# CHAPTER 2 RULES AND DEFINITIONS

## 2.1 Character Set

A program unit is written using the following characters:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U,
V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,

and the following special characters:

| | |
|---|---|
| or b | Space |
| = | Equals |
| + | Plus |
| - | Minus |
| * | Asterisk |
| / | Slash |
| ( | Left Parenthesis |
| ) | Right Parenthesis |
| , | Comma |
| . | Radix Point |
| ' | Apostrophe |

## 2.2 Source Program Format

A source program file is made up of statements and comments. A statement may be contained on one or several lines. A comment is contained on one line, it is not considered as a statement, and merely provides information for documentary purposes. Comment lines may be placed freely in the program file.

Every program unit (main program, subprogram, etc.) must terminate with an end line. This line contains an END statement and serves to separate individual program units. Any subsequent units must begin on a new line.

Lines in FORM format have the following characteristics:

1. Comment lines are recognized by a C in character position 1.
2. Continuation lines are recognized by a non-blank, or non-zero character in position 6.
3. Character position 73-80 may be used for sequence identification information. This field is not considered part of the statement, it is provided for programmer convenience.
4. No more than 80 characters will be processed.

Figure 2-1 illustrates the appearance and general properties of a FORTRAN program written on a coding sheet. This example illustrates the FORM format.

FORM format is for IBM 80 columns cards in this textbook.

```
C     THIS IS AN EXAMPLE OF A SIMPLE FORTRAN PROGRAM
    1 READ(5,18) A,B,C
      IF(A) 3,20,3
    3 D=B*B-4.*A*C
      IF(D.LT.0.0) GO TO 2
      X1=(-B+SQRT(D))/(2.*A)
      X2=(-B-SQRT(D))/(2.*A)
      WRITE(6,14) A,B,C,X1,X2
      GO TO 1
    2 WRITE(6,16) A,B,C
      GO TO 1
C     THE FOLLOWING THREE CARDS SHOW THE INPUT/OUTPUT FORMAT
   14 FORMAT(1H ,3F12.4,10HROOTS ARE ,2F12.4)
   16 FORMAT(1H ,3F12.4,'ROOTS ARE IMAGINARY')
   18 FORMAT(3F12.4)
   20 STOP
      END
```

Fig. 2.1 An Example of a Simple Fortran Program

## 2.3 Simbol Formation And Data Type

A simbolic name consists of one to eight* alphanumeric characters, the first of which must be alphabetic. Data types may be associated with a symbolic name either implicitly or explicitly. The implicit associations are determined by the first character of the symbol; integer if the name begins with the letters I,J,K,L,M,or N; otherwise real.

An explicit declaration of type for some symbol always overrides its implicit type. Data type is explicitly associated with a simbol when it appears in one of the statements: INTEGER, REAL,and so on, or when it appears in a FUNCTION statement with a type prefix (e.g., REAL FUNCTION MART(A,B)).

A simbolic name representing a function, variable, or array has only one data type association for each program unit. Once associated with a particular data type, a specific name implies that type for any usage of that symbolic name that requires a data type association throughout the program unit in which it is defined.

The mathematical and representational properties for each of the data types are defined below.

1. An integer datum is always an exact representation of an integer value. It may assume positive, negative, or zero integral values. Each integer datum requires one 36 bit word of storage in fixed point format* The permissible range of values for integer type is $-2^{35}$ to $2^{35}-1$*
2. A real datum is an approximation to the value of a real number. It may assume positive, negative, or zero values, possibly fractional. A real datum requires one 36 bit bit word of storage in floating point format* The permissible range of values for real type is approximately $\pm 10^{-38}$ to $\pm 10^{38}$ with a precision of eight digits*

* These descriptions are valid for TOSBAC-5600 in PHRI.

## 2.4 Constants

A constant is a value that is known prior to writing a FORTRAN statement and which does not change during program execution.

An integer constant consists of 1 to 11 decimal digits with an accuracy of 10 digits* The decimal point of integers are always omitted; however, it is always assumed to be immediately to the right of the last digit in the string. An integer constant may be as large as $2^{35}-1$ (=$3.4\times10^{10}$)* except when used for the value of a subscript or as an index of a DO or a DO parameter, in which case the maximum value of the integer is $2^{18}-1$ (=$2.6\times10^{5}$)*

Examples: -5 8 567 53684

A real constant is in floating-point mode and is contained in one computer word (single precision). This constant consists of one of the following;

1. One to nine* significant decimal digits written with a decimal point, but not followed by a decimal exponent.
2. One to nine* significant decimal digits written with or without a decimal point, followed by a decimal exponent written as the letter E followed by a signed or unsigned one or two digit integer constant. When the decimal point is omitted, it is always assumed to be immediately to the right of the rightmost digit. The exponent value may be explicitly 0, and the field following the E may not be blank.

Examples: 35. 6503. 45.638 -3.05 0.0008
6.0E2 (means $6.0\times10^{2}$; 600)
6E-3 (means $6.0\times10^{-3}$; 0.006)

A real constant has precision to eight digits*. The magnitude must be between the approximate limits of $10^{-38}$ and $10^{38}$*, or must be zero.

## 2.5 Variables

A variable is any quantity that is reffered to by name rather than by value. A variable may take on many values and may be changed during the execution of the program.

The type of a variable is specified implicitly by its name, or explicitly by use of a type statement.

1. Default implicit type association enables the declaration of real and integer variables and function names according to the following rules;
   a. If the first character of the name is I,J,K,L,M,or N, it is an integer name.
   b. If the first character is any other alphabetic character, it is a real name.
2. The explicit type statements (described in this textbook) are used to assign a type to a variable or function subprogram.
3. Function subprogram names may be typed on the FUNCTION statement by use of the type prefix.

An integer variable and a real variable occupy the same number of storage locations as a constant of the same type.

An external variable is the name of a subprogram that appears as an actual argument in the calling sequence to some subprogram. It must appear in an EXTERNAL statement before its first use in the source program.

An array is an ordered set of data with from one to seven* dimensions. The array is referenced by a symbolic name. Identification of the entire ordered set is achieved by the use of the array name.
An item of data in an array is called an array element. It is identified by immediately following the array name with a subscript that points to the particular element of the array.

A variable may be made to represent any element of an array containing from one to seven dimensions by appending one to seven subscripts to the variable name. Subscript expressions are separated by commas. The number of subscript expressions must correspond with the declared dimensionality except in an EQUIVALENCE statement.

A subscript expression may take the form of I, c*I, I±d, c*I±d, where c and d are integer constants.

Examples:

| | |
|---|---|
| IMAS | 8*IQUAN |
| J9 | 5*L+7 |
| K2 | 3*M-3 |
| N+3 | J-9 |

The value of a subscript expression must be greater than zero and not greater than the corresponding array dimension.

Subscripted Variables

A subscripted variable consists of a variable name,followed by parentheses, enclosing one to seven subscripts separated by commas.

Examples:

A(I)
K(3)
BETA (8*J+2,K-2,L)
MATH(I,J,K,L,M,N)

1. During execution, the subscript is evaluated so that the subscripted variable refers to a specific member of the array.
2. Each variable that appears in subscripted form must have the size of the array specified. This must be done by DIMENSION statements that contains the dimension information. The specification of dimensionality must precede the first reference to the array.

## 2.6 Expressions

### 2.6.1 Arithmetic Expressions

An arithmetic expression consists of certain legal sequences of constants, subscripted and nonsubscripted variables, and arithmetic function references separated by arithmetic operation symbols, commas and parentheses.

The following arithmetic operation symbols denote addition, subtraction, multiplication, division, and exponentiation, respectively:

+ - * / **

The rules for constructing arithmetic expressions are:

Figures 2-2 and 2-3 indicate which constants, variables, and functions may be combined by the arithmetic operators to form arithmetic expressions. The intersection of a row and column gives the type of the result of such an expression. Figure 2-2 gives the valid combinations with respect to the arithmetic operators +, -, *, and /. Figure 2-3 gives the valid combinations with respect to the arithmetic operators **.

| | I | R |
|---|---|---|
| I | I | R |
| R | R | R |

Figure 2-2

| BASE \ POWER | I | R |
|---|---|---|
| I | I | R |
| R | R | R |

Figure 2-3

Legend
I-Integer
R-Real

2. Any expression may be enclosed in parentheses.

3. Expressions may be connected by the arithmetic operation symbols to form other expressions, provided that:

   a. No two operators appear in sequence except **, which is a single operator and denotes exponentiation.

   b. No operation symbol is assumed to be present. For example, (X) (Y) is not valid.

4. The expression A**B**C is evaluated as A**(B**C).

5. Preceding an expression by a plus or minus sign does not affect the type of the expression.

6. In the hierarchy of operations, parentheses may be used in arithmetic expressions to specify the order in which operations are to be computed. Where parentheses are omitted, the order is understood to be as follows:

   a. Function Reference

   b. ** Exponentiation

   c. * and / Multiplication and Division

   d. + and - Addition and Subtruction

   This hierarchy is applied first to the expression within the innermost set of parentheses in the statement; this procedure continues through the outer parentheses until the entire expression has been evaluated.

7. Operations on the same level (e.g.,A*B/C) are evaluated left to right. Parentheses may be used to reorder this sequence if necessary.

The FORTRAN expression

```
A*6+Z/Y**(W+(A+B)/X**K)
```

represents the mathematical expression

$$6A+\frac{Z}{Y^{\left(W+\frac{(A+B)}{X^K}\right)}}$$

## 2.6.2 Relational Expressions

A rerational expression consists of two arithmetic expressions connected by a relational operator. Relational expressions always result in a true or false evaluation. Relational expressions are logical operands and can be used in a logical IF statement.

The six relational operation symbols are:

| Symbol | Definition |
|---|---|
| .GT. | Greater than |
| .GE. | Greater than or equal to |
| .LT. | Less than |
| .LE. | Less than or equal to |
| .EQ. | Equal to |
| .NE. | Not equal to |

Examples: A.GT.B A.EQ.B A.LE.B

### 2.6.3 Logical Expressions

A logical expression consists of certain sequences of logical constants, logical variables, references to logical functions, and relational expressions separated by logical operation symbols. A logical expression always results in a true or false evaluation.

The logical operation symbols (where a and b are logical expressions) are:

| Symbol | Definition |
|---|---|
| .NOT.a | This has the value .TRUE. only if a is .FALSE.; it has the value .FALSE. only if a is .TRUE. |
| a.AND.b | This has the value .TRUE. only if a and b are both .TRUE.; it has the value .FALSE. if a or b or both are .FALSE. |
| a.OR.b | (INCLUSIVE OR) This has the value .TRUE. if either a or b or both are .TRUE.; it has the value .FALSE. only if both a and b are .FALSE. |

.NOT., .AND., and .OR. must always be preceded and followed by a period.

Example: `IF (F(X).GT.3.5.OR. L.EQ.3) GO TO 100`

Unary Operators

The unary operators, negative, positive, and logical not, may immediately precede a constant or a variable in an expression; however, if the placement causes the unary negative or positive operator to be adjacent operator, it must be enclosed in parentheses with the constant or variable.

Examples:

```
A=+1.6
C=D/(-Z)*W
IF(-3.+T4)1,2,3
L1=R2.GT.(-2.)
L2=.NOT.P.GT.Q
A=B**(-2)
```

## 2.7 FORTRAN Statements

The basic unit of FORTRAN is the statement. Statements are classified according to the following uses:

1. Assignment statements specifying numerical value assignment.

2. Control statements governing the order of execution in the object program.

3. Input/Output statements and input/output formats which describe the form of the data.

4. Subprogram statements enabling the programmer to define and use subprograms.

5. Specification statements providing information about variables used in the program, information about storage allocation and data assigned.

6. Compiler control statements direct the compilation activety.

# CHAPTER 3 FORTRAN STATEMENTS

## 1. Assignment Statements

### 1.1 Arithmetic Assignment Statement

The arithmetic assignment statement has the general form:

v = e

where v is an unsigned variable name or array element name of an arithmetic type (integer, real) and e is an arithmetic expression. An arithmetic assignment statement causes FORTRAN to compute the value of the expression on the right and to give that value to the variable on the left of the equal sign.

The following examples show various arithmetic assignment statement: where

R1 and R2 are real variables
I is an integer variable

| | |
|---|---|
| R1 = R2 | R2 replaces R1 |
| I = R2 | R2 is truncated to an integer, and stored in I |
| R1 = I | I is converted to a real variable and stored in R1. |

### 1.2 Label Assignment Statement

A label assignment statement is of the form:

ASSIGN k TO i

where K is a statement number and i is a nonsubscripted integer variable. The statement number must refer to an executable statement in the same program unit in which the ASSIGN statement appears. For example:

```
ASSIGN 24 TO M
    .
    .
    .
GO TO M,(1,22,41,24,36)
```

## 1.3 Arithmetic Statement Function

An arithmetic statement function is defined to the program unit in which it is referenced. It is defined by a single statement similar in form to the arithmetic assignment statement.

In a given program unit, all arithmetic statement function definitions must precede the first executable statement of the program unit. The name of an arithmetic statement function must not appear in EXTERNAL, COMMON or EQUIVALENCE statements, as a scalar name, or as an array name in the same program unit.

An arithmetic statement function definition has the form:

$f(a_1,a_2,\ldots,a_n) = e$

where f is the function name, the $a_i$ are distinct symbolic name (called dummy arguments of the function), and e is an expression. Since the $a_i$ are dummy arguments, their names, which serves only to indicate number, and order of arguments, may be the same as actual variable names appearing elsewhere in the program unit.

For example:

A function can be defined to compute one root of the quadratic equation, $ax^2+bx+c=0$, given values a, b, and c as follows:

```
ROOT (A,B,C)=(-B+SQRT(B**2-4*A*C))/(2*A)
```

This is the definition of the function. This definition can be used by supplying values for a, b, and c. An example of the use of the function using 16.9 for a, 20.5 for b, and T for c follows:

```
ANS = ROOT(16.9,20.5,T)
```

## 2. Input/Output Statements

### 2.1 READ Statement

READ(u,f)  list

This statement, formatted file READ, includes a reference to a file reference (u) and format information (f).
The file reference (u) may be an integer constant, or variable. A designation of 5 implies the system standard input card reading device. The FORMAT reference (f) refers to the statement label of a FORMAT statement, which must be an integer constant.

List Specifications

When arrays or variables are to be transmitted, an ordered list of the quantities to be transmitted must be included in the input/ output statements. The order of the input/output list must be the same as the order in which the information exists or is to exist on the input/output medium.

An input/output list is a string of list items separated by commas. A list item may be:

1. A scalar
2. An array name
3. An array element
4. An implied DO

An input/output list is processed from left to right. Parenthesized sublists are permitted only with implied DO's; redundant parentheses will result in a fatal diagnostic.

Examples: A, B, I, J, F(K,L)

Consider the following input/output list:

```
A,B(3), (C(I),D(I,K),I=1,10),
((E(I,J),I=1,10,2),F(J,3),J=1,K)
```

This list implies that the information in the external input/output medium is arranged as follows:

```
A,B,(3),C(1),D(1,K),C(2),D(2,K),....,C(10),D(10,K),
E(1,1), E(3,1),...,E(9,1),F(1,3),
E(1,2),E(3,2),....,E(9,2),F(2,3),E(1,3),...,F(K,3)
```

The execution of an input/output list is exactly that of a DO loop, as though each left parentheses (except expression and subscripting parentheses) were a DO, with indexing given immediately before the matching right parentheses, and the DO range extending up to that indexing information. The order of the input/output list above may be considered equivalent to the followings:

```
  A
  B(3)
  DO 5 I=1,10
  C(I)
5 D(I,K)
  DO 9 J=1,K
  DO 8 I=1,10,2
8 E(I,J)
9 F(J,3)
```

By specifying an array name in the list of an input/output statement an entire array can be designated for transmi  ion between core storage and an input/output device. Only the name of the array need be given and the indexing information may be omitted. For example:

```
DIMENSION A(3,5)
    .
    .
READ(u,f) A
```

In the above example, the READ statement shown will read the entire array A; the array is stored in column order in increasing storage locations, with the first subscript varying most rapidly, and the last varying least rapidly. So that, in this case, the above READ statement is the same as

```
READ(u,f) ((A(I,J),I=1,3),J=1,5)
```

## 2.2 WRITE Statement

WRITE (u,f) list

The formatted file WRITE statement must include a file reference (u) and a FORMAT reference (f).

The file reference (u) may be an integer constant, or variable. Adesignation of 6 implies the system standard output printing device.

The FORMAT reference (f) refers to the statement label of a FORMAT statement, which must be an integer constant.

## 2.3 FORMAT Statement

The FORMAT statement is used in conjunction with formatted input/output statements.

A FORMAT statement has the form:

m FORMAT $(t_1,t_2,t_3,\ldots,t_n)$

where

m is the statement number
t is a field descriptor

In this form, the following field descriptors are permitted:

rFw.d
rEw.d
rGw.d
rIw
} ---------- Numeric Field Descriptors

$wHh_1h_2\ldots h_w$
$'h_1h_2\ldots h_w'$
} ------Character Field Descriptors

wX ------------ Field Positioning Descriptors

where

r is an optional repeat count

w is the field width, expressed in number of characters

d is the number of fractional places (characters)

$h_i$ is a single character

The F, E, and G descriptors are for REAL values, I is for INTEGER,H is for CHARACTER values, X is for skipping over text. The following briefly describes how these descriptors are formed. Note that the last two, H, and X, do not require a variable in the input/output list; all others do.

Fw.d = Real mode without exponent

Ew.d = Real mode with exponent

Gw.d = F or E editing code is taken dependent on value of output item

Iw = Integer mode and field occupies w print positions

wH = Hollerith field to occupy w print positions

wX = Field of width w is blank filled on output, skipped on input

OUTPUT DEVICE CONTROL

The spacing of the printing on the output device is controlled by the first chracter of the line of output. The first character is not printed but is examined to determine if it is a control character to regulate the spacing of the output device. If the first character is recognized as a control character, the line is printed after the proper spacing has been effected. In any event, it is deleted when the line is printed. This control affects printers, teletypewriters, and displays.

The effects produced by control characters are :

| Character | Effect |
|---|---|
| 0 | 1 blank line prior to print |
| + | Overprint |
| 1 | Slew to top of page before printing |
| (blank) | Space to next line |

## 2.4 REWIND Statement

This statement refers only to sequential files. It causes the specified file to be positioned at its initial point.
The statement has the following form:

REWIND u

where u is the file reference. u may be an integer constant or variable.

If the file is an output file, an EOF is written before rewinding.

## 2.5 ENDFILE Statement

This statement is operable only for sequential files. Its execution causes the indicated file to be closed with an end-of-file signal. For an output file, the buffer(s) is flushed and a file mark is written. Nothing is done for an input file.(The end-of-file signal is a unique record indicating demarcation of a sequential file.) This statement has the form:

ENDFILE u

where u is the file designator (unit number). u may be an integer constant or variable.

## 3. Control Statements

### 3.1 GO TO Statement

#### 3.1.1 GO TO, Unconditional

The unconditional GO TO indicates the next statement to be executed. It has the form:

GO TO K

where K is the statement number of another statement in the program. When this statement is encountered, the next statement to be executed will be the statement having statement number K. This statement can be any executable statement in the program either before or after the GO TO statement subject to the rules for transferring into and out of DO loops. For example:

```
GO TO 5
```

The program continues execution with statement number 5. Control is transferred unconditionally to statement number 5.

