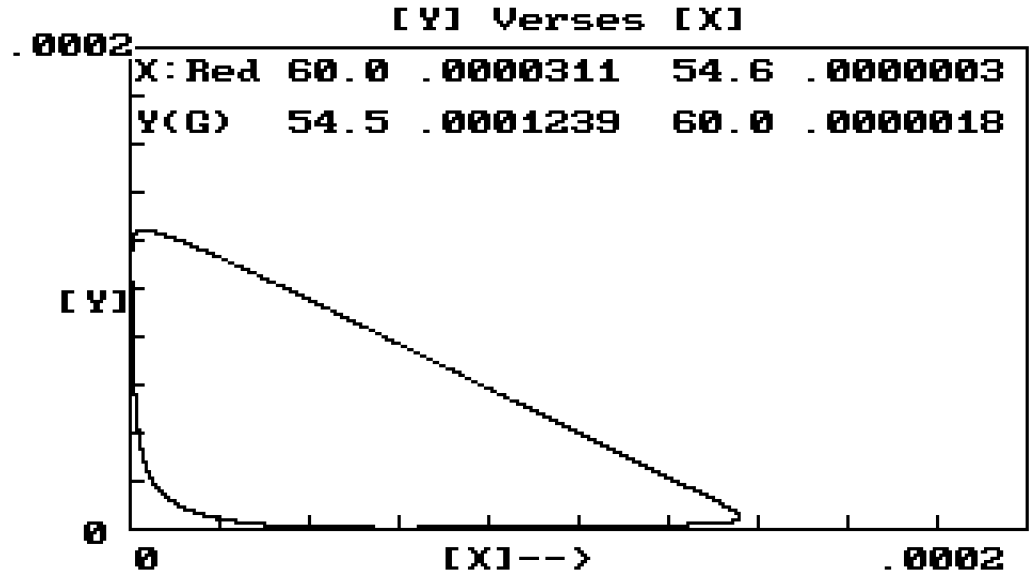
```
REM Indentifing maxima and minima in the plots of [X] &[Y]
DEF FNM (Q, P1, PMAX, PMIN, S)
IF S = 4 THEN C$ = "R" ELSE C$ = "G"
ON Q GOTO 15, 20
15 IF P1 > PMAX THEN
PMAX = P1: TPMAX = T: LOCATE S, 11
PRINT USING " ##.#"; TPMAX; : PRINT USING " .#######"; PMAX
ELSEIF P1 < PMAX THEN
LOCATE 5, 16: PRINT "Max("; C$; ")": A$ = INPUT$(1)
LOCATE 5, 16: PRINT "        ": PMIN = A0: Q = 2
END IF
GOTO 25
20 IF P1 < PMIN THEN
PMIN = P1: TPMIN = T: LOCATE S, 26
PRINT USING " ##.#"; TPMIN; : PRINT USING " .#######"; PMIN
ELSEIF P1 > PMIN THEN
LOCATE 5, 31: PRINT "Min("; C$; ")": A$ = INPUT$(1)
LOCATE 5, 31: PRINT "        ": PMAX = 0: Q = 1
END IF
25 IF S = 4 THEN M = Q: XMAX = PMAX: XMIN = PMIN: GOTO 26
M1 = Q: YMAX = PMAX: YMIN = PMIN
26 END DEF
DEF FNA (X, Y) = R1 * CA - R2 * X - R3 * X * Y ^ 2
DEF FNB (X, Y) = R2 * X + R3 * X * Y ^ 2 - R4 * Y
IF PLOT$ = "Y" THEN
SCREEN 1: COLOR 15, 0: VIEW (38, 20)-(318, 170)
ON TYP GOTO 1, 2, 3
1 XI = 0: XF = TF: YI = 0: YF = A0: GOTO 4
2 XI = 0: XF = A0: YI = 0: YF = A0: A$ = "[X]-->"
LOCATE 13, 2: PRINT "[Y]": GOTO 4
3 XI = 0: XF = TF: YI = -A0: YF = A0
4 WINDOW (XI, YI)-(XF, YF): LINE (XI, YI)-(XF, YF), , B
LOCATE 2, 1
ON TYP GOTO 31, 32, 33
31 PRINT " [X] and [Y] in an Oscillatory Reaction": GOTO 34
32 PRINT "                  [Y] Verses [X]": GOTO 34
33 PRINT "                  Variations of Rates"
34 LOCATE 3, 1: PRINT USING ".####"; YF
LOCATE 4, 6: PRINT "X:Red": LOCATE 6, 6: PRINT "Y(G)"
LOCATE 22, 4: PRINT "0"
LOCATE 23, 6: PRINT "0             "; A$; "            "; XF
IF TYP = 3 THEN LINE (XI, 0)-(XF, 0)
FOR I = 1 TO 9
```

```
IF TYP = 3 THEN
XS = I * XF / 10: YS = YI + (YF - YI) * I / 10
LINE (XS, YI)-(XS, YI + .00001)
LINE (XI, YS)-(XI + 1, YS)
ELSE
LINE (I * XF / 10, 0)-(I * XF / 10, YF * .03)
LINE (XI, YF * I / 10)-(XF / 70, YF * I / 10)
END IF
NEXT I
ELSE
PRINT STRING$(30, "-"): PRINT " t/min    [X]*10000    [Y]*10000"
PRINT STRING$(30, "-")
END IF
T = 0: XA = X0: YB = Y0
FOR N = 0 TO NMAX
K1 = H * FNA(XA, YB)
L1 = H * FNB(XA, YB)
K2 = H * FNA(XA + .5 * K1, YB + .5 * L1)
L2 = H * FNB(XA + .5 * K1, YB + .5 * L1)
K3 = H * FNA(XA + .5 * K2, YB + .5 * L2)
L3 = H * FNB(XA + .5 * K2, YB + .5 * L2)
K4 = H * FNA(XA + K3, YB + L3)
L4 = H * FNB(XA + K3, YB + L3)
X1 = XA + (1 / 6) * (K1 + 2 * K2 + 2 * K3 + K4)
Y1 = YB + (1 / 6) * (L1 + 2 * L2 + 2 * L3 + L4)
IF PLOT$ = "Y" THEN
'FOR I = 1 TO 100 STEP .1: NEXT I
ON TYP GOTO 6, 7, 8
6 PSET (T, XA), 2: PSET (T, YB), 1: GOTO 9
7 PSET (X1, Y1): GOTO 9
8 PSET (T, K1 * 4 / H), 2: PSET (T, L1 * 4 / H), 1
9 ELSE
IF N = 20 * NN THEN A$ = INPUT$(1): NN = NN + 1
PRINT USING "   ##.#"; T;
ON TYP GOTO 11, 11, 12
11 PRINT USING "    ##.######"; XA; YB: GOTO 13
12 PRINT USING "    ##.####"; K1 * 1000 / H; L1 * 1000 / H
13 END IF
T = T + H: XA = X1: YB = Y1
IF PLOT$ = "Y" THEN
DUM = FNM(M, X1, XMAX, XMIN, 4)
DUM = FNM(M1, Y1, YMAX, YMIN, 6)
END IF
```

