AN INTERACTIVE LIFE CYCLE COST
FORECASTING TOOL

THESIS

Nicolas M. Habash
Captain, USAF

AFIT/GOR/ENS/92D-01

THESIS APPROVAL

STUDENT: Nicolas M. Habash CLASS: GOR-92D

THESIS TITLE: An Interactive Life Cycle Cost Forecasting Tool.

DEFENSE DATE: 12 November 1992

COMMITEE:	NAME/DEPARTMENT	SIGNATURE
Advisor:	Dr. John P. Cain/ENS	....
Co-advisor:	Maj. Kenneth W. Bauer/ENS	....

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

AFIT/GOR/ENS/92D-01

AN INTERACTIVE LIFE CYCLE FORECASTING TOOL

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Nicolas M. Habash, M.S.

Captain, USAF

December 1992

Preface

This study is to develop software capable of computing life cycle cost of systems using constant cost elements, stochastic cost elements, and cost estimating relationships. In addition, the software is to incorporate file functions to include reading cost files, creating cost files, and viewing and editing the data within the cost files. Emphasis of the study is placed on structured software code, simplified user interaction, and efficiency of algorithm.

I chose this research topic because of my interest in economic related computation and my strength in structured programming.

I am indebted to Dr Cain for his guidance and patience. I take pleasure in the opportunity to work with him in developing this quality software.

I would also like to extend my appreciation to Major Bauer for his part in the completion of this effort as my sole reader.

My final thanks goes to my family for their support and understanding during these taxing times while I completed this degree.

Nicolas M. "Nick" Habash

Table of Contents

Table of Content (Cont'd)

List of Figures and Tables

Abstract

This thesis is about computing life cycle cost and culminates with a life cycle cost forecasting tool (software). The software is written in FORTRAN 77 and uses IMSL subroutine extensively for random number generation from various distributions. The algorithm can compute life cycle cost for the following types of cost elements: constant, stochastic, and cost estimating relationships (CERs). The use of a trapezoid approximation of payment allocation is allowed. Percent allocation of payments is also allowed and is recommended where the trapezoid fails to correctly represent the payment schedule. The CER algorithm handles cost data that was estimated *in* base 10, in natural logarithm, or in logarithm.

File management capability is an integral part of the software. The software can read, create, view and edit cost files. The output of the software includes two files. File filename_LC.OUT contains a annual cost and total cost at 85, 90, 95, and 99 percent confidence levels. File filename_SAS.DAT contains all randomly generated annual costs. A screen output of filename_LC.OUT occurs after executing the algorithm on the cost data.

AN INTERACTIVE LIFE CYCLE COST FORECASTING TOOL

I. Introduction

Background

The acquisition of a weapon system by the Department of Defence (DoD), the investment in equipment by a commercial enterprise, or a new change in public welfare policy, all require assessment of cost. Cost associated with such ventures may be continuous or short term and may be composed of an initial outlay of funds followed by annual payments or returns over the life of the venture. In any case, the true cost or gain is the distribution of the cash flow over time taken at present value.

An important consideration pertinent to the cost of a venture is the timing of the money spent. Since time is money, a system that includes a later outlay of cash may be less expensive than an alternative system even though it has an equal or greater total cost. On the other hand, two ventures having the same initial outlay of funds followed by periodic returns may have different present worth. The venture that has the larger present value of its periodic returns has the higher worth. The value of money today is always more than the value of the same amount in the future.

Costing the system requires the calculation of its present value over its life, in term of dollars. This allows for budgeting, for planning, and for the comparison between other like ventures. To capture the inherent uncertainty in a cost estimate, Monte Carlo methods must be used in the estimating process. Monte Carlo methods involves the approximation of a system cost through repeated random sampling.

In ventures that require new technologies, the cash flow is not known and requires estimation. The cost estimation of the new venture is most often based on the cost of older similar ventures with adjustments made for differences. This is known as costing by analogy. The adjustments made for the differences is often based on the expected cost of research and development. Thus the cost of a venture can be known (a constant), stochastic (based on probability), or a cost estimating relationship (based on a relationship between such variables as speed, capacity, ability and the like), which may be known or stochastic in nature.

DoD programs require federal funding in order for them to survive. The acquisition of major weapon systems is a costly and long-term effort, often requiring the commitment of billions of dollars over many years (4:2). The funding necessary for a single weapon system can span decades (like the B-52 bomber or the C-141 transport aircraft). This occurs because weapon systems are not off-the-shelf items, used for a period of time and then disposed of. These weapon systems are highly specialized and are not readily available. Their life begin with conception and ends with retirement. The cost over their life incorporates all facets of development, testing, acquisition, operating, repairing, upgrading, modifying, and final retirement. Therefore, a weapon system requires a constant flow of funds throughout its life, hence the term LCC. The AF Systems Command Manual 173-1, Cost Estimating Procedures, defines LCC as:

> the government's total cost of owning a system, subsystem, or component over its full life. It includes development, production, operating, and support costs.

Statement of Problem

Systems Analyst Engineers at the Air Force Institute of Technology (AFIT) use life cycle cost (LCC) as an input to the decision of selecting between alternative systems. LCC is also used in budgeting and planning. A previous model, although operative in its current form, is bulky, lacks programming structure and is not user friendly. More specific, the model lacks the flexibility of easily editing cost elements. It does not incorporate in its algorithm the correct distributions for approximating from less than thirty degrees of freedom (number of observations minus number of parameters associated with the least square regression equation). Its ability to handle a diverse type of cost estimating relationships (CERs) is also limited. In addition, the execution of the model on a given data file is time consuming. AFIT requires a computer model that can input multiple data files, has data-file editing ability, use the correct distributions in its algorithms, provide for additional CER computing ability, and is efficient in computation. Such a model would allow costing systems to be based on more diverse data, easy updating of the respective data files as more accurate information is obtained, and higher utility due to efficiency. The program will be made available to all DoD departments with interest in cost estimation of systems in the conceptual phase.

Summary of Current Knowledge

Life cycle cost estimations are usually done by any of three methods: estimation by analogy, parametric modeling, or engineering estimation. Estimating by analogy, the interpolation of a new systems LCC from an existing system, is generally considered the easiest of the three. It requires the least amount of detail, knowing a similar base system from which the new system cost can be estimated. Parametric modeling requires more work than the analogy estimation due to the addition of specific characteristics in the cost estimating relationships. Engineering, or bottom up, estimates require a great deal of detail and are generally not recommended by the DoD. The use of life cycle costing in the conception phase of a project provides management with a reasonable and quick estimate of the expected cost of the project, as well as with a model that is easily modified as changes occurs in other phases of the project.

Scope

The LCC model will incorporate a structured technique of programming. This technique uses the calling of different subroutines to perform specific subfunctions of the main program.

The program will be able to "read-in" existing data files, create new data files, view and edit the data, and execute the program to output a similarly named "out-file".

The program will be able to process constant cost estimates (CCEs), stochastic cost estimates (SCEs), and cost estimating relationships (CERs). All types of cost elements will be in base 10; however, CERs will also be allowed to be in logarithm of base 10 and base "e".

The use of Internal Math Subroutines Library (IMSL), available on the VAX computer system, will be used extensively to keep the program length to a minimum. The various random number generating IMSL subroutines will be used in the stochastic aspects of the program.

Research Approach

I have completed three college level classes in the Fortran programming language and programming structure. I shall reference notes and texts on the programming language as difficulties arise. I will research Monte Carlo methods to become more familiar with the intricacies of the procedure. I will research the computation methods of CCEs, SCEs, and CERs. In addition, I will research the computation method of CERs that are in logarithm of base 10 and base "e". On-hand texts will be used to reference the various probability distributions to be used in the computations.

Material and Equipment

The program, Life Cycle Cost, will be written in Fortran 77 programming language on the VAX computer system located at the Air Force Institute of Technology (AFIT). Access to the VAX will be through a personal computer linked to the VAX through a local area network. The IMSL routines will be used throughout the program for speed of calculation, and to keep the program length at a minimum.

Results

The program is intended to serve the USAF in cost estimation and budgeting of various acquisition of systems. However, the program will be available to AFIT students to be used throughout their studies as the need arise. Emphasis in the program will be on making it user friendly in the input and manipulation of data as well as in the correction of data. It will be helpful in catching input errors and prompting the user for corrections. Also, emphasis will be placed on speed of algorithm, to minimize user waiting time, and computational accuracy.

II. LITERATURE SEARCH

Introduction

The purpose of this literature search is to familiarize the reader with life cycle costing (LCC) research. Until recently, management has justified the economy of an expenditure primarily by the purchase price. They would normally select the item that would provide the required service at the lowest purchase price. Current economics extends the cost of an acquisition to all expenditures associated with it, from its development to its retirement. Thus the manager should recognize that the cost associated with an acquisition includes the cost of development, procurement, and ownership, and that the purchase price may be less significant than the subsequent costs associated with ownership.

LCC provides a realistic estimation of the actual cost associated with an acquisition over its life, and it is to be used to better distinguish between the economy of alternatives. Most of the cost associated with a new acquisition can be determined or approximated using current cost figures. These may include the cost of shipping, tax incentives, and operating. However, "costing" the development and maintaining of a system may require more than the mere referencing of current costs since no specific long term data may be available. For these types of estimates, three types of LCC estimating tools are available: estimating by analogy, estimating by parametric, and estimating by engineering.

Life Cycle Cost Models

Dr. Joseph Cavinato, in his journal article entitled, " Product Costs From Cradle To Grave", asserts that the purchase price of an item is only part of its total cost, just the start of a series of costs to be accumulated by the firm, its downstream customers, and users until the end of the product's life. It reflects the cumulative costs of developing and maintaining supplier relationships, transportation costs, sales and freight term, payment terms, downstream service costs, ultimate user costs, and salvage cost (1:51-52). In other words, any and all costs associated with a system may be considered total. In the Department of Defense Directive 5000.1, LCC is believed to be a crucial element in deciding which alternative system is to be implemented. They state that "it is sometimes the only factor that can be directly compared between competing systems". The Program Analysis and Evaluation office of the Assistant Secretary of Defense states, "Cost estimates are as important as operational effectiveness measures in a cost and operational effectiveness analysis" (4:15). There are three types of LCC models:

1. Estimating by analogy.
2. Parametric methods.
3. Engineering or bottom up estimates.

"Estimates by analogy and the parametric methodology are both 'top down' methods, because they examine the program as a whole" (5:16). The engineering estimate is bottom up method because it includes detailing the cost of every component that is contained within the whole program (5:16). The meaning of "top down" and "bottom up" will become apparent in the following review.

Estimating by Analogy

Estimating by analogy is the process of estimating a new system cost based on an existing system. It is the simplest of the LCC model because it requires the least amount of details. In the DoD Directive 5005.1 they state that "Analogy estimates are conducted by adjusting the cost of a known system similar to the one in question to arrive at a cost projection" (4:15). Therefore, the cost of a current project is determined by examining the cost of known projects and adjusting for differences. If data are available on more than one past program, all relevant data will be included in the estimate by providing a general relationship on those characteristics (both similar and different) of all systems concerned (5:15-16).

Analogy estimates are usually conducted very early in the development of a future system to try to gage the approximate order of magnitude of the expected total cost. "By starting LCC analysis at the conceptual phase, it is possible to ask whether it would be wiser to produce a new system or to modify an older existing one" (5:15). Estimating a new system's cost of development, for example, may begin by consulting the engineering department to determine how much engineering and management the new system will require with respect to a similar completed system. From the design department, estimates are obtained on the design of the new system based on the design cost of the existing system. The same method is used for fabrication and testing of the new system to determine an estimated cost. Differences in the costs between the proposed and old system may reflect inflation, size of the task, enhance capabilities, differences in materials used, advancement in technology , and the like. In similar fashion, all relevant other system costs may be estimated (5:23-25).

Table 1 contains an example of how analogy estimates are conducted by comparing a new program for airborne electronic equipment against an already existing system (taken from Seldon's book on Life Cycle Costing: A Better Method of Government Procurement).

Table 1

Cost Estimate by Analogy (million $)

Cost Element	Base Program	New Program
System engineering and management	2.0	2.2 (a)
Design	8.0	11.2 (b)
Prototype fabrication and material	1.5	1.1 (c)
Flight and laboratory test	3.0	1.5 (d)
Total	14.5	16.0

Notes:

(a) Systems engineering and program management for the New Program are similar to those of the Base Program; the 10% increase in the New program is due to larger task of design monitoring.
(b) In the New Program, the additional design task of the moving target indicator adds 20%, and higher performance requirements will require another 20%, for a total of 140% of the cost of the Base Program.
(c) The New Program requires only two prototypes versus three for the Base Program; tooling and other fixed costs were about $300,000 for the Base Program, with each prototype at $400,000.
(d) The New Program requires only one aircraft model qualification; flight test personnel estimate the cost one-half that of the Base Program.

If conducted early in the conceptual phase, the method of estimating by analogy provides management with a quick and easy way to determine if the cost of a new project is prohibitive or not. This method allows periodic updates to be made as they occur, with relatively minor effort. "Surprisingly, such estimates are usually accurate if all the significant changes between programs are understood and accounted for" (5:25).

Parametric Methodology

"Parametric cost analysis involves the development and utilization of estimating relationships between historical costs and physical and/or performance characteristics of a system" (2:5-4; 4:15). The system characteristics (such as aircraft speed, the number of maintenance personnel required for support) are commonly referred to as parameters; whereas, "the historical costs reflects the impact of growth, engineering changes, program over runs, and any other cost, schedule, or performance difficulties encountered in comparable programs" (2:5-4). If the characteristics of the current or previous system can be quantified by some statistical means (linear regression for example), a parametric relationship or cost estimating relationship (CER) can be developed. The purpose of a CER is to attempt to predict the future based upon information received from past occurrences (4:79). A CER accomplishes this by transforming the problem from one of estimating dollars to one of estimating a more familiar and more accessible variable" (5:25). A major concern with the use of CER's as forecasting tools is the level of confidence that can be placed in the estimates derived from the CER.

There are several advantages associated with using CER's to estimate costs. These advantages are as follows:

1. The cost estimates are based on general system characteristics, no detailed information is necessary;
2. The model is very fast and easy to use;
3. The model is resistant to user bias;
4. The confidence intervals can be placed on forecasts since parametric statistics are used in generating the forecasts (6:8).

The concern regarding the level of confidence that can be placed in CER estimates is addressed by the fourth advantage.

Figure 1 shows an example of a parametric CER model. Notice the costs reflect general characteristics and an error is associated with each algorithm.

COST REPRESENTATION FOR ONE C-130 AIRCRAFT
USING THE PARAMETRIC METHODOLOGY

```
Airframe   =  200,000 + 50 * X1 + e
Engine     =  2,000 + 60 * X2 + e
Electronic=  530 + 2000 * X3 + e
Manpower   =  300,000 + 400,000 * X4 + e
Operating =  500,000 + 12,000 * X5 +e
```

X1	=	airframe weight	X4	=	crew size
X2	=	thrust	X5	=	annual flight hours
X3	=	number of radios	e	=	error; IID, N(0,MSE)

Figure 1 Parametric CER cost model example (6:9).

Engineering Methodology.

The engineering method (a.k.a. the bottom up method) for cost estimating is the costing of the system through the costing of its parts and sub-components. It is the breaking down of a system to its smallest components, estimating the cost of their production, and the cost of their assembly into the system; hence, the name "bottom up". To conduct an engineering estimate, a detailed description of the task to be completed must be understood by every level involved in the project. Each level estimates the cost of doing its particular task assigned to it by using specific hardware-to-cost relationships (5:31,181). The sum of the cost estimate derived from each level represents the total cost of the project, hence the term 'bottom up'. Obviously this requires more information, and indeed the DoD does not recommend this as a method for preliminary cost analysis (4:54).

Summary

Life cycle cost modeling is usually done through one of three methods. Estimating by analogy is generally considered the easiest of the three. This is because it requires the least amount of detail and uses a similar base system from which the new system cost can be estimated. Parametric modeling requires more work than the analogy estimation due to the addition of specific characteristics in the cost estimating relationships. Engineering or bottom up estimates require a great deal of detail and are generally not recommended by the DoD. The use of life cycle costing in the conception phase of a project provides management with a reasonable and quick estimate of the expected cost of the project, as well as with a model that is easily modified as changes occurs in other phases of the project.

III. Methodology

Overview

The purpose of this chapter is to familiarize the reader with the general methods used in the development of the program (LC.FOR). (See Figure 2. LC.FOR Program Structure). In addition, the reader will also be acquainted with the computational algorithms and the distributions employed.

Program Structure.

LC.FOR is written in Fortran 77, VAX language. LC.FOR is a menu driven program requiring user interaction through various prompts and questions. The Program structure is based on a main routine that calls upon subroutines to accomplish user-selected tasks. The routine "MAIN" allows for any of five selections:

1. READ A COST FILE
2. CREATE A NEW COST FILE
3. VIEW/EDIT A COST FILE
4. EXECUTE LIFE CYCLE COST
5. EXIT PROGRAM

Read Existing Cost File.

User selection of "read a cost file" invokes main program "MAIN" to call subroutine "READ". "READ" lists the last version of all data files of the form "..._LC.DAT" (example: B52_LC.DAT). After the user makes an entry of the file name, "READ" continues to read the cost file and returns to the main program. The cost file remains intact and is closed. At this time the data is still available in the cost file and is also loaded in the program.

Create a New Cost File.

User selection of "create a new cost file" invokes "MAIN" to call subroutine "CREATE". "CREATE" prompts the user for the file name under which the data is to be stored in the default directory. Once the file name is given, "CREATE" then prompts the user for the life of the system for which LCC is to be forecasted, and the expected discount rate. Then "CREATE" prompts the user for the number of constant cost elements (CCEs), stochastic cost elements (SCEs), and cost estimating relationships (CERs) to be used in computing LCC. "CREATE" then calls subroutines "INCCE", "INSCE", and "INCER" if the number of CCEs, SCEs, or CERs is greater then zero, respectively. "INCCE", "INSCE" and "INCER" each prompt the user for the entry of CCE, SCE, and CER type data respectively. Once all data items are entered into the program, the program saves the data in a new file and returns to "MAIN" routine. At this time the data is saved into a new cost file and is also available in the program. Saving of the data is accomplished by subroutine "CREATE" calling subroutine "WRITE".

View/Edit a Cost File.

Once data is available in the program, it may be viewed and edited. Upon user selection of "view/edit a cost file" from the "MAIN" menu, "MAIN" calls subroutine "VIEW" to perform the displaying of the data. An entry to edit a data item results in the calling of "INCCE", "INSCE" or "INCER" depending on the cost element type being viewed at the time. Once all viewing of data is completed and if editing has occurred, the program saves the updated data into a new version of the cost file with the same name or a new name depending on a user entry. This is done by the subroutine "VIEW" calling subroutine "WRITE". The program then returns to "MAIN".

Execute Life Cycle Cost.

Upon user selection of "Execute Program", "MAIN" calls subroutine "EXEC". "EXEC" prompts the user for the number of number random variables desired, in order to perform a Monte Carlo simulation. It then performs the computation of the system's LCC. It writes to the screen and a file named filename_LC.OUT the summary data and to a file named filename_SAS.DAT all the annual cost estimates. The program, following a 'RETURN' entry, returns to "MAIN" menu.