#### 3.1.2 GO TO, Assigned

The assigned GO TO statement indicates which statement is the next to be executed. The assigned GO TO has the form:

GO TO I, $(K_1,K_2,\ldots,K_n)$

or GO TO I

where

I is an integer switch variable

$K_i$ are statement numbers

The K's are optional. If present, then at the time of execution of an assigned GO TO statement the variable I must have been assigned the value of one of the statement numbers in the parentheses. The next statement to be executed will be the one whose statement number in the parentheses has the same value as the variable I. If statement I is not in the list of K's, a run time diagnostic is generated.

When the K's are not present (GO TO I, above), no validation of I takes place. Control transfers directly to statement I.

For example:

```
ASSIGN 17 TO J
GO TO J, (5,4,17,2)
```

Statement number 17 is executed next.

### 3.1.3 GO TO, Computed

The computed GO TO indicates the statement that will be executed next. This is determined by using a computed integer value. It has the following form:

GO TO $(K_1, K_2, \ldots, K_n)$,e

Where the $K_i$ are statement labels or switch variables. The expression e is truncated to an integer at the time of execution. The next statement to be executed will be $K_i$ where i is the integral value of the expression e. If i is out of range, a message is outputed and execution is terminated.

For example:

```
J=3
GO TO (5,4,17,1), J
```

Statement 17 is executed next.

## 3.2 IF Statement

### 3.2.1 IF, Arithmetic

The arithmetic IF statement causes a change in the execution sequence of statements depending on the value of an arithmetic expression. It has the following form:

IF (e) $K_1, K_2, K_3$

where e is an arithmetic expression and the $K_i$ are statement numbers

The arithmetic IF is a three-way branch. Execution of this statement causes a transfer to one of the statements $K_1$, $K_2$, or $K_3$. The statement identified by $K_1$, $K_2$, or $K_3$ is executed next depending on whether the value of e is less than zero, zero, or greater than zero, respectively

Example:

```
IF  (A(J, K)-B) 10, 4, 30

IF  (A(J, K)-B) 0 control goes to statement 10
IF  (A(J, K)-B)=0 control goes to statement 4
IF  (A(J, K)-B) 0 control goes to statement 30
```

### 3.2.2 IF, Logical

The logical IF statement causes conditional execution of a certain statement depending on whether or not a logical expression is true or false. It has the following form:

IF(e)s

where e is a logical or relational expression and s is any executable statemant except a DO statement or another logical IF statement. Upon execution of this statement, the logical or relational expression e is evaluated. If the value of e is true, statement s is executed. If the value of e is false, control is transferred to the next sequential statement.

Example:

```
IF(A.GT.B) GO TO 3
```

If A is arithmetically greater than B, the execution of the user program continues with the statement labeled with 3. Otherwise execution continues with the next sequential executable statement.

If e is true and s is a CALL statement that does not take a nonstandard return, control is transferred to the next sequential statement upon return from the subprogram.

The following examples illustrate several uses of the logical IF.

1. IF (NOT.A.G T.B) F = SIN (R)
2. IF (16.GT.L) GO TO 24
3. IF (D.G T.E.OR.X.LE.Y) GO TO (18,10),I
4. IF (K.E Q.0) CALL SUB

In example 1, if (NOT.A.G T.B) is true, compute F and return to the statement following IF.

In example 2, if (16.GT.L), control transfers to statement 24.

In example 3, if (D.G T.E.OR.X.LE.Y) is true, control transfers to statement 18 or 20 depending upon whether I is 1 or 2.

In example 4, if K=0, control goes to the subprogram SUB. Return is to the statement following the IF.

### 3.3 DO Statement

This statement enables the user to execute a section of a program repeatedly, with automatic changes in the value of a variable between repetitions. The DO statement may be written in either of these forms:

DO n i = $m_1$,$m_2$,

or

DO n i = $m_1$,$m_2$,$m_3$

In these statements n must be a statement number of an executable statement, i must be a nonsubscripted integer variable, and $m_1$,$m_2$,$m_3$ may be any valid arithmetic expression. If $m_3$ is not stated, it is understood to be 1 (first form). These parameters ($m_1$,$m_2$,$m_3$) are truncated to integers before use.

i: the induction variable
$m_1$: the initial parameter
$m_2$: the terminal parameter
$m_3$: the step parameter

The statements following the DO up to and including the one with statement number n are executed repeatedly. They are executed first with i=m; before each succeeding repetition i is increased by $m_3$ (when present, otherwise by 1); when i exceeds $m_2$ execution of the DO is ended.

1. The terminal statement (n) may not be a GO TO (of any form), IF, RETURN, STOP or DO statement.

2. The range of a DO statement includes the executable statements from the first executable statement following the DO to and including the terminal statement (n) associated with the DO.

3. Another DO statement is permitted within the range of a DO statement. In this case, the range of the inner DO must be a subset of the range of the outer DO.

4. The values of $m_1$, $m_2$ and $m_3$ must all be positive and $m_3$ may not be zero; $m_1$ cannot be the constant zero but can be a variable whose value is zero. If $m_2$ is less than or equal to $m_1$ the loop will be processed once.

5. None of the control parameters, i, $m_2$, or $m_3$, may be redefined within the loop or in the extended range of the loop, if such exists.

A completely nested set of DO statements is a set of DO statements and their ranges such that the first occurring terminal statement of any of those DO statements physically follows the last occurring DO statement.

If a statement is the terminal statement of more than one DO statement, the statement number of that terminal statement may not be used in any GO TO or arithmetic IF statement that occurs anywhere but in the range of the innermost DO with that as its terminal statement.

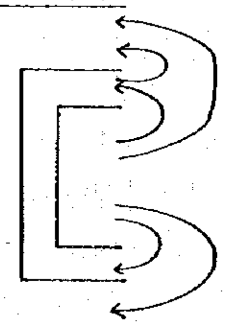
A DO statement is used to define a loop. The action succeeding execution of a DO statement is described n the following steps:

1. The induction variable, i, is assigned the value represented by the initial parameter ($m_1$).
2. Instruction in the range of the DO are executed.
3. After execution of the terminal statement the induction variable of the most recently executed DO statement associated with the terminal statement is changed by the value represented by the associated step parameter ($m_3$).
4. If the value of the induction variable after change is less than or equal to the terminal value, then the action described starting at the 2nd step is repeated, with the understanding that the range in question is that of the DO, whose induction variable has been most ecently changed. If the value of the induction variable is greater than the terminal value, then the DO is said to have been satisfied.
5. At this point, if there were one or more other DO statements referring to the terminal statement in question, the induction variable of the next most recently executed DO statement is changed by the value represented by its assosiated step parameter and the action described in the 4th step is repeated until all DO statements referring to the particular termination statement are satisfied, at which time all such nested DO's are said to be satisfied and the first executable statement following the terminal statement is executed.

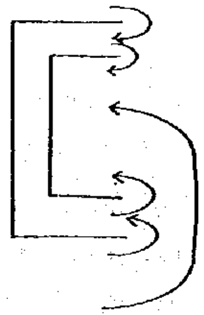
Transfer of Control

The following configurations show permitted and nonpermitted transfers.

Permitted



Not Permitted



An example of the DO statement follows:

```
    K = 0
    DO 10 I = 1,3
    DO 10 J = 1,2
    K = K + I + J
 10 CONTINUE
```

where the K values are computed as:

```
       OLD        NEW
        K   I   J   K

    K = 0
    K = 0 + 1 + 1 = 2
    K = 2 + 1 + 2 = 5
    K = 5 + 2 + 1 = 8
    K = 8 + 2 + 2 = 12
    K =12 + 3 + 1 = 16
    K =16 + 3 + 2 = 21
```

### 3.4 CONTINUE Statement

The CONTINUE statement is a dummy statement most often used as the last statement in the range of a DO, when the last statement would otherwise have been a GO TO or IF. It has the following form:

```
CONTINUE
```

For example:

```
10 DO 12 I = 1, 10
   IF (ARG - VAL(I)) 12, 13, 12
12 CONTINUE
```

Execution of this statement causes a continuation of the normal execution sequence.

### 3.5 STOP Statement

The STOP statement causes the object-program to halt and control to be returned to the operating system. It has the forms:

```
STOP
```

## 4. Subprogram Statements

### 4.1 SUBROUTINE Statement

The SUBROUTINE statement must be the first statement of a SUBROUTINE subprogram. The SUBROUTINE statement has the following form:

SUBROUTINE name ($a_1$, $a_2$, ..., $a_i$)
or
SUBROUTINE name

where

Name is the symbolic name of a subprogram and must be unique within the first six characters.

Each $a_i$ (if present) is a nonsubscripted variable name, the dummy name of a SUBROUTINE or FUNCTION subprogram.

Examples:

```
SUBROUTINE COMP (X,Y,P)
SUBROUTINE QUADEQ (B,A,C,ROOT1, ROOT2)
SUBROUTINE OUTPUT
```

1. The SUBROUTINE statement must be the first statement of a SUBROUTINE subpragram.
2. The arguments may be considered dummy variable names that are replaced at the time of execution by the actual arguments supplied in the CALL statement which refers to the SUBROUTINE subprogram. The actual arguments must correspond in number, order, size and type with the dummy arguments.
3. When a dummy argument is an array name, a statement containing dimension information must appear in the SUBROUTINE subprogram; the corresponding actual argument in the CALL statement must be a dimensioned array name.
4. No argument in a SUBROUTINE statement may also be included in COMMON, EQUIVALENCE or DATA statements in the subprogram.
5. The SUBROUTINE subprogram must be logically terminated by a RETURN statement and physically by an END statement.
6. The SUBROUTINE subprogram may contain any FORTRAN statements except FUNCTION or another SUBROUTINE statement. Using functions, or calling other subroutine in the SUBROUTINE SUBPROGRAM is permitted.

## 4.2 CALL Statement

The CALL statement is used to refer to a SUBROUTINE subprogram.

A CALL statement is one of the forms:

CALL s($a_1$, $a_2$,..., $a_n$)

or

CALL s

where s is the name of a SUBROUTINE and the $a_i$ are actual arguments.

The execution of a CALL statement references the designated subroutine. Execution of the CAll statement is completed upon return from the designated subroutine.

Example: CALL MATMISL(A,B,C,I,J,K)

Execution of the user program continues with the first executable statement of the SUBROUTINE (or SUBROUTINE entry point) MATMISL.

Additional examples:

```
CALL MATMPY(X,5,10,Y,7,2)
CALL OUTPUT
```

The CALL statement transfers control to a SUBROUTINE subprogram and presents it with the actual arguments.

The arguments may be any of the following:

1. A constant.
2. A subscripted or nonsubscripted variable or an array name.
3. The name of a FUNCTION or SUBROUTINE subprogram.

The arguments presented by the CALL statement must agree in number, order, type, and array size (except as explained under the DIMENSION statement) with the corresponding dummy arguments in the SUBROUTINE statement of the called subprogram.

### 4.3 RETURN Statement

The logical termination of any subprogram is the RETURN statement, which returns control to the calling program. There may be any number of RETURN statements in the program.

A RETURN statement is of the form:

RETURN

### 4.4 FUNCTION Statement

The FUNCTION statement is the first statement of a FUNCTION subprogram. The type of the function may be explicitly stated by preceding the word FUNCTION with the type, by the subsequent appearance of the function name in a type statement, or implicitly by the first letter of the function name. The FUNCTION statement has the forms:

FUNCTION name ($a_1$, $a_2$, ..., $a_n$)

REAL FUNCTION name ($a_1$, $a_2$, ..., $a_n$)

INTEGER FUNCTION name ($a_1$, $a_2$, ..., $a_n$)

where

name is the symbolic name of a single-valued function

the arguments $a_1$, $a_2$, ..., $a_n$ (there must be at least one) are non-subscripted variable names or the dummy name of a SUBROUTINE or FUNCTION subprogram.

Examples:

```
FUNCTION ARCSIN (RADIAN)
REAL FUNCTION ROOT (A,B,C)
INTEGER FUNCTION CONST (ING,SG)
```

1. The FUNCTION statement must be the first statement of a FUNCTION subprogram. At least one dummy variable must be enclosed in parentheses.
2. The name of the function must appear at least once in some definitional context. This name cannot be used in a COMMON statement.

Examples:

```
FUNCTION CALC (A,B)
.
.
.
CALC=Z+B
.
.
.
RETURN
END
```

By this method the output value of the function is returned to the calling program.

The calling program is the program in which the function is referred to or called.

The called program is the subprogram that is referred to or called by the calling program.

3. The arguments may be considered dummy variable names that are replaced at the time of execution by the actual arguments supplied in the function reference in the calling program.The actual arguments must correspond in number, order, size and type with the dummy arguments.
4. When a dummy argument is an array name, a statement with dimension information must appear in the FUNCTION subprogram; also, the corresponding actual argument must be a dimensioned array name.
5. None of the dummy arguments may appear in an EQUIVALENCE, or COMMON statement in the FUNCTION subprogram.
6. The FUNCTION subprogram must be logically terminated by a RETURN statement and physically terminated by an END statement.
7. The FUNCTION subprogram may contain any FORTRAN statements except SUBROUTINE, or another FUNCTION statement. Using other functions or calling subroutines in the FUNCTION SUBPROGRAM is permitted.
8. The actual arguments of a FUNCTION subprogram may be any of the following:

   a. A constant.

   b. A subscripted or nonsubscripted variable or an array name.

   c. The name of a FUNCTION subprogram.

9. A FUNCTION subprogram is referred to by using its name as an operand in an arithmetic expression and following it with the required actual arguments enclosed in parentheses.

10. A FUNCTION subprogram may not call itself, either directly or indirectly through some other called subprogram.

11. A FUNCTION name must be unique to 6 chracters.

The following example shows the use of a FUNCTION subprogram:

Calling Program

```
      .
      .
      .
X=Y  2+D CALC(F,G)
      .
      .
      .
```

Called Program

```
FUNCTION CALC (A,B)
      .
      .
CALC=...
      .
      .
      .
RETURN
END
```

## Adjustable Dimensions

The name of an array and the constants that are its dimensions may be passed as arguments to a subprogram. In this way a subprogram may perform calculations on arrays whose sizes are not determined until the subprogram is called. The following rules apply to the use of adjustable dimensions:

1. Variables may be used as dimensions of an array only in the array declarator of a FUNCTION or SUBROUTINE subprogram. For any such array, the array name and all the variables used as dimensions must appear as dummy arguments in at least one FUNCTION, or SUBROUTINE statement.
2. The adjustable dimensions may not be altered within the subprogram.
3. The true dimensions of an actual array must be specified in a DIMENSION statement of some calling program.
4. The calling program passes the specific dimensions to the subprogram. These specific dimensions are those that appear in the DIMENSION statement of the calling program. Variable dimension size may be passed through more than one level of subprogram. The specific dimensions passed to the subprogram as actual arguments cannot exceed the true dimensions of the indicated array.
5. Variables used as dimensions must be integers. If the variables are not implicitly typed by their initial letters, a type statement must precede the dimension statement in which they are used as adjustable dimensions.
6. If an adjustable array name or any of its adjustable dimensions appears in a dummy argument list of a FUNCTION, or SUBROUTINE statement, that array name and all its adjustable dimensions must appear in the same dummy argument list.

Example:

```
DIMENSION K(4,5),J(2,3)          SUBROUTINE SETFLG(K,J,I,L,M,N)
      .                          DIMENSION K(I,L),J(M,N)
      .                                .
      .                          DO 20 NO=1,I
CALL SETFLG (K,J,4,5,2,3)        DO 20 MO=1,L
      .                       20 K(NO,MO)=0
      .                                .
```

Supplied FUNCTION Subprogram

Usual FORTRAN compiler supplies the following basic FUNCTION mathematical subprograms. So, users may use these FUNCTION subprograms without the FUNCTION statements.

| | | |
|---|---|---|
| 1. Absolute value | x | ABS(X) |
| | k | IABS(K) |
| 2. Remaindering | x(mod y) | AMOD(X,Y) |
| | i(mod j) | MOD(I,J) |
| 3. Exponential | $e^x$ | EXP(X) |
| 4. Natural Logarithm | ln(x) | ALOG(X) |
| 5. Trigonometic Sine | sin(x) | SIN(X) |
| 6. Trigonometic Cosine | cos(x) | COS(X) |
| 7. Square Root | x | SQRT(X) |
| 8. Arctangent | $\tan^{-1}x$ | ATAN(X) |

where i, j, and k are integer type variables or constants and x is a real type variable or a constant. Only IABS and MOD are integer valued FUNCTIONS, and all other FUNCTIONS are real valued.

## 5. Specification Statements

### 5.1 DIMENSION Statement

The DIMENSION statement provides the information necessary to allocate storage for arrays in the object program, and it defines the maximum size of the arrays. An array may be declared to have from one to seven dimensions by placing it in a DIMENSION statement with the appropriate number of subscripts appended to the variable. The DIMENSION statement has the form:

DIMENSION $v_1(i_1)$, $v_2(i_2)$, $v_3(i_3)$, ..., $v_n(i_n)$

Each $v_i$ is an array declarator with each $v$ being an array name. Each $i_i$ is composed of from one to seven unsigned integer constants, integer parameters, or integer variables separated by commas. Integer variables may be a component of $i_i$ only when the DIMENSION statement appears in a subprogram.

1. The DIMENSION statement must precede the first use of the array in any executable statement.
2. A single DIMENSION statement may specify the dimensions of any number of arrays.
3. If a variable is dimensioned in a DIMENSION statement, it must not be dimensioned elsewhere.

In the following examples A, B, and C are declared to be array variables with 4, 1, and 7 dimensions respectively.

```
DIMENSION A(1, 2, 3, 4), B(10)
DIMENSION C(2, 2, 3, 3, 4, 4, 5)
```

### 5.2 COMMON Statement

A COMMON statement is of the form:

COMMON/$x_1$/$a_1$/$\cdots$/$x_n$/$a_n$

where each $a_i$ is a nonempty list of variable names, array name or array declarators (no dummy arguments are permitted) and each $x_i$ is a symbolic name or is empty. If $x_i$ is empty, the first two slashes are optional. Each $x_i$ is a block name that bears no relationship to any variable or array having the same name. COMMON assigns two elements in different subprograms or in a main program and a subprogram to the same location(s).

All variables named in a COMMON statement are assigned to storage in the sequence in which the names appear in the COMMON statement. For example if the following statement appeared in the main program:

```
COMMON A,B,C,D
```

the four variables are assigned to storage locations in the order named in a special section of storage called unnamed or blank common. Thus A is a specific storage location followed by B, etc. If in a subprogram we have the statement:

```
COMMON W,X,Y,Z
```

This means W is assigned the first location in blank common, and X the next, etc. Since the storage assigned to blank common is the same for the subprogram as the main program, A and W, B and X, C and Y, and D and Z share the same locations.

Additional blocks of storage can be established by l beling COMMON. Labeled COMMON is established by writing the label between two slashes as follows:

```
COMMON/X/A,B,C
```

L beled and blank COMMON may be included in the same statement. For example, if the following two statements were to appear in a main program and in a subprogram:

```
COMMON A,B,C/Y1/D,E/Y2/F(50),G(3,10)
COMMON H,I,J/Y1/K,L/Y2/M(50),N(3,10)
```

Blank COMMON would contain A, B, C (in that order) in the program containing the first COMMON statement and H, I, J in the program containing the second. A and H would be assigned the same location as would B and I, and C and J.