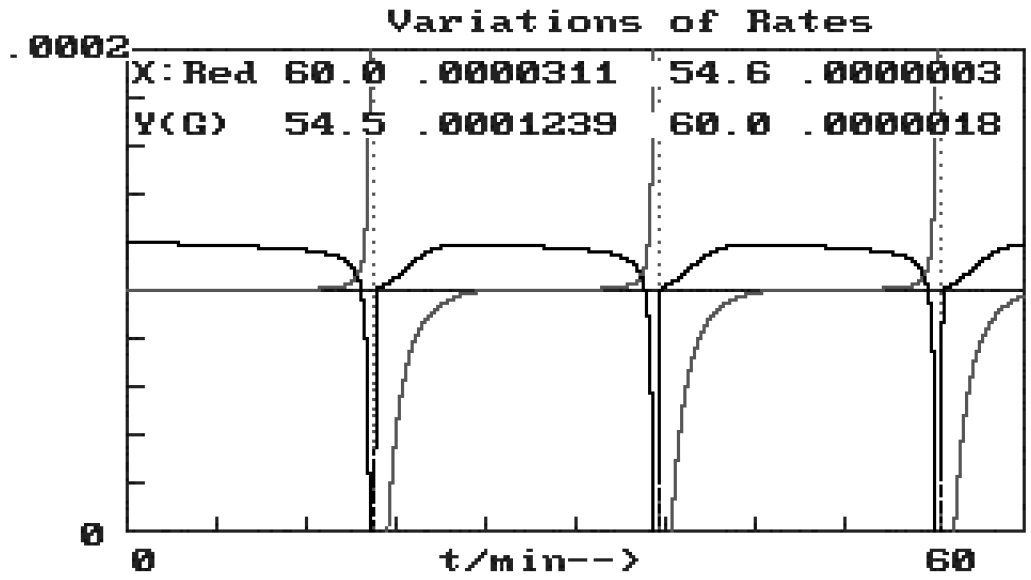
```
40 NEXT N
IF PLOT$ <> "Y" THEN PRINT STRING$(30, "-")
END
```

Output





Press any key to continue



Comments The maxima and minima in the plots of [X] and [Y] with time are as follows

Max. [X]	t/min →	15.7	34.6	53.6
Min. [X]	t/min →	16.8	35.7	54.7
Max. [Y]	t/min →	16.6	35.5	54.5
Min. [Y]	t/min →	25.0	44.0	–