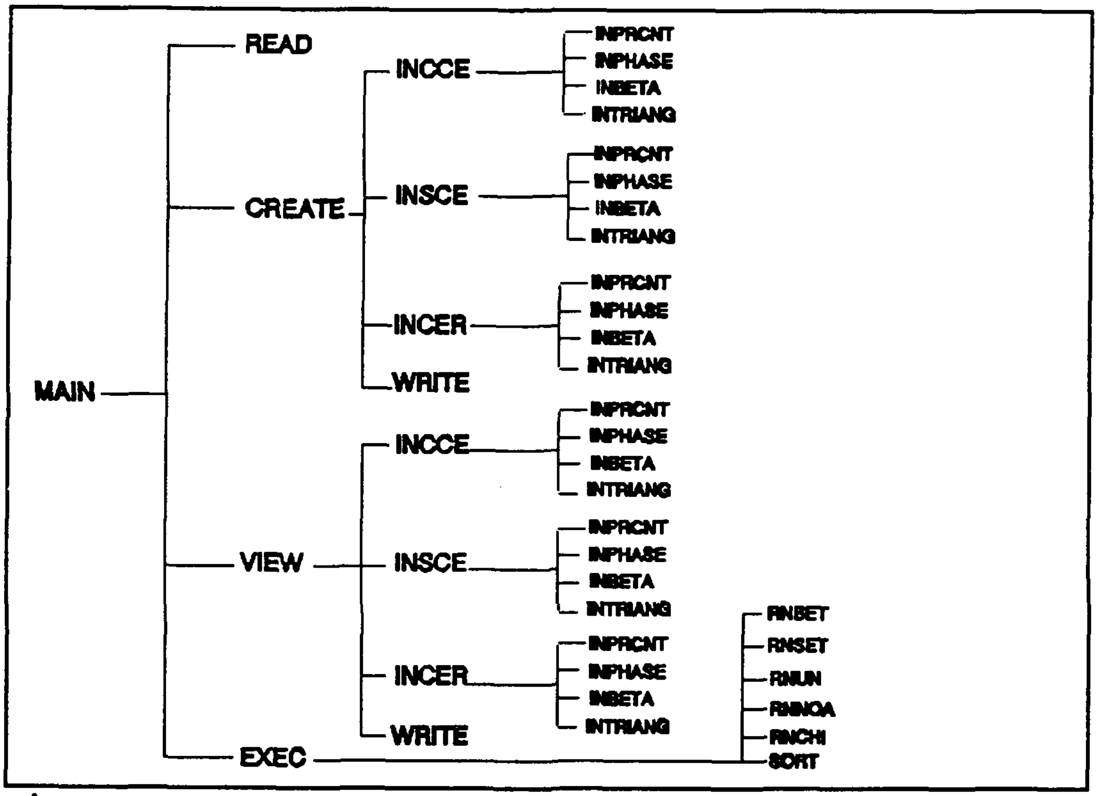


Figure 2. LC.FOR Program structure.

Computational Algorithms.

IMSL subroutines are used for sampling from the Beta, the Normal, the Uniform, and the Chi-square distributions.

T-Distribution. To obtain a sample from a "t" distribution when a CER's degree of freedom is less than thirty the following relationship is used:

RT = RN * SQRT(DF/RX)

RT Random number from a Student-t distribution.
RN Random number from a Normal (0,1) distribution. RN is obtained from the IMSL routine RNNOA.
DF Degree of Freedom for each CER.

DF = N - [K + 1], If an intercept term exists.
DF = N - K, If there is no intercept term.

N Number of observations.
K Number of slope parameters.

RX Random number from a Chi-square distribution. RX is obtained from IMSL routine RNCHI.
SQRT Square root.

Computing Cost for a CCE.

Trapezoid Allocation of Cost (Discounted).

$$PV = C * Tr_{yr} * 1/(1+i)^{yr}$$

Percent Allocation of Cost (Discounted).

$$PV = C * Pr_{yr} * 1/(1+i)^{yr}$$

PV	Present value.
C	Cost (Fixed for CCE, Stochastic for SCE and CER).
Tr	Fraction based on trapezoid distribution (See Figure 3).
Pr	Fraction based on percent allocated to the year.
i	Discount Rate.
yr	Year at which this payment fraction is made.

A loop is created to sum the PV of all fractional parts of the cost.

Computing Cost for a SCE.

Triangular Probability Distribution.

A: Smallest estimation of Cost.
B: Largest estimation of Cost.
D: Mode of best guess of Cost.

For RU <= ((D-A)/(B-A))

C = A + SQRT((D-A) * (B-A) * RU)

For RU > ((D_A)/(B_A))

C = B - SQRT((B-D) * (B-A) * (1-RU))

RU: Random Uniform number of range 0 to 1, obtained from IMSL subroutine.

Beta Probability Distribution.

C = A + RB * (B-A)

RB: Random Beta number of range 0 to 1, obtained from the IMSL subroutine. RB is a function of the beta type designated by the user in the data. There are nine types of Beta (See Figure 4).

Allocation of Cost. See Trapezoid Allocation of Cost and Percent Allocation of cost under Computing Cost for a CCE.

Computing Cost for a CER.

Base 10 (whole numbers)

$$Chat = b_0 + b_1 x_1 + b_2 x_2 + \ldots$$

For DF => 30: C = Chat + RN SQRT(Var + X' Var/Cov X)

For DF < 30: C = Chat + t SQRT(Var + X' Var/Cov X)

TC = SCL * C

SCL Scalar multiple (from a fixed value, a triangular dist., or a beta dist.)
X Vector of x_{is}
Chat Expected value of the cost of the CER.
b_0 Value of the Intercept term.
b_i slope parameter (or co-efficient) of driver.
x_i Driver. A variable that may be a know constant, or a stochastic from a Beta or a Triangular distribution.
TC Total cost.
Var/Cov Variance Covariance matrix of the intercept and slope parameters.

Log base 'e' (Natural log)

$x_i = Ln(x_i)$ For all i

$Chat = b_0 + b_1 x_1 + b_2 x_2 + \ldots$

For DF => 30: C = Chat + RN SQRT(Var + X' Var/Cov X)

For DF < 30: C = Chat + t SQRT(Var + X' Var/Cov X)

TC = SCL * Exp(C) Exp Exponential

Log base 10 (Log)

$x_i = Log(x_i)$ For all i

$Chat = b_0 + b_1 x_1 + b_2 x_2 + \ldots$

For DF => 30: C = Chat + RN SQRT(Var + X' Var/Cov X)

For DF < 30: C = Chat + t SQRT(Var + X' Var/Cov X)

$TC = SCL * 10^C$

Allocation of Cost. See Trapezoid Allocation of Cost and Percent Allocation of cost under Computing Cost for a CCE.

Annual Cash Flow.

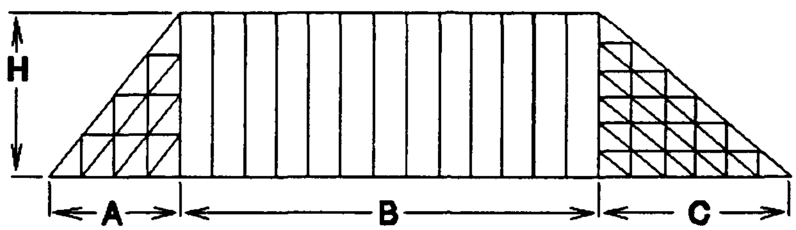
Based on a Trapezoid.

A trapezoid can be used to approximate a payment profile. The total area of a trapezoid represents a normalized cost (ie. the area sums to 1) (See Fig. 3). The measurement of the height of the trapezoid accomplishes normalization. 'A' of the trapezoid represents a period when a larger portion of cost is paid each successive year (phase-in period). 'B' of the trapezoid represents a period when the portion of cost paid is constant for each successive year (constant cost period). 'C' of the trapezoid represents a period when a smaller portion of cost is paid each successive year (phase-out period). Thus, for the first year, the payment percent allocated to that year is represented by the area of the far left small triangle show within 'A' of figure 3 below. The total cost imputed to that year is the percent multiplied by the cost of the component. For the second year, the payment percent is represented by the area formed by the three small triangles stacked immediately to the right of the far left small triangle in area 'A'. During the constant cost periods, area 'B', each rectangle in area 'B' represent the fraction of cost paid in a given year. During the phase-out period the reverse of the phase-in period occurs. For each successive year, a smaller percentage of cost is paid until the last fraction, represented by the far right small triangle, is paid. At that time no more payments are made for that particular cost element.

A = 4 yrs Phase-in period.
B = 13 yrs Constant payment period.
C = 6 yrs Phase-out period.

Area = 100% of Cost

$H = 2/(A + 2B + C)$



Phase-in period: $I = 1, A$; $Tr_I = (2I - 1)H/(2A)$

Constant payment period: $I = 1, B$; $Tr_I = H$

Phase-out period: $I = 1, C$; $Tr_I = [(2C + 1) - 2I]H/(2C)$

Figure 3. Annual cash flow based on a Trapezoid.

<u>Based on a Percentage.</u>

A percentage of the sampled cost is distributed each year for a designated number of years. Thus, cash flow for a fiscal year is equal to that years percentage of cost. For example, if cost is $100,000 and for year two 25 percent is designated, then the cash payment for that year is $25,000.

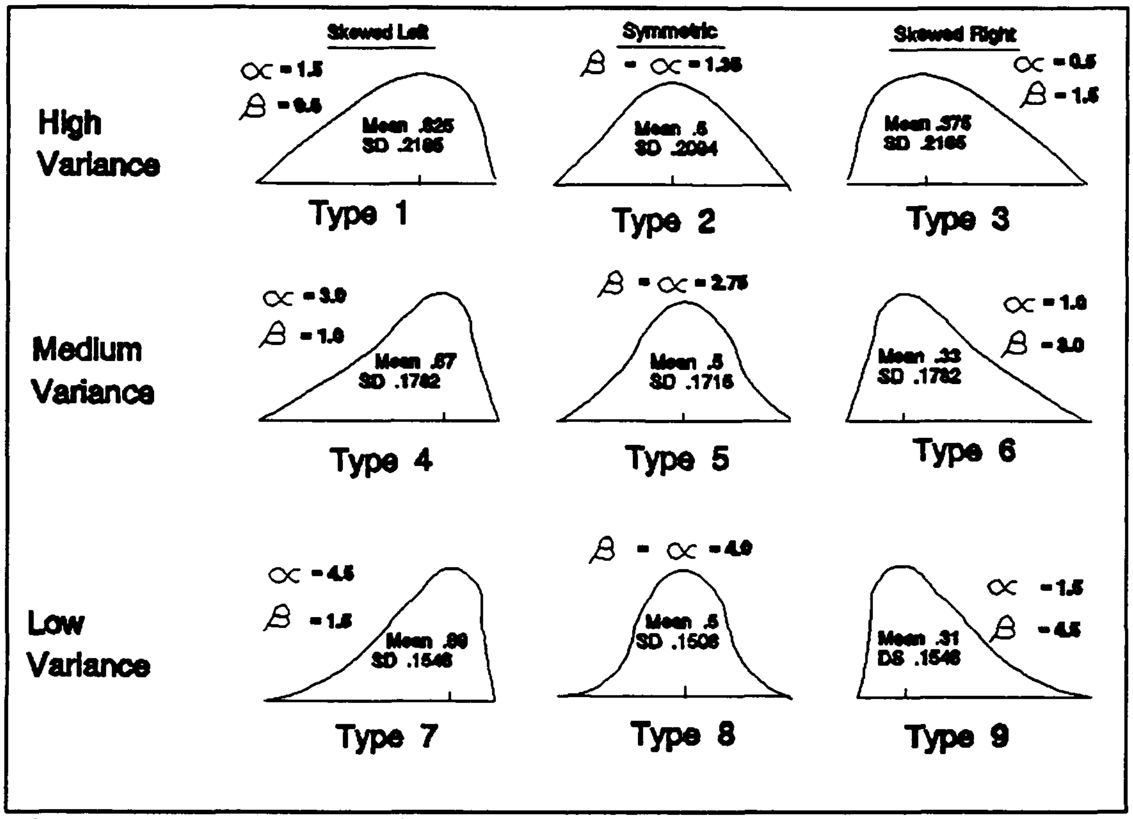


Figure 4. Beta distribution types.

Probability Distributions.

Beta.

The nine beta probability distribution functions show in figure 4 above give a visual representation of the uncertainty of components could be best described. Beta distributions of type 1, 4, and 7 are all skewed left. This indicates that a sample taken from 1, 4, or 7 is more likely to be closer to the high estimate than the low estimate. The main difference between 1, 4, and 7 is the variance. 4 has a smaller variance than 1, while 7 has a smaller variance than 4. A smaller variance indicates a tighter distribution about the mean. If the Beta distribution is too confusing (ie. the estimator is uncomfortable with the concept of variance), then the triangular distribution may suffice for estimation.

Triangular.

The triangular distribution operates much like the Beta distribution except only a lower limit, an upper limit, and a best guess (mode) at the cost is required. A particular advantage of the triangular distribution is that it's more intuitive than the Beta distribution. The concept of "best guess" replaces variance with the triangular distribution. And, for a large number of sampling, the triangular distribution sampled uniformly approaches a symmetric Beta distribution.

IV. Conclusions and Recommendations

Conclusions

The purpose of this effort was to produce a user friendly program for calculating LCC of systems given CCEs, SCEs, and CERs. Cost estimates were to be sampled from a fixed constant, a Triangular or a Beta distribution. Payments were to be based on a trapezoid or a user designated percentage per given year. The program was to incorporate a Normal and a Student-t distribution of the variance of the CERs. Of primary importance is the program to allow the user to view and edit existing data files and to save the data under new names if desired. The program was to be structured in a fashion to ease future incorporation of other types of cost elements, promote easy de-bugging of code, and for its algorithm to be accurate and efficient. All these have been achieved if not more.

Recommendations. Recommend to incorporate multiplicative cost elements (MCE). MCEs are cost elements were the quantity and the price of a system or subsystem are each being a known constant, based on a triangular distribution, or based on a Beta distribution or any combination of which. For example a purchase of a system may include a quantity of the same subsystems. If the quantity of subsystems is unknown, then the number may be described by a triangular or a Beta distribution. If the cost of the subsystem is also unknown, then the cost of each subsystem may be represented by a triangular or a Beta distribution. Any combination of a constant, triangular distribution, and Beta distribution may be used to represent the purchase of a quantity at a price. Thus the total cost is the product of the sampled quantity and the sampled price.

Appendix A

User Manual

Connecting to the VAX. From a personal computer that is attached to the VAX (a.k.a. csc) via a local area network, type 'c csc'. If you are connected directly to the VAX, simply type 'login'. Either of these commands will connect you to the VAX.

Getting the Program. Make sure you have a copy of LC.EXE in the directory you wish to store the cost files. The output files will be created in the same directory but may be moved later if desired. You may have to copy LC.EXE into your desired directory. Copy the program from the floppy disk into the directory you wish to operate in. For users not at AFIT that have access to a VAX system, you require only the executable file LC.EXE. If the program is to run on a personal computer, then you need to obtain a copy of the fortran source code (LC.FOR). The program needs to be compiled and linked with IMSL subroutines to be executable. If IMSL subroutines are not available, then the user must write his own subroutines to obtain random samples from a Normal (N(0,1)), a Chi-square, a Uniform, and a Beta distribution.

Start Program LC. Type 'Run LC' and carriage return (CR). LC.EXE will run and the first prompt will appear. (Note: LC.EXE has some Y/N prompts. The program is expecting upper case Y or N responses; although, it may operate using lower case on some machines.)

```
** MAIN MENU **
                                                    (1)
1. READ DATA FILE.
2. CREATE NEW DATA FILE.
3. VIEW/EDIT DATA.
4. EXECUTE LIFE CYCLE COST.
5. EXIT.

ENTER SELECTION (1 - [5]):
```

Numbers and characters appearing in parenthesis indicate that you may only select from those options. Thus (1 - [5]) indicates that you may select a number from one to five. Any other number will return you to the same prompt. Numbers located in brackets as [5] imply that a CR with no other input will default to the bracketed number. thus a CR alone from the main menu will terminate the program. (Y/[N]) indicate to select 'Y' or 'y' for a yes answer, and 'N' or 'n' for a no answer followed by a CR. A CR alone or anything other than a Y would be understood as a No.

Once you have reached the Main Menu, your next step is to read-in a data file or create a data file. Selecting '3' or

'4' from 'MAIN MENU' when no data is in the program will return the program to the same prompt.

READ A DATA FILE. Selecting '1' and CR from the main menu will invoke the 'READ' subroutine. A listing of available files from the directory in which LC.EXE resides will be presented. All data files pertinent to this program will end with the suffix '_LC.DAT'. To select a data file, enter the data file name excluding the suffix. If the read function was not successful, an 'Unable to open file' followed by the indicated file name message will be displayed. Otherwise, the program will return to the main menu prompt. Any file pertinent to LC.EXE may be read into the program from the 'Main Menu' at any time. Appending of files through 'READ' is not possible. Also, appending of files through VMS commands will be futile.

CREATE A DATA FILE. Selecting '2' from 'MAIN MENU' will invoke the 'CREATE' subroutine. At this time you should be ready to enter data. The following is the sequence of inputs to the program: the systems life in years, the discount rate to be used in discounting future payments to present value, the number of CCEs, the number of SCEs, the number of CERs. If zero was entered for the number of CCEs, the number of

SCEs, and the number of CERs, then the program will return to the 'Main Menu'.

CREATE CCEs. For number of CCEs equal one or more, the program will prompt to enter all CCEs. The first CCE prompt is:

```
CONSTANT COST ELEMENT (CCE) #
ENTER COST:                                              (2)
```

Enter the cost of the first CCE followed by CR. The second prompt regarding the CCE is the cash allocation (dispersement).

Cost Allocation. Cost allocation can be based on a trapezoid or a percentage for a number of years:

```
CONSTANT COST ELEMENT #                          (3)
1. ALLOCATE COST BASED ON A PERCENTAGE.
2. ALLOCATE COST BASED ON A TRAPEZOID.

ENTER (1 - [2]):
```

Allocation Cost Based on a Percentage. 1 and CR will invoke prompts for the allocation of cost based on a percentile distribution. The first prompt for an allocation based on a percentile is:

(4)

ENTER THE TOTAL NUMBER OF PAYMENTS FOR CCE # (1 - Nyrs):

At this time enter the number of payments associated with CCE 1. Nyrs is the maximum that can be entered and 1 is the minimum. Nyrs is the life of the system. Thus, the number of payments can not exceed the system life. The next set of prompts are:

ENTER YEAR PAYMENT 1 IS MADE FOR CCE #: (5)

At this time enter the year the first payment for CCE # is made. The next prompt is:

(6)

ENTER PERCENTAGE OF COST PAID AT END OF YEAR 1 FOR CCE 1 (%):

At this time enter the percentage of cost for CCE 1 to be paid at the end of year 1 in percent (i.e. 25 for 25%). The last two prompts, prompts 4 and 5, will repeat for as many time as the number of payments. If the input total is not equal to 100.0 percent, then a error message will appear and you will be required to re-enter the data regarding the percentages.

Allocation Cost Based on a Trapezoid. 2 and CR will invoke prompts for the allocation of cost based on a trapezoid distribution. The first prompt for an allocation based on a trapezoid is:

ENTER THE PHASE-IN PERIOD FOR CCE 1: (7)

Enter the period in whole years that the allocation of cost is being phased in or increasing. This is indicated by 'A' in figure 3. The next prompt is:

ENTER THE CONSTANT-COST PERIOD FOR CCE 1: (8)

Enter the period in whole years in which the allocation of cost is constant. This is indicated by 'B' in figure 3. The next prompt is:

ENTER THE PHASE-OUT PERIOD FOR CCE 1: (9)

Enter the period in whole years in which the allocation of cost is being phased-out or decreasing. This is indicated by 'C' in figure 3. The next prompt is:

ENTER THE YEAR PAYMENTS START FOR CCE 1: (10)

Enter the year payments start for CCE 1 in whole years. This is the number of years until the phase-in period starts. Thus, the first cost is imputed to the next year after the number entered for (10). (ie. If you enter 3 for prompt (10) above, then the first cost of this component is assigned to year 4).

Once the data entry is complete for CCE 1, the program will go through the same sequence of prompts for the remaining CCEs. After All CCE data has been entered, the program will continue and prompt for SCEs, given that the number of SCEs entered is one or more.

<u>CREATE SCEs.</u> For the number of SCEs equal to one or more, the program will prompt to enter all SCEs. The first SCE prompt is:

```
STOCHASTIC COST ELEMENT (SCE) #                         (11)

1. TRIANGULAR DISTRIBUTION.
2. BETA DISTRIBUTION

ENTER DISTRIBUTION SELECTION ([1] - 2):
```

The prompt is asking whether the SCE is to be modeled by a triangular distribution or a Beta distribution. Enter 1 and CR to select a triangular distribution, or 2 and CR to select a Beta distribution.

Triangular Distribution. 1 and CR will invoke the prompts to enter values to describe a triangular distribution. The first prompt in entering values for a triangular distribution is:

ENTER MODE (best guess) OF TRIANG. DIST. SCE #: (12)

At this time enter the mode of the SCE. The mode is the your best guess as to what the actual cost will be. The next prompt for the triangular distribution is:

ENTER LOW VALUE OF TRIANG. DIST. SCE #: (13)

At this time enter the least that you think the price can be. The next prompt for the triangular distribution is:

ENTER HIGH VALUE FOR TRIANG. DIST. SCE #: (14)

At this time enter the most that you think the price will be. This concludes all entries for a triangular distribution.