The common block labelled Y1 would establish D and E in the same locations as K and L. Y2 in the first program contains the 50 locations of F and the 30 locations of G. The same 80 locations will be assigned to M and N in the second program.

## 5.3 EQUIVALENCE Statement

The EQUIVALENCE statement is of the form:

EQUIVALENCE $(k_1),(k_2),\ldots,(k_n)$

where each k is a list of the form:

$a_1, a_2, \ldots , a_m$

Each a is either a variable name or an array element name (not a dummy argument), the subscript of which contains only integer constants or parameter symbols, and m is greater than or equal to 2. The EQUIVALENCE statement causes two or more variables, or arrays, to be assigned to the same storage location(s). EQUIVALENCE differs from common in that EQUIVALENCE assigns variables within the same program or subprogram to the same storage location; common assigns variables in different subprograms or a main program and a subprogram to the same locations.

One EQUIVALENCE statement can establish equivalence between any number of sets of variables. For example:

```
DIMENSION B(5), C(10,10), D(5,10,15)
EQUIVALENCE (A,B(1),C(5,4)), (D(1,4,3),E)
```

In this example, part of the arrays C and D are to be shared by other variables. Specifically, the variable A is to occupy the same location as the array element C(5,4), and the array B is to begin in this same location; the variable E shares location D(1,4,3) of the D array.

The following rules apply:

1. Each pair of parentheses in the statement list encloses the names of two or more variables that are to be assigned the same location during execution of the object program; any number of equivalences (sets of parentheses) may be given.

2. When using the EQUIVALENCE statement with subscripted variables, two methods may be used to specify a single element in the array.

For example, D(1,2,1) or D(p) may be used to specify the same element, where D(p) references the $p_i$ element of the array in storage.

3. Quantities or arrays that are not mentioned in an EQUIVALENCE statement will be assigned unique locations.

4. Locations can be shared only among variables, not among constants.

5. The sharing of locations requires a knowledge of which FORTRAN statements will cause a new value to be stored in a location. There are five such statements:

   a. Execution of an arithmetic statement stores a new value in the location assigned to the variable on the left side of the equal sign.

   b. Execution of a DO statement or an implied DO in an input/output list sometimes stores a new indexing value.

   c. Execution of a READ statement stores new values in the locations assigned to the variables mentioned in the input list.

   d. Execution of a CALL statement may assign new values to variables in COMMON or to arguments passed to that subprogram.

   e. An initial value may be stored in some location via a DATA statement.

6. Variables brought into a COMMON block through EQUIVALENCE statements may increase the size of the block indicated by the COMMON statements, as in the folloeing example:

```
COMMON /X/A,B,C
DIMENSION D(3)
EQUIVALENCE (B,D(1))
```

   the layout of core storage indicated by this example (extending from the lowest location of the block to the highest location of the block) is:

```
A
B,D(1)
C,D(2)
  D(3)
```

7. Since arrays must be stored in concecutive forward locations, a variable may not be made equivalent to an element of an array in such a way as to cause the array to extend below the beginning of a COMMON block.

8. Two variables in one COMMON block or in two different COMMON blocks must not be made equivalent.

9. The EQUIVALENCE statement does not make two or more elements mathematically equivalent.

10. Equivalenced variables must not appear as dummy arguments in a FUNCTION, or SUBROUTINE statement.

## 5.4 DATA Statement

A data initialization statement is of the following form:

DATA $k_1/d_1/,k_2/d_2/,\cdots,k_n/d_n/$

where each $k_i$ is a list containing names of variables, arrays, array elements and implied DOs. Each $d_i$ is a list of optionally signed constants of the form:

C or J* C

where C is a constant and J is a repeat modifier which specifies that constant C is to be used J times. J must be an integer constant.

The DATA statement enables the programmer to enter data into the program at the time of compilation. For example:

```
DATA A,B,C/14.7,62.1,1.5E-20/
```

or

```
DATA A/14.7/,B/62.1/,C/1.5E-20/
```

will initially assign the value 14.7 to A, 62.1 to B and 1.5E-20 to C.

The following is an additional example:

```
DATA P, (A(I), I=1,5), A(9)/0.0, 5 1.0, 100.5/
```

This will make P the value zero; put 1.0 in the first five elements of A; and 100.5 in A(9).

The following rules apply:

1. There must be a one-to -one correspondence between the list items and the data constants.

2. When J* appear ahead of a constant it indicates the constant is to be applied J times, i.e., it will initialize the next J items in the list with C.

3. DATA defined variables that are redefined during execution assume their new values regardless of the DATA statement.

4. Where data is to be compiled into an entire array, the name of the array (with indexing information omitted) can be placed in the list. The number of data literals must exactly equal the size of the array. For example, the statements

```
DIMENSION B(25)
DATA A,B,C/24 4.0,3.0,2.0,1.0/
```

   define the values of A, B(1),···,B(23) to be 4.0 and the values of B(24), B(25), and C to be 3.0, 2.0, and 1.0 respectively.

5. Dummy arguments and names in blank common may not appear in the list $k_i$.

### 5.5 EXTERNAL Statement

The EXTERNAL statement has the following form:

EXTERNAL $a_1$, $a_2$,···,$a_n$

where

$a_i$ is a function name whose characteristics are being qualified by this statement.

FORTRAN permits the use of a function name as an argument in a subprogram call. When this is done, the name must be included in an EXTERNAL statement in the calling program to disting ish the FUNCTION name from a variable name. The following example illustrates this use in a main calling program and a subroutine subprogram:

Main Program

```
      EXTERNAL SIN, COS
      CALL SUBR (2.0, SIN, RESULT)
      WRITE (6, 10) RESULT
   10 FORMAT ('0 SIN(2.0) = ',F10.6)
      CALL SUBR (2.0, COS, RESULT)
      WRITE (6, 20) RESULT
   20 FORMAT ('0 COS(2.0) = ', F10.6)
      STOP
      END
```

SUBROUTINE Subprogram

```
      SUBROUTINE SUBR (X,F,Y)
      Y = F(X)
      RETURN
      END
```

## 5.6 INTEGER Statement

The INTEGER statement is an explicit type statement with the following form:

INTEGER $a_1, a_2, \cdots, a_n$

where

$a_i$ is a variable, array, or FUNCTION subprogram name whose types are integer type by this statement, even if the first character is neither I, J, K, L, M nor N.

## 5.7 REAL Statement

The REAL statement is one of the explicit type statements with the following form:

REAL $a_1, a_2, \cdots, a_n$

where

$a_i$ is a variable, array, or FUNCTION subprogram name whose types are real type by this statement, even if the first character is I, J, K, L, M or N.

6. Compiler Control Statement

The END statement specifies the phisical end of the source program. It must be the last statement of every program and must be completely contained on that line. END creates no object-program instructions. It has the form:

END

## CHAPTER 4 PROGRAMMING

In this chapter, several complete sample programs and partial programs will be presented. In addition all of the necessary steps in the sequence of preparing and processing the programs will be demonstrated: analyzing, flow charting, coding, preparing input, and obtaining output.

### 4.1 Analyzing The Problem

After the problem is read, a general procedure for solution must be determined. Thought must be given to such questions as:

Can data be generated or must they be supplied ?

Is an arithmetic formula needed? Is it used once or can it be used many times by placing it within a DO loop ?

How can the program execution be terminated ?

What information will be desired in the print out of the solution ?

### 4.2 Flow charting

Since the flow chart technique so conveniently demonstrates visually the logical sequence of steps to be perfomed in a problem, the beginner is strongly advised to construct a flow chart for each problem he attempts before coding the program.

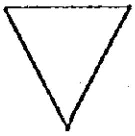
Statement numbers sometimes appear in the flow charts. This has been done to make the logic easier to follow and to ease the transition from flow charting to coding. However, when constructing flow charts of his own, he will find that statement numbers are necessary for coding but not necessary for flow charting and should find it much more convenient to omit them from the flow chart.

The beginner will find it instructive to follow a few numerical values through the step-by-step flow of the problem to assure, before coding, that the sequencing is correct and that the execution can terminate.

Although almost any geometric figures can be used to indicate the type of activity to be performed, the following set of geometric figures will be used in this textbook.

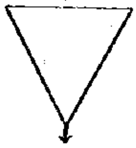
## FLOW CHART SYMBOLS

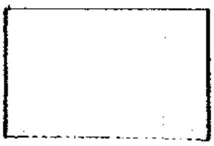
SYMBOL | DEFINITION | EXAMPLE



**Enyry Point**

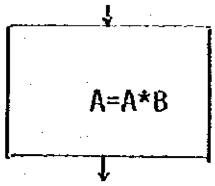
Used to indicate entry into the program.

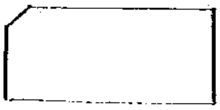




**Process Boxes**

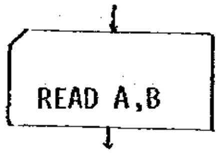
Used for assignment, computation or modification.

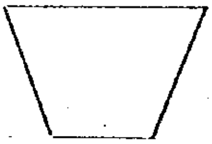


**Card Designation**

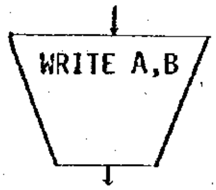
Used to indicate input.

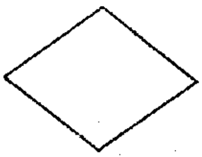


**Output Designation**

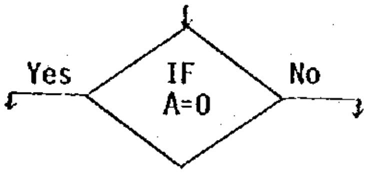
Used to indicate output.

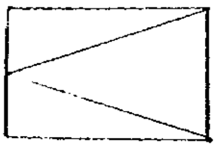


**Decision Box**

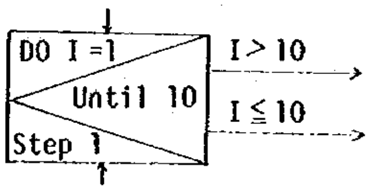
Used to indicate test points.

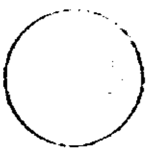


**Loop Control**

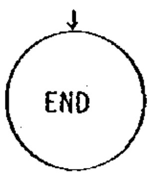
Used to indicate a looping operation.



**Exit Point**

The symbol indicates all executable statements are listed.



## 4.3 Sample Programs

### PROBLEM 1

We have data $x_1, x_2, \ldots, x_n$ ($n < 1000$). Find the average, the variance, and the standard deviation.

**Analysis**

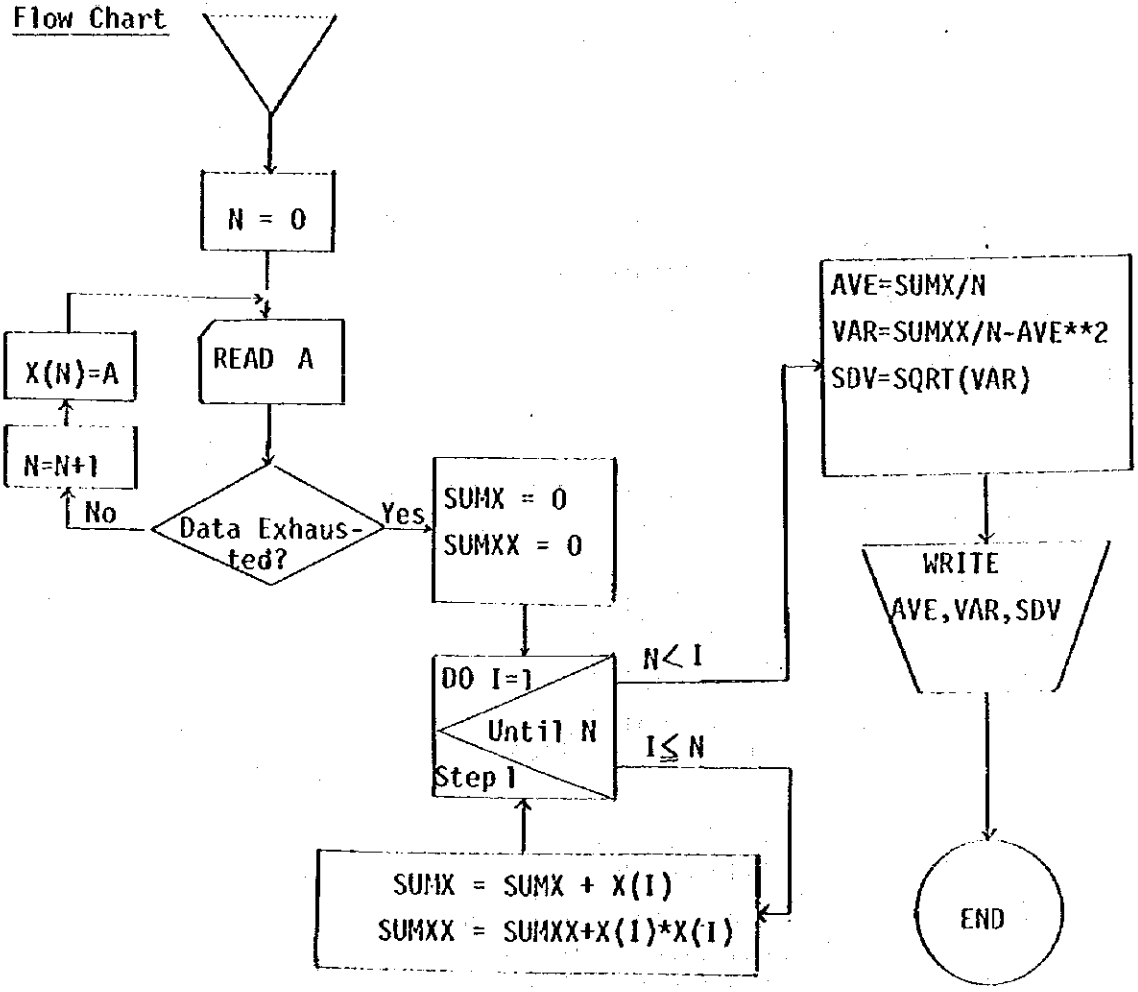
average $\bar{x} = (x_1+x_2+\ldots+x_n)/n = (\sum x_i)/n$

variance $\sigma^2 = \sum(x_i-\bar{x})^2/n = \sum x_i^2/n - \bar{x}^2$

standard deviation $\sigma = \sqrt{\sigma^2}$

**Input Data** Input data are prepared as 1 card for 1 datum with FORMAT F10.3, that is, first 7 columns of the 80 columns IBM card are integer part of the datum, and the following 3 columns are decimal part, and other 70 columns are meaningless. When all data are exhausted, the following card is punched 99999.0 in first 10 columns, which is not one of data.

**Flow Chart**



Complete Program for PROBLEM 1

```
      DIMENSION X(1000)
      N=0
    2 READ(5,10) A
   10 FORMAT(F10.3)
      IF(A.GE.99999.0) GO TO 1
      N=N+1
      X(N)=A
      GO TO 2
    1 SUMX=0.0
      SUMXX=0.0
      DO 20 I=1,N
      SUMX=SUMX+X(I)
   20 SUMXX=SUMXX+X(I)*X(I)
      AVE=SUMX/N
      VAR=SUMXX/N-AVE**2
      SDV=SQRT(VAR)
      WRITE(6,30) AVE,VAR,SDV
   30 FORMAT('1   AVE=',F10.3,'   VAR=',F10.3,'   SDV=',F10.3)
      STOP
      END
```

Another Complete Program for PROBLEM 1

```
      DIMENSION X(1000)
      DO 2 N=1,1000
      READ(5,10) A
      IF(A.GE.99999.0) GO TO 1
    2 X(N)=A
    1 N=N-1
      CALL STAT(X,N,A,B,C)
      WRITE(6,30) A,B,C
   30 FORMAT(1H1,6H  AVE=,F10.3,
     1           6H  VAR=,F10.3,
     2           6H  SDV=,F10.3)
      STOP
      END
```

```
      SUBROUTINE STAT(P,I,X,Y,Z)
      DIMENSION P(1000)
      A=0.
      B=0.
      DO 1 K=1,I
      A=A+P(K)
    1 B=B+P(K)**2
      X=X/I
      Y=Y/I-X**2
      Z=SQRT(Y)
      RETURN
      END
```

In the SUBROUTINE STAT, the second statement can be replaced by the statement DIMENSION P(I) described in Adjustable DIMENSION.

FOR SCA INTERNAL
USE ONLY

No. 3

# BASIC METHODOLOGIES FOR WORLD ECONOMY, TRADE AND TRANSIT ANALYSIS AND FORECASTING

## NO. 3 BASIC METHODOLOGIES FOR WORLD ECONOMY, TRADE AND TRANSIT ANALYSIS AND FORECASTING

### PREFACE

This No.3 curriculum text is a lecture notebook on the subject of "Basic methodologies for World Economy, Trade and Transit Analysis and Forecasting," to be prepared and distributed in advance to the participants of the Suez Canal Authority for the training program in Japan which will be held on November 6-24, 1978 at the KANSAI Training Center.

The present training textbook is prepared for the SCA's participants to be used as an introductory guidebook of II-1 ; Review Analysis and II-4 ; Information Management System, TASK II ; Systems Analysis, which was defined in the Inception Report on the Technical Cooperation Program to the Economic Planning Unit, submitted to the SCA in July 1978.

The present text is composed of three Parts : Part I : Managerial Economics, Part II : Basic Methodologies for World Economy, Trade and Transit Analysis and Forecasting and Part III : Special Lecture on Systems Analysis. In Part I basic concepts and analytical methods of managerial economics are explained in order to help the SCA's participants understand how the SCA's problems, internal and external, should be interpreted within the framework of applied micro-economics. In Part II explanations are made at a introductory level on the basic methods and concepts which are commonly used for analyzing and forecasting world economy, trade and the Canal transit volume. In Part III a brief description is given the systems analysis method to assist the SCA's participants to understand how the systems analysis method must be used to tackle the SCA management problems.

This textbook is set out in order to achieve the following objectives : 1) all of the SCA's participants will become familiarilized with basic concpts and methods of business economics ; 2) they will acquire theoretical knowledge and technical skills at an introductory level as to how world economy, trade and the Canal transit volume should be analyzed and forecast ; and 3) the necessity of a problem-solving approach will be well recognized by all of the participants.

However, it must be understood that the present text is written as a lecture notebook and that the explanations provided on each of the subject matters are as brief as possible and more detailed explanations will be given during the lecture sessions.

Each part of this text was written by the following Professors and training staff:

| | |
|---|---|
| Part I | Prof. H. Yamada (Kyoto University) |
| | Prof. T. Ihara (Kagawa University) |
| Part II | Prof. S. Myojin (Okayama University) |
| | Prof. I. Wakai (Kyoto University) |
| | Prof. M. Haruna ( ditto ) |
| | Prof. K. Yamamoto ( ditto ) |
| | Staff member of Mitsubishi Research Institute |
| Part III | Prof. K. Yoshikawa (Kyoto University) |

ACKNOWLEDGEMENTS

In preparing the present textbook we are deeply indebted to the British Petroleum Company Limited, for the use of their statistical series on world oil industry and trade. Their statistical source of information is indicated in the text.