Beta Distribution. 2 and CR will invoke the prompts to enter values to determine the Beta type, the least possible cost, and the most possible cost. The first prompt in entering values for a Beta distribution is:

ENTER BETA DIST. TYPE (1 - 9) FOR SCE #: (15)

At this time you are required to enter the beta type from which sampling for this cost element will be made. Reference figure 4 for the nine beta types available in this program. Beta types that symmetric indicate that the expect cost lies in the center of the distribution. Beta types that are skewed left or right have the expected cost closer to the most the price can be or the least the price can be, respectively. The beta distribution shown at the top of figure 4 have larger variance than those shown at the bottom. Smaller variance indicates that the price is more likely to be about the expected value, and the reverse for a larger variance. Therefore, enter a integer number from 1 to 9 to select the beta type most representing the uncertainty associated with the SCE. The next prompt for the Beta distribution is:

ENTER THE LOW VALUE FOR BETA DIST. SCE #: (16)

At this time enter the least amount that you expect the price of this SCE to be. The next prompt for the Beta distribution is:

ENTER THE HIGH VALUE FOR THE BETA DIST. SCE #: (17)

Enter the most you expect the price of this SCE to be. This concludes entries for the Beta distributions.

After the cost distribution for the SCE is entered, entries regarding the allocation of cost is required. The same prompts used for entering the allocation of cost for a CCE is also used for the allocation of cost for a SCE. Therefore, reference 'CCE Allocation of Cost Based on a Percentage' (prompts 4 through 6) and reference 'CCE Allocation of Cost Base on a Trapezoid' (prompts 7 through 10) when entering the cost allocation type and data for a SCE. After all SCE data are entered, the program will continue to prompt for CERs if the number of CERs entered is one or more.

CREATE CERs. For the number of CERs equal to one or more, the program will prompt to enter all CERs. The first CER prompt is:

```
*** CER # SCALAR MULTIPLE ***                         (18)
1. SCALAR MULTIPLE > 1
ENTER ([0] - 1):
```

This prompt is asking you whether you are estimating average cost per unit and, in which case, you need to multiply by the number of units such that the estimate is of total cost. Thus, if the CER you are about to enter represent an average cost, enter 1 and CR. On the other hand, if the CER you are about to enter represents total cost of system or subsystem, enter 0 and CR.

Scalar Multiple. The CER is assumed to represent average cost and the number of units must be entered such that total cost can be computed. The scalar to be entered for the number of units can be represented by a fixed value, a triangular distribution, or a beta distribution. The prompt for the entry of the scalar is:

```
TYPE OF SCALAR (SCL) ESTIMATION.                        (19)
1. TRIANGULAR DISTRIBUTION.
2. BETA DISTRIBUTION.
3. FIXED VALUE.
ENTER (1 - [3]):
```

At this time you must enter 1 for a scalar represented by a triangular distribution, 2 for a scaler represented by a Beta distribution, or 3 for a scaler that is a fixed value and CR.

<u>Scalar Triangular Distribution.</u> See prompts 12 through 14 for examples of prompts for triangular distribution entries.

<u>Scalar Beta Distribution.</u> See prompts (15) through (17) in the SCE section for examples of prompts for Beta distribution entries.

<u>Scalar Fixed Value.</u> Having selected 3 for the scalar type, the program will ask for the scalar value with the following prompt:

```
ENTER FIXED SCALAR VALUE FOR CER #:                     (20)
```

Enter an integer representing the number of units to be purchased such that when multiplied with the CER the total cost is determined.

Number of Slope Parameters. Once the scalar data is entered, the program will prompt for the number of slope parameters. the prompt for the number of slope parameters is:

```
ENTER THE NUMBER OF SLOPE PARAMETERS (1 - #):        (21)
```

The number of slope parameters is the number of parameters associated with the drivers. It is the same number as the number of drivers and it does not include the intercept. An integer number is required for the number of slope parameters.

Base. This program allows for the estimation of CERs in base 10 (whole numbers), logarithm of base 10 and logarithm of base 'e' (natural log). The next prompt provides for the selection between the different bases:

```
** CER # IS ESTIMATED IN **                           (22)
1. WHOLE DOLLARS.
2. LOG BASE E (Ln).
3. LOG BASE 10 (Log).

ENTER ([1] - 3):
```

An input of 1, 2, or 3 will determine the proper type of computation to be accomplished in the EXEC phase. See Chapter III computation of CERs using base '10', logarithm base 'e', and logarithm base '10. Once a base is selected there is no difference in the prompts. The next question is regarding the sample size from which the least square regression was accomplished.

Sample Size. To enter the sample size associated with this CER the following prompt is displayed:

```
                                                             (23)
ENTER THE NUMBER OF OBSERVATIONS FROM WHICH CER # WAS OBTAINED
([0] IF UNKNOWN, 1 - 999):
```

The CER being entered was derived using least square regression from a number of observations. Enter this number. 1 is the minimum and 999 is the maximum that can be entered. If the number of observations is not known, enter 0 and CR. A 0 entry will imply that a sample from a Normal(0,1) distribution will be used in place of a t(DF) distribution in computing cost or average cost. This will bias the estimation is the DF is less than thirty. After the sample size is entered, the program asks for the value of the CER's intercept term.

Intercept. The next prompt is asking for the intercept value (b_0).

(24)

ENTER VALUE OF CER # INTERCEPT TERM ([0] if none):

The intercept (b_0) represents a fixed cost that may represent research and development. b_0 is usually determined from the same least square regression as does the parameters of the CER. In some cases b_0 is less than or equal to zero.

Variance. The next prompt asks for the variance of the CER.

(25)

ENTER THE ESTIMATED VARIANCE (S-squared) FOR CER #:

Variance is the error sum of squares (or the sum of squared error) divided by DF. This provides for an unbiased variance.

Drivers. The next set of prompts is to collect information on the drivers. There is as many drivers as there are slope parameters. Each driver can be represented by a constant value, and triangular distribution, or a beta distribution. The following prompt is for the distribution of the driver.

```
DISTRIBUTION OF DRIVER (DRV) #                        (26)
1. CONSTANT VALUE
2. TRIANGULAR DISTRIBUTION
3. BETA DISTRIBUTION
ENTER ([1] - 3):
```

At this time enter 1, 2, or 3 depending on the distribution of the driver being entered.

Constant Driver Value. The following prompt is used to enter a constant driver value.

```
ENTER VALUE OF CONSTANT DRIVER CER #:                 (27)
```

A real number is required for driver values. therefore, enter the constant value and CR.

Driver Triangular Distribution. See prompts 12 through 14 in the SCE section for examples of prompts for triangular distribution entries.

Driver Beta Distribution. See prompts 15 through 17 in the SCE section for examples of prompts for Beta distribution entries.

Slope Parameters. Once all driver information in entered, the next set of prompts is for the entry of parameter. All parameters are entered as a fixed value. The prompt for parameter entry is:

ENTER VALUE OF SLOPE PARAMETER #: (28)

At this time enter the value of slope parameter #. This is a real number. This prompt repeats as many times as the number of slope parameters. The next required entry is the Variance/Covariance matrix.

Variance/Covariance Matrix. The variance/covariance (var/cov) matrix contains the variance of each parameter, including the intercept if on exists, and the covariance between parameters. Since this matrix is symmetric about the diagonal, only the diagonal and one of its triangles are required to be entered. In the var/cov matrix position (1,1) holds the variance of the intercept term b_0. Position (i,i) holds the variance of parameter b_i for i = 2, (P+1) (P is the number of slope parameters). Positions (i,j) , j = 1, (P+1), hold the covariance between all parameter including b_0. If b_0 is zero then positions (1,1), (1,j), and (j,1) are all set to zero and the prompt would begin at position (2,2). The prompt for entering the var/cov matrix is:

(29)

PARAMETER VARIANCE/COVARIANCE MATRIX POSITION (i,j).
ENTER VARIANCE/COVARIANCE:

A real number input is required. If the CER's base is logarithm base 'e' or logarithm base '10', then the entered var/cov values must also be in logarithm base 'e' or logarithm base '10', respectively. Once the var/cov matrix is entered all that remains to be entered is the allocation of the cost. As in CCE and SCE, cost of a CER can be allocated based on a percentage or based on a trapezoid. Thus, the next prompt for the CER is for the allocation of cost over time. An example of the prompt and required entries are available in the 'CREATE CCE' subsection 'Allocation of Cost Based on a Percentage' and subsection 'Allocation of Cost Based on a Trapezoid'. (See prompts 12 - 14, and 15 - 18). Once the data entry is complete for CER 1, the program will go through the same sequence of prompts for the remaining CERs. After all CER data has been entered, the program will have completed the 'CREATE' sequence.

Save File as. After all data entry is complete, a prompt requesting verification of the file name in which the data is to be saved. The prompt is:

SAVE FILE AS 'Filename' ? ([Y]/N]): (30)

A simple CR, a 'Y' CR, or a 'y' CR entry implies you agree with the shown file name. The program will continue to save the file and return to the main menu. However, a 'N' CR or 'n' CR entry will imply that a different name is desired. The following prompt will appear requesting a new file name:

ENTER THE DATA FILE NAME (_LC.DAT understood): ________ (31)

The program will accept up to an eight character string representing the data file's name. Any additional characters will not be read. _LC.DAT will be appended to your designated name when the data file is opened. The program will continue to save the data and return to the main menu.

Although the data has been stored, it still exist in the program. From the main menu, the user may read in new data from a data file or create a new data file. In either of these cases any existing data in the program will sees to exist. The user may chose to view/edit the data in the program or run the executable algorithm on the data. In either of the last two choices, the original data remains in the program. The last option is the user may simply exit.

VIEW/EDIT. This portion of the program allows the user to view and edit all cost elements in the program. Before entering 'View/Edit' a data file must exist in the program. This can only be done through 'READ A DATA FILE' or 'CREATE A DATA FILE'. If there is no data in the program, selecting 'VIEW/EDIT A DATA FILE' will cause a message to appear on the screen and the program to return the user to the main menu. After the user is satisfied with the data in the program, he will be given a chance to save the data into a file under the original file name or under a new file name. Under no circumstance does this program delete any files. All files are closed once read. The first prompt that appears when selecting 'View/Edit a Data File' is:

```
** VIEW/EDIT MAIN MENU **                                    (32)

1. CCE.      #
2. SCE.      #
3. CER.      #
4. SYSTEM LIFE (yrs):  #
5. DISCOUNT RATE (%):  #
6. EXIT MENU.

ENTER (1 - [6]):
```

Prompted are the number of CCEs, the number of SCEs, the number of CERs, the systems life, and the discount rate. Selecting 1, 2, or 3 will lead to viewing and editing CCEs, SCEs, or CERs, respectively. Selecting 4 will allow the user to edit the system's life. While, selecting 5 will allow the user to change the discount rate.

System Life. The prompts for the system life is:

(33)

ENTER NEW SYSTEM LIFE (Mn # - Mx # [Old Yrs]):

Shown in parenthesis are the minimum acceptable system life (Mn #), the maximum acceptable system life (Mx #), and the current system life. The minimum number of years is derived from all the data pertinent to this file. The rule is the system can not have a life smaller than the period associated with the allocation of any cost element. The maximum number of years is fixed internally to 999 and requires the editing of the parameter 'NY' in the source code to change it. 'Old Yrs' is the current system life. Enter a new system life (integer) and CR, or CR and the old system life remains. After entering the system life, the program will check for the upper and lower bounds than return to the main view/edit menu.

Discount Rate. The prompt for the discount rate is:

```
ENTER THE DISCOUNT RATE (%):                                  (34)
```

At this time enter the new discount rate and CR. The program will check if the new discount rate is between 0.0 and 100.0 and will return to the view/edit menu. This assumes that we will never have hyper inflation. If we do, interest rates can go higher than 100 percent. Note also: if all cost data are in real dollars, then use a "real" discount rate. If, on the other hand, constant dollars are used in the cost data, then inflation rate should be used for discounting.

CCE. To arrive here you have selected '1' from VIEW/EDIT MAIN MENU. In this section you will be allowed to view and, if you desire, edit all CCE data. You may also add and delete a CCE from this section. The menu that drives this section is:

```
** CCE VIEW/EDIT MENU **                           (35)

1. VIEW/EDIT A CCE.
2. ADD A CCE.
3. DELETE A CCE.
4. EXIT THIS MENU.

ENTER (1 - [4]):
```

VIEW/EDIT A CCE. To view and edit existing data you must select '1' and CR from the above menu. This will display all CCEs on the program sequentially and will allow you to edit the values upon making a numerical selection. Adding or deleting a CCE is not possible through the '1' selection. The displays for CCE is as follows:

```
            CONSTANT COST ELEMENT  #                         (36)
            *************************

            1. COST:                   ####
                                          (percentage allocation)
            2. NUMBER OF PAYMENTS:     #

            PAYMENT   YEAR   PERCENT
            -------   ----   -------

               1        #       #
               2        #       #
               3        #       #
               .        .       .
               .        .       .
               etc..
(or)                                       (trapezoid allocation)
            2. TRAPEZOID ALLOCATION:

               PHASE-IN PERIOD:        #
               CONSTANT PERIOD:        #
               PHASE-OUT PERIOD:       #
               YEAR PAYMENT START:     #

            ENTER ( 1 - 2, [NEXT]]):
```

At this time data of CCE is displayed for your inspection. You may select '1' if you wish to change the cost of the CCE, or you may select '2' if you wish to change the allocation of the cost associated with the CCE displayed. Selecting '1' will invoke a prompt (see prompt number 2) to enter the new cost value. Selecting '2' will invoke a prompt to select from the percentage allocation of cost or the trapezoid allocation of cost (see prompt number 3). Once selection of the allocation of cost is made, input for the allocation will be required. See prompts 4 through 6 for allocation of cost based on percentages, and see prompts 7 through 10 for allocation of cost based on a trapezoid.

Add a CCE. To add a CCE you must select '2' from the 'CCE VIEW/EDIT MENU'. Upon selecting '2' the program will cycle through all the same prompts used had you selected 'CREATE A DATA FILE' and is entering data for a CCE. Thus, for adding a CCE through the above menu, reference 'CREATE CCEs' under 'CREATE A DATA FILE'. (See prompts 2 through 10).

Delete a CCE. To delete a CCE you must select '3' from 'CCE VIEW/EDIT MENU'. Upon selecting '3' the program will prompt the following:

(37)

ENTER THE NUMBER OF THE CCE TO DELETE (1 - #, [EXIT]):

At this time you must enter a CCE number within the limits displayed in the prompt. To exit this prompt with no CCE deleted simply enter a CR. However, once a valid number has been entered, the following prompt will appear for confirmation:

(38)

ARE YOU SURE YOU WANT TO DELETE CCE #? (Y/[N]):

If you are sure of this deletion, then enter 'Y' or 'y' and the CCE will be deleted. Otherwise, enter CR or 'N' or 'n' and no deletion will occur. In any case, the program will return to 'CCE VIEW/EDIT MENU'.

SCE. To arrive here you have selected '2' from the VIEW/EDIT MAIN MENU. In this section you will be allowed to view and, if you desire, edit all SCE data. You may also add and delete a SCE from this section. The menu that drives this section is:

```
** SCE VIEW/EDIT MENU **                    (39)

1. VIEW/EDIT A SCE.
2. ADD A SCE.
3. DELETE A SCE.
4. EXIT THIS MENU.

ENTER (1 - [4]):
```

VIEW/EDIT A SCE. To view and edit existing data you must select '1' from the above menu. This will display all SCEs on the program sequentially and will allow you to edit the values upon making a numerical selection. Adding or deleting a SCE is not possible through the '1' selection. The displays for SCE is as follows:

```
          STOCHASTIC COST ELEMENT    #                      (40)
          ****************************

          1. DISTRIBUTION:            TRIANGULAR
             MODE VALUE:              #######
(or)
          1. DISTRIBUTION:            BETA
             BETA TYPE:               #
(and)
             LOWER LIMIT:             #######
             UPPER LIMIT:             #######

                                     (percentage allocation)
          2. NUMBER OF PAYMENTS:  #

          PAYMENT   YEAR   PERCENT
          -------   ----   -------

             1        #       #
             2        #       #
             3        #       #
             .        .       .
             .        .       .
             etc..

(or)                                  (trapezoid allocation)
          2. TRAPEZOID ALLOCATION:

             PHASE-IN PERIOD:     #
             CONSTANT PERIOD:     #
             PHASE-OUT PERIOD:    #
             YEAR PAYMENT START:  #

          ENTER ( 1 - 2, [NEXT]]):
```

At this time data of the SCE is displayed for your inspection. You may select '1' if you wish to change the cost distribution of the SCE, or you may select '2' if you wish to change the allocation of the cost associated with the SCE displayed. Selecting '1' will invoke a prompt (see prompt 11) to enter the new cost distribution type. For a triangular distribution prompts 12 through 14 will be displayed. For a beta distribution, prompts 15 through 17 will be displayed. Selecting '2' will invoke a prompt to select from the percentage allocation of cost or the trapezoid allocation of cost (see prompt 3). Once selection of the allocation of cost is made, input for the allocation type will be required. (See prompts 4 through 6 for allocation of cost based on percentages, and see prompts 7 through 10 for allocation of cost based on a trapezoid).

Add a SCE. To add a SCE you must select '2' from the 'SCE VIEW/EDIT MENU'. Upon selecting '2' the program will cycle through all the same prompts used had you selected 'CREATE A DATA FILE' and is entering data for a SCE. Thus, for adding a SCE through the above menu, reference 'CREATE SCEs' under 'CREATE A DATA FILE'. (See prompts 11 - 17 and 7 - 10).

Delete a SCE. To delete a SCE you must select '3' from the 'SCE VIEW/EDIT MENU'. Upon selecting '3', the program will display prompt 37. At this time you must enter a SCE number within the limits shown in prompt 37. To exit this prompt with no SCE deleted simply enter a CR. However, once a valid number has been entered, the following prompt 38 will appear for confirmation. If you are sure of this deletion, then enter 'Y' or 'y' and the SCE will be deleted. Otherwise, enter CR or 'N' or 'n' and no deletion will occur. In any case, the program will return to 'SCE VIEW/EDIT MENU'.

CER. To arrive here you have selected '3' from the VIEW/EDIT MAIN MENU. In this section you will be allowed to view and, if you desire, edit all CER data. You may also add and delete a CER from this section. The menu that drives this section is:

```
** CER VIEW/EDIT MENU **                              (41)

1. VIEW/EDIT A CER.
2. ADD A CER.
3. DELETE A CER.
4. EXIT THIS MENU.

ENTER (1 - [4]):
```

VIEW/EDIT A CER. To view and edit existing data you must select '1' from the above menu. This will display all CERs on the program sequentially and will allow you to edit the values upon making a numerical selection. Adding or deleting a CER is not possible through the '1' selection. The displays for CER is as follows:

```
               COST ESTIMATING RELATIONSHIP  #                  (42)
               ********************************

               1. SCALAR VALUE:    1: FIXED
(or)
               1. SCALAR DISTRIBUTION:   TRIANGULAR
                  MODE VALUE:             #########
(or)
               1. SCALAR DISTRIBUTION:   BETA
                  BETA TYPE:              #
(and for triangular and beta only)
                  LOW VALUE:              #########
                  HIGH VALUE:             #########
(or)
               1. SCALER DISTRIBUTION:   CONSTANT
                  SCALER VALUE:           #########

               2. VIEW/EDIT COST DRIVERS & SLOPE PARAM.: #
               3. ESTIMATION BASED ON:                   WHOLE DOLLAR
(or)                                                      NATURAL LOG
(or)                                                      LOG BASE 10
               4. SAMPLE SIZE (n):                        #
               5. INTERCEPT PARAM. (b0):                  ##########
               6. VARIANCE (Ssq):                         ##########
               7. VIEW/EDIT COST ALLOCATION (TRAP/PRCNT).
               8. EXIT THIS MENU.

               ENTER (1 - 8, [NEXT]):
```

At this time data of the CER is displayed for your inspection. You may select '1' to change the scaler multiple data, '2' to view and edit driver and parameter data, '3' to change the base of the data, '4' to change the sample size, '5' to change the value of the intercept parameter, '6' to change the estimated variance, '7' to view and edit the payment schedule, or '8' to return to the last menu. A simple CR will bring up the data of the next CER if one exists.

Edit Scalar Data. you have selected '1' from the above menu. To edit the scaler multiple data, the program will retrace the prompts used in the 'CREATE A DATA FILE' where the scaler data is entered. Thus, to gain information on the entry requirements and prompts, reference the 'Scaler Multiple' section under 'CREATE CER' and prompts 18 through 20, 12 through 14, and 15 through 17.

View/Edit Cost Drivers & Slope Parameters. You have selected '2' from the above menu. The following is an example of the display for the cost drivers and slope parameters:

DRIVER INFORMATION FOR CER # (43)

	SLOPE	CONST	LW-BND	UP-BND	MODE	BETA
1	####	#####				
2	####		######	######	######	
3	####		######	######		#

1. CHANGE A COST DRIVER ONLY.
2. CHANGE A SLOPE PARAMETER ONLY.

ENTER (1 - 2, [EXIT]):

Prompt 43 an example of how the three type of possible drivers are displayed with their slope parameters. At line 1 is an example of a 'constant' driver. At line 2 is an example of a driver that has a triangular distribution. And at line 3 is an example of a driver that has a beta distribution. A user has the option to change the value as well as the distribution of the drivers, and change the value of the slope parameter. The user does not have the option to delete or add a driver or a slope parameter as this requires the reassessment of the var/cov matrix and sample variance.

Change a Cost Driver. You have selected option '1' from prompt 43. The distribution and the values of the cost driver can be changed. The first prompt is:

(44)
ENTER THE NUMBER OF THE COST DRIVER TO CHANGE (1 - #, [EXIT]):

You must enter the line number of the cost driver. The line numbers are the numbers on the far left side in prompt 43. Once the line number has been entered, the program will cycle all driver related prompts used in the 'CREATE CERs' routine. Thus reference 'CREATE CERs' subsection 'Driver' and prompts 26, 27, 12 through 14, and 15 through 17. After the new driver data in entered, the program returns to prompt 43.

Change a Slope Parameter. You have selected '2' from prompt 43. The value of the slope parameter can be changed through this selection. The first prompt is:

ENTER LINE NUMBER OF SLOPE PARAM. TO CHANGE (1 - #, [EXIT]): (45)

You must enter the line number located at the far left in prompt 43 of the slope parameter you desire to change. Once a valid line number is entered, the following prompt will appear:

ENTER NEW SLOPE PARAM. # VALUE: (46)

You must enter a none zero value for the new slope parameter value. A zero value is no acceptable since it would alter the driver effect resulting in invalidating var/cov and sample variance data. After the new slope parameter is entered, the program returns to prompt 43.

Change Base of the DATA. You have selected '3' from prompt 42. This selection is to change the base of the data. It is not advisable to change the base of your data unless an error was made while creating the CER data originally. You should see a message to that effect on the screen if you have selected '3'. The following prompt is for your confirmation:

** DO YOU STILL WISH TO CHANGE THE BASE? (Y/N): (47)

If an error was made in the original CER data and you wish to change the base, then enter a 'y' or a 'Y' to confirm. A prompt will follow to select the base of this CER. See 'CREATE CERs' subsection 'Base' and prompt 22 regarding changing the base of the CER data.

Change the Sample Size. You have selected '4' from prompt 43. The following prompt will appear to change the sample size:

ENTER A NEW SAMPLE SIZE FOR CER # [OLD: ##]: (48)

Enter the sample size from which the CER was determined. If you wish to keep the 'old' sample size, simply enter CR. The program will then return to prompt 43.

Change the Intercept Term. You have selected '5' from prompt 43. The following prompt will appear to change the value of the intercept term:

ENTER THE VALUE OF INTERCEPT TERM FOR CER # ([0] IF NONE): (49)

Enter the value of the new intercept term or zero if none exists. A value less than zero will not be accepted. If the intercept term was changed to zero, then the intercept variance and covariance terms in the var/cov matrix will zeroed also. The program will then return to prompt 43.

Change the Variance. You have selected '6' from prompt 43. The following prompt will appear for you to enter the new sample variance:

ENTER NEW SAMPLE VARIANCE FOR CER #: (50)

Enter the new sample variance at this time. The program will then return to prompt 43.

Note: you could also change the variance/covariance matrix by editing the ASCii file filename_LC.DAT.

Change Cost Allocation. You have selected '7' from prompt 43. This selection will display with the current cost allocation whether it is based on a trapezoid or a percentage. At the bottom of the display is an 'ENTER' command from which you may select to re-enter the cost allocation. The new cost allocation may be either based on a trapezoid or a percent. The display of the cost allocation is as in prompt 40 line 2. However, the 'ENTER' line is different. it is:

ENTER (1 = NEW COST ALLOCATION, [NEXT]): (51)

A CR here will return you to prompt 43 with no change in the cost allocation data. However, a '1' entry will evoke the cycle of entering the cost allocation data for the CER. See 'CREATE CCE' subsection 'Cost Allocation'. An example of the prompt and required entries are also available in the 'CREATE CCE' subsection 'Allocation of Cost Based on a Percentage' and subsection 'Allocation of Cost Based on a Trapezoid'. After entering of the CER cost allocation data, the program will return to prompt 43. This concludes all possible editing options for the CERs.

ADD A CER. You have selected '2' from 'CER VIEW/EDIT MENU'. Upon selecting '2' the program will cycle through all the same prompts used had you selected 'CREATE A DATA FILE' and is entering data for a CER. Thus, for adding a CER, reference 'CREATE A DATA FILE' subsection 'CER'. (See prompts 18 - 29, 12 - 14, and 15 - 18).

DELETE A CER. You have selected '3' from 'CER VIEW/EDIT MENU'. Upon selecting '3', the program will display prompt 37. At this time you must enter a CER number within the limits shown in prompt 37. To exit this prompt with no CER deleted simply enter a CR. However, once a valid number has been entered, prompt 38 will appear for confirmation. If you are sure of this deletion, then enter 'Y' or 'y' and the CER will be deleted. Otherwise, enter CR, 'N' CR, or 'n' CR and no deletion will occur. In any case, the program will return to 'CER VIEW/EDIT MENU'.

EXECUTE LIFE CYCLE COST. You have selected '4' from 'MAIN MENU'. Data must exist in the program for the program to execute the algorithm. If data has not been read or created in the program, then you will be returned to 'MAIN MENU' with a no data message. If data does exist in the program, then you will be asked for the number of random variables to be generated. The number of random variables (NRV) is the number of times the program will execute the data to provide different confidence levels of the total cost. Currently nrv is limited to 800 in the program. This may be changed through editing of the source code (See source parameter editing in chapter IV). The following prompt is for the NRV:

```
                                                                  (52)
     ENTER THE NUMBER OF RANDOM VARIABLES DESIRED (Max: ##):
```

The larger the NRV is the better the representation of the actual cost will be. Upon entering the NRV desired, the program will execute the algorithm. After completion of the execution, the program will display end-information regarding the cost in the following format:

```
   !!! THE OUT-FILE NAME IS: ..._LC.OUT  !!!                      (53)
   !!! THE SAS-FILE NAME IS: ..._SAS.DAT !!!
  ## YEARS @ ##.# % DISCOUNT RATE.
  % CONFIDENCE       ANNUAL COST               TOTAL COST
  ------------       -----------               ----------
       85            ###########.           #############.
       90            ###########.          ##############.
       95           ############.          ##############.
       99           ############.         ###############.

  MINIMUM POSSIBLE                          #############.
  MAXIMUM POSSIBLE                        ###############.

  [ENTER RETURN TO CONTINUE]:
```

As you can tell from the above display, two out-files are created. The ...LC_OUT file contains the same information as displayed above. The ..._SAS.DAT file contains all the sample annual cost data from which the confidence data was obtained. This last file is for the convenience of the user if he or she wishes to accomplish SAS type analysis on the sample costs. Once you are finished observing the displayed data, enter CR to return to 'MAIN MENU'.