TABLE OF CONTENTS

# PART I MANAGERIAL ECONOMICS

## PREFACE OF PART I

This textbook is intended for trainees from the Arab Republic of Egypt who are taking in their fundamental course Managerial Economics: Basic Methodologies for World economy, Trade and Transit Analysis and forecasting, Part I.

Its purpose is to give a sound understanding of the basic aspects of these subjects and some insight into approaches that can be used for making sound economic decisions in the types of problems they are likely to encounter in their careers. The subjects included in this volume are "Cost Analysis", "Demand Analysis", together with "Pricing", and "Project Evaluation". The primary emphasis throughout is on the practical applications and implications of the analytical methods and concepts. An economic way of thinking is presented which effectively permits the reader to deepen his understanding and thereby exploit the full advantage of the techniques and methods discussed.

We gratefully acknowledge extensive our debt to many excellent books which have helped us in the definition and presentation of this compact textbook. We are particularly indebted to M. H. Spencer's "Managerial Economics", J. L. Riggs's, "Economic Decision Models" and R. de. Neufuille & J.H. Stafford's, "Systems Analysis for Engineers and Managers".

# CHAPTER 1 APPROACHES TO ECONOMIC DECISION-MAKING

The basic purpose of this textbook is to give to the reader a sound introduction to the very important way of thinking called economic-decision-making.

A decision is simply a selection from two or more courses of action. Yet, decision-making can be an exciting and fascinating experience. Some choices are trivial or largely automatic, while others have far-reaching effects. Most major decisions require an economic analysis. Either on a personal basis or on behalf of an organization, we are always called upon to judge the most advantageous use for limited resources.

When considering the merits of different expansion schemes for the Suez Canal, we must take account of the following factors explicitly.

(1) Cost Analysis

Since any investment decision to enlarge the Suez Canal will inevitably incurr some costs, we must, first of all, consider the crucial cost concepts for decision-making in Chapter II, and clarify them in turn.

(2) Demand Analysis

Benefits or returns which can be expected from the development project are highly dependent upon the demand structures, which are in turn affected by the world economy, and/or the international shipping economy, etc. We therefore, consider the demand concept in Chapter III, and then the setting of tariffs in Chapter IV.

(3) Project Evaluation

Finally, we deal with the evaluation of the different expansion schemes for the Suez Canal Authority in Chapter V.

The mutual relationships among these factors are shown in Figure 1-1, where the supply side (described in Chapter II) is for convenience, separated from the demand side (developed in Chapter's III through IV) and then, both are synthesized in the final chapter, Chapter V.

Throughout his considerations of our methodological framework, the reader should bear in mind that the "controllable" variables of our decision-making are the sizes of investments and levels of tariffs for the various vessels which may use the enlarged Suez Canal.

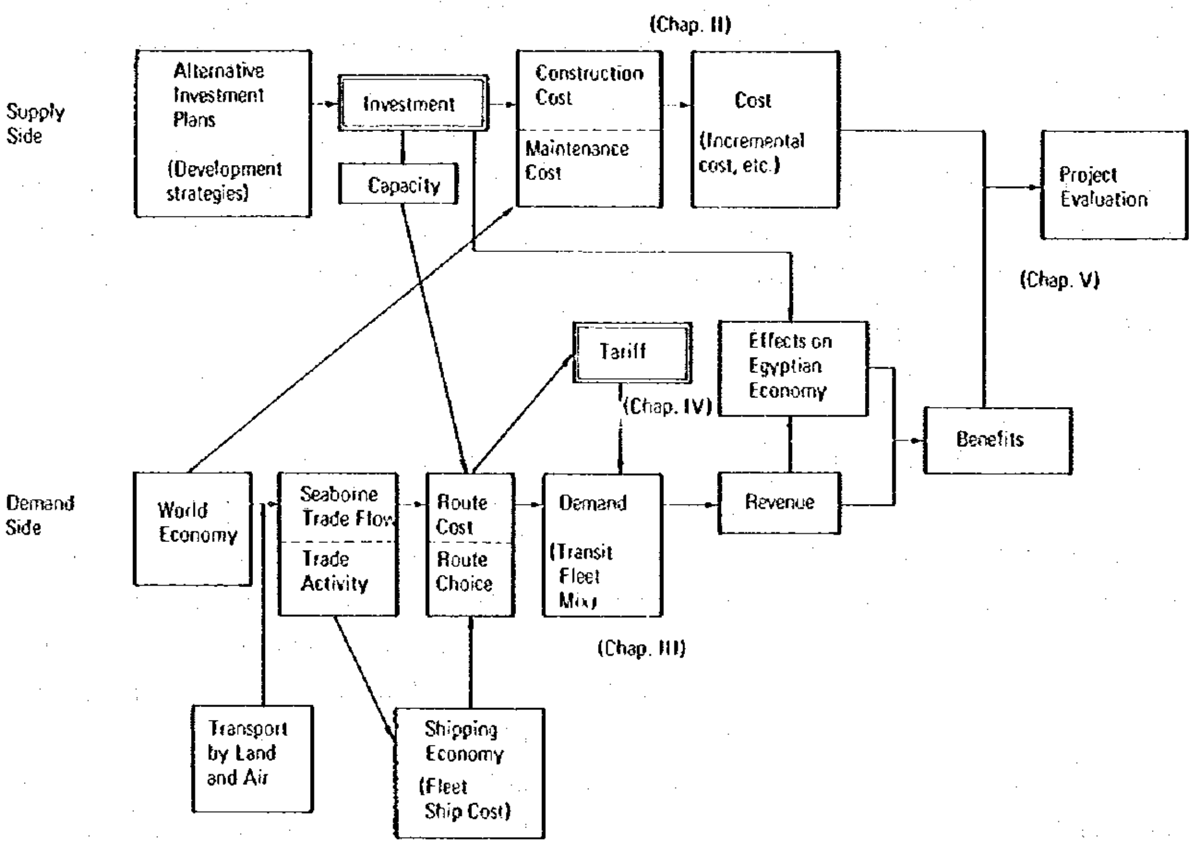



Fig. I.1 Diagrammatic Representation of Methodological Framework

# CHAPTER 2 COST ANALYSIS

Many economic decisions require us to balance cost against revenue in an optimal way. While information about demand conditions is bound to be scarce and difficult to obtain, cost information is usually plentiful. It is important that we should make the best use of it, something which is not always done.

The successful decision-maker will always ask himself which costs are relevant to the particular decision. This chapter considers how relevant costs can be identified, and introduces some of the basic cost concepts which are of interest for economic decision-making.

## 2.1 Cost Concepts for Decision-Making

A classification of important cost concepts for decision-making should dispel the notion that conventional accounting practices provide us with all our necessary cost information, and should convince as of the fact that cost concepts differ depending on managerial uses and viewpoints. In practical work, the historical costs provided by accounting are often sufficient to fulfill certain legal and financial requirements, but for economic decision-making where the concern is with predicting the costs which would be incurred in taking alternative courses of action, conventional accounting usually leaves much to be desired. As will be seen below, the most useful estimates are frequently those that are derived by combinations and adjustments in data, evidencing the fact that the accounts are a source of basic information rather than an end in themselves.

### (1) Absolute Costs and Opportunity Costs

One of the most fundamental distinctions between two general classes of ideas of costs is that between absolute (or outlay) costs and opportunity (or alternative) costs. Absolute costs involve an outlay of funds or, in fact, all reductions in assets such as wages paid, material expense, rents, interest charges, and so on. Opportunity costs, on the other hand, concern the cost of foregone opportunities, or in other words a comparison between the policy that was chosen and the policy that was rejected.

A subdivision of opportunity costs is imputed costs. These never appear in the accounting records but are nevertheless important for certain types of decisions. Interest (never paid or received) on idle land, depreciation on fully depreciated property still in use, interest on equity capital, and rent on company-owned facilities are examples of imputed costs.

(2) Direct Costs and Indirect Costs

Direct costs are costs that are readily identified and visibly traceable to a particular product, class of products, operation, process, or plant. The concept may also be extended beyond the sphere of manufacturing costs: thus, overhead may be direct as to departments, and manufacturing costs are frequently direct as to product lines, sales territories, customer classes, and the like.

Indirect costs are costs that are not readily identified nor visibly traceable to specific goods, services, operations, etc., but are nevertheless charged to the product in standard accounting practice. The importance of the distinction between direct and indirect costs from an economic standpoint is that some indirect costs, even though not traceable to a product, may nevertheless bear a functional relation to production and vary with output in some definite way. Examples of such costs are electric power, heat, light, and depreciation based on output.

(3) Fixed Costs and Variable Costs

Economists generally distinguish between two major categories of costs; fixed costs and variable costs. Fixed costs are those costs that do not vary with (are not a function of) output. They are costs that require a fixed outlay of funds each period such as rent, property taxes and similar "franchise" payments, interest on bonds, and depreciation when measured as a function of time (without any relation to output).

It should be emphasized that the term "fixed" refers to those costs that are fixed in total with respect to volume, for they may still be a function of capacity and hence vary with plant size. In other words, fixed costs are not fixed in the sense that they do not vary: they may vary and frequently do, but from causes that are independent of volume.

A term synonymous with fixed cost, at least to the economist, is overhead cost. To the cost accountant, however, the meaning of this term is virtually the same as indirect cost.

Variable costs are those costs that are a function of output in the production period. Unlike fixed costs -- the resource services of which are given off in a constant flow irrespective of the output quantity, variable costs are incurred by the stock services that are transformed or "used up" as output is produced. The sum of these two categories of costs, total fixed costs and total variable cost, at any given output level, yields the total cost at that output level, i.e., TFC + TVC = TC. When derived for successive levels of output, the resulting TC series thus represents a functional relationship between total cost and output. Examples of variable costs include materials utilized, power, direct labor, factory supplies, salesmen's commissions, and depreciation on a production basis (rather than time basis).

Since total variable costs comprise the only changing portion of total cost in the above equation, any change in the aggregate will be equal to the change in total variable cost. These changes, due to changes in output, are called marginal costs. That is, marginal cost is the change in total cost resulting from a unit change in output. In economic theory, marginal cost is of significance for decisions involving resource allocation and in product pricing. At the present time, however, it is sufficient to note that the concept of marginal cost should not be confused with the notion of incremental cost discussed below.

(4) Short-Run Costs and Long-Run Costs

The above distinction between fixed and variable costs has a close relation with another kind of cost dimension used by economists, short-run costs and long-run costs.

Short-run costs are costs that can vary with the degree of utilization of plant and other fixed factors, i.e., vary with output, but not with plant capacity. The short-run is thus a period in which fixed costs remain unchanged but variable costs can fluctuate with output; accordingly, it is a period in which a flow of output emerges from a fixed stock of resources.

Long-run costs, in contrast, are costs that can vary with the size of plant and with other facilities normally regarded as fixed in the short run. The latter is an interval of time in which plant, equipment, labor force, etc., can be expanded or contracted to meet demand requirements. It is thus a period in which the output emanates from a variable stock of resources and, therefore, a period in which there are no fixed costs, i.e., all costs are variable.

(5) Incremental Costs and Sunk Costs

When a decision has to be made involving a change in the volume of business, the difference in cost between the two policies may be considered to be the cost really incurred due to the change in business activity. This change in cost is the incremental cost (or differential cost) of a given amount of business. It represents the change in costs resulting from a change in business activity, where the latter may include any type of change such as the introduction of new machinery, development of a new product, or expansion into different markets.

Evidently, certain costs will not be altered as a result of a decision that will change business activity. These costs that are not assignable by differential method are called sunk costs, and hence are not relevant to the future effects of the decision. Since the incremental cost includes interest on investment, that is, the interest on additional capital that may be required because of the added business, sunk cost would include interest on the entire investment to the extent that it is not traceable differentially to some part of the product. It should be noted, however, that incremental costs need not be variable with output, nor traceable to a product, nor absolute costs. They may in some situations be considered to be as the same kind as opportunity costs, the incremental cost then becomes the foregone opportunity of using limited resources in their present way, as compared to their most profitable alternative activity.

(6) Avoidable Costs and Unvoidable Costs

A cost that may not only be postponed but may be avoided entirely as a result of a contraction of business activity is called an avoidable cost (or an escapable cost). It is important to note that such cost is conceived as a net figure: the decrease in cost by curtailing or terminating an activity, less any added cost incurred by other operating units as a result thereof.

An Unavoidable cost (or an inescapable cost) is a cost that must be continued in the face of a business contraction. Occasionally the avoidable-unavoidable grouping is employed in place of the more usual fixed-variable classification by some accountants and businessmen. From an economic standpoint, before the enterprise is started and resources are committed, all costs may be viewed as avoidable and all expected costs as variable.

The foregoing classification of cost concepts reveals various distinctions from both the economic and accounting standpoints. There thus cannot be found any single meaning for "cost" that would be universally applicable in all situations. At best, we can only attempt to translate the many current usages into consistent language in order to be certain that the given concept is being used for its proper purpose.

## 2.2 Interest

Nearly everyone is at least acquainted with the term interest, the money paid for the use of borrowed capital or, equivalently, money gained from the use of loaned capital. Today, the practice of charging interest is an established principle.

### 2.2.1 Interest Rate

An interest rate is the ratio of the amount gained from an investment to the amount invested for a specified period (usually 1 year). The level of the interest rate is a function of the borrower's and lender's viewpoints. For the borrower interest paid is a cost which he seeks to minimize: for the lender interest received is a gain which he hopes to maximize. The actual rate for a transaction is a compromise within the broader framework of risk, availability of capital, and investment opportunities.

#### (1) Simple Interest

When a simple interest rate is quoted, the interest earned is directly proportional to the capital involved in the loan. Expressed as a formula, the interest earned, I, is calculated by

$I = Pin$

where $P$ : present amount, or principal,

$i$ : interest rate per period,

$n$ : number of interest periods.

Since the principal, P, is a fixed value, the annual interest charged is constant. Therefore the total amount a borrower is obligated to pay a lender is

$F = P + I = P(1+in)$

where $F$ : future sum of money.

(2) Compound Interest Rate

Most economic studies are based on compound interest. With compound interest the life of a loan is divided into a number of interest periods. At the end of the first period the earned interest is calculated and added to the initial value of the loan. The sum of P+I is then considered to be the value of the loan for the second period. In this way the interest earned during each period in turn earns interest during the following periods. Eventually the present value P is multiplied by a compound-interest factor, $(1+i)^n$, to obtain the future sum F.

### 2.2.2 Equivalence and Conversion

When we evaluate the time value of money, we must clarify the concept of equivalence. Let us say that two things are equivalent when they produce the same effect. In other words, equivalent values can be determined by calculating the compound amount of each sum for each period. In this sense, say $1,000 today is equivalent to $1,791 received 10 years from now, or it is equivalent to a $135.90 annuity for 10 years from now, as long as we apply an interest rate at 6% compounded annually.

Note that there are two basic types of factors. The one converts a single amount to a present or future value. The other type is for a series of uniform values called an annuity. The six most commonly used time-value conversions are summarized in Table 2-1. The first letter of the symbol indicates the quantity being sought and the second letter is the known amount. The superscript shows the applicable interest rate per period, (i x 100)%, and the subscript is the number of interest periods (n) involved in the conversion.

Table 2.1 Conversion Descriptions and Symbols

| To Find | Given | Symbol |
|---|---|---|
| Future worth | Present amount | $(f/p)_n^i$ |
| Present worth | Future amount | $(p/f)_n^i$ |
| Future worth | Annuity of A amounts | $(f/a)_n^i$ |
| Worth of annuity A | Future amount | $(a/f)_n^i$ |
| Present worth | Annuity of A amounts | $(p/a)_n^i$ |
| Worth of annuity A | Present amount | $(a/p)_n^i$ |

A better understanding of the conversion process is achieved by developing the following interest formulas.

(1) Single-payment Future-worth Factor: $(f/p)_n^i$

The future worth of a present payment P when interest is accumulated at a specific rate (i) for a given number of periods (n), is expressed as

$$F = P(1+i)^n$$

where F : future worth at the end of n years.

The ratio of future worth to present payment is then expressed as

$$(f/p)_n^i = \frac{F}{P} = (1+i)^n$$

which is called the "single-payment future-worth factor".

(2) Single-payment Present-worth Factor: $(p/f)_n^i$

Rearranging the single-payment future-worth formula $F=P(1+i)^n$ to express P in terms of F, we have the ratio of present-worth to future value:

$$(p/f)_n^i = \frac{P}{F} = \frac{1}{(1+i)^n}$$

which is called the "single-payment present-worth factor".

(3) Sinking-fund Factor: $(a/f)_n^i$

A fund established to accumulate a given future amount through the collection of a uniform series of payments is called a sinking fund. Each payment has a constant value A and is made at the end of an interest period. Then we can derive the "sinking fund factor" as

$$(a/f)_n^i = \frac{A}{F} = \frac{i}{(1+i)^n - 1}$$

(4) Series-payment Future-worth Factor: $(f/a)_n^i$

From the above formula of the sinking-fund factor, we can readily obtain the following relation.

$$F = A \frac{(1+i)^n - 1}{i}$$

Then the time value for the future worth of the annuity is expressed as

$$(f/a)_n^i = \frac{F}{A} = \frac{(1+i)^n - 1}{i}$$

which we call the "series-payment future-worth factor".

(5) Capital-recovery Factor: $(a/p)_n^i$

The capital-recovery factor is used to determine the amount of each future annuity payment required to accumulate a given present value when the interest rate and number of payments are known. Using symbols to represent the conversions, we can obtain

$$A = P \frac{i(1+i)n}{(1+i)n - 1}$$

from which comes the expression for the "capital-recovery factor".

$$(a/p)_n^i = \frac{A}{P} = \frac{i(1+i)n}{(1+i)n - 1}$$

(6) Series-payment Present-worth Factor: $(p/a)_n^i$

Lastly, since the present value of a series of uniform end-of-period payments can be determined by

$$P = A \frac{(1+i)^n - 1}{i(1+i)^n}$$

we have the time-value expression for the present worth of an annuity,

$$(p/a)_n^i = \frac{P}{A} = \frac{(1+i)^n - 1}{i(1+i)^n}$$

which is called the "series-payment present-worth factor".

It should be noted that the following equalities always hold for our conversion symbols:

$$(f/p)_n^i = \frac{1}{(p/f)_n^i}, \quad (f/p)_n^i \times (p/a)_n^i = (f/a)_n^i$$

$$(f/a)_n^i = \frac{1}{(a/f)_n^i}, \quad (f/a)_n^i \times (a/p)_n^i = (f/p)_n^i$$

$$(p/a)_n^i = \frac{}{(a/p)_n^i}, \quad (a/f)_n^i + i = (a/p)_n^i$$

The concept of equivalence is the cornerstone for time value of money comparisons. To have a precise meaning, income and expenditures must be identified with time as well as amount.

The calculation of equivalent values and the pictorial representation of all the interest factors are summarized by using the aforementioned example: "How can $1,000 today be translated into equivalent alternative expressions of cash flow where the interest rate is say 6% compounded annually?".