Appendix B

Source Code

```
***************************************************************
*  LC is a Life Cycle Cost model design to calculate cost based on*
*  Constant Cost Elements (CCE), Stochastic Cost Elements (SCE),  *
*  and Cost Estimating Relationships (CER).                       *
***************************************************************
*  By: Habash, Nicolas M., Capt,. U.S.A.F. (MAY 1984 - SEP 1992)  *
***************************************************************
      PROGRAM LC

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*3 CET
      CHARACTER*1 RESP

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

*      NC........UPPER LIMIT FOR EACH OF NUMBER OF CCEs, SCEs, AND*
*                CERs. (CHANGE WHAT NC IS SET TO IN ALL PARAMETER *
*                STATEMENTS THROUGHOUT THE PROGRAM (INCLUDING     *
*                SUBROUTINES) TO CHANGE THE NUMBER OF CCEs, SCEs, *
*                AND CERs).                                      *
*     ND........UPPER LIMIT FOR THE NUMBER OF DRIVERS AND SLOPE  *
*                PARAMETERS. (CHANGE WHAT 'ND' IS SET TO IN ALL   *
*                 PARAMETER STATEMENTS THROUGHOUT THE PROGRAM     *
*                (INCLUDING SUBROUTINES) TO CHANGE THE MAXIMUM    *
*                NUMBER OF ALLOWABLE COST DRIVERS AND SLOPE      *
*                PARAMETERS FOR CER'S.                           *
*     NY........UPPER LIMIT FOR THE SYSTEM LIFE IN YEARS.        *
*                 (CHANGE WHAT NY IS SET TO IN ALL PARAMETER      *
*                STATEMENTS THROUGHOUT THE PROGRAM (INCLUDING     *
*                SUBROUTINES) TO CHANGE THE MAXIMUM ALLOWABLE    *
*                SYSTEM LIFE).                                   *
*      NV........UPPER LIMIT FOR THE NUMBER OF COST ESTIMATES     *
*                COMPUTED.  (CHANGE WHAT NV IS SET TO IN ALL      *
*                PARAMETER STATEMENTS THROUGHOUT THE PROGRAM      *
*                (INCLUDING SUBROUTINES) TO CHANGE THE MAXIMUM   *
*                ALLOWABLE NUMBER OF COST ESTIMATES COMPUTED).   *
***************************************************************
      INTEGER   NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
```

```
      INTEGER    CCENP(NC), SCEYR(NC,NY), SCENP(NC)
      INTEGER    CERNP(NC), CERYR(NC,NY), OUT, INN
*
*
*     IRPLY      INTEGER TYPE USER REPLY                             *
*     NCCE       NUMBER OF COST ESTIMATORS THAT ARE CONSTANT         *
*     NSCE       NUMBER OF COST ESTIMATORS THAT ARE STOCHASTIC       *
*     NCER       NUMBER OF COST ESTIMATORS THAT ARE COST             *
*                 ESTIMATING RELATIONSHIPS                            *
*     NYRS       NUMBER OF YEARS OF THE SYSTEM                       *
*     ...YR( , ) YEAR PAYMENT IS MADE FOR PERCENTILE BASED           *
*                PAYMENTS                                            *
*     ...NP( )   NUMBER OF PAYMENTS TO BE MADE FOR PERCENTILE        *
*                BASED PAYMENTS                                      *
*     OUT        HOLDS THE OUTPUT TERMINAL NUMBER OBTAINED FROM      *
*                IMSL ROUTINE 'UMACH'.                               *
*     INN        HOLDS THE INPUT TERMINAL NUMBER.                    *
*****************************************************************
      REAL       CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^           RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^           CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY)
*
*     DSC        DISCOUNT RATE USED FOR PRESENT VALUE                *
*     RRPLY      REAL TYPE USER REPLY                                *
*                                                                    *
*     CONSTANT COST ELEMENTS                                         *
*     CCE( ,1)   CCE PHASE-IN PERIOD                                 *
*        ( ,2)   CCE CONSTANT COST PERIOD                            *
*        ( ,3)   CCE PHASE-OUT PERIOD                                *
*        ( ,4)   CCE YEAR PAYMENTS BEGIN                             *
*        ( ,5)   VALUE OF THE CONSTANT COST ESTIMATOR                *
*        ( ,6) 1 = COST IS ALLOCATED BY PERCENTILE BASIS             *
*              2 = COST IS ALLOCATED BY TRAPEZOID BASIS              *
*                                                                    *
*     FOR CCE( ,6) = 1                                               *
*     CCENP( )   CCE NUMBER OF PAYMENTS                              *
*     CCEYR( , ) YR A PAYMENT IS MADE FOR THE CCE                    *
*     CCEPR( , ) PERCENT OF COST PAYED AT YEAR CCEYR( , )            *
*                                                                    *
*  STOCHASTIC COST ELEMENTS                                          *
*     SCE( ,1)   SCE PHASE-IN PERIOD                                 *
*        ( ,2)   SCE CONSTANT COST PERIOD                            *
*        ( ,3)   SCE PHASE-OUT PERIOD                                *
*        ( ,4)   SCE YEAR PAYMENTS BEGIN                             *
*        ( ,5)   1 = TRIANGULAR DIST, 2 = BETA DIST                  *
*        ( ,6)   LOW VALUE                                           *
*        ( ,7)   HIGH VALUE                                          *
*        ( ,8)   MODE VALUE IF SCE( ,5) = 1                          *
*                BETA TYPE  IF SCE( ,5) = 2                          *
```

```
*         ( ,9) 1 = COST IS ALLOCATED BY PERCENTILE BASIS          *
*               2 = COST IS ALLOCATED BY TRAPEZOID BASIS           *
*                                                                  *
*     FOR SCE( ,9) = 1                                             *
*     SCENP( )  SCE NUMBER OF PAYMENTS                             *
*     SCEYR( , ) YRA PAYMENT IS MADE FOR THE SCE                   *
*     SCEPR( , ) PERCENT OF COST PAYED AT YEAR SCEYR( , )          *
*                                                                  *
*  COST ESTIMATING RELATIONSHIPS ELEMENTS                          *
*                                                                  *
*     CER( ,1)  SCE PHASE-IN PERIOD                                *
*        ( ,2)  SCE CONSTANT COST PERIOD                           *
*        ( ,3)  SCE PHASE-OUT PERIOD                               *
*        ( ,4)  SCE YEAR PAYMENTS BEGIN                            *
*        ( ,5)  NUMBER OF SAMPLES FROM WHICH THE CER WAS           *
*               DETERMINED                                         *
*               IF < 30, THEN THE STUDENT T DIST. IS USED          *
*               IF UNKNOWN OR >= 30, THEN THE NORMAL DIST. IS      *
*               USED.                                              *
*        ( ,6)  INTERCEPT VALUE (b0)                               *
*        ( ,7)  NUMBER OF SLOPE PARAMETERS                         *
*        ( ,8)  0 = NO SCALAR MULTIPLE                             *
*               1 = SCALAR MULTIPLE                                *
*        ( ,9)  FOR ( ,8) = 1                                      *
*               1 = TRIANGULAR DIST VALUE.                         *
*               2 = BETA DIST VALUE.                               *
*               3 = FIXED VALUE                                    *
*       ( ,10)  FOR ( ,9) = 1 .OR. 2                               *
*               LOWER VALUE                                        *
*       ( ,11)  FOR ( ,9) = 1 .OR. 2                               *
*               UPPER VALUE                                        *
*       ( ,12)  FOR ( ,9) = 1 .OR. 3, MODE VALUE                   *
*               FOR ( ,9) = 2, BETA TYPE                           *
*       ( ,13)  1 = BASE 10 DATA (WHOLE DOLLARS)                   *
*               2 = Ln DATA                                        *
*               3 = Log10 DATA                                     *
*       ( ,14)  1 = COST IS ALLOCATED BY PERCENTILE BASIS          *
*               2 = COST IS ALLOCATED BY TRAPEZOID BASIS           *
*                                                                  *
*  FOR CER( ,14) = 2                                               *
*     CERNP( )     NUMBER OF PAYMENTS                              *
*     CERYR( , )   YEAR PAYMENT IS MADE                            *
*     CERPR( , )   PERCENT OF COST MADE AT CERYR( , )              *
*                                                                  *
*    DRVR( , ,1)  LOW VALUE OF DRIVER ESTIMATE                     *
*    DRVR( , ,2)  HIGH VALUE OF DRIVER ESTIMATE                    *
*    DRVR( , ,3)  MODE VALUE OF DRVR IF DRVR( ,4) = 1 OR 2         *
*                 BETA TYPE  IF DRVR( ,4) = 3                      *
*    DRVR( , ,4) = 1 IF DRIVER IS A CONSTANT                       *
```

```
*                  = 2 IF DRIVER IS TRIANGULARLY DIST.                *
*                  = 3 IF DRIVER IS BETA DIST.                        *
*        VAR ( )    VARIANCE OF THE CER                               *
*       PARA( , )   SLOPE PARAMETER VALUES                            *
*                                                                     *
*    VRCV( , , )   VARIANCE COVARIANCE MATRIX OF THE PARAMETERS       *
*******************************************************************
      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
*
*     ADDCCE     .TRUE. IF A CCE IS BEING ADDED                       *
*     ADDSCE     .TRUE. IF A SCE IS BEING ADDED                       *
*     ADDCER     .TRUE. IF A CER IS BEING ADDED                       *
*     CHANGE     .TRUE. IF ANY CHANGE IS MADE ON THE DATA (WRITE      *
*                 DATA)                                               *
*     DATA       .TRUE. IF DATA EXISTS                                *
*******************************************************************

      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS

      EXTERNAL UMACH

*  OBTAIN THE TERMINAL NUMBER AND THE KEYBOARD NUMBER.

      CALL UMACH(1,OUT)
      CALL UMACH(2,INN)

*  LOAD CLEAR SCREEN SYSTEM COMMAND

      CLS = 'WRITE SYS$OUTPUT CLEAR_SCREEN'

      STATUS = LIB$SPAWN('CLS')

1     WRITE (OUT,10)
10    FORMAT (//,3X,'** LIFE CYCLE COST **',/,
     ^           3X,'**    MAIN MENU    **',//,
     ^           3X,'1. READ COST FILE.',/,
     ^           3X,'2. CREATE NEW COST FILE.',/,
     ^           3X,'3. VIEW/EDIT COSTS.',/,
     ^           3X,'4. EXECUTE LIFE CYCLE COST.',/,
     ^           3X,'5. EXIT.',//,
     ^           3X,'ENTER SELECTION (1 - [5]): ',$)
      READ (INN,'(I1)',ERR=1) IRPLY
```

```
      IF (IRPLY .EQ. 1) THEN
         CALL READ
         GO TO 1
      ENDIF

      IF (IRPLY .EQ. 2) THEN
         CALL CREATE
         GO TO 1
      ENDIF

      IF (IRPLY .EQ. 3) THEN

         IF (DATA .EQ. .FALSE.) THEN
           WRITE (OUT,25)
25         FORMAT (//,3X,'****** NO DATA PRESENT ******',/,
     ^                3X,'** READ OR CREATE THE DATA **',/)
           GO TO 1
         ENDIF

         CALL VIEW
         GO TO 1

      ENDIF

      IF (IRPLY .EQ. 4) THEN

         IF (DATA .EQ. .FALSE.) THEN
           WRITE (OUT,25)
           GO TO 1
         ENDIF

         CALL EXEC
         GO TO 1

      ENDIF

      END

******************************************************************
* SUBROUTINE READ: READS DATA FROM THE INPUT FILE                *
*                    'filename_LC.DAT'                           *
******************************************************************

      SUBROUTINE READ

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)
```

```
      INTEGER    NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
      INTEGER    CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY)
      INTEGER    CERNP(NC), OUT, INN, STATUS, LIB$SPAWN

      REAL       CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^           RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^           CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY)

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET
      CHARACTER*1  RESP

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS

      LOGICAL     DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER


*       LIB$SPAWN        -       SPAWN FUNCTION COMMAND

*
*   DO A DIRECTORY OF ALL THE *_LLC.DAT FILE
*

         STATUS = LIB$SPAWN ( 'DIRECTORY *_LC.DAT;')

      WRITE (OUT,10)
10    FORMAT (//,51X,'_________',/,
     ^          3X,'ENTER THE COST FILE NAME ( _LC.DAT understood):
     ^ ',$)
      READ (INN,'(A8)') ACCOUNT


         STATUS = LIB$SPAWN ( 'CLS')

*   CREATE THE FILENAME TO ..

        J = INDEX (ACCOUNT,' ')
        J=J-1
        FILENAME = ACCOUNT ( :J)//'_LC.DAT'
```

```
      OPEN (1, FILE=FILENAME, STATUS='OLD', ERR=95)

*   READ THE CONSTANT COST ESTIMATES
*   READ THE STOCHASTIC COST ESTIMATES
*   READ THE CER TYPE DATA

*
*   READ ALL THE INFORMATION REGARDING COST ELEMENTS
*
      READ (1,*) NCCE, NSCE, NCER, NYRS, DSC

*
*   CONSTANT COST ELEMENT INFORMATION
*
      DO 20, J = 1, NCCE
        READ (1,*) CCE(J,1), CCE(J,2), CCE(J,3), CCE(J,4),
     ^         CCE(J,5), CCE(J,6), CCENP(J)

        IF (CCE(J,6) .EQ. 1.0) THEN

         READ (1,*) (CCEYR(J,K), CCEPR(J,K),K = 1, CCENP(J))

        ENDIF

20    CONTINUE

*
*   STOCHASTIC COST ELEMENT INFORMATION
*

      DO 30, J = 1, NSCE
        READ (1,*) SCE(J,1), SCE(J,2), SCE(J,3), SCE(J,4), SCE(J,5)
        READ (1,*) SCE(J,6), SCE(J,7), SCE(J,8), SCE(J,9), SCENP(J)

        IF (SCE(J,9) .EQ. 1.0) THEN

         READ (1,*) (SCEYR(J,K), SCEPR(J,K),K = 1, SCENP(J))

        ENDIF

30    CONTINUE
*
*   COST ESTIMATING RELATIONSHIP ELEMENT INFORMATION
*
      DO 50, J=1,NCER

        READ (1,*) CER(J,1), CER(J,2), CER(J,3), CER(J,4),
     ^         CER(J,5), CER(J,6), CER(J,7), CER(J,8)
```

```
      READ (1,*) CER(J,9), CER(J,10), CER(J,11), CER(J,12),
    ^            CER(J,13), CER(J,14), CERNP(J), VAR(J)

      DO 35, I = 1, INT(CER(J,7))
        READ (1,*) DRVR(J,I,1), DRVR(J,I,2), DRVR(J,I,3),
    ^              DRVR(J,I,4)
35    CONTINUE

      READ (1,*) (PARA(J,I), I = 1, INT(CER(J,7)))

*  CER(J,7) + 1 BECAUSE OF THE INTERCEPT TERM. IF NO INTERCEPT,
*  THEN COLUMN 1 AND ROW 1 OF VRCV( ) = 0

      DO 40, I = 1, INT(CER(J,7)+1.0)

        READ (1,*) (VRCV(J,I,K), K = 1, INT(CER(J,7)+1.0))

40    CONTINUE

      IF (CER(J,14) .EQ. 1.0) THEN

       READ (1,*) (CERYR(J,K), CERPR(J,K),K = 1, CERNP(J))

      ENDIF

50    CONTINUE

      DATA = .TRUE.
      CLOSE (1, STATUS = 'KEEP')
      RETURN

95    WRITE (OUT,100) FILENAME
100   FORMAT (///,3X,'*** UNABLE TO OPEN ',A16,' ***',//)
      RETURN
      END
```

```
*******************************************************************
*    SUBROUTINE CREATE: CREATES A NEW DATA FILE FROM SCRATCH      *
*******************************************************************

      SUBROUTINE CREATE

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER   NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
      INTEGER   CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY)
      INTEGER   CERNP(NC), STATUS, LIB$SPAWN, CHGCER, OUT, INN

      REAL      CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^          RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^          CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY)

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS


        CHANGE = .FALSE.

        STATUS = LIB$SPAWN ( 'CLS')

      WRITE (OUT,10)
10    FORMAT (//,3X,'YOU ARE CREATING A NEW COST FILE.',/,
     ^        51X,8('-'),/,
     ^       3X,'ENTER THE COST FILE NAME ( _LC.DAT understood): ',$)
      READ (INN,'(A8)') ACCOUNT

*  CREATE THE FILENAME TO OPEN

        J = INDEX (ACCOUNT,' ')
        J=J-1
        FILENAME = ACCOUNT ( :J)//'_LC.DAT'
```

```
      WRITE (OUT,11)
11    FORMAT (//,3X,'** REMEMBER - ERRORS MADE WHILE ENTERING DATA
     ^ **',
     ^         /,3X,'** CAN BE CORRECTED BY SELECTING VIEW/EDIT
     ^ **',
     ^         /,3X,'** FROM THE MAIN MENU.',24X,'**',//)


15    WRITE (OUT,16) NY
16    FORMAT (/,3X,'ENTER THE SYSTEM LIFE IN WHOLE YEARS (MAX:',I3,
     ^              '): ',$)
      READ (INN,'(I2)',ERR=15) NYRS

      IF (NYRS .LE. 0) THEN

        WRITE (OUT, 17)
17      FORMAT (//,3X,'** SYSTEM LIFE MUST BE GREATER THAN 0. **',
     ^           /,6X,'DO YOU WISH TO EXIT ? (Y/[N]): '$)
        READ (INN,'(A1)') RESP

        IF (RESP .EQ. 'y' .OR. RESP .EQ. 'Y') RETURN

        GO TO 15

      ENDIF

      WRITE (OUT, 18)
18    FORMAT (/,3X,'ENTER THE DISCOUNT RATE (e.g: for 10.5%,
     ^enter 10.5): ',$)
      READ (INN,*) DSC

19    WRITE (OUT,20) NC
20    FORMAT (/,3X,'ENTER THE NUMBER OF CONSTANT COST ELEMENTS
     ^ (CCE) ([0] - ',I2,'): ',$)
      READ (INN,'(I2)',ERR=19) IRPLY

      IF (IRPLY .LT. 0 .OR. IRPLY .GT. NC) THEN
         WRITE (OUT,21)
21       FORMAT (//,3X,'*** INPUT OUT OF RANGE. TRY AGAIN. ***')
         GO TO 19
      ENDIF
      NCCE = IRPLY

24    WRITE (OUT,25) NC
25    FORMAT (/,3X,'ENTER THE NUMBER OF STOCHASTIC COST ELEMENTS
     ^ (SCE) ([0] - ',I2,'): ',$)
      READ (INN,'(I2)',ERR=24) IRPLY

      IF (IRPLY .LT. 0 .OR. IRPLY .GT. NC) THEN
```

```
            WRITE (OUT,21)
            GO TO 24
         ENDIF
         NSCE = IRPLY

29       WRITE (OUT,30) NC
30       FORMAT (/,3X,'ENTER THE NUMBER OF COST ESTIMATING
        ^ RELATIONSHIPS (CER) ([0] - ',I2,'): ',$)
         READ (INN,'(I2)',ERR=29) IRPLY

         IF (IRPLY .LT. 0 .OR. IRPLY .GT. NC) THEN
            WRITE (OUT,21)
            GO TO 29
         ENDIF
         NCER = IRPLY

         IF ((NCCE + NSCE + NCER) .EQ. 0) THEN
            STATUS = LIB$SPAWN ( 'CLS')
           RETURN
         ENDIF

         WRITE (OUT,35)
35       FORMAT (//,3X,
        ^'** NOTE: ALL COST ENTRIES MUST BE IN THE SAME DOLLAR
        ^ **',/,3X,
        ^'** UNIT (ie: ONE $, THOUSAND $, OR MILLION $)         **',//,
        ^3X,'[ENTER RETURN TO CONTINUE] ',$)
         READ (INN,*)


         IF (NCCE .GT. 0) THEN
            ADDCCE = .FALSE.
            CHGCCE = 0
            CALL INCCE (I, CHGCCE)
         ENDIF

         IF (NSCE .GT. 0) THEN
            ADDSCE = .FALSE.
            CHGSCE = 0
            CALL INSCE (I, CHGSCE)
         ENDIF

         IF (NCER .GT. 0) THEN
            ADDCER = .FALSE.
            CHGCER = 0
            CALL INCER (I, CHGCER)
         ENDIF

         CALL WRITE
```

```
      RETURN
      END

********************************************************************
*    SUBROUTINE WRITE: WRITES THE DATA INTO THE FILE               *
*                    'filename_LC.DAT'                              *
********************************************************************

      SUBROUTINE WRITE

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER   NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
      INTEGER   CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY)
      INTEGER   CERNP(NC), OUT, INN, STATUS, LIB$SPAWN

      REAL      CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^          RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^          CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY)

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET
      CHARACTER*1  RESP

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS


*
*  PROMPT FOR A POSSIBLE NEW FILE NAME
*
1     WRITE (OUT,5) FILENAME
5     FORMAT(/,3X,'SAVE FILE AS ',A16,'? ([Y]/N): ',$)
      READ (INN,'(A1)',ERR=1)RESP

      IF (RESP.EQ.'N'.OR.RESP.EQ.'n') THEN
```

```
      WRITE (OUT,10)
10    FORMAT (//,51X,'________',/,
     ^        3X,'ENTER THE COST FILE NAME ( _LC.DAT understood):
     ^ ',$)
      READ (INN,'(A8)') ACCOUNT


         STATUS = LIB$SPAWN ( 'CLS')

*  CREATE THE FILENAME TO OPEN

        J = INDEX (ACCOUNT,' ')
        J=J-1
        FILENAME = ACCOUNT ( :J)//'_LC.DAT'

      ENDIF

*
*  OPEN A FILE NAMED IN FILENAME
*
      OPEN (1, FILE=FILENAME, STATUS='NEW')

*
*  WRITE ALL THE INFORMATION REGARDING FILE 'FILENAME'
*
      WRITE (1,*) NCCE, NSCE, NCER, NYRS, DSC

*
*  CONSTANT COST ELEMENT INFORMATION
*
      DO 20, J = 1, NCCE
        WRITE (1,*) CCE(J,1), CCE(J,2), CCE(J,3), CCE(J,4),
     ^              CCE(J,5), CCE(J,6), CCENP(J)

        IF (CCE(J,6) .EQ. 1.0) THEN

         WRITE (1,*) (CCEYR(J,K), CCEPR(J,K), K=1,CCENP(J))

        ENDIF

20    CONTINUE

*
*  STOCHASTIC COST ELEMENT INFORMATION
*

      DO 30, J=1, NSCE
        WRITE (1,*) SCE(J,1), SCE(J,2), SCE(J,3), SCE(J,4),
     ^              SCE(J,5)
```

```
          WRITE (1,*) SCE(J,6), SCE(J,7), SCE(J,8), SCE(J,9),
     ^                SCENP(J)

          IF (SCE(J,9) .EQ. 1.0) THEN

           WRITE (1,*) (SCEYR(J,K), SCEPR(J,K), K= 1, SCENP(J))

          ENDIF

30      CONTINUE
*
*  COST ESTIMATING RELATIONSHIP ELEMENT INFORMATION
*
        DO 50, J = 1, NCER

          WRITE (1,*) CER(J,1), CER(J,2), CER(J,3), CER(J,4),
     ^                CER(J,5), CER(J,6), CER(J,7), CER(J,8)

          WRITE (1,*) CER(J,9), CER(J,10), CER(J,11), CER(J,12),
     ^                CER(J,13), CER(J,14), CERNP(J), VAR(J)

          DO 35, I = 1, INT(CER(J,7))
            WRITE (1,*) DRVR(J,I,1), DRVR(J,I,2), DRVR(J,I,3),
     ^                  DRVR(J,I,4)
35        CONTINUE

          WRITE (1,*) (PARA(J,I), I = 1, INT(CER(J,7)))

*  CER(J,7) + 1 BECAUSE OF THE INTERCEPT TERM. IF NO INTERCEPT,
*  THEN COLUMN 1 AND ROW 1 OF VRCV( ) = 0

          DO 40, I = 1, INT(CER(J,7) + 1.0)

            WRITE (1,*) (VRCV(J,I,K), K = 1, INT(CER(J,7)+1.0))

40        CONTINUE

          IF (CER(J,14) .EQ. 1.0) THEN

           WRITE (1,*) (CERYR(J,K), CERPR(J,K), K=1,CERNP(J))

          ENDIF

50      CONTINUE

        DATA = .TRUE.

        CLOSE (1, STATUS = 'KEEP')
```

```
      RETURN

      END

*******************************************************************
*   SUBROUTINE INCCE: ALLOWS THE USER TO ENTER NEW DATA OF CCE    *
*******************************************************************

      SUBROUTINE INCCE (I, CHGCCE)

      INTEGER     NC, NY, NV, ND
      PARAMETER   (NC=10, ND=10, NY=99, NV=800)

      INTEGER    NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY),
     ^           CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY),
     ^           CERNP(NC), STRT, YEAR, OUT, INN, CHGCCE STATUS,
     ^           LIB$SPAWN

      REAL       CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^           RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^           CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY), SUMPR,
     ^           PRCNT

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     'CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      LOGICAL     DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET
      CHARACTER*1  RESP

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS


      CET = 'CCE'

      STRT = 1

*  INCASE THE SUBROUTINE IS BEING USED TO ADD A CCE

      IF (ADDCCE .EQ. .TRUE.) STRT = NCCE
```

```
      GO TO (9, 14) CHGCCE

         STATUS = LIB$SPAWN ( 'CLS')


      DO 50, I = STRT, NCCE
        WRITE (OUT,8) I
8       FORMAT (3X,'CONSTANT COST ELEMENT (CCE) ',I2)
9       WRITE (OUT,10)
10      FORMAT (/,3X,'ENTER COST: ',$)
        READ (INN,*) CCE(I,5)

        IF (CHGCCE .EQ. 1) RETURN

14      WRITE (OUT,15) CET, I
15      FORMAT (/,3X,A3,X,I2,/,
     ^             3X,'1. ALLOCATE COST BASED ON A PERCENTAGE.',/,
     ^             3X,'2. ALLOCATE COST BASED ON A TRAPEZOID.',//,
     ^             3X,'ENTER (1 - [2]): ',$)
        READ (INN,'(I1)',ERR=14) IRPLY

        IF ( IRPLY .LT. 0 .OR. IRPLY .GT. 2) GO TO 14

        IF (IRPLY .EQ. 0) IRPLY = 2

        CCE(I,6) = REAL(IRPLY)

*  USE PERCENT ALLOCATION (INPRCNT) OR TRAPEZOIDAL ALLOCATION
*   (INPHASE)

        IF (CCE(I,6) .EQ. 1.0) THEN

30      WRITE (OUT,35) CET, I, NYRS
35      FORMAT (/,3X,'ENTER THE TOTAL NUMBER OF PAYMENTS FOR
     ^ ',A3,X,I2,'(Max: ',I2,'): ',$)
        READ (INN,'(I2)',ERR=30) IRPLY

        IF (IRPLY .GT. NYRS) THEN

          WRITE (OUT, 36) NYRS
36        FORMAT (//,
     ^    3X,'** ERROR. NUMBER OF PAYMENTS MUST BE LESS THAN **',/,
     ^    3X,'** OR EQUAL TO THE SYSTEM LIFE (',I3,').',10X,'**',/,
     ^    3X,'****************** TRY AGAIN ********************',/)
          GO TO 30

        ENDIF

        CCENP(I) = IRPLY
```

```
          SUMPR = 0.0

          DO 40, J = 1, CCENP(I)

            CALL INPRCNT (J,CET,PRCNT,YEAR,NYRS)
            CCEPR(I,J) = PRCNT
            CCEYR(I,J) = YEAR
            SUMPR = SUMPR + CCEPR(I,J)

40        CONTINUE

          IF (SUMPR .NE. 100.0) THEN
           WRITE (OUT, 45) SUMPR
45         FORMAT (//,3X,'*** THE SUM OF THE PERCENTAGE OF PAYMENTS
     ^ ***',/,
     ^                3X,'EQUALS ',F5.1,/,
     ^                3X,'PLEASE RECALCULATE AND RE-ENTER THE DATA')


           GO TO 30
          ENDIF

          ELSE

            CALL INPHASE (I, CET, CCE(I,1), CCE(I,2), CCE(I,3),
     ^ CCE(I,4), NYRS)

          ENDIF

           IF (CHGCCE .EQ. 2) RETURN

           STATUS = LIB$SPAWN ( 'CLS')

50      CONTINUE

        ADDCCE = .FALSE.

        RETURN
        END

*******************************************************************
*   SUBROUTINE INPHASE: ALLOWS THE USER TO ENTER THE TIME PHASING *
*******************************************************************

        SUBROUTINE INPHASE (I,CET,INPHS, CNPHS, OUPHS, STPHS, NYRS)

        REAL  INPHS, CNPHS, OUPHS, STPHS

        INTEGER   I, NYRS, OUT, INN
```

```
      COMMON /TRMNL/ OUT, INN

      CHARACTER*3  CET

5     WRITE (OUT,10) CET, I
10    FORMAT (/,3X,'ENTER THE PHASE-IN PERIOD FOR ',A3,X,I2,': ',$)
      READ (INN,*) INPHS

      WRITE (OUT,20) CET, I
20    FORMAT (/,3X,'ENTER THE CONSTANT-COST PERIOD FOR ',A3,X,I2,':
     ^ ',$)
      READ (INN,*) CNPHS


      WRITE (OUT,30) CET, I
30    FORMAT (/,3X,'ENTER THE PHASE-OUT PERIOD FOR ',A3,X,I2,': ',$)
      READ (INN,*) OUPHS

      WRITE (OUT,40) CET, I
40   FORMAT (/,3X,'ENTER THE YEAR PAYMENTS START FOR ',A3,X,I2,': '$)
      READ (INN,*) STPHS


      IF ((INPHS + CNPHS + OUPHS + STPHS) .GT. NYRS) THEN

        WRITE (OUT, 45)
45      FORMAT (//,3X,'*** WARNING!! ***',//,
     ^  3X,'NUMBER OF YEARS FOR THE SYSTEM EXCEEDED.',/,
     ^  3X,'DO YOU WISH TO RE-ENTER LAST PHASE DATA ? ([Y]/N): ',$)
        READ (INN,'(A1)') RESP

        IF (RESP.EQ.'N'.OR.RESP.NE.'n') THEN
          NYRS = INPHS + CNPHS + OUPHS + STPHS
          RETURN
        ENDIF

        GO TO 5

      ENDIF

      RETURN
      END

*******************************************************************
*    SUBROUTINE INPRCNT:  ALLOWS THE USER TO ENTER A DISTRIBUTIONS*
*    OF COST BASED ON PERCENTILE.                                 *
*******************************************************************
```

```
      SUBROUTINE INPRCNT (J,CET,PRCNT,YEAR,NYRS)

      REAL           PRCNT
      INTEGER        YEAR, J, NYRS, OUT, INN
      CHARACTER*3    CET

      COMMON /TRMNL/ OUT, INN

15     WRITE (OUT, 25) J, CET
25      FORMAT (/,3X,'ENTER YEAR PAYMENT ',I2,' IS MADE FOR ',A3,':
',$)
       READ (INN,'(I3)',ERR=15) YEAR

       IF (YEAR .GT. NYRS) THEN
        WRITE (OUT, 27) NYRS, J, YEAR
27        FORMAT  (//,3X,'**  ERROR:    SYSTEM  LIFE  IS  ',I3,'
     ^YEARS.**',/,
     ^           3X,'**  PAYMENT  ',I2,'  IS  MADE  AT  YEAR  ',I3,'.
     ^**',/,
     ^        3X,'** YEAR PAYMENT IS MADE MUST BE NO GREATER **',/,
     ^        3X,'** THAN SYSTEM LIFE.  [PLEASE RE-ENTER]    **')

         GO TO 15

       ENDIF

30     WRITE (OUT,35) YEAR, J
35     FORMAT (3X,'ENTER PERCENTAGE OF COST PAYED AT END OF YEAR
     ^ ',I2,
     ^           ' FOR PAYMENT ',I2,' (%): ',$)
       READ (INN,*) PRCNT

      RETURN
      END

*****************************************************************
*    SUBROUTINE INTRIANG: ALLOWS THE USER TO ENTER TRIANGULAR       *
*    DISTRIBUTION DATA.                                              *
*****************************************************************

      SUBROUTINE INTRIANG (I,CET,LVAL,HVAL,MODE)

      REAL  LVAL, HVAL, MODE

      INTEGER  I, OUT, INN

      COMMON /TRMNL/ OUT, INN

      CHARACTER*3 CET
```

```
30        WRITE (OUT, 35) CET, I
35        FORMAT(/,3X,'ENTER MODE (best guess) OF TRIANG. DIST.
     ^ ',A3,X,I2,': ',$)
          READ (INN,*,ERR=30) MODE

40        WRITE (OUT,45) CET, I
45        FORMAT(/,3X,'ENTER LOW VALUE FOR TRIANG. DIST. ',A3,X,I2':
     ^ ',$)
          READ (INN,*,ERR=40) LVAL

50        WRITE (OUT,55) CET, I
55        FORMAT(/,3X,'ENTER HIGH VALUE FOR TRIANG. DIST. ',A3,X,I2':
     ^ ',$)
          READ (INN,*,ERR=50) HVAL

          IF (MODE .LT. LVAL .OR. MODE .GT. HVAL) THEN

            WRITE (OUT,60)
60         FORMAT (//,3X,'*** ERROR IN TRIANGULAR DISTRIBUTION DATA
     ^ ***',/
     ^                  3X,'*************** TRY AGAIN
     ^ *******************')
            GO TO 30

          ENDIF
          RETURN
          END

******************************************************************
*    SUBROUTINE INBETA: ALLOWS THE USER TO ENTER TRIANGULAR      *
*    DISTRIBUTION DATA.                                          *
******************************************************************

        SUBROUTINE INBETA (I, CET, LVAL, HVAL, BTYPE)

        REAL LVAL, HVAL, BTYPE

        INTEGER I, IRPLY, OUT, INN

        COMMON /TRMNL/ OUT, INN

        CHARACTER*3 CET


20        WRITE (OUT, 25) CET, I
25       FORMAT (/,3X,'ENTER BETA DIST. TYPE (1 - 9) FOR ',A3,X,I2,':
',$)
          READ (INN,'(I1)',ERR=20) IRPLY
```

```
      IF (IRPLY .LT. 1 .OR. IRPLY .GT. 9) THEN

         WRITE (OUT,30)
30       FORMAT (//,3X,'*** INPUT OUT OF RANGE. TRY AGAIN.
     ^ ***',//)
         GO TO 20

      ENDIF

      BTYPE = REAL(IRPLY)

40    WRITE (OUT,45) CET, I
45    FORMAT (/,3X,'ENTER LOW VALUE FOR BETA DIST. ',A3,X,I2':
     ^ ',$)
      READ (INN,*,ERR=40) LVAL

50    WRITE (OUT,55) CET, I
55    FORMAT (/,3X,'ENTER HIGH VALUE FOR BETA DIST. ',A3,X,I2':
     ^ ',$)
      READ (INN,*,ERR=50) HVAL

      IF (LVAL .GT. HVAL) THEN
        WRITE (OUT,60)
60      FORMAT (//,3X,'*** ERROR IN BETA DISTRIBUTION DATA
     ^ ***',/,
     ^            12X,'LOW VALUE > HIGH VALUE',/,
     ^            3X,'************ TRY AGAIN ****************')
        GO TO 20
      ENDIF

      RETURN
      END

*******************************************************************
*    SUBROUTINE INSCE: ALLOWS THE USER TO ENTER STOCHASTIC COST   *
*    ELEMENTS                                                     *
*******************************************************************

      SUBROUTINE INSCE (I, CHGSCE)

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER   NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
      INTEGER   CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY)
      INTEGER   CERNP(NC), YEAR, OUT, INN, CHGSCE, STATUS
      INTEGER   LIB$SPAWN

      REAL      CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4)
```

```
      REAL      RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^          CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY), SUMPR

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET
      CHARACTER*1  RESP

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS


      CET = 'SCE'

      STRT = 1

*   IF ADDING A SCE

      IF (ADDSCE .EQ. .TRUE.) STRT = NSCE

*   IF CHANGING A SCE

      GO TO (5, 24) CHGSCE

         STATUS = LIB$SPAWN ( 'CLS')


      DO 100, I = STRT, NSCE

5        WRITE (OUT,10) I
10       FORMAT (/,3X,'STOCHASTIC COST ELEMENT (SCE) ',I2,//,
     ^              3X,'1. TRIANGULAR DISTRIBUTION.',/,
     ^              3X,'2. BETA DISTRIBUTION',//,
     ^              3X,'ENTER DISTRIBUTION SELECTION ([1] - 2): ',$)
         READ (INN,'(I1)',ERR=5) IRPLY

         IF (IRPLY .LT. 0 .OR. IRPLY .GT. 2) THEN

            WRITE (OUT, 20)
```

```
20          FORMAT (//,3X,'*** INPUT OUT OF RANGE. TRY AGAIN.
     ^ ***',//)
            GO TO 5

         ENDIF

         IF (IRPLY .EQ. 0) IRPLY = 1

         SCE(I,5) = REAL(IRPLY)

         CET = 'SCE'

         IF (SCE(I,5) .EQ. 1.0) THEN

            CALL INTRIANG ( I, CET, SCE(I,6), SCE(I,7), SCE(I,8))

         ELSEIF (SCE(I,5) .EQ. 2.0) THEN

            CALL INBETA ( I, CET, SCE(I,6), SCE(I,7), SCE(I,8))

         ENDIF

         IF (CHGSCE .EQ. 1) RETURN

24       WRITE (OUT,25) CET, I
25       FORMAT (//,3X,A3,X,I2,/,
     ^                3X,'1. ALLOCATED COST BASED ON A PERCENTAGE.',/,
     ^                3X,'2. ALLOCATED COST BASED ON A TRAPEZOID.',//,
     ^                3X,'ENTER (1 - [2]): ',$)
         READ (INN,'(I1)',ERR=24) IRPLY

         IF ( IRPLY .LT. 0 .OR. IRPLY .GT. 2) GO TO 24

         IF (IRPLY .EQ. 0) IRPLY = 2

         SCE(I,9) = REAL(IRPLY)

*  USE PERCENT ALLOCATION (INPRCNT) OR TRAPEZOIDAL ALLOCATION
*  (INPHASE)

         IF (SCE(I,9) .EQ. 1.0) THEN

30       WRITE (OUT,35)CET, I, NYRS
35       FORMAT (/,3X,'ENTER THE TOTAL NUMBER OF PAYMENTS FOR
     ^ ',A3,X,I2,' (Max: ',I2,'): ',$)
         READ (INN,'(I2)',ERR=30) IRPLY

         IF (IRPLY .GT. NYRS) THEN
```

```
         WRITE (OUT, 36) NYRS
36       FORMAT (//,
     ^   3X,'** ERROR. NUMBER OF PAYMENTS MUST BE LESS THAN **',/,
     ^   3X,'** OR EQUAL TO THE SYSTEM LIFE (',I3,').',10X,'**',/,
     ^   3X,'***************** TRY AGAIN *******************',/)
         GO TO 30

       ENDIF

       SCENP(I) = IRPLY

       SUMPR = 0.0
       DO 40, J = 1, SCENP(I)

         CALL INPRCNT (J,CET,PRCNT,YEAR,NYRS)
         SCEPR(I,J) = PRCNT
         SCEYR(I,J) = YEAR
         SUMPR = SUMPR + SCEPR(I,J)

40     CONTINUE

       IF (SUMPR .NE. 100.0) THEN

        WRITE (OUT, 60) SUMPR
60      FORMAT (//,3X,'*** ERROR.  THE SUM OF THE PERCENTAGES
     ^ ***',/,3X,'*** OF THE PAYMENTS EQUALS ',F5.1,6X,'***',//,
     ^             3X,'PLEASE RECALCULATE AND RE-ENTER THE
     ^ DATA',/)

        GO TO 30

       ENDIF


       ELSEIF (SCE(I,9) .EQ. 2.0) THEN

         CALL INPHASE (I,CET,SCE(I,1),SCE(I,2),SCE(I,3),SCE(I,4),
     ^   NYRS)

       ENDIF

        IF (CHGSCE .EQ. 2) RETURN

        STATUS = LIB$SPAWN ( 'CLS')

100    CONTINUE

       RETURN
       END
```

```
***********************************************************************
*    SUBROUTINE INCER: ALLOWS THE USER TO ENTER CER TYPE DATA        *
***********************************************************************

      SUBROUTINE INCER (I, CHGCER)

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER   NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
      INTEGER   CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY)
      INTEGER   CERNP(NC), JSTRT, STRT, YEAR, CHGCER, OUT, INN
      INTEGER   STATUS, LIB$SPAWN

      REAL      CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^          RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^          CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY), SUMPR

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET
      CHARACTER*1  RESP

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS


      CET = 'CER'
      STRT = 1
      JSTRT = 0

      GO TO (5,77,45,170) CHGCER

      IF (ADDCER .EQ. .TRUE.) STRT = NCER

         STATUS = LIB$SPAWN ( 'CLS')


      DO 500, I = STRT, NCER
```

```
          WRITE (OUT,4) I
4         FORMAT (/,3X,'COST ESTIMATING RELATIONSHIP (CER) ',I2)

5         WRITE (OUT,10)
10        FORMAT (/,3X,'** SCALAR MULTIPLE **')
          IF (I .EQ. 1) THEN
            WRITE (OUT,11)
11          FORMAT(/,
     ^ 3X,'If your CER needs to be multiplied by a scalar
     ^ multiple enter 1.',/,
     ^ 3X,'EXAMPLE - Your dependent variable is Average Cost per
     ^ Radio',/,
     ^ 3X,'and you need to multiply by say 60 radios to convert
     ^ to ',
     ^  /,3X,'Total Cost of 60 radios')

         ENDIF

         WRITE (OUT,12)
12       FORMAT (/,3X,'1. SCALAR MULTIPLE > 1',//,
     ^               3X,'ENTER ([0] - 1): ',$)
          READ (INN,'(I1)',ERR=5) IRPLY

          IF (IRPLY .LT. 0 .OR. IRPLY .GT. 1) THEN
             WRITE (OUT,15)
15           FORMAT (//,3X,'*** INPUT OUT OF RANGE. TRY AGAIN.
     ^ ***',//)
             GO TO 5
          ENDIF

          CER(I,8) = REAL(IRPLY)

          IF (CER(I,8) .EQ. 0.0) THEN
             CER(I,9) = 3
             CER(I,12)= 1
             IF (CHGCER .EQ. 1) RETURN
             GO TO 35

*   SCALAR MULTIPLE EXISTS FOR CER 'I'
*
*   PROMPT FOR SCALAR TYPE
*
          ELSEIF (CER(I,8) .EQ. 1.0) THEN

19           WRITE (OUT,20)