The equivalent outcomes are shown as follows:

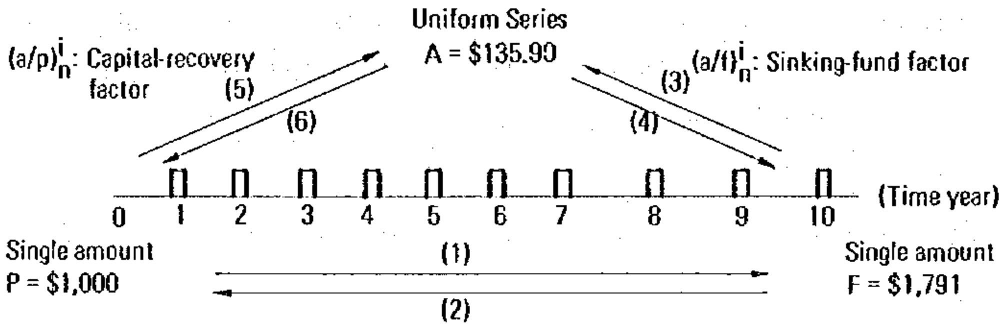



Fig. 2.1 Equivalent Outcomes with an Interest Rate of 6% Compounded Annually

Where the numerical values for the interest symbols are from Table 1 of Appendix I.

(1) $F = P(f/p)^i_n = 1{,}000(f/p)^{0.06}_{10} = 1{,}000 \times \boxed{1.7908} \doteqdot 1{,}791$

reciprocal (between 1.7908 and 0.5584)

(2) $P = F(p/f)^i_n = 1{,}791(p/f)^{0.06}_{10} = 1{,}791 \times \boxed{0.5584} \doteqdot 1{,}000$

(3) $A = F(a/f)^i_n = 1{,}791(a/f)^{0.06}_{10} = 1{,}791 \times \boxed{0.0759} \doteqdot 135.90$

reciprocal (between 0.0759 and 13.1808)

(4) $F = A(f/a)^i_n = 135.90(f/a)^{0.06}_{10} = 135.90 \times \boxed{13.1808} \doteqdot 1{,}791$

(5) $A = P(a/p)^i_n = 1{,}000(a/p)^{0.06}_{10} = 1{,}000 \times \boxed{0.1359} = 135.90$

reciprocal (between 0.1359 and 7.3601)

(6) $P = A(p/a)^i_n = 135.90(p/a)^{0.06}_{10} = 135.90 \times \boxed{7.3601} \doteqdot 1{,}000$

Hence we may give the following interpretations to channels (1), (3) and (5) in Figure 2-1.

(1) $1,000 today is equivalent to $1,791 received 10 years from now.
(2) $1,791 received 10 years from now is equivalent to $135.90 received at the end of each year for 10 years.
(5) $1,000 today is equivalent to $135.90 annuity for 10 years from now.

### 2.3 Depreciation

Depreciation is the decrease in value of physical properties with the passage of time. Although the fact that depreciation does occur is easily ascertained and recognized, the determination of its magnitude in advance is not easy. But since depreciation is a cost, we must consider it properly in our economic studies.

#### 2.3.1 Importance of Depreciation

Primarily, it is necessary to consider depreciation for two reasons:

1) To provide for the recovery of capital that has been invested in physical property.
2) To enable the cost of depreciation to be charged to the cost of producing products or services that result from the use of the property.

Depreciation differs from other costs in several respects. First, although its actual magnitude cannot be determined until the asset is retired from service, it always is paid or committed in advance. Thus when we purchase an asset, we are prepaying all the future depreciation cost. Second, throughout the life of the asset we can only estimate what the annual or periodic depreciation cost is. Consequently, we must estimate the depreciation cost in economic studies. A third difference is the fact that while much usually can be done to control the ordinary out-of-pocket costs, relatively little can be done to control depreciation cost, once an asset has been acquired. Further, many of the factors that affect depreciation costs are external to the owner of the asset. Thus the decrease in value is inexorable.

2.3.2 Causes of Depreciation

Being aware of the potential reasons that assets decrease in value helps to estimate the pattern of the decrease.

(1) Physical Depreciation

Everyday wear and tear of operation gradually lessens the physical ability of an asset to perform its intended function. A good maintenance program retards the rate of decline, but it seldom maintains the precision expected from a new machine. In addition to the normal wear, accidental physical damage can also impair functioning.

(2) Functional Depreciation

Demands made on an asset may increase beyond its capacity to produce. A central heating plant unable to meet the increased heat demands of a new building no longer serves its intended function. At the other extreme, the demand for services may cease to exist, as with a machine which produces a product no longer in demand.

(3) Technological Depreciation

Newly developed means of accomplishing a function may make the present means uneconomical. Steam locamotives lost value rapidly as railroads turned to diesel power. Current product styling, new materials, improved safety, and better quality at lower cost from new developments make old designs obsolete.

(4) Depletion

Consumption of an exhaustible natural resource to produce products or services is termed depletion. Removal of oil, timber, rock, or minerals from a site decreases the value of the holding. This decrease is compensated for by a proportionate reduction in earnings derived from the resource.

(5) Monetary Depreciation

A change in price level is a subtle but troublesome cause of depreciation. Customary accounting practices relate depreciation to the original price of an asset, not to its replacement. If prices rise during the life of an asset, a comparable replacement becomes more expensive. This means that the capital recovered will be insufficient to provide an adequate substitute for the worn-out asset.

### 2.3.3 Depreciation Methods

The great variety of ways that an asset can lose value makes the estimation of an accurate life a difficult problem. The economic life of an asset is the number of years of use that minimizes the equivalent annual cost of holding the item.

Concurrent with the identification of an asset's economic life is the choice of a method to determine the annual charge for depreciation. From the depreciation methods available, we will investigate four which account for most industrial practices:

1) The straight-line method,
2) The sum-of-digits method,
3) The declining-balance method,
4) The sinking-fund method.

The symbols used in the formulas are as follows:

- P : purchase price (or present worth at time zero) of the asset,
- S : salvage value or future value at the end of the asset's economic life,
- n : economic life in years,
- N : number of years of depreciation or use from the time of purchase,
- i : interest rate received on invested capital.

It should be noted that the salvage value is positive if the item is resold or scrapped, zero if it is discarded at no cost, or negative if removal costs exceed resale value.

#### (1) Straight-line Method

Straight-line depreciation is the simplest to apply and the most widely used of the depreciation methods. The annual depreciation is constant. The book value is the difference between the purchase price and the product of the number of years of use times the annual depreciation charge:

$$\text{Annual depreciation charge} = \frac{P - S}{n}$$

Book value at end of year N = $P - \frac{N}{n}(P-S)$

(2) Sum-of-digits Method

The sum-of-digits method provides a larger depreciation change during the early years of ownership than in the later years. The name is taken from the calculation procedure. The annual charge is the ratio of the digit representing the remaining years of life (n-N+1) to the sum of the digits for the entire life (1+2+ ... +n) multiplied by the initial price minus the salvage value (P-S). Thus the annual charge decreases each year from a maximum the first year.

Annual depreciation charge = $\frac{2(n-N+1)}{n(n+1)}(P-S)$

Book value at end of year N = $\frac{2[1+2+\ldots+(n-N)]}{n(n+1)}(P-S)+S$

(3) Declining-balance Method

The declining-balance method is another means of amortizing an asset at an accelerated rate early in its life, with corresponding lower annual charges near the end of service. An important point with this method is that the salvage value must be greater than zero. A depreciation rate is calculated from $1-(S/P)^{1/n}$ which requires a positive value for S in order to be realistic. This constant rate is applied to the book value for each depreciation period.

Annual depreciation charge = book value at (N-1) x $\left(1 - \sqrt[n]{\frac{S}{P}}\right)$

Book value at end of year N = $P\left(\frac{S}{P}\right)^{N/n}$

(4) Sinking-fund Method

The sinking-fund concept has already been introduced in the compound-interest calculations. While a firm rarely deposits sinking-fund payments in an outside business, it is a potentially useful model for an asset that loses value slowly during the first years and more rapidly during late years. The annual depreciation charge is constant if earned interest is not included. The amount of accumulated depreciation reserve is equal to the future worth of the series of uniform depreciation-charge payments at a given date. Then the book value is the difference between the purchase price and the depreciation reserve.

$$\text{Annual depreciation charge} = (P-S)(a/f)_n^i$$

$$\text{Books value at end of year } N = P-(P-S)(a/f)_n^i(f/a)_N^i$$

Each depreciation method has unique features which appeal to different management philosophies. The broad patterns of capital recovery for the above four methods are shown in Figure 2-2. The curves are based on depreciation charges without taxes or profit on the investment. The sinking-fund method has the slowest rate of capital recovery. If the interest rate used for the sinking-fund calculations is zero, $(a/f)_n^0$, the sinking-fund curve coincides with the values for the straight-line method. Both the sum-of-digits and the declining-balance methods recover a large share of the initial investment early in the depreciable life. In the first half of an asset's economic life, about three-fourths of the depreciation cost

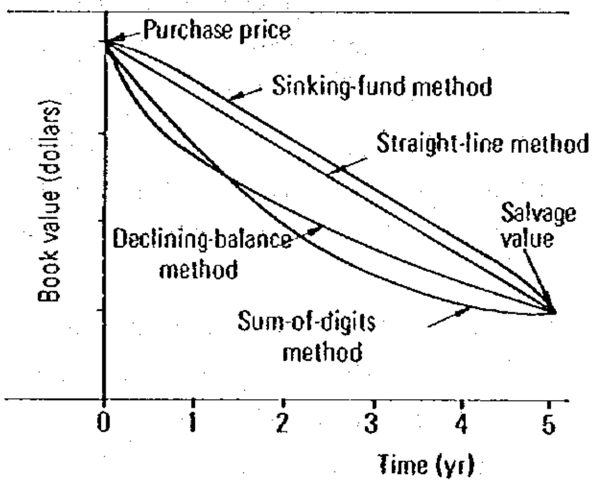



Fig. 2.2 Book Value of an Asset Calculated by Different Depreciaion Mthods

is written off by the sum-of-digits method and two-thirds is written off by the declining-blance method.

# CHAPTER 3 DEMAND ANALYSIS

## 3.1 Purposes of Demand Analysis

From a managerial viewpoint, a main objective of demand analysis is to determine and measure the forces that affect the sales of a product, and to establish relationships between sales and these controlling forces. In this way demand analysis becomes a basis for:

1) Sales forecasting and profit budgeting,
2) Controlling and manipulating demand,
3) Adjusting production to future sales.

## 3.2 Demand Concept (1) - Demand Curve

To a professional economist, the term demand with reference to market demand, has a specific meaning: it is a dependent or functional relationship revealing the quantity that will be purchased of a particular commodity at various prices, at a given time and place. In elementary economics this relationship is portrayed both arithmetically in the form of a demand schedule, which is a table showing prices and corresponding quantities, and graphically in the form of a chart, which is a pictorial representation of a demand schedule and reveals what is commonly called a demand curve.

Several features of the demand curve should be noted:

1) It is customary to represent the price level on the vertical axis and the quantity demanded on the horizontal axis.
2) The graph depicts the situation at a single point in time, say 10:10 A.M. on November 8. Hence, all but one of the prices and quantities must be hypothetical - the curve must generally answer the "conditional" question: "If the price were $4.00 per bushel how much would this (these) consumer(s) buy?"
3) The curve is generally assumed to have a negative slope. In economic terms, this is the plausible assertion that, other things being equal, more of the commodity would be demanded if the price were lower.

Table 3.1 Hypothetical Demand Schedule

| Row | Unit Price | Quantity Demanded (Millions of Units per Month) | Value (Millions of Dollars) |
|---|---|---|---|
| A | $4.00 | 50 | $200 |
| B | 3.50 | 60 | 210 |
| C | 3.00 | 70 | 210 |
| D | 2.50 | 80 | 200 |
| E | 2.00 | 90 | 180 |
| F | 1.50 | 100 | 150 |
| G | 1.00 | 110 | 110 |

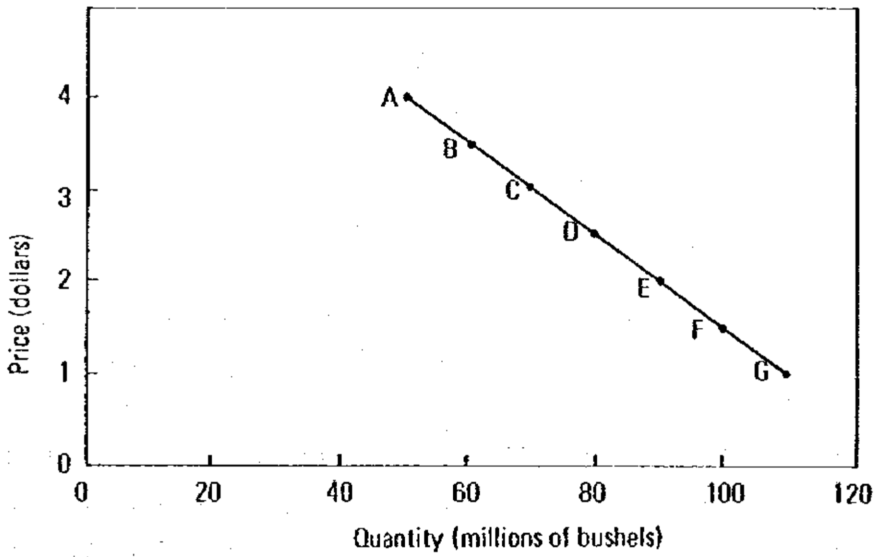



Fig. 3.1 Hypothetical Demand Curve for Wheat: Purchases Depend on Price

It should be evident that since demand is a functional concept, it may be expressed not only arithmetically in such a form as in Table 3-1 or graphically in such a form as in Figure 3-1, but also algebraically in the form of an equation such as Y=f(X), which means simply that "demand Y is a function of its explaining variable X".

### 3.3 Demand Concept (2) - Shift in Demand Curve

The second demand-curve characteristic -- its temporary nature, implies that the shape and position of the curve is likely to change with the passage of time. See Figure 3-2. At one moment DD' is the relevant demand curve, but at another instant the curve has the form EE'. Such a change is described as a shift in the demand curve. This is contrasted with a movement along a demand curve, say from point $D_0$ to $D_1$.

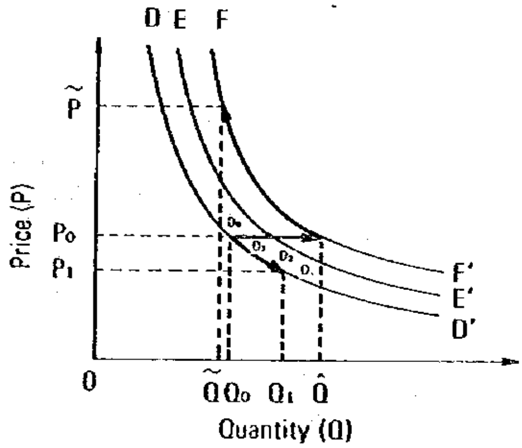



Fig. 3.2 Shift in Demand Curve

A shift in a demand curve is normally accounted for by a change in the value of some of the other variables which affect demand. For example, a rise in customer's income can lead to an upward shift in the demand curve from DD' to FF'. This means that at any given price, $OP_0$, the customer will demand more than before the shift $O\hat{Q}$. It should be noted, however, that if price happens to rise sufficiently at the same time, customer may end up buying less, $O\tilde{Q}$, despite an outward shift in the demand curve. In such a case, the shift in demand is accompanied by an offsetting movement along the curve.

Besides income, many other variables can affect the position of the demand curve. A change in the amount of advertising, a change in price or quality, or the advertising approach of a competing product - even a change in the weather - can shift the curve. Some of the relevant variables may even be intangible and unquantifiable - for example, a change in customer's tastes can cause a shift in a demand curve - although we may prefer to go behind this phenomenon and seek the variables which account for the taste change.

To summarize, demand is a function of many variables such as price, advertising, and decisions relating to competing products. The relationship which describes this entire many-variable interconnection is called the demand function. By contrast, the demand curve deals only with two of these variables, price and quantity demanded, and ignores the others. Indeed, the distinction between a movement along and a shift in a demand curve may be described in terms of the variables involved. Any change in quantity demanded which results only from a variation in price is a movement along the curve, whereas change in the value of any other variable in the demand function is likely to shift the demand curve.

Accordingly, a demand function expressed conceptually in the term of a simple relation as above may be inadequate to explain the variations in demand, and a multiple relation may therefore be necessary. The latter would be expressed as $Y=f(X_1, X_2, X_3, \ldots, Xn)$, where each of the X's denotes a specified independent variable. Thus, if we allow price, income, advertising expenditure, and the price of substitutes to stand for the independent variables, and the quantity of coffee purchased as the dependent variable (Y), the above equation would be read, "The demand for coffee (Y) is a function of, or dependent upon the price ($X_1$), income levels ($X_2$), advertising expenditure ($X_3$), price of substitutes, and certain unspecified factors."

## 3.4 Elasticity: A Measure of Responsiveness

The most obvious piece of information we desire of a demand function (or from economic relationships of other varieties) is an indication of the effect on the dependent variable of a change in the value of one of the other independent variables. In the case of the demand curve, this involves measurement of the response in quantity demanded which can be expected to result from a given change in the price of the commodity.

One measure of responsiveness is what we may call the marginal demand contribution of a price change, $\Delta Q/\Delta P$, or the corresponding derivative, $dQ/dP$, the change (fall) in quantity demanded caused by a unit change (rise) in price. This measure will be observed as the reciprocal of the slope of the demand curve $\Delta P/\Delta Q$ (or $dP/dQ$).

However, the measures of responsiveness, $\Delta Q/\Delta P$ and $dQ/dP$, are subject to a drawback which has led theorists to employ another measure which we call elasticity. The difficulty with, say, $\Delta Q/\Delta P$ is that it deals with the absolute changes in quantity and price, which makes it difficult to compare the responsiveness of different commodities. Thus, an appropriate measure of responsiveness of demand to price changes should employ percentage rather than absolute change figures.

Employing percentage terms we have the following definition for price elasticity of demand:

$$\text{Price elasticity of demand for item x} = \frac{\text{percentage change in quantity of x demanded}}{\text{percentage change in the price of x}}$$

It should be noted, however, that the value of elasticity is always defined so as to make it non-negative. However, when the demand curve is negatively inclined, a rise in price will lead to a fall in quantity so we must therefore also change the sign taking an absolute value of the right-hand side in our definition.

Hence, we have,

$$\text{Price elasticity of demand} = -\frac{100\ \Delta Q/Q}{100\ \Delta P/P} = -\frac{\Delta Q/Q}{\Delta P/P}$$

Moreover, since division by a fraction, $\Delta P/P$, is the same as multiplication by its reciprocal, $P/\Delta P$, we obtain the expression

$$\text{Price elasticity of demand} = -\frac{\Delta Q}{Q}\cdot\frac{P}{\Delta p} = -\frac{\Delta Q}{\Delta P}\cdot\frac{P}{Q}$$

When it comes to measuring elasticity, a distinction between point elasticity and arc elasticity should be made clearly. Namely we can say that point elasticity measures the elasticity at a particular point on the curve, whereas arc elasticity measures the elasticity over a range or segment of the curve. Hence, the arc price elasticity of our demand curve indicated by DD' in Figure 3-2 can be defined as follows:

$$\text{Arc price elasticity of demand} = -\frac{Q_1 - Q_0}{P_1 - P_0}\cdot\frac{P_1 + P_0}{Q_1 + Q_0}$$

Point elasticity of demand is the corresponding concept for each particular point on the demand curve. But, at any such point there is no change in price ($\Delta P=0$) or in quantity. We therefore define point elasticity in the same way as the derivative. That is, we take point elasticity to be the limit of the arc elasticity figure as the arc $D_0D_1$ is made smaller and smaller, first being cut down to $D_0D_2$, then to $D_0D_3$, and so on. We thereby arrive at the definition.

$$\text{Point price elasticity of demand} = -\frac{dQ}{dP}\cdot\frac{P}{Q},$$

Where the derivative $dQ/dP$ has been substituted for $\Delta Q/\Delta P$ in the above definition of arc price elasticity of demand.

When the value for price elasticity of demand is less than one, we say we have inelastic demand over that range of prices. When we obtain a value equal to one, we say we have unitary elasticity of demand over that price range, and when we obtain a value greater than one, we say we have elastic demand over that range of prices.

If we consider the function $Y=f(X)$, which may represent a demand function, a cost function, or any functional relationship in economics, we might say in a more general way that elasticity is the percentage change in the dependent variable, Y, resulting from a one per cent change in the independent variable, X. In other words, we might say that the elasticity, $\xi$, of the function $Y=f(X)$ at the point X is the rate of percentage change in the dependent variable, Y, relative to a small percentage change in the independent variable, X. Thus, we arrive at the following general definition of elasticity.

$$\xi = \left(\frac{\Delta Y}{Y}\right)\Big/\left(\frac{\Delta X}{X}\right) = \frac{X\Delta Y}{Y\Delta X}$$

## 3.5 Elasticity and Revenue

We now take a step further and consider the relationship between elasticity and revenue.

Let R represent revenue, then we have:

$$R = PQ$$

Where P stands for price, and Q stands for quantity demanded.

Needless to say Q is a function of P written,

$Q = f(P)$

which we have already called the demand function. Now, taking the derivative of the revenue R with respect to the price variable P we obtain.

$$\frac{dR}{dP} = Q + P\frac{dQ}{dP} = Q(1 + \frac{P}{Q}\frac{dQ}{dP}) = Q(1-\varepsilon)$$

From this, we can eventually obtain the following relations:

$$\frac{dR}{dP} \gtreqless 0 \text{ according to } \varepsilon \lesseqgtr 1 \quad \left(\text{of } \begin{array}{l}\text{inelastic demand}\\ \text{unitary elasticity}\\ \text{elastic demand}\end{array}\right)$$

From this relationship, we can readily observe the following properties of the concept of elasticity:

1) Given any segment of the demand curve, a change in price within that segment will have no effect on the product PQ if and only if the elasticity of demand throughout the range is exactly equal to unity. More specifically, a change in price from $P_0$ to $P_1$ will yield $P_0Q_0 = P_1Q_1$ if and only if the elasticity of the arc $D_0D_1$ is unity, and each and every intermediate price change will also leave PQ unaffected if and only if the point elasticity is unity at every point along this arc.

2) If a demand curve has elasticity less than unity (it is inelastic), a rise in price will increase revenue, PQ, and *vice versa*. If the curve has an elasticity greater than unity (it is elastic), a fall in price will increase revenue and *vice versa*.

The total revenue curve and its associated demand curve are shown in Figure 3-3, which is based on the data of our hypothetical demand schedule given in Table 3-1.

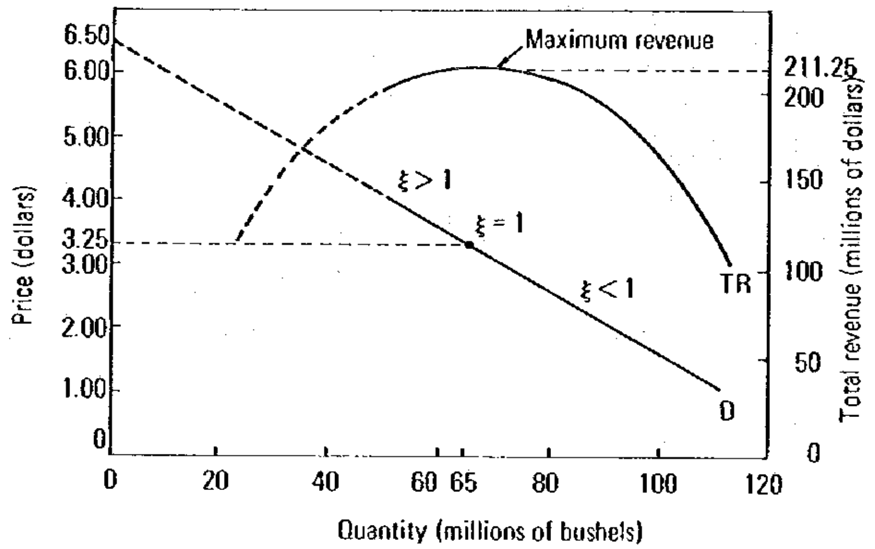



Fig. 3.3 Total Revenue is at a Maximum – Where Elasticity of Demand is Unitary

3.6 Quantity and Income

When we consider longer time periods in demand analysis, price should be regarded as only one of many factors which affect the demand under study. Particularly income becomes a factor of considerable importance in explaining demand over longer periods. In such a case, we conventionally treat income as a factor which shifts the demand curve. For most products, it is expected that increases in consumer income will result in a shift of the price-quantity relationship upwards and to the right. Table 3-2 gives data on quantity, price, and income for a hypothetical product over five years, from which Figures 3-4 and 3-5 are drawn, respectively:

Table 3.2 Price, Income, Quantity Schedule

| Year | Price (Dollars) | Income (Dollars per capita) | Quantity (Thousands of units) |
|---|---|---|---|
| 1 | $1.00 | $2,000 | 1,000 |
| 2 | 1.00 | 2,500 | 1,250 |
| 3 | 1.00 | 3,000 | 1,500 |
| 4 | 1.00 | 3,500 | 1,750 |
| 5 | 1.00 | 4,000 | 2,000 |

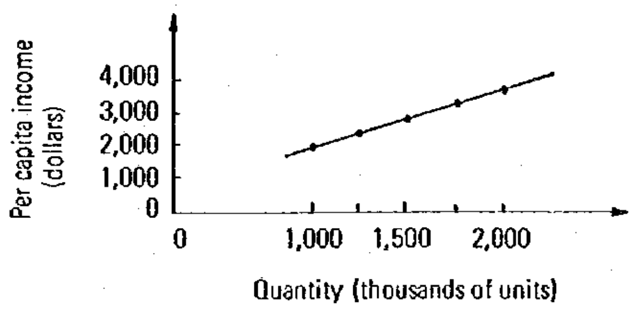



Fig. 3.4 Relationship between Quantity and Income -- Price Remaining Constant

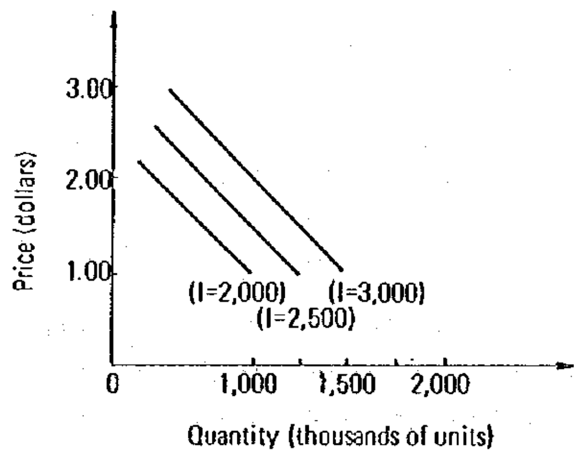



Fig. 3.5 Increase in Income -- Shifts Demand Upward

# CHAPTER 4 PRICING: DETERMINATION OF TARIFFS

## 4.1 Pricing Objectives

Since pricing is not an end in itself, but a means to an end, an explicit formulation of a company's pricing objectives is essential. The fundamental guides to pricing are the company's overall goals.

The most typical pricing objectives conventionary employed are as follows;

1) Pricing to achieve a target return on investment,
2) Pricing to stabilize prices and outputs,
3) Pricing to realize a target market share,
4) Pricing to meet or match competition.

In most companies, one of these goals predominates, but as the listing of collateral objectives indicates, price-making by any one company is not always ruled by a single policy objective. What would you think are the pricing objectives for the Suez Canal Authority?

## 4.2 Pricing Theory

### 4.2.1 Monopoly Pricing

A market with a single seller is called a monopoly in economics. Hence, the term monopoly defines a situation where a single firm produces a commodity for which there are no close substitutes. In such a market, consider how the single most profitable price is determined by reference to Figure 4-1.

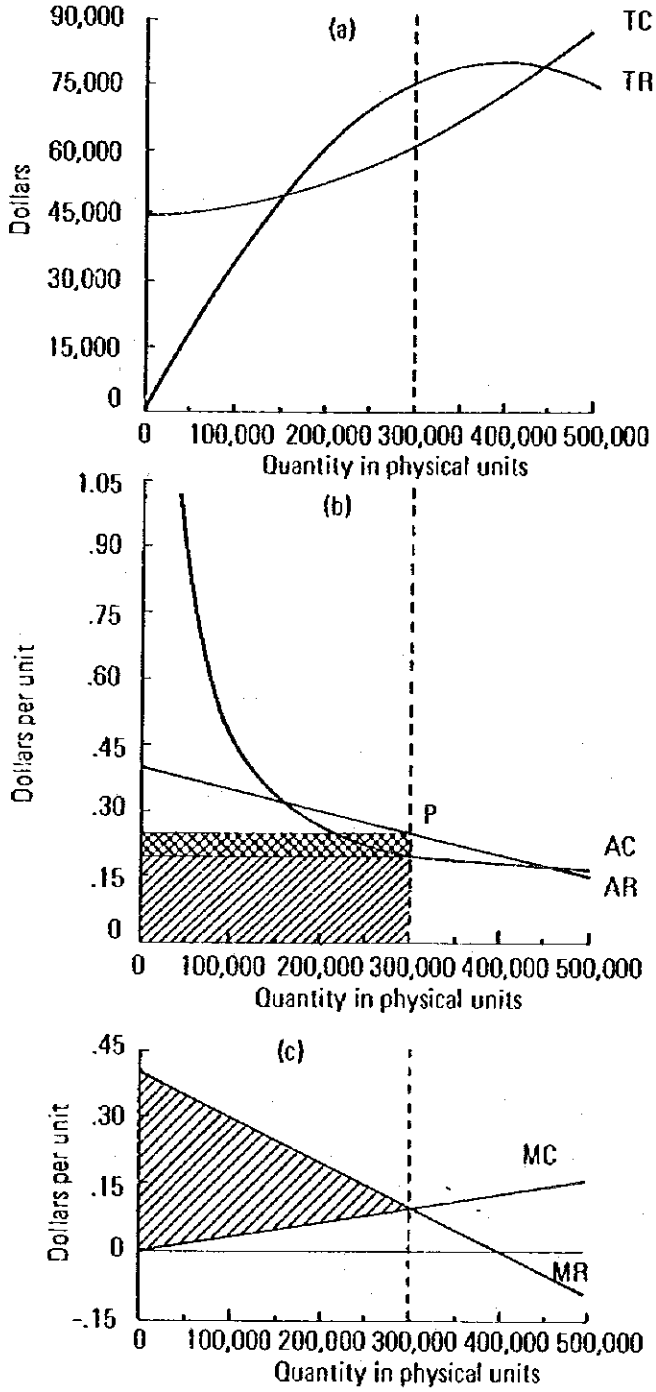



Fig. 4.1 Theory of Monopoly Price

In graph (a) of Figure 4-1, TR shows the seller's total revenue per month (or another time-unit) for various output rates, and TC shows the total cost of producing the indicated output. The point of maximum total profit is at an output of 300,000 units per month, where the spread between the total revenue and the total cost curve is greatest ($15,000).

In graph (b), AR shows the same demand curve in the form of average revenue per unit (price) associated with a series of output rates. AC is the corresponding average unit cost curve. The price that will maximize profits (25¢) can be determined from this diagram by finding the unit profit which, when multiplied by output, gives the largest total profit, as indicated by the crosshatched area of the rectangle in the diagram ($15,000).

Graph (c) shows the same demand and cost behavior in marginal form. MR depicts the additional revenue produced by the sale of an additional unit at each output rate. Similarly, MC shows, for the relevant range of output, the additional cost of making an additional unit. The point of maximum profit is the output rate at which marginal cost and marginal revenue are equal (10¢). The triangular area between the two marginal curves equals contribution profits, the excess of total revenue over incremental costs (in this case, $60,000). Deducting the fixed cost ($45,000) gives net profits ($15,000).

The main contributions to practical pricing of this kind of theoretical analysis are to point out the kinds of demand and cost relations that need to be guessed, and to show how these functional relations should be used to indicate the most profitable price.

This solution is logical enough so far as it goes, but it does not go very far. It should not be taken very literally in practice as a description or a prescription. A major qualification of its usefulness is the restraints on profit maximization as a pricing objective. These restraints are partly political. On even purely economic grounds, however, these methods are an oversimplification and difficult to apply. For the pure monopoly situation assumed here, there are formidable econometric problems of estimating the demand and cost curves. For more provisional monopolies, the analysis cannot take account of potential competition and the reactions of substitute competitors, or of the effects on future prices or patronage, all of which are vital factors in most monopoly pricing situations.

4.2.2 Differential Pricing

The term differential pricing means a method that can be used by some sellers to tailor their prices to the specific purchasing situations or circumstances of the buyer.

Specifically, differential pricing may be defined as the practice by a seller of charging different prices to the same or to different buyers for the same good, without corresponding differences in cost.

Three practical conditions are necessary if a seller is to practice price discrimination effectively:

1) Multiple demand elasticities,
2) Market segmentation,
3) Market sealing.

First of all, there must be differences in demand elasticity among buyers due to differences in income, location, available alternatives, tastes, or other factors. If the underlying conditions that normally determine demand elasticity are the same for all purchasers, the separate demand elasticities for each buyer or group of buyers will be approximately equal and a single rather than multiple price structure may be warranted.

Secondly, the seller must be able to partition (segment) the total market by segregating buyers into groups or submarkets according to elasticity. Profits can then be enhanced by charging a different price in each submarket.

Lastly, the seller must be able to prevent - or natural circumstances must exist which will prevent - any significant resale of goods from the lower - to the higher - priced submarket. Any leakage in the form of resale by buyers between submarkets will, beyond minimum critical levels, tend to neutralize the effect of differential prices and narrow the effective price structure to where it approaches that of a single price to all buyers.

(1) Perfect Price Discrimination

It is instructive to consider the extreme case of perfect price discrimination in order to understand the motivation behind more practicable schemes of discrimination. Whereas the usual assumption employed in the economic theory of monopoly is that the single most profitable price under existing conditions is charged, the notion of perfect price discrimination requires that every unit is to be sold at a different price and that this is to be the highest price at which that unit will be purchased.

The usual geometrical demonstration of perfect price discrimination is given in Figure 4-2, which contrasts it with one-price monopoly pricing. It should be noted that the demand curve becomes the marginal revenue curve under perfect discrimination. This follows from the assumption that each unit sold is independent of the others; that is, to sell an additional unit, the monopolist does not have to set a lower price on the earlier units. Whatever price he gets for a particular unit adds exactly that amount to his revenue. The average revenue curve lies above the marginal revenue curve, since whatever quantity is considered on the horizontal axis, the earlier units in this quantity will bring a higher price than the later ones and hence will hold the average price above the price at the margin. Maximum profit is secured by selling OA units, where marginal cost equals marginal revenue. Further sales would not be profitable because more would be added to cost than to revenue. Total revenue would amount to OA times AK.

Total revenue under perfect discrimination is also equal to $OP_1MQ_1$, the revenue under a single-price monopoly, plus the shaded areas. The vertically shaded area to the left of M represents the additional amounts paid for units that would have been sold at price M under a one-price system. The shaded area to the right of M is revenue from units that could not be sold profitably under a one-price system since the extra revenue gained from their sale would be more than offset by the loss in revenue due to lower prices on all sales plus the additions to costs.

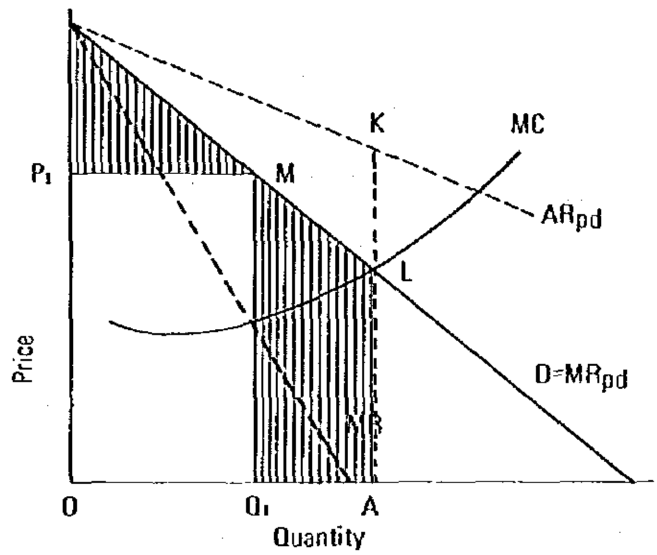


Fig. 4.2 Perfect Price Discrimination Contrasted with Single Monopoly Price

(2) Discrimination by Customer Classification

Instead of attempting to charge the maximum that the demand will bear for each unit, a firm with some monopoly power may divide customers up by some identifiable characteristic. It is then instructive to concentrate on a model for the two-price case. For specific examples, one may think of adults and children at a movie house, or unrelated individuals and family members on an airline that is using a family plan.

In Figure 4-3 we show the simplest case in which marginal costs remain the same over the relevant output. In (a) we show the conventional monopoly or monopolistic competition price (P) and output (Q) determined by the intersection of MR and MC. The demand curves in (b) and (c) added horizontally at each price equal the total demand shown in Graph (a). Graph (b) depicts the less elastic demand. Graph (c) shows the more elastic demand. Note also that under discrimination the total quantity sold $OQ_1$ plus $OQ_2$ is greater than that under the one price system, OQ. Fuller use of capacity is frequently one result of price discrimination.

The profitability of price discrimination is dependent upon the differing elasticities of demand. If the elasticity had been identical in both markets, MR would have equaled MC at price P in the two markets. Only one price would have been charged.

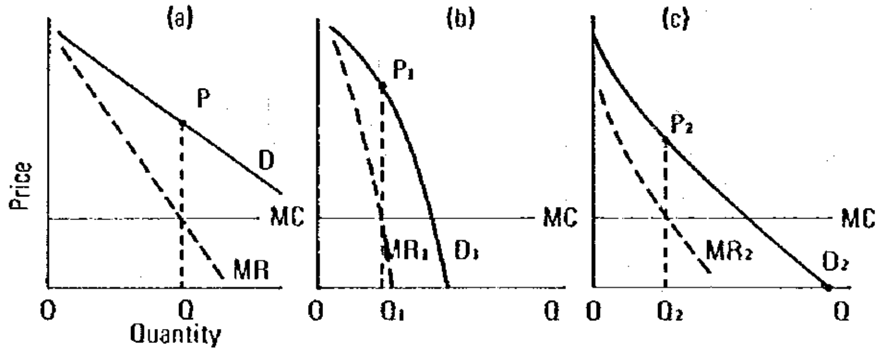



Fig. 4.3 Price Discrimination in Two Separate Markets

## 4.3 Pricing Policy

Pricing policy can be established with many purposes in mind. Tariffs for example, do not only provide revenues for operating, maintaining, and financing a facility, but also aid the rational allocation of capital among competing modes. They are regarded as a possible means of redistributing income and/or promoting the growth of particular regions and industries.