20           FORMAT (//,3X,'TYPE OF SCALAR (SCL) ESTIMATION.',/,
     ^                    3X,'1. TRIANGULAR DISTRIBUTION.',/,
     ^                    3X,'2. BETA DISTRIBUTION.',/,
     ^                    3X,'3. FIXED VALUE.',//,
```

```
     ^              3X,'ENTER (1 - [3]): ',$)
            READ (INN,'(I1)',ERR=19) IRPLY

            IF (IRPLY .LT. 0 .OR. IRPLY .GT. 3) THEN
              WRITE (OUT,15)
              GO TO 19
            ENDIF

            IF (IRPLY .EQ. 0) IRPLY = 3

            CER(I,9) = REAL(IRPLY)
*
*  FOR A SCALAR MULTIPLE, GET THE MODE FOR FIXED OR TRIANG DIST.
*  SCALAR, OR GET THE BETA TYPE OF THE SCALAR
*
            IF (CER(I,9) .EQ. 1.0 ) THEN
              CET = 'SCL'
              CALL INTRIANG(I,CET,CER(I,10),CER(I,11),CER(I,12))

            ELSEIF (CER(I,9) .EQ. 2.0) THEN
              CET = 'SCL'
              CALL INBETA(I,CET,CER(I,10),CER(I,11),CER(I,12))

            ELSEIF (CER(I,9) .EQ. 3.0) THEN

25            WRITE (OUT,30) I
30          FORMAT (/,3X,'ENTER FIXED SCALAR VALUE FOR CER ',I2,':
     ^ ',$)
              READ (INN,*,ERR=25) CER(I,12)


            ENDIF

          ENDIF

        IF (CHGCER .EQ. 1) RETURN

*
*  OBTAIN NUMBER OF SLOPE PARAMETERS
*

35      WRITE (OUT,40) ND
40      FORMAT (/,3X,'ENTER THE NUMBER OF SLOPE PARAMETERS (1 - ',
     ^          I2,'): ',$)
        READ (INN,'(I2)',ERR=35) IRPLY

        IF (IRPLY .LT. 1 .OR. IRPLY .GT. ND) THEN
          WRITE (OUT,15)
          GO TO 35
```

```
      ENDIF

      CER(I,7) = REAL(IRPLY)

*
*  BASE OF THE DRIVERS
*

45          WRITE (OUT,50) I
50         FORMAT (//,3X,'CER',I2,' IS ESTIMATED IN:',/,
     ^                3X,'1. WHOLE DOLLARS.',/,
     ^                3X,'2. LOG BASE e (Ln).',/,
     ^                3X,'3. LOG BASE 10 (Log).',//,
     ^                3X,'ENTER ([1] - 3): ',$)
            READ (INN,'(I1)',ERR=45) IRPLY

            IF (IRPLY .LT. 0 .OR. IRPLY .GT. 3) THEN
              WRITE (OUT,15)
              GO TO 45
            ENDIF

            IF (IRPLY .EQ. 0) IRPLY = 1

            CER (I,13) = REAL(IRPLY)

            IF (CHGCER .EQ. 3) RETURN

70          WRITE (OUT,71) I
71          FORMAT (/,3X,'ENTER THE NUMBER OF OBSERVATIONS FROM
     ^ WHICH CER' ,I2,' WAS OBTAINED',/,
     ^                3X,'([0] IF UNKNOWN, 1 - 999): ',$)
            READ (INN,'(I3)',ERR=70) IRPLY

            IF (IRPLY .LT. 0 .OR. IRPLY .GT. 999) THEN
              WRITE (OUT,15)
              GO TO 70
            ENDIF

*  IF SAMPLE SIZE IS NOT KNOWN THEN SET SAMPLE SIZE = 999 TO USE
*  THE NORMAL DISTRIBUTION

            IF (IRPLY .EQ. 0) IRPLY = 999

            CER(I,5) = REAL(IRPLY)

72      WRITE (OUT, 73) I
73      FORMAT (/,3X,'ENTER VALUE OF CER ',I2,' INTERCEPT TERM',
     ^               ' ([0] if none): ',$)
```

```
      READ (INN,*,ERR=72) CER(I,6)


      IF (CER(I,6) .LT. 0.0) THEN

        WRITE (OUT,74)
74      FORMAT (//,3X,'*** INTERCEPT MUST BE GREATER THAN',
     ^               ' OR EQUAL TO 0 ***')
        GO TO 72

      ENDIF

75    WRITE (OUT, 76) I
76    FORMAT (/,3X,'ENTER THE ESTIMATED VARIANCE (S-squared)',
     ^              'FOR CER',X,I2,': ',$)
      READ (INN,*,ERR=75) VAR(I)

*  COLLECT DRIVER INFORMATION

77       CET = 'DRV'

          IF (CHGCER .EQ. 2) THEN
            J = IRPLY
            GO TO 78
          ENDIF

       DO 100, J = 1, INT(CER(I,7))

78         WRITE (OUT, 79) J
79         FORMAT (//,3X,'DISTRIBUTION OF DRIVER (DRV) ',I2,/,
     ^                 3X,'1. CONSTANT VALUE'/,
     ^                 3X,'2. TRIANGULAR DISTRIBUTION',/,

     ^                 3X,'3. BETA DISTRIBUTION',//,
     ^                 3X,'ENTER ([1] - 3): ',$)
           READ (INN,'(I1)',ERR=78) IRPLY

           IF (IRPLY .LT. 0 .OR. IRPLY .GT. 3) THEN
             WRITE (OUT,15)
             GO TO 78
           ENDIF

           IF (IRPLY .EQ. 0) IRPLY = 1

           DRVR(I,J,4) = REAL(IRPLY)

         IF (DRVR(I,J,4) .EQ. 1.0 ) THEN

           WRITE (OUT,83) CET, J
```

```
83          FORMAT (/,3X,'ENTER CONSTANT VALUE OF ',A3,X,I2,': ',$)
            READ (INN,*) DRVR(I,J,3)

          ELSEIF (DRVR(I,J,4) .EQ. 2.0 ) THEN

             CALL INTRIANG(J,CET,DRVR(I,J,1),DRVR(I,J,2),
     ^                     DRVR(I,J,3))

          ELSEIF (DRVR(I,J,4) .EQ. 3.0) THEN

             CALL INBETA(J,CET,DRVR(I,J,1),DRVR(I,J,2),DRVR(I,J,3))

          ENDIF

        IF (CHGCER .EQ. 2) RETURN

100     CONTINUE

*  INPUT THE SLOPE PARAMETERS

        DO 130, J = 1, INT(CER(I,7))

115       WRITE (OUT, 120) J
120       FORMAT (/,3X,'ENTER VALUE OF SLOPE PARAMETER ',I2,': ',$)
          READ (INN,*,ERR=115) PARA(I,J)

130     CONTINUE

*  IF THERE IS NO INTERCEPT VALUE PLACE 0s IN INTERCEPT POSITION
*  OF VAR/COV MATRIX, AND START IN POSITION (2,2)

        STRT = 0

        IF (CER(I,6) .EQ. 0.0) THEN

          DO 142, J = 1, INT(CER(I,7)) + 1
            VRCV(I,1,J) = 0.0
            VRCV(I,J,1) = 0.0
142       CONTINUE

          STRT = 1

        ENDIF

*  ENTER FOR THE VAR/COV MATRIX

144       WRITE (OUT,148)
148       FORMAT (//,3X,'** NOTE: INTERCEPT HOLDS POSITION (1,1) IN
     ^ VAR/COV MATRIX **')
```

```
         DO 160, J = STRT + 1, INT(CER(I,7)) + 1
           DO 160, K = J, INT(CER(I,7)) + 1

150             WRITE (OUT, 155) J, K
155        FORMAT (/,3X,'PARAMETER VARIANCE/COVARIANCE MATRIX POSITION
     ^ (',I2,',',I2,').',/,3X,'ENTER VARIANCE/COVARIANCE: ',$)
                READ (INN,*,ERR=150) VRCV(I,J,K)
                VRCV(I,K,J)=VRCV(I,J,K)

160      CONTINUE

*  PROMPT TO RE-ENTER THE VRCV MATRIX INCASE AN ERROR WAS MADE

161     WRITE (OUT,162)
162     FORMAT (/,3X,'** DO YOU WISH TO RE-ENTER THE VAR/COV
     ^  MATRIX',
     ^              ' (Y/[N]); ',$)
        READ (INN,'(A1)',ERR=161)RESP

        IF (RESP.EQ.'Y'.OR.RESP.EQ.'y') GO TO 144

170      WRITE (OUT,175) I
175      FORMAT (//,3X,'CER',X,I2,/,
     ^                3X,'1. ALLOCATE COST BASED ON A PERCENTAGE.',/,
     ^                3X,'2. ALLOCATE COST BASED ON A TRAPEZOID.',//,
     ^                3X,'ENTER (1 - [2]): ',$)
         READ (INN,'(I1)',ERR=170) IRPLY

         IF ( IRPLY .LT. 0 .OR. IRPLY .GT. 2) GO TO 170

         IF (IRPLY .EQ. 0) IRPLY = 2

         CER(I,14) = REAL(IRPLY)

*  USE PERCENT ALLOCATION (INPRCNT) OR TRAPEZOIDAL ALLOCATION
*   (INPHASE)

         IF (CER(I,14) .EQ. 1.0) THEN

180      WRITE (OUT,185) I, NYRS
185      FORMAT (/,3X,'ENTER THE TOTAL NUMBER OF PAYMENTS FOR
     ^  CER',X,I2,'(Max: ',I2,'): ',$)
         READ (INN,'(I2)',ERR=180) IRPLY

         IF (IRPLY .GT. NYRS) THEN

           WRITE (OUT, 190) NYRS
190        FORMAT (//,
```

```
     ^    3X,'** ERROR. NUMBER OF PAYMENTS MUST BE LESS THAN **',/,
     ^    3X,'** OR EQUAL TO THE SYSTEM LIFE (',I3,').',10X,'**',/,
     ^    3X,'****************** TRY AGAIN *********************',/)
          GO TO 180

        ENDIF

        CERNP(I) = IRPLY

        SUMPR = 0.0
        DO 200, J = 1, CERNP(I)

          CET = 'CER'
          CALL INPRCNT (J,CET,PRCNT,YEAR,NYRS)
          CERPR(I,J) = PRCNT
          CERYR(I,J) = YEAR
          SUMPR = SUMPR + CERPR(I,J)

200     CONTINUE

        IF (SUMPR .NE. 100.0) THEN
         WRITE (OUT, 210) SUMPR
210       FORMAT (//,3X,'*** THE SUM OF THE PERCENTAGE OF PAYMENTS
     ^***',/,
     ^               3X,'EQUALS ',F5.1,/,
     ^               3X,'PLEASE RECALCULATE AND RE-ENTER THE DATA')


         GO TO 180
        ENDIF

        ELSEIF (CER(I,14) .EQ. 2.0) THEN

          CET = 'CER'
          CALL INPHASE (I,CET,CER(I,1),CER(I,2),CER(I,3),CER(I,4),
     ^                  NYRS)

        ENDIF

         IF (CHGCER .EQ. 4) RETURN

         STATUS = LIB$SPAWN ( 'CLS')

500     CONTINUE

        RETURN
        END

*********************************************************************
```

```
*     SUBROUTINE VIEW: ALLOWS THE USER TO VIEW AND EDIT ALL COST    *
*     ELEMENTS.  ALSO ALLOWS FOR THE ADDITION AND DELETION OF       *
*    ANY AND ALL COST ELEMENT.                                      *
*******************************************************************

      SUBROUTINE VIEW

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER    NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY),
     ^           CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY),
     ^           CERNP(NC), YEAR, CHGCER, MINYRS, OUT, INN, CHGSCE,
     ^           CHGCCE, STATUS, LIB$SPAWN

      REAL       CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^           RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^           CCEPR(NC,NY), SCEPR(NC,NY), CERPR(NC,NY), SUMPR,
     ^           TEMP

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /TRMNL/ OUT, INN

      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER

      CHARACTER*29 CLS
      CHARACTER*15 FILENAME
      CHARACTER*12 BASE
      CHARACTER*9  ACCOUNT
      CHARACTER*3  CET
      CHARACTER*1  RESP

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS


*  FIRST MENU IN VIEW/EDIT

1         STATUS = LIB$SPAWN ( 'CLS')

      WRITE (OUT,5) NCCE,NSCE,NCER,NYRS,DSC
5     FORMAT (//,3X,'** VIEW/EDIT MAIN MENU **'//,
     ^           3X,'1. CCE.',5X,I3,/,
     ^           3X,'2. SCE.',5X,I3,/,
     ^           3X,'3. CER.',5X,I3,/,
     ^           3X,'4. SYSTEM LIFE (yrs):'X,I3,/,
```

```
     ^              3X,'5. DISCOUNT RATE (%):'X,F6.2,/,
     ^              3X,'6. EXIT MENU.',//,
     ^              3X,'ENTER (1 - [6]): ',$)
      READ (INN,'(I1)',ERR=1) IRPLY

      IF (IRPLY .EQ. 6 .OR. IRPLY .EQ. 0) THEN
        STATUS = LIB$SPAWN ( 'CLS')
        IF (CHANGE .EQ. .TRUE.) CALL WRITE
        CHANGE = .FALSE.
         RETURN
      ENDIF

      IF (IRPLY .EQ. 4) THEN

*  DETERMINE THE MINIMUM 'NYRS' CAN BE

      MINYRS = 0

*  CCE
******
      DO 6, I = 1, NCCE

        IF (CCE(I,6) .EQ. 2.0) THEN

                  M I N Y R S    =    M A X 0    ( M I N Y R S ,
INT(CCE(I,1)+CCE(I,2)+CCE(I,3)+CCE(I,4)))


        ENDIF

        DO 6, J = 1, CCENP(I)
         MINYRS = MAX0 (MINYRS, CCEYR(I,J))

6     CONTINUE

*  SCE
******
      DO 7, I = 1, NSCE

        IF (SCE(I,9) .EQ. 2.0) THEN

        MINYRS = MAX0 (MINYRS, INT(SCE(I,1) + SCE(I,2) + SCE(I,3) +
                         SCE(I,4)))

        ENDIF

        DO 7, J = 1, SCENP(I)
          MINYRS = MAX0 (MINYRS, SCEYR(I,J))
```

```
7     CONTINUE

*  CER
******
      DO 8, I = 1, NCER

        IF (SCE(I,14) .EQ. 2.0) THEN

                  M I N Y R S   =   M A X 0   ( M I N Y R S ,
INT(CER(I,1)+CER(I,2)+CER(I,3)+CER(I,4)))

        ENDIF

        DO 8, J = 1, CERNP(I)
          MINYRS = MAX0 (MINYRS, CERYR(I,J))

8     CONTINUE

9     WRITE (OUT,11) MINYRS, NY, NYRS
11    FORMAT (/,3X,'ENTER NEW SYSTEM LIFE (',I3,' - ',I3,' ['
     ^          ,I3,']): ',$)
      READ (INN,'(I3)',ERR=9) IRPLY

*  NO CHANGE IN 'NYRS'

      IF (IRPLY .EQ. 0) GO TO 1

*  CHECK FOR OUT OF RANGE ON 'NYRS'

      IF (IRPLY .LT. MINYRS .OR. IRPLY .GT. NY) THEN
        WRITE (OUT,12)
12      FORMAT (/,3X,'** ENTER A SYSTEM LIFE WITHIN LIMITS.**',/,
     ^             3X,'** LOWER LIMIT IS BASED ON EXISTING  **',/,
     ^             3X,'** DATA.                             **',/)

        GO TO 9
      ENDIF

*  ACCEPT NEW 'NYRS'

        NYRS = IRPLY

        CHANGE = .TRUE.
        GO TO 1

*  CHANGE IN DISCOUNT RATE
**************************

      ELSEIF (IRPLY .EQ. 5) THEN
```

```
14        WRITE (OUT,15)
15        FORMAT (/,3X,'ENTER THE DISCOUNT RATE (%): ',$)
          READ (INN,*,ERR=14) DSC

          IF (DSC .GT. 100.00 .OR. DSC .LT. 0.0) THEN

           WRITE (OUT,25)
25        FORMAT (//,3X,'*** DISCOUNT RATE IS LIMITED: [0.0 - 100.0]
     ^ ***')
           GO TO 14

          ENDIF

          CHANGE = .TRUE.
          GO TO 1

        ENDIF

        GO TO (100, 200, 300) IRPLY

*
*  CONSTANT COST ELEMENT
************************

100         STATUS = LIB$SPAWN ( 'CLS')

      WRITE (OUT, 105)
105   FORMAT (//,3X,'** CCE VIEW/EDIT MENU **',//,
     ^           3X,'1. VIEW/EDIT A CCE.',/,
     ^           3X,'2. ADD A CCE.',/,
     ^           3X,'3. DELETE A CCE.',/,
     ^           3X,'4. EXIT THIS MENU.',//,
     ^           3X,'ENTER (1 - [4]): ',$)
      READ (INN,'(I1)',ERR=100) IRPLY

      IF (IRPLY .EQ. 4 .OR. IRPLY .EQ. 0)  GO TO 1

      GO TO (110, 169, 170) IRPLY

*
*  VIEW/EDIT A CCE
*

110   DO 165, I = 1, NCCE

         STATUS = LIB$SPAWN ( 'CLS')

          CET = 'CCE'
```

```
114     WRITE (OUT, 115) I,CCE(I,5)
115     FORMAT (//,3X,'CONSTANT COST ELEMENT ',I2,/,
   ^            3X,'***************************',//,
   ^            3X,'1. COST:                      ',F14.0,/)

        IF (CCE(I,6) .EQ. 1.0) THEN

          WRITE (OUT,116) 2, CCENP(I)
116       FORMAT (/,3X,I1,'. NUMBER OF PAYMENTS: ',I2,//,
   ^               3X,'PAYMENT   YEAR   PERCENT',/,
   ^               3X,'-------   ----   -------',/)

          WRITE (OUT,117) (J, CCEYR(I,J), CCEPR(I,J), J = 1,
   ^      CCENP(I))
117       FORMAT(5X,I2,6X,I2,6X,F5.1)

        ELSE

         WRITE (OUT,119) 2, INT(CCE(I,1)), INT(CCE(I,2)),
   ^           INT(CCE(I,3)), INT(CCE(I,4))
119      FORMAT (/,3X,I1,'. TRAPEZOID ALLOCATION:',//,
   ^             3X,'   PHASE-IN PERIOD:      ',I3,/,
   ^             3X,'   CONSTANT PERIOD:      ',I3,/,
   ^             3X,'   PHASE-OUT PERIOD:     ',I3,/,
   ^             3X,'   YEAR PAYMENT STARTS: ',I3)

        ENDIF

        WRITE (OUT,120)
120     FORMAT(//,3X,'ENTER ( 1 - 2, [NEXT]): ',$)
        READ (INN,'(I1)',ERR=114) IRPLY

        IF (IRPLY .EQ. 1) THEN

          CHGCCE = 1
          CALL INCCE (I, CHGCCE)
          CHGCCE = 0
          CHANGE = .TRUE.
          STATUS = LIB$SPAWN ( 'CLS')
          GO TO 114

        ELSEIF (IRPLY .EQ. 2) THEN

          CHGCCE = 2
          CALL INCCE (I, CHGCCE)
          CHGCCE = 0
          CHANGE = .TRUE.
          STATUS = LIB$SPAWN ( 'CLS')
          GO TO 114
```

```
         ENDIF

165      CONTINUE

         IF (NCCE .EQ. 0) THEN
           WRITE (OUT, 166) CET
166        FORMAT (/,3X,'*** NO ',A3,' DATA EXIST ***',/)
         ENDIF

         GO TO 100

*
*    ADD A CONSTANT COST ELEMENT
*

169      ADDCCE = .TRUE.
         NCCE = NCCE + 1
         CALL INCCE (I, CHGCCE)
         CHANGE = .TRUE.
         GO TO 100

*
*    DELETE A CONSTANT COST ELEMENT
*

170      WRITE (OUT, 175) CET, NCCE
175      FORMAT (/,3X,'ENTER THE NUMBER OF THE ',A3,' TO DELETE '
     ^                  '(1 - ',I2,', [EXIT]): ',$)
         READ (INN,'(I2)',ERR=170) IRPLY

         IF (IRPLY .GT. NCCE .OR. IRPLY .EQ. 0) GO TO 100

180      WRITE (OUT, 185) CET, IRPLY
185      FORMAT (/,3X,'ARE YOU SURE YOU WANT TO DELETE ' ,A3,X,I2,
     ^                  '? (Y/[N]): ',$)
         READ (INN, '(A1)',ERR=180) RESP

         IF (RESP.NE.'Y'.AND.RESP.NE.'y') GO TO 100

         CHANGE = .TRUE.

*   CASE WHERE DELETING THE LAST OF THE CCE ENTRIES

         IF (IRPLY .EQ. NCCE) THEN
           NCCE = NCCE - 1
           GO TO 100
         ENDIF

*   CASE WHERE DELETING OTHER THAN THE LAST CCE
```

```
         NCCE = NCCE - 1

         DO 190, I = IRPLY, NCCE

           CCENP(I) = CCENP(I+1)

           DO 187, J = 1, CCENP(I)

             CCEYR(I,J) = CCEYR(I+1,J)
             CCEPR(I,J) = CCEPR(I+1,J)

187        CONTINUE

           DO 190, J = 1, 6

             CCE(I,J) = CCE(I+1,J)

190      CONTINUE

         GO TO 100

*
*  STOCHASTIC COST ELEMENT
*************************


200     STATUS = LIB$SPAWN ( 'CLS')

        CET = 'SCE'

        WRITE (OUT, 205)
205     FORMAT (//,3X,'** SCE VIEW/EDIT MENU **',//,
     ^             3X,'1. VIEW/EDIT A SCE.',/,
     ^             3X,'2. ADD A SCE.',/,
     ^             3X,'3. DELETE A SCE.',/,
     ^             3X,'4. EXIT THIS MENU.',//,
     ^             3X,'ENTER (1 - [4]): ',$)
        READ (INN,'(I1)',ERR=200) IRPLY

        IF (IRPLY .EQ. 4 .OR. IRPLY .EQ. 0)  GO TO 1

        GO TO (210, 240, 270) IRPLY


*
*  VIEW/EDIT STOCHASTIC COST ELEMENTS
*

210     DO 235, I=1, NSCE
```

```
          STATUS = LIB$SPAWN ( 'CLS')

214       WRITE (OUT, 215) I
215       FORMAT (//,3X,'STOCHASTIC COST ELEMENT ',I2,/,
     ^              3X,'*****************************',/)

          IF (SCE(I,5) .EQ. 1.0) THEN

219         WRITE (OUT, 220) SCE(I,8)
220         FORMAT (3X,'1. DISTRIBUTION:          TRIANGULAR',/,
     ^              6X,'MODE VALUE:',8X,F14.0)

          ELSEIF (SCE(I,5) .EQ. 2.0) THEN

            WRITE (OUT,225) INT(SCE(I,8))
225         FORMAT (3X,'1. DISTRIBUTION:          BETA',/,
     ^              3X,'   BETA TYPE:',15X,I1)

          ENDIF

          WRITE (OUT,230) SCE(I,6), SCE(I,7)
230       FORMAT (  6X,'LOWER LIMIT:',7X,F14.0,/,
     ^              6X,'UPPER LIMIT:',7X,F14.0)

          IF (SCE(I,9) .EQ. 1.0) THEN

            WRITE (OUT,116) 2, SCENP(I)

            WRITE (OUT,117) (J, SCEYR(I,J), SCEPR(I,J), J = 1,
     ^            SCENP(I))

          ELSEIF (SCE(I,9) .EQ. 2.0) THEN

           WRITE (OUT,119) 2,INT(SCE(I,1)),INT(SCE(I,2)),INT(SCE(I,3)),
     ^                       INT(SCE(I,4))

          ENDIF

          WRITE (OUT,120)

          READ (INN,'(I1)',ERR=214) IRPLY

          IF (IRPLY .EQ. 1) THEN

            CHGSCE = 1
            CALL INSCE (I,CHGSCE)
            CHGSCE = 0
            CHANGE = .TRUE.
            STATUS = LIB$SPAWN ( 'CLS')
```

```
          GO TO 214

        ELSEIF (IRPLY .EQ. 2) THEN

          CHGSCE = 2
          CALL INSCE (I,CHGSCE)
          CHGSCE = 0
          CHANGE = .TRUE.
          STATUS = LIB$SPAWN ( 'CLS')
          GO TO 214

        ENDIF

235    CONTINUE

        IF (NCCE .EQ. 0) THEN
          WRITE (OUT, 166) CET
        ENDIF

      GO TO 200

*
*   ADD A STOCHASTIC COST ELEMENT
*

240     ADDSCE = .TRUE.
        CHGSCE = 0
        NSCE = NSCE + 1
        CALL INSCE (I, CHGSCE)
        CHANGE = .TRUE.
        ADDSCE = .FALSE.
        GO TO 200

*
*   DELETE A STOCHASTIC COST ELEMENT
*

270     WRITE (OUT, 175) CET, NSCE
        READ (INN,'(I2)',ERR=270) IRPLY

        IF (IRPLY .EQ. 0 .OR. IRPLY .GT. NSCE) GO TO 200

280     WRITE (OUT, 185) CET, IRPLY
        READ (INN, '(A1)',ERR=280) RESP

        IF (RESP.NE.'Y'.AND.RESP.NE.'y') GO TO 200

        CHANGE = .TRUE.
```

```
*   CASE WHERE DELETING THE LAST SCE