Economists usually start with one basic precept concerning prices which allows that everyone may consume any service whose incremental cost of production he is willing to pay. This incremental cost pricing, however, is almost immediately modified in actual practice by another conflicting principle according to which to the greatest extent possible, the total cost of producing a specific service should be paid by those actually consuming it.

Still another important dictum is that only one price should usually be charged for any single homogeneous service, since such a market is cleared by a price that equates demand and supply, and this price provides a simple, unmistakable indicator for the evaluation of investments under study.

In the idealized conditions of economic theory-such as perfect competition, specific income distribution, and no scale economy - we can state confidently that charging each consumer according to the incremental cost of providing the service he consumes will lead to an optimal use of resources.

The real world, of course, does not always reflect such an idealized world of economic theory. For example, market imperfections do exist, and much can be made of this fact by those who wish to criticize the implications of incremental cost pricing and its related economic theory.

In general, because of problems of cost estimation, imperfect markets, and multiple and conflicting objectives, pricing policy seems to have differed quite widely. One policy arises from experience and pragmatic adaptation to observed circumstances (Recall the policy to share the benefits equally between the Suez Canal Authority and the International Shipping Community). The other policy represents concepts derived from theoretical analysis. In short, the most fundamental difference between alternative pricing policies should be ascribable to the degree to which a particular price is demand as opposed to cost oriented, although the distinction is not always clear-cut.

#### 4.3.1 Full-Cost Pricing

The most widely used method of cost-oriented pricing is known as full-cost pricing (or cost-plus pricing). It is procedure whereby the price is determined by adding a fixed mark-up of some kind to the cost of the goods. Thus, if a manufacturer desires a 20 per cent mark-up, his pricing by this method would price at $12.00 a good whose cost was $10.00. Evidently, two decision problems confront the manager who uses full-cost pricing: (1) arriving at an estimate of cost, and (2) selecting the appropriate margin or mark-up. How are these done by most firms?

In practice, most manufacturers using full-cost pricing usually employ some notion of standard cost as their basic cost figure. They arrive at the figure by estimating unit costs of labor and materials and by computing unit overhead costs for operations at some arbitrary percentage of capacity. In other words, they typically calculate their costs for a "standard output", commonly between two-thirds and four-fifths of capacity, irrespective of the actual volume of operations. Other cost measures sometimes used, however, are actual cost, or the cost for the most recent accounting period, and expected cost, which is a forecast of actual cost for the future pricing period based on a forecast of operating rates for that period. In any case, regardless of the method employed to estimate costs, the over-all nature of the full-cost pricing formula is essentially the same.

How is the ratio of the mark-up determined? Numerous surveys of pricing practices have been made by economists, and hundreds of pricing methods have been reported. On the basis of all this information, the evidence continues to indicate no definite answer to this question, other than the feeling on the part of businessmen that their margins represent what they believe to be a "fair" reasonable profit. The evidence also indicates that there are wide variations in the percentage of mark-up both within industries and between industries, due to differences in pricing objectives, competition, cost structures, accounting methods, inventory turnover, and industry custom.

### 4.3.2 Advantages of Full-Cost Pricing

What are the reasons given by businessmen for the wide prevalence of full-cost pricing? Some of the chief ones are these:

1) It offers a relatively simple and expedient method of setting price by the mechanical application of a formula.
2) It provides a method for obtaining adequate ("fair") profits when demand is unknown.
3) It is a method of establishing a stable price uninfluenced by fluctuations in demand, which is particularly important to firms that commit themselves on price through their catalogs, advertising, etc.
4) It is desirable for public relations purposes even at the expense of short-run profits, presumably because customers will accept price increases when costs rise.

### 4.3.3 Disadvantages of Full-Cost Pricing

Despite its prevalence, full-cost pricing has at least three important disadvantages to firms employing it as a pricing method:

1) It fails to take account of demand as measured in terms of buyers' desire and purchasing power. Moreover, where price planning for the future is involved, what is needed is a forecast of both future costs and future demand if the best pricing decision is to be made.
2) It attempts to make an accurate measure of what usually amounts to the wrong cost concept, rather than even an approximate measure of the right cost concept. What is frequently needed, for example, is at least a rough estimate of opportunity costs and of incremental costs, neither of which are readily available from accounting records, rather than accurate estimates of irrelevant concepts such as past or present costs.
3) It fails to reflect competition in terms of rivals' reactions and the possible entry of new firms. For example, in an industry that prices by the full-cost method, if company margins are above the level necessary to cover operating costs and yield "normal profits" per unit at capacity, new firms will tend to enter the industry as long as no considerable excess capacity is already present. The result will be a smaller market share for each firm, and therefore, higher unit overhead costs and lower profits per firm.

# CHAPTER 5 PROJECT EVALUATION

This chapter defines the methods needed to evaluate and compare individual projects. A problem exists because the flows of costs and benefits that would accrue at different times to each project are not directly commensurable, even when they are all measured in the same monetary units. In effect, money spent or gained now has more value than that paid or received later. Capital has a time value whose implications must be carefully traced out in each project evaluation.

As a first step we present the formulas which enable us to compare the different, and often irregular series of benefits and costs that are associated with alternative projects. Second we present the methods for determining the appropriate time value of money to ascribe to capital at any time. With this background we are finally able to explore the meaning of the several evaluation criteria which have been proposed and are in use.

## 5.1 Comparison of Time Streams of Benefits and Costs

For the purpose of presenting the formulas for making different time streams of costs and benefits comparable, a few definitions must be clarified. The opportunity cost of money over time is the cost, at the margin, of not having an extra amount available. The interest rate of money is the rate of return that is currently paid for money in the money markets. The interest rate may sometimes be equal to the opportunity cost for a decision - making group. But, particularly if this group is faced with several alternatives, the interest rate is generally not the opportunity cost of capital. The discount rate is the time value that is imputed to money for the purposes of comparison. If we evaluate some projects, the discount rate should equal its opportunity cost for capital.

### (1) Discounting Formulas

Comparable measures of time streams of money are obtained through the application of one of four basic formulas. They apply for all rates of opportunity cost and of interest, and thus can be used both for project evaluation and for the calculation of financing charges.

The rotation for the discounting formulas can be explained with reference to Figure 5-1. The discounting rate is r % per period over n periods, which constitute the life of the project. The present value of future revenues is P. The future value is F, which represents the sum of money that would be obtained by investing at r % for n periods.

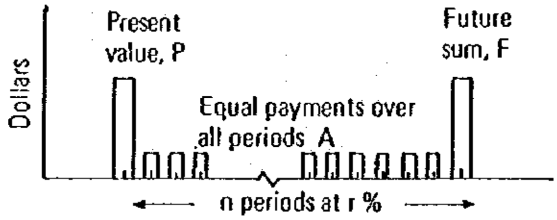


Fig. 5.1 Notation for Discounting Formulas

The four basic discounting formulas are summarized in Table 5-1.

Table 5.1 Four Basic Discounting Formulas

| Situation | Formula | Factor |
|---|---|---|
| Future value F of current sum P | $F = P(1+r)^n$ | Future-worth factor: $(f/p)^r_n = (1+r)^n$ |
| Present value P of future sum F | $P = F\frac{1}{(1+r)^n}$ | Present-worth factor: $(p/f)^r_n = \frac{1}{(1+r)^n}$ |
| Future value F of series payments A | $F = A\frac{(1+r)^n-1}{r}$ | Series future worth factor: $(f/a)^r_n = \frac{(1+r)^n-1}{r}$ |
| Present value P of series payments A | $P = A\frac{(1+r)^n-1}{r(1+r)^n}$ | Capital-recovery factor: $(a/p)^r_n = \frac{1}{(p/a)^r_n} = \frac{r(1+r)^n}{(1+r)^n-1}$ |

Each of the essential factors are conveniently tablated for different values of r and n in most standard references. These tables are recommended for use in discounting calculations.

(2) Present Values and Annual Costs

In practice, comparisons between time streams of benefits and costs are expressed in terms of present values or of equivalent annual costs. The present value method refers all revenues to a common time, usually the present. The equivalent annual cost method calculates the series of equal payments that would have the same present value as the time stream of costs and benefits that would actually occur if the project were implemented. For the same discount rate and project life, both methods are numerically equivalent.

The choice between the present value and the equivalent annual cost method of discounting depends principally upon convenience. For simple problems, where a constant stream of disbursements and revenues may occur, the annual cost method is said to be preferable for presentation to decision-makers. For the analysis of complex systems, however, the present value method is to be preferred. A principal advantage of this method is that it focuses attention on the quantity which is the appropriate fundamental quantity that should be maximized.

(3) Implication of Different Discount Rates and Project Life

The formulas for developing comparisons between different time streams of benefits and costs are functions only of the discount rate r and of the project life n. As the estimation of these parameters is largely a matter of judgement, it is useful to explore the implications of alternative choices of the quantities. It turns out that comparisons of projects are highly sensitive to these parameters, and thus that particular care should be given to their selection.

The present worth of future benefits or costs drops off rapidly for larger discount rates and more distant returns, as indicated in Figure 5-2. As a rule of thumb, the rate can be estimated by the so-called "72 rule", which says that the quantity $(1+r)^n$ doubles about every $n^*$ years, where $n^*=72/r$. Thus, as shown in Figure 5-2, the present worth is halved in about 15 years when the discount rate is 5%, and in about 7 years when it is 10%.

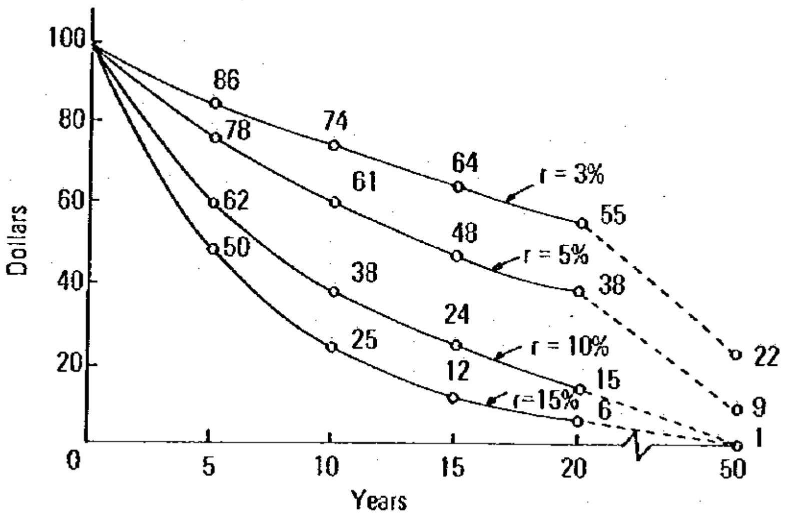



Fig. 5.2 Present Worth of $100, n Years Hence, for Discount Rates of 3, 5, 10, and 15% per Year

Benefits that occur beyond 20 years in the future have an increasingly negligible contribution to the present value of a project. The overall desirability of any project is also highly sensitive to the choice of discount rate and project life. Hence, the choice of the discount rate and of the life of the project are clearly major determinants in large-scale systems because expenses and benefits typically occur at quite different times. Large initial construction expenses are generally followed by a long series of fairly low benefits. Comparisons between these amounts can be markedly affected by the choice of r and n. A higher discount rate drastically reduces the present value of the benefits, cuts down the number of projects that can be justified, and results in significantly less construction than would a lower rate.

## 5.2 Choice of Discount Rate

The theoretically correct basis for the discount rate to be applied to project evaluation is the opportunity cost of capital. As has been shown, the opportunity costs of money are the returns lost by diverting funds to the project under consideration. This is not an easy quantity to determine. Because of lack of competition in the development of many public systems and because of dissimilar rates of taxation on different kinds of activity, the discount rate is not the same for all situations. It depends on what the time opportunity costs are, and on where project resources come from. The purpose of this section is thus to indicate the basic guidelines on how the time value of money can be determined and to briefly describe current practice.

(1) Lower Bound: Risk-free Interest Rates

The first guide to the determination of the appropriate discount rate for project evaluation is simple. The opportunity cost of capital for all investments should be at least as great as the risk-free constant dollar interest rates currently being paid. That is, no investment is worth undertaking unless it is just as productive as any of the comparable investments which are commonly available, such as long-term government bonds.

(2) Social Discount Rate

The social discount rate is the discount rate which should apply to public projects. The determination of its value is of central interest because so many large-scale systems are public projects. The capital required for government investments is drawn from the private sector where it would otherwise be spent on consumer goods or invested by the private sector itself. To determine the opportunity cost of this capital, one needs first to examine the rates of return that are appropriate for these alternative uses. Second, it is necessary to weigh the proportion of capital obtained from both sources to obtain a composite opportunity cost.

(3) Risk Premium

A risk premium is necessary to compensate, in an expected-value sense, for probable failures of projects. The projected benefits of public investments are almost invariably optimistic. They assume that the system will develop and succeed as planned. In reality, the expected value of a project may be lower than that anticipated. Such risky projects should be required to show a higher rate of return i.e., a risk premium, in order to make their expected return equal to the return on risk-free investments. Many economists agree that such an adjustment for risk should be made.

### (4) Current Practice

Discount rates of 10% or more on government projects are commonly used for the evaluation of projects throughout the world. For example, the World Bank uses about 10% in feasibility studies of highways and port facilities; the Ministry of Transport in England uses 10 to 12%; the National Road Board of Sweden uses 9%; and the Ministry of Public Works in Mexico uses 12%. The justification given for such rates is that they reflect the real opportunity costs of the capital over time.

## 5.3 Evaluation Criteria

The essential issue to be resolved in the evaluation of individual projects is: which project is the most productive? Which gives the highest returns? There are essentially two indices of economic merit which are used to resolve this question. The first is the benefit-cost criterion and the second is the internal rate of return. Each of these is discussed in turn. As will be seen, these criteria should be supplemented by a third, that of net present value, whenever it is required to choose a group of projects for investment.

### (1) Benefit-Cost Criterion

The benefit-cost criterion of project evaluation ranks projects by the ratio, on a present value basis, of their total benefits to costs. That is,

$$\text{Benefit-cost ratio} = \frac{\text{Present value of all benefits}}{\text{Present value of all costs}}$$

Although the benefit-cost criterion provides an attractive measure of merit for simple projects, it is subject to a major theoretical objection for the analysis of complex systems. The benefit-cost ratio is only concerned with total benefits and costs and is insensitive to their distribution. It assumes that a dollar of benefit is equally as useful to the nation if it is given to a millionaire as if to a tenant farmer. According to the benefit-cost formula, a project is desirable so long as total benefits exceed costs even if all the benefits go to the millionaire and are paid by the farmers. This should shock our sense of equity.

The benefit-cost ratio is therefore, like any single-valued objective function, altogether too simple to be adequate for the evaluation of large-scale projects. Systems with many objectives should have several measures of effectiveness. We should therefore not limit ourselves to a single economic criterion when evaluating projects with many objectives.

The benefit-cost ratio usually underrates the productivity of a project with high annual costs. The extent of the distortion introduced by the benefit-cost criterion for projects with high annual costs can be presented formally. First we define K as the present value of an initial capital investment, A as that of the annual costs, and B as that of the annual benefits. The usual benefit-cost ratio is then

$$\text{Benefit-cost ratio} = \frac{B}{A + K}$$

The ratio of the sum of the annual net benefits to total costs is then

$$\text{Net benefit-cost ratio} = \frac{B - A}{K}$$

To indicate the extent of the distortion introduced by the benefit-cost ratio, we take the ratio of the former to the latter to obtain

$$\frac{\text{Total benefit-cost ratio}}{\text{Net benefit-cost ratio}} = \frac{1}{(1+A/K)(1-A/B)}$$

This formula is plotted in Figure 5-3. As suggested by this discussion the benefit-cost ratio often cannot measure real productivity. A different criterion is required.

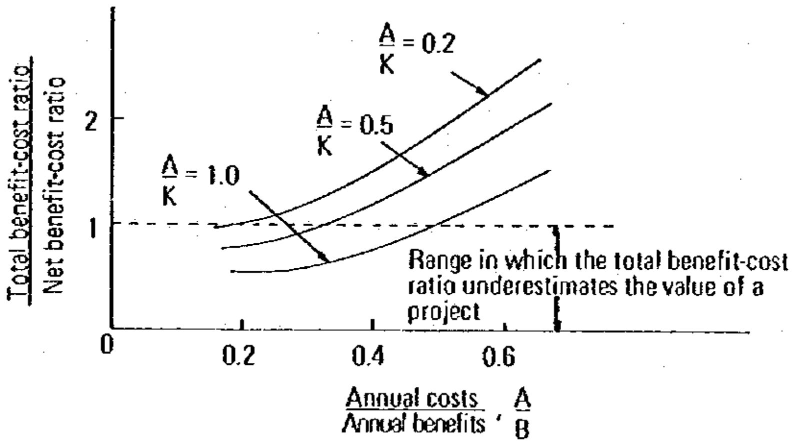



Fig. 5.3 Distortion Introduced by the Benefit-Cost Ratio in the Evaluation of Projects with High Annual Costs A with respect to the Annual Benefits B and the Initial Capital Invested K.

(2) <u>Internal Rate of Return</u>

The concept of the internal rate of return has been proposed as an index of the desirability of projects. The higher the rate, the better a project. By definition, it is the discount rate at which the net present value of the benefits equals the net present value of the costs. That is,

Internal rate of return = r*

such that, using r* as a discount rate,

Present value of all benefits = Present value of all costs

As such it is naturally open to all the difficulties associated with significant annual costs, as indicated in the discussion of the benefit-cost ratio. It has nevertheless been proposed as a measure of the true productivity of a project.

The intuitive appeal of the internal rate of return as a criterion of evaluation is enhanced by the fact that its use would eliminate the direct need to determine the appropriate discount rate, which is a difficult task. This criterion is therefore used by sophisticated design agencies in a number of countries. The concept, however, has three significant weaknesses: it can provide ambiguous values: it may provide a distorted understanding of productivity; and it can alter the ranking of projects from that indicated by the technically more correct net present value approach.

Ambiguous values for the internal rate of return can be obtained when a project is forced to incur high costs at the end of its life. It could happen for a system which operates effectively and provides high benefits at first but later, as with many public services, provides inferior service while incurring high maintenance costs. As an illustration consider a project which costs $200 in the beginning, $310 in its sixth year, and creates total benefits of $500 at the rate of $100 a year in between, as shown in Figure 5-4. For a discount rate of zero the net present value of the project is - $10. For a discount rate infinitely large it is also negative. It is, however, positive for intermediate values. The internal rate of return is then ambiguous.

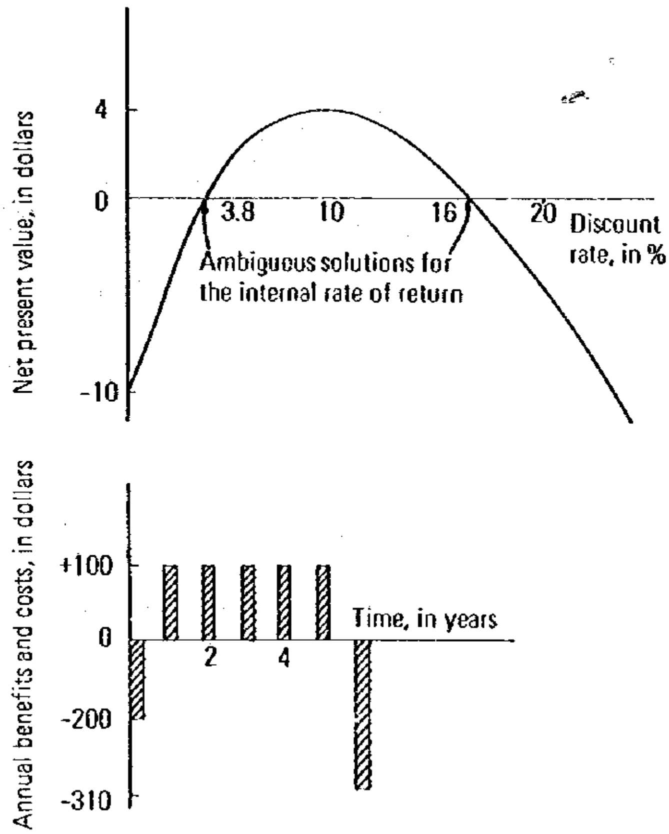



Fig. 5.4 An Ambiguous Solution for the Internal Rate of Return can be Obtained for Projects with High Termination Costs

The calculation of the internal rate of return can give a false sense of productivity when important elements of cost are not included in the calculation. This would occur whenever properties which belong to the government are taken as free resources. When these inputs can actually be used for another valuable purpose, they have an opportunity cost.

Finally, it is possible to construct reasonable numerical examples for which the ranking of projects based on the actual opportunity costs of capital are different from those produced by the internal rate of return criterion. To illustrate this, consider the comparison of two hypothetical projects A and B, in either of which $1,000 can be invested. Project A will return $2,000 in 10 annual increments of $200 each; Project B will return $2,500 in 20 annual increments of $125 each. It can be seen that Project B is more desirable at some discount rates, say r=3%, while Project A has the higher internal rate of return, as shown in Table 5-2.

Table 5.2 Different Ranking of Projects using Benefit – Cost and Internal Rate of Return Criteria

| Projects | Benefit-Cost Ratio at r = 3% | Internal Rate of Return % |
|---|---|---|
| A | 1.73 | ~ 15 (best) |
| B | 1.86 (best) | ~ 11 |

If the discount rate on which the evaluation is based indeed reflects the true opportunity cost, the use of the internal rate of return criterion may lead to the wrong answer. Thus we must conclude that the internal rate of return is not a perfect criterion for the evaluation of projects.

(3) Net Present Value Criterion

Ultimately, we are concerned with the maximization of the value of a system, that is of its net present value. This quantity is

Net present value = present value of all benefits - present value of all costs.

If we are forced to operate within a budget constraint, as may be supposed, then the maximization of net present value is equivalent to the maximization of the benefit-cost ratio or of the rate of return since the cost is fixed. For the situation where we must operate under a fixed budget, the net present value criterion is the single best criterion for the evaluation of projects.

Use of the net present value criterion avoids each of the economic difficulties associated with the benefit-cost and the internal rate of return criteria. Firstly, it avoids the complication introduced by the question of annual operating costs by the simple expedient of focusing directly on the net benefits. Secondly, because the implied reinvestment rates are consistent it does not produce ambiguous results as can the internal rate of return. Third, it does not conjure up false perspectives of productivity. Finally, by charging the capital at its real opportunity cost rather than at an internal rate of return, it provides a more accurate assessment of the net contribution of any project to present value.

In short, the net present value criterion is particularly useful in situations where the capital resources are limited and must be allocated to the most productive projects.

(4) Summary

Each of the two major criteria available for the evaluation of individual projects suffer from substantial theoretical difficulties. The benefit-cost criterion takes no account of distributional impacts and generally overestimates the effects of operating costs. The internal rate of return criterion, which manages to avoid the questions of which discount rate to choose, does raise the possibility, however, that ambiguous results may occur and that the ranking of the projects may deviate from that which would result if the true discount rate were used.

The net present value criterion is therefore recommended as the best single criterion for the evaluation of projects. Whenever the total resources are constrained the net present value avoids the difficulties associated with the other criteria and provides an accurate assessment of the economic merit of a plan. This criterion of economic efficiency should, however, be supplemented by other distributional criteria for the evaluation of large-scale projects.

By Prof. H. Yamada
Kyoto University
Prof. I. Ihara
Kagawa University

## APPENDIX I. INTEREST AND ANNUITY TABLES FOR DISCRETE COMPOUNDING

Various values of the factors for the interest (or discount) rate (6% and 10%) and number of periods from 1 to 100 are shown in Tables 1 and 2, respectively. The conversion symbols commonly used in such tables are given as follows:

(1) Single Payment

$(F/P)_N^i = (1+i)^N$ Single-payment future-worth factor

$(P/F)_N^i = \frac{1}{(1+i)^N}$ Single-payment present-worth factor

(2) Uniform Series

$(F/A)_N^i = \frac{(1+i)^N-1}{i}$ Series-payment future-worth factor

$(P/A)_N^i = \frac{(1+i)^N-1}{i(1+i)^N}$ Series-payment present-worth factor

$(A/F)_N^i = \frac{i}{(1+i)^N-1}$ Sinking-fund factor

$(A/P)_N^i = \frac{i(1+i)^N}{(1+i)^N-1}$ Capital-recovery factor

## Table 1: Discrete Compounding; i = 6%

| | SINGLE PAYMENT | | UNIFORM SERIES | | | | |
|---|---|---|---|---|---|---|---|
| | Compound Amount Factor | Present Worth Factor | Compound Amount Factor | Present Worth Factor | Sinking Fund Factor | Capital Recovery Factor | |
| N | To find F Given P F/P | To find P Given F P/F | To find F Given A F/A | To find P Given A P/A | To find A Given F A/F | To find A Given P A/P | N |
| 1 | 1.0600 | 0.9434 | 1.0000 | 0.9434 | 1.0000 | 1.0600 | 1 |
| 2 | 1.1236 | 0.8900 | 2.0600 | 1.8334 | 0.4854 | 0.5454 | 2 |
| 3 | 1.1910 | 0.8396 | 3.1836 | 2.6730 | 0.3141 | 0.3741 | 3 |
| 4 | 1.2625 | 0.7921 | 4.3746 | 3.4651 | 0.2286 | 0.2886 | 4 |
| 5 | 1.3382 | 0.7473 | 5.6371 | 4.2124 | 0.1774 | 0.2374 | 5 |
| 6 | 1.4185 | 0.7050 | 6.9753 | 4.9173 | 0.1434 | 0.2034 | 6 |
| 7 | 1.5036 | 0.6651 | 8.3938 | 5.5824 | 0.1191 | 0.1791 | 7 |
| 8 | 1.5938 | 0.6274 | 9.8975 | 6.2098 | 0.1010 | 0.1610 | 8 |
| 9 | 1.6895 | 0.5919 | 11.4913 | 6.8017 | 0.0870 | 0.1470 | 9 |
| 10 | 1.7908 | 0.5584 | 13.1808 | 7.3601 | 0.0759 | 0.1359 | 10 |
| 11 | 1.8983 | 0.5268 | 14.9716 | 7.8869 | 0.0668 | 0.1268 | 11 |
| 12 | 2.0122 | 0.4970 | 16.8699 | 8.3838 | 0.0593 | 0.1193 | 12 |
| 13 | 2.1329 | 0.4688 | 18.8821 | 8.8527 | 0.0530 | 0.1130 | 13 |
| 14 | 2.2609 | 0.4423 | 21.0151 | 9.2950 | 0.0476 | 0.1076 | 14 |
| 15 | 2.3966 | 0.4173 | 23.2760 | 9.7122 | 0.0430 | 0.1030 | 15 |
| 16 | 2.5404 | 0.3936 | 25.6725 | 10.1059 | 0.0390 | 0.0990 | 16 |
| 17 | 2.6928 | 0.3714 | 28.2129 | 10.4773 | 0.0354 | 0.0954 | 17 |
| 18 | 2.8543 | 0.3503 | 30.9056 | 10.8276 | 0.0324 | 0.0924 | 18 |
| 19 | 3.0256 | 0.3305 | 33.7600 | 11.1581 | 0.0296 | 0.0896 | 19 |
| 20 | 3.2071 | 0.3118 | 36.7856 | 11.4699 | 0.0272 | 0.0872 | 20 |
| 21 | 3.3996 | 0.2942 | 39.9927 | 11.7641 | 0.0250 | 0.0850 | 21 |
| 22 | 3.6035 | 0.2775 | 43.3923 | 12.0416 | 0.0230 | 0.0830 | 22 |
| 23 | 3.8197 | 0.2618 | 46.9958 | 12.3034 | 0.0213 | 0.0813 | 23 |
| 24 | 4.0489 | 0.2470 | 50.8155 | 12.5504 | 0.0197 | 0.0797 | 24 |
| 25 | 4.2919 | 0.2330 | 54.8645 | 12.7834 | 0.0182 | 0.0782 | 25 |
| 26 | 4.5494 | 0.2198 | 59.1563 | 13.0032 | 0.0169 | 0.0769 | 26 |
| 27 | 4.8223 | 0.2074 | 63.7057 | 13.2105 | 0.0157 | 0.0757 | 27 |
| 28 | 5.1117 | 0.1956 | 68.5281 | 13.4062 | 0.0146 | 0.0746 | 28 |
| 29 | 5.4184 | 0.1846 | 73.6397 | 13.5907 | 0.0136 | 0.0736 | 39 |
| 30 | 5.7435 | 0.1741 | 79.0581 | 13.7648 | 0.0126 | 0.0726 | 30 |
| 35 | 7.6861 | 0.1301 | 111.435 | 14.4982 | 0.0090 | 0.0690 | 35 |
| 40 | 10.2857 | 0.0972 | 154.762 | 15.0463 | 0.0065 | 0.0665 | 40 |
| 45 | 13.7646 | 0.0727 | 212.743 | 15.4558 | 0.0047 | 0.0647 | 45 |
| 50 | 18.4201 | 0.0543 | 290.336 | 15.7619 | 0.0034 | 0.0634 | 50 |
| 55 | 24.6503 | 0.0406 | 394.172 | 15.9905 | 0.0025 | 0.0625 | 55 |
| 60 | 32.9876 | 0.0303 | 533.128 | 16.1614 | 0.0019 | 0.0619 | 60 |
| 65 | 44.1449 | 0.0227 | 719.082 | 16.2891 | 0.0014 | 0.0614 | 65 |
| 70 | 59.0758 | 0.0169 | 967.931 | 16.3845 | 0.0010 | 0.0610 | 70 |
| 75 | 79.0568 | 0.0126 | 1300.95 | 16.4558 | 0.0008 | 0.0608 | 75 |
| 80 | 105.796 | 0.0095 | 1746.60 | 16.5091 | 0.0006 | 0.0606 | 80 |
| 85 | 141.579 | 0.0071 | 2342.98 | 16.5489 | 0.0004 | 0.0604 | 85 |
| 90 | 189.464 | 0.0053 | 3141.07 | 16.5787 | 0.0003 | 0.0603 | 90 |
| 95 | 253.546 | 0.0039 | 4209.10 | 16.6009 | 0.0002 | 0.0602 | 95 |
| 100 | 339.301 | 0.0029 | 5638.36 | 16.6175 | 0.0002 | 0.0602 | 100 |
| ∞ | | | | 18.182 | | 0.0600 | ∞ |

Table 2: Discrete Compounding; i = 10%

| | SINGLE PAYMENT | | UNIFORM SERIES | | | | |
|---|---|---|---|---|---|---|---|
| | Compound Amount Factor | Present Worth Factor | Compound Amount Factor | Present Worth Factor | Sinking Fund Factor | Capital Recovery Factor | |
| | To find F Given P | To find P Given F | To find F Given A | To find P Given A | To find A Given F | To find A Given P | |
| N | F/P | P/F | F/A | P/A | A/F | A/P | N |
| 1 | 1.1000 | 0.9091 | 1.0000 | 0.9091 | 1.0000 | 1.1000 | 1 |
| 2 | 1.2100 | 0.8264 | 2.1000 | 1.7355 | 0.4762 | 0.5762 | 2 |
| 3 | 1.3310 | 0.7513 | 3.3100 | 2.4869 | 0.3021 | 0.4021 | 3 |
| 4 | 1.4641 | 0.6830 | 4.6410 | 3.1699 | 0.2155 | 0.3155 | 4 |
| 5 | 1.6105 | 0.6209 | 6.1051 | 3.7908 | 0.1638 | 0.2638 | 5 |
| 6 | 1.7716 | 0.5645 | 7.7156 | 4.3553 | 0.1296 | 0.2296 | 6 |
| 7 | 1.9487 | 0.5132 | 9.4872 | 4.8684 | 0.1054 | 0.2054 | 7 |
| 8 | 2.1436 | 0.4665 | 11.4359 | 5.3349 | 0.0874 | 0.1874 | 8 |
| 9 | 2.3579 | 0.4241 | 13.5795 | 5.7590 | 0.0736 | 0.1736 | 9 |
| 10 | 2.5937 | 0.3855 | 15.9374 | 6.1446 | 0.0627 | 0.1627 | 10 |
| 11 | 2.8531 | 0.3505 | 18.5312 | 6.4951 | 0.0540 | 0.1540 | 11 |
| 12 | 3.1384 | 0.3186 | 21.3843 | 6.8137 | 0.0468 | 0.1468 | 12 |
| 13 | 3.4523 | 0.2897 | 24.5227 | 7.1034 | 0.0408 | 0.1408 | 13 |
| 14 | 3.7975 | 0.2633 | 27.9750 | 7.3667 | 0.0357 | 0.1357 | 14 |
| 15 | 4.1772 | 0.2394 | 31.7725 | 7.6061 | 0.0315 | 0.1315 | 15 |
| 16 | 4.5950 | 0.2176 | 35.9497 | 7.8237 | 0.0278 | 0.1278 | 16 |
| 17 | 5.0545 | 0.1978 | 40.5447 | 8.0216 | 0.0247 | 0.1247 | 17 |
| 18 | 5.5599 | 0.1799 | 45.5992 | 8.2014 | 0.0219 | 0.1219 | 18 |
| 19 | 6.1159 | 0.1635 | 51.1591 | 8.3649 | 0.0195 | 0.1195 | 19 |
| 20 | 6.7275 | 0.1486 | 57.2750 | 8.5136 | 0.0175 | 0.1175 | 20 |
| 21 | 7.4002 | 0.1351 | 64.0025 | 8.6487 | 0.0156 | 0.1156 | 21 |
| 22 | 8.1403 | 0.1228 | 71.4027 | 8.7715 | 0.0140 | 0.1140 | 22 |
| 23 | 8.9543 | 0.1117 | 79.5430 | 8.8832 | 0.0126 | 0.1126 | 23 |
| 24 | 9.8497 | 0.1015 | 88.4973 | 8.9847 | 0.0113 | 0.1113 | 24 |
| 25 | 10.8347 | 0.0923 | 98.3470 | 9.0770 | 0.0102 | 0.1102 | 25 |
| 26 | 11.9182 | 0.0839 | 109.182 | 9.1609 | 0.0092 | 0.1092 | 26 |
| 27 | 13.1100 | 0.0763 | 121.100 | 9.2372 | 0.0083 | 0.1083 | 27 |
| 28 | 14.4210 | 0.0693 | 134.210 | 9.3066 | 0.0075 | 0.1075 | 28 |
| 29 | 15.8631 | 0.0630 | 148.631 | 9.3696 | 0.0067 | 0.1067 | 29 |
| 30 | 17.4494 | 0.0573 | 164.494 | 9.4269 | 0.0061 | 0.1061 | 30 |
| 35 | 28.1024 | 0.0356 | 271.024 | 9.6442 | 0.0037 | 0.1037 | 35 |
| 40 | 45.2592 | 0.0221 | 442.592 | 9.7791 | 0.0023 | 0.1023 | 40 |
| 45 | 72.8904 | 0.0137 | 718.905 | 9.8628 | 0.0014 | 0.1014 | 45 |
| 50 | 117.391 | 0.0085 | 1163.91 | 9.9148 | 0.0009 | 0.1009 | 50 |
| 55 | 189.059 | 0.0053 | 1880.59 | 9.9471 | 0.0005 | 0.1005 | 55 |
| 60 | 304.481 | 0.0033 | 3034.81 | 9.9672 | 0.0003 | 0.1003 | 60 |
| 65 | 490.370 | 0.0020 | 4893.71 | 9.9796 | 0.0002 | 0.1002 | 65 |
| 70 | 789.746 | 0.0013 | 7887.47 | 9.9873 | 0.0001 | 0.1001 | 70 |
| 75 | 1271.89 | 0.0008 | 12708.9 | 9.9921 | * | 0.1001 | 75 |
| 80 | 2048.40 | 0.0005 | 20474.0 | 9.9951 | * | 0.1000 | 80 |
| 85 | 3298.97 | 0.0003 | 32979.7 | 9.9970 | * | 0.1000 | 85 |
| 90 | 5313.02 | 0.0002 | 53120.2 | 9.9981 | * | 0.1000 | 90 |
| 95 | 8556.67 | 0.0001 | 85556.7 | 9.9988 | * | 0.1000 | 95 |
| 100 | 13780.6 | * | 137796 | 9.9993 | * | 0.1000 | 100 |
| ∞ | | | | 10.0000 | | 0.1000 | ∞ |

* Less than 0.0001.

## APPENDIX II. BIBLIOGRAPHY


1. Barish, N. N., "Economic Analysis: for Engineering and Managerial Decision-Making,"McGraw-Hill, 1962.
2. Baumol, W. J., "Economic Theory and Operations Analysis", Prentice-Hall, 1965.
3. Chiang, A. C., "Fundamental Methods of Mathematical Economics", McGraw-Hill, 1967.
4. Dean, J., "Managerial Economics", Prentice-Hall, 1951.
5. DeGarmi, E. P. & Canada, J. R., "Engineering Economy", Collier-Macmillan, 1973.
6. Hague, D. C., "Managerial Economics: Analysis for Business Decisions", Longmans, 1969.
7. Neufville, R. de, & Stafford, J. H., "Systems Analysis for Engineers and Managers", McGraw-Hill, 1971.
8. Riggs, J. L., "Economic Decision Models: for Engineers and Managers", McGraw-Hill, 1968.
9. Spencer, M. H., "Managerial Economics: Text, Problems, and Short Cases", Richard D. Irwin, 1968.
10. Taylor, G. A., "Managerial and Engineering Economy: Economic Decision-Making", D Van Nostrand, 1964.
11. Theil, H., Boot, J.C.G. & Kloek T., "Operations Research and Quantitative Economics: an Elementary Introduction", McGraw-Hill, 1965.