          IF (IRPLY .EQ. NSCE) THEN
            NSCE = NSCE - 1
            GO TO 200
          ENDIF

*   CASE WHERE DELETING OTHER THAN THE LAST SCE

          NSCE = NSCE - 1

          DO 290, I = IRPLY, NSCE

            SCENP(I) = SCENP(I+1)

            DO 287, J = 1, SCENP(I)

              SCEYR(I,J) = SCEYR(I+1,J)
              SCEPR(I,J) = SCEPR(I+1,J)

287         CONTINUE

              DO 290, J = 1, 9

              SCE(I,J) = SCE(I+1,J)

290         CONTINUE

          GO TO 200

*
*   COST ESTIMATING RELATIONSHIP
*******************************

300           STATUS = LIB$SPAWN ( 'CLS')
        CET = 'CER'
        CHGCER = 0

        WRITE (OUT, 305)
305     FORMAT (//,3X,'** CER VIEW/EDIT MENU **',//,
     ^             3X,'1. VIEW/EDIT A CER.',/,
     ^             3X,'2. ADD A CER.',/,
     ^             3X,'3. DELETE A CER.',/,
     ^             3X,'4. EXIT THIS MENU.',//,
     ^             3X,'ENTER (1 - [4]): ',$)
        READ (INN,'(I1)',ERR=200) IRPLY

        IF (IRPLY .EQ. 4 .OR. IRPLY .EQ. 0)  GO TO 1
```

```
      GO TO (310, 440, 570) IRPLY

*
*  VIEW/EDIT COST ESTIMATING RELATIONSHIPS
*

310     DO 439, I=1, NCER

311       STATUS = LIB$SPAWN ( 'CLS')

        WRITE (OUT, 315) I
315     FORMAT (//,3X,'COST ESTIMATING RELATIONSHIP ',I2,/,
     ^             3X,'**********************************',/)

        IF (CER(I,8) .EQ. 0.0) THEN

          WRITE (OUT, 317)
317       FORMAT (3X,'1. SCALAR VALUE:  1; FIXED',/)

        ELSE

          IF (CER(I,9) .EQ. 1.0) THEN

            WRITE (OUT, 319)INT(CER(I,12)),INT(CER(I,10)),
     ^            INT(CER(I,11))
319         FORMAT (3X,'1. SCALAR DISTRIBUTION:  TRIANGULAR',/,
     ^              3X,'     MODE VALUE:         ',I14,/,
     ^              3X,'     LOW VALUE:          ',I14,/,
     ^              3X,'     HIGH VALUE:         ',I14,/)

          ELSEIF (CER(I,9) .EQ. 2.0) THEN

            WRITE (OUT, 320)INT(CER(I,12)),INT(CER(I,10)),
     ^                      INT(CER(I,11))
320         FORMAT (3X,'1. SCALAR DISTRIBUTION:  BETA',/,
     ^              3X,'     BETA TYPE:          ',I1,/,
     ^              3X,'     LOW VALUE:          ',I14,/,
     ^              3X,'     HIGH VALUE:         ',I14,/)

          ELSEIF (CER(I,9) .EQ. 3.0) THEN

            WRITE (OUT,321) INT(CER(I,12))
321         FORMAT (3X,'1. SCALAR DISTRIBUTION:  CONSTANT',/,
     ^              3X,'     SCALAR VALUE        ',I14,/)

          ENDIF

        ENDIF
```

```
      IF (CER(I,13) .EQ. 1.0) THEN
         BASE = 'WHOLE DOLLAR'
      ELSEIF (CER(I,13) .EQ. 2.0) THEN
         BASE = 'NATURAL LOG'
      ELSEIF (CER(I,13) .EQ. 3.0) THEN
         BASE = 'LOG BASE 10'
      ENDIF


      WRITE (OUT, 322) INT(CER(I,7)), BASE, INT(CER(I,5)),
   ^                 CER(I,6), VAR(I)
322   FORMAT (3X,'2. VIEW/EDIT COST DRIVERS & SLOPE PARAM.:
   ^ ',I2,/,
   ^          3X,'3. ESTIMATION BASED ON:  '15X,A12,/,
   ^          3X,'4. SAMPLE SIZE (n):      '15X,I3,/,
   ^          3X,'5. INTERCEPT PARAM. (b0):',15X,F18.7,/,
   ^          3X,'6. VARIANCE (Ssq):  '20X,F18.7,/,
   ^          3X,'7. VIEW/EDIT COST ALLOCATION (TRAP/PRCNT).',/,
   ^          3X,'8. EXIT THIS MENU.',//,
   ^          3X,'ENTER (1 - 8, [NEXT]): ',$)

      READ (INN, '(I1)',ERR=311) IRPLY

      IF (IRPLY .LT. 0 .OR. IRPLY .GT. 8)THEN
         PRINT*,'*** INPUT OUT OF RANGE ***'
         PRINT*,'******* TRY AGAIN ********'
         GO TO 311
      ENDIF

      IF (IRPLY .EQ. 8) GO TO 300
*
*  CHANGE INFORMATION REGARDING THE SCALAR
*
    IF (IRPLY .EQ. 1) THEN
       CHGCER = 1
       ADDCER = .FALSE.
       CALL INCER (I, CHGCER)
       CHGCER = 0
       CHANGE = .TRUE.
       GO TO 311
    ENDIF

*
*  DISPLAY COST DRIVERS AND PARAMETERS
*

    IF (IRPLY .EQ. 2) THEN

*  HEADER FOR TRIANGULAR DISTRIBUTION
```

```
323     STATUS = LIB$SPAWN ( 'CLS')
        WRITE (OUT,324) I
324              FORMAT    (/,28X,'DRIVER    INFORMATION    FOR    CER
',I2,/,28X,29('-'),//,
     ^  10X,'SLOPE',10X,'CONST',9X,'LW-BND',9X,'UP-BND',11X,'MODE',
     ^  2X,'BETA',/,10X,5('-'),10X,5('-'),9X,6('-'),9X,6('-'),
     ^  11X,4('-'),2X,4('-'))

*   WRITE PARAMETERS AND DRIVERS TO SCREEN

       DO 335, J=1,CER(I,7)

*   CONSTANT DRVR
****************

         IF (DRVR(I,J,4) .EQ. 1.0) THEN

*   WHOLE NUMBER TYPE DISPLAY

          IF (CER(I,13) .EQ. 1.0) THEN
           WRITE (OUT,325) J, PARA(I,J), DRVR(I,J,3)
325        FORMAT (X,I2,X,F11.1,X,F14.1)

*   LOG10 OR Ln TYPE DISPLAY

          ELSE

           WRITE (OUT,326) J, PARA(I,J), DRVR(I,J,3)
326        FORMAT (X,I2,X,F11.7,X,F14.1)

          ENDIF

*   TRIANGULAR DIST. DRVR
************************

         ELSEIF (DRVR(I,J,4) .EQ. 2.0) THEN

*   WHOLE NUMBER TYPE DISPLAY

          IF (CER(I,13) .EQ. 1.0) THEN

           WRITE (OUT,327)J,PARA(I,J),DRVR(I,J,1),DRVR(I,J,2),
     ^                    DRVR(I,J,3)
327        FORMAT (X,I2,X,F11.1,15X,3(X,F14.1))

*   LOG10 OR Ln TYPE DISPLAY

          ELSE
```

```
          WRITE (OUT,328)J,PARA(I,J),DRVR(I,J,1),DRVR(I,J,2),
     ^                        DRVR(I,J,3)
328       FORMAT (X,I2,X,F11.7,15X,3(X,F14.1))

         ENDIF

*  BETA DIST. DRVR
******************

        ELSEIF (DRVR(I,J,4) .EQ. 3.0) THEN

*  WHOLE NUMBER TYPE DISPLAY

         IF (CER(I,13) .EQ. 1.0) THEN

          WRITE (OUT,329)J,PARA(I,J),DRVR(I,J,1),DRVR(I,J,2),
     ^                  INT(DRVR(I,J,3))
329       FORMAT (X,I2,X,F11.1,15X,2(X,F14.1),18X,I2)

*  LOG10 OR Ln TYPE DISPLAY

         ELSE

          WRITE (OUT,330)J,PARA(I,J),DRVR(I,J,1),DRVR(I,J,2),
     ^                  INT(DRVR(I,J,3))
330       FORMAT (X,I2,X,F11.7,15X,2(X,F14.1),18X,I2)

         ENDIF

        ENDIF

335     CONTINUE


*  PROMPT FOR CHANGES TO PARAMETERS AND DRIVERS

      WRITE (OUT,336)
336   FORMAT (//,3X,'1. CHANGE A COST DRIVER ONLY.',/,
     ^           3X,'2. CHANGE A SLOPE PARAMETER ONLY.',//,
     ^           3X,'ENTER (1 - 2, [EXIT]): ',$)
      READ (INN,'(I1)',ERR=323) IRPLY

      IF (IRPLY .LT. 1 .OR. IRPLY .GT. 2) GO TO 311
*
*  CHANGE A COST DRIVER AND SLOPE PARAMETER
*

       IF (IRPLY .EQ. 1) THEN
```

```
         WRITE (OUT, 337) INT(CER(I,7))
337      FORMAT (/,3X,'ENTER THE NUMBER OF THE COST DRIVER TO
     ^ CHANGE (1 - ',I2,', [EXIT]): ',$)
         READ (INN, '(I2)',ERR=311) IRPLY

         IF (IRPLY .EQ. 0) GO TO 311

          CHGCER = 2
          ADDCER = .FALSE.
          CALL INCER (I,CHGCER)
          CHGCER = 0
          CHANGE = .TRUE.

          GO TO 323

       ELSEIF (IRPLY .EQ. 2) THEN

340     WRITE (OUT,341) INT(CER(I,7))
341     FORMAT (/,3X,'ENTER LINE NUMBER OF SLOPE PARAM. TO CHANGE',
     ^  ' (1 - ',I2,', [EXIT]): ',$)
        READ (INN,'(I2)',ERR=340) IRPLY

        IF (IRPLY .LT. 1 .OR. IRPLY .GT. INT(CER(I,7))) THEN

          GO TO 323
        ENDIF

        TEMP = PARA(I,IRPLY)
        WRITE (OUT,343) IRPLY
343     FORMAT(/,3X,'ENTER NEW SLOPE PARAM. ',I2,' VALUE: ',$)
        READ(INN,*) PARA(I,IRPLY)

* ZERO PARAMETERS NOT ALLOWED

        IF (PARA(I,IRPLY) .EQ. 0.0) THEN
          PARA(I,IRPLY) = TEMP
          GO TO 323
        ENDIF

        IF (TEMP .NE. PARA(I,IRPLY)) CHANGE = .TRUE.

          GO TO 323

       ENDIF
      ENDIF

*
*  CHANGE THE ESTIMATING BASE OF THE A CER DRIVES.
*
```

```
      IF (IRPLY .EQ. 3) THEN

344     WRITE (OUT, 345)
345   FORMAT (//,3X,'** YOU SHOULD NOT CHANGE THE BASE OF YOUR
     ^ **',/,
     ^           3X,'** DATA UNLESS AN ERROR WAS MADE IN FILE
     ^ **',/,
     ^           3X,'** CREATION.
     ^ **',//,
     ^           3X,'** DO YOU STILL WISH TO CHANGE THE BASE?',
     ^              ' (Y/[N]): ',$)
        READ (INN,'(A1)',ERR=344)RESP

        IF (RESP.NE.'Y'.AND.RESP.NE.'y') GO TO 311

        CHGCER = 3

        TEMP = CER(I,13)
        ADDCER = .FALSE.
        CALL INCER(I, CHGCER)
        CHGCER = 0

        IF (CER(I,13) .EQ. TEMP) GO TO 311

        CHANGE = .TRUE.

        GO TO 311

      ENDIF

*
*  CHANGE THE SAMPLE SIZE OF CER(I)
*

      IF (IRPLY .EQ. 4) THEN

365     WRITE (OUT, 366) I,INT(CER(I,5))
366     FORMAT (/,3X,'ENTER A NEW SAMPLE SIZE FOR CER ',I2,
     ^                ' [OLD: ',I3,']: ',$)

         READ (INN,'(I3)',ERR=365) IRPLY
         IF (IRPLY .EQ. 0) GO TO 311
          CHANGE = .TRUE.
          CER (I,5) = IRPLY
          GO TO 311

      ENDIF
```

```
*
*  CHANGE VALUE OF INTERCEPT TERM
*

      IF (IRPLY .EQ. 5) THEN

        TEMP = CER(I,6)

369     WRITE (OUT, 370) I
370     FORMAT (/,3X,'ENTER VALUE OF INTERCEPT TERM FOR CER ',I2,'
     ^ ([0] IF NONE): ',$)
        READ (INN,*,ERR=369) RRPLY

        IF (RRPLY .LT. 0.0) THEN
         PRINT*
         PRINT*,' ** INTERCEPT MUST BE GREATER THAN OR EQUAL TO 0.0
     ^ **'
         GO TO 369
        ENDIF

        CER(I,6) = RRPLY

        IF (TEMP .EQ. RRPLY) GO TO 311

        CHANGE = .TRUE.

*  IF THE INTERCEPT WAS SET TO 0, THEN REMOVE IT FROM THE VAR/COV
*  MATRIX

        IF (CER(I,6) .EQ. 0.0 .AND. TEMP .NE. 0.0) THEN

          DO 372, K = 1, CER(I,7) + 1

             VRCV(I,1,K) = 0.0
             VRCV(I,K,1) = 0.0

372       CONTINUE

        ENDIF

          GO TO 311

      ENDIF

*
*  CHANGE ESTIMATED VARIANCE.
*

      IF (IRPLY .EQ. 6) THEN
```

```
        TEMP = VAR(I)

376     WRITE (OUT,377) I
377     FORMAT (/,3X,'ENTER NEW SAMPLE VARIANCE FOR CER ',I2,':
     ^ ',$)
        READ (INN,*,ERR=376) VAR(I)

        IF (TEMP .EQ. VAR(I)) GO TO 311

        CHANGE =.TRUE.
        GO TO 311

      ENDIF

*
*  CHANGE THE COST ALLOCATION OF THE CER
*

      IF (IRPLY .EQ. 7) THEN

         STATUS = LIB$SPAWN ( 'CLS')

*  DISPLAY PHASING OR PERCENTAGE DATA

380     IF (CER(I,14) .EQ. 2.0 ) THEN

          WRITE (OUT,315) I
          WRITE (OUT,119) 1,INT(CER(I,1)),INT(CER(I,2)),
     ^                    INT(CER(I,3)),
     ^                    INT(CER(I,4))

        ELSEIF (CER(I,14) .EQ. 1.0) THEN

          WRITE (OUT,315) I
          WRITE (OUT, 116) 1, CERNP(I)
          WRITE (OUT,117)(J, CERYR(I,J), CERPR(I,J), J=1,CERNP(I))

        ENDIF

389     WRITE (OUT,390)
390     FORMAT (3X,'ENTER (1 = NEW COST ALLOCATION,
     ^  [NEXT]): ',$)
        READ (INN,'(I1)',ERR=389) IRPLY

        IF (IRPLY .EQ. 1 ) THEN

*  PROMPT FOR NEW COST ALLOCATION (TRAPEZOID/PERCENTAGE)

           CHGCER = 4
```

```
            ADDCER = .FALSE.
            CALL INCER (I, CHGCER)
            CHGCER = 0
            CHANGE = .TRUE.
            STATUS = LIB$SPAWN ('CLS')
            GO TO 380

         ENDIF

        GO TO 311

       ENDIF

439    CONTINUE

         IF (NCCE .EQ. 0) THEN
           WRITE (OUT, 166) CET
         ENDIF


*  ADD/VIEW SEQUENCE COMPLETED. RETURN TO MENU

       GO TO 300

*
*  ADD A CER
*

440     NCER = NCER + 1
        CHGCER = 0
        ADDCER = .TRUE.
        CALL INCER (I, CHGCER)
        CHANGE = .TRUE.
        ADDCER = .FALSE.

* EDIT/VIEW NEW CER

        I = NCER
        GO TO 311

*
*  DELETE A CER
*

570      WRITE (OUT, 175) CET, NCER
         READ (INN,'(I2)',ERR=570) IRPLY

         IF (IRPLY .EQ. 0 .OR. IRPLY .GT. NCER) GO TO 300
```

```
580     WRITE (OUT, 185) CET, IRPLY
        READ (INN, '(A1)',ERR=580) RESP

        IF (RESP.NE.'Y'.AND.RESP.NE.'y') GO TO 300

        CHANGE = .TRUE.

*  CASE WHERE THE LAST CER IS TO BE DELETED

        IF (IRPLY .EQ. NCER) THEN
            NCER = NCER - 1
            GO TO 300
        ENDIF

*  CASE WHERE A CER OTHER THEN THE LAST CER IS TO BE DELETED

        NCER = NCER - 1

        DO 595, I = IRPLY, NCER

          CERNP(I) = CERNP(I+1)

          DO 591, J = 1, CERNP(I)

            CERYR(I,J) = CERYR(I+1,J)
            CERPR(I,J) = CERPR(I+1,J)

591       CONTINUE

          VAR (I) = VAR (I + 1)

          DO 592, J = 1, 14

            CER(I,J) = CER(I+1,J)

592       CONTINUE

          DO 595, J = 1, INT(CER(I,7))

            DRVR (I,J,1) = DRVR (I+1,J,1)
            DRVR (I,J,2) = DRVR (I+1,J,2)
            DRVR (I,J,3) = DRVR (I+1,J,3)

            PARA (I,J) = PARA (I+1,J)

            DO 595, K = 1, INT(CER(I,7)+1.0)

            VRCV (I,J,K) = VRCV (I+1,J,K)
```

```
595     CONTINUE

        GO TO 300

        END

*******************************************************************
*     SUBROUTINE EXEC: CALCULATES NRV COST ESTIMATIONS BASED ON   *
*     THE DATA OF CCEs, SCEs, AND CERs.                           *
*******************************************************************

      SUBROUTINE EXEC

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER    NCCE, NSCE, NCER, IRPLY, J, I, NYRS, CCEYR(NC,NY)
      INTEGER    CCENP(NC), SCEYR(NC,NY), SCENP(NC), CERYR(NC,NY)
      INTEGER    CERNP(NC), NRV, OUT, INN

      REAL       CCE(NC,6), SCE(NC,9), CER(NC,14), DRVR(NC,ND,4),
     ^           RRPLY, DSC, VRCV(NC,NC,NC), VAR(NC), PARA(NC,ND),
     ^           CCEPR(NC,NY),
     ^           SCEPR(NC,NY), CERPR(NC,NY), SUMCST(NV), TRPCST,
     ^           CCEDSC, SCEDSC, CERDSC, YRCST, H, A, B, D, DF,
     ^           SCST, RR, MD, PI, RU(NV), RN(NV), RX, PVCST(4),
     ^           SMLCST, LRGCST

      REAL       XSTR(21), SB(21), SBSQ, SD, CHAT, COST

      LOGICAL    DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /LOGIC/ DATA, CHANGE, ADDCCE, ADDSCE, ADDCER
      COMMON /SRT/ SUMCST, NRV

      COMMON /TRMNL/ OUT, INN

      CHARACTER*29 CLS
      CHARACTER*16 FILEOUT2
      CHARACTER*15 FILENAME, FILEOUT
      CHARACTER*9 ACCOUNT
      CHARACTER*3 CET
      CHARACTER*1 RESP

      COMMON /PDATA/ CCE, SCE, CER, PARA, DSC, RRPLY, IRPLY,
     ^NCCE, NSCE, NCER, NYRS, VAR, VRCV, DRVR, CCEPR,
     ^CCEYR, CCENP, SCEPR, SCEYR, SCENP, CERPR, CERYR, CERNP

      COMMON /NAMES/ ACCOUNT, FILENAME, RESP, CET, CLS
```

```
      EXTERNAL  RNSET, RNUN, RNNOA, RNCHI

*  SET THE SEED NUMBER EQUAL TO THE CLOCK.

      CALL RNSET(0)

*  GET THE NUMBER OF RANDOM VARIABLES DESIRED.

1     WRITE (OUT,2) NV
2     FORMAT (/,3X,'ENTER THE NUMBER OF RANDOM VARIABLES DESIRED',
     ^' (Max: 'I4'): ',$)
      READ (INN,'(I3)',ERR=1) NRV

      IF (NRV .GT. NV .OR. NRV .LT. 0) GO TO 1

*  GET NRV MANY RANDOM UNIFORM AND NRV MANY RANDOM NORMAL NUMBERS

      CALL RNUN(NRV,RU)
      CALL RNNOA(NRV,RN)

*  COLLECT TOTAL LIFE CYCLE COST APPROXIMATION NRV TIMES
********************************************************


*  CONSTANT COST ELEMENTS CALCULATIONS
*  CALCULATE CCE PHASING (TRAP OR PERCENT) AND DISCOUNTING
**********************************************************

*  COMPUTATION ON ALL CCEs IS DONE ONE TIME SINCE THE RESULT IS
*  CONSTANT BETWEEN COMPUTATIONS.

       CCEDSC = 0.0

       DO 50, J = 1, NCCE

*
*  PERCENT METHOD PHASE-IN
*
        IF (CCE(J,6) .EQ. 1.0) THEN

           DO 40, I = 1, CCENP(J)

             YRCST = (CCEPR(J,I)/100) * CCE(J,5)

             CCEDSC = CCEDSC + YRCST / ((1 + (DSC/100))**CCEYR(J,I))

40         CONTINUE
```

```
*
*  TRAPEZOID METHOD PHASE-IN
*

        ELSE

           CALL TRAP(CCE(J,1),CCE(J,2),CCE(J,3),CCE(J,4),
     ^     CCE(J,5),DSC,TRPCST)

            CCEDSC = CCEDSC + TRPCST

        ENDIF

50      CONTINUE

*  SCE AND CER ARE STOCHASTIC THUS REPETITION OF COMPUTATION IS
REQUIRED

      DO 600, K = 1, NRV

       SCEDSC = 0.0
       CERDSC = 0.0

*  STOCHASTIC COST ELEMENTS CALCULATIONS
*  GET MONTI CARLO VALUE BASED ON DISTRIBUTION (BETA OR TRIANGULAR),
*  THEN CALCULATE SCE PHASING AND DISCOUNTING
*******************************************************************


      DO 100, J = 1, NSCE

*  FOR TRIANGULAR DISTRIBUTION

        IF (SCE(J,5) .EQ. 1.0) THEN

            A = SCE(J,6)
            B = SCE(J,7)
            D = SCE(J,8)

            IF (RU(K) .LE. (D-A)/(B-A)) THEN

              SCST = A + SQRT((D-A) * (B-A) * RU(K))

            ELSE

              SCST = B - SQRT((B-D) * (B-A) * (1-RU(K)))

            ENDIF
```

```
*  FOR BETA DISTRIBUTION

         ELSEIF (SCE(J,5) .EQ. 2.0) THEN

             CALL BETA(SCE(J,6),SCE(J,7),SCE(J,8),SCST)

         ENDIF

*
*  TRAPEZOID METHOD FOR DISPERSION OF COST
*
         IF (SCE(J,9) .EQ. 2.0) THEN

          CALL TRAP(SCE(J,1),SCE(J,2),SCE(J,3),SCE(J,4),SCST,DSC,
     ^              TRPCST)

          SCEDSC = SCEDSC + TRPCST

*
*  FOR PERCENTAGE METHOD FOR DISPERSION OF COST
*

         ELSEIF (SCE(J,9) .EQ. 1.0) THEN

            DO 95 I = 1, SCENP(J)

              YRCST = SCEPR(J,I)/100 * SCST

                SCEDSC    =    SCEDSC    +    YRCST    /    (1    +
     ^           (DSC/100))**(SCEYR(J,I))

95          CONTINUE

         ENDIF

100   CONTINUE

*   COST ESTIMATING RELATIONSHIPS CALCULATIONS
*
*******************************************************************

      DO 500, J = 1, NCER

*  'CHAT' IS SET TO THE VALUE OF THE INTERCEPT FIRST SINCE THE
*  INTERCEPT IS FIXED.  THE REMAINING PARAMETERS AND DRIVERS ARE
*  ADDED TO 'CHAT' LATER DEPENDING ON EACH DRIVER'S DIST. TYPE.
```

```
      CHAT = CER(J,6)

*  SET THE IMPLIED VALUE OF THE INTERCEPT'S DRIVER IN THE DESIGN
*  MATRIX

      XSTR(1) = 0

      IF (CER(J,6) .GT. 0) XSTR(1) = 1

*  SET 'XSTR' EQUAL TO DRIVER WHILE ACCOUNTING FOR DISTRIBUTION

      DO 210, I = 1, CER(J,7)

*  CONSTANT DRIVER

        IF (DRVR(J,I,4) .EQ. 1.0) THEN

          XSTR(I+1) = DRVR(J,I,3)

*  TRIANGULAR DIST. DRIVER

        ELSEIF (DRVR(J,I,4) .EQ. 2.0) THEN

           A = DRVR(J,I,1)
           B = DRVR(J,I,2)
           D = DRVR(J,I,3)

           IF (RU(K) .LE. (D-A)/(B-A)) THEN

             XSTR(I+1) = A + SQRT((D-A) * (B-A) * RU(K))

           ELSE

             XSTR(I+1) = B - SQRT((B-D) * (B-A) * (1-RU(K)))

           ENDIF

*  BETA DIST. DRIVER

        ELSEIF (DRVR(J,I,4) .EQ. 3.0) THEN

         CALL BETA(DRVR(J,I,1),DRVR(J,I,2),DRVR(J,I,3),XSTR(I+1))

        ENDIF

*  TAKE THE Ln OR LOG10 OF XSTR IF REQUIRED

        IF (CER(J,13).EQ.2.0) XSTR(I+1) = ALOG(XSTR(I+1))
```

```
         IF (CER(J,13).EQ.3.0) XSTR(I+1) = ALOG10(XSTR(I+1))
*  SUM-UP MEAN COST; 'CHAT'.
         CHAT = CHAT + PARA(J,I) * XSTR(I+1)
210     CONTINUE
        SBSQ = 0.0
*  DETERMINE THE STANDARD DEVIATION
        DO 250, I = 1, CER(J,7)+1
          SB(I) = 0.0
          DO 240, L = 1, CER(J,7)+1
            SB(I) = SB(I) + XSTR(L) * VRCV(J,L,I)
240       CONTINUE
          SBSQ = SBSQ + SB(I) * XSTR(I)
250     CONTINUE
        SD = SQRT(VAR(J) + SBSQ)
*  DEGREE OF FREEDOM
********************
*   W/ INTERCEPT TERM
        IF (CER(J,6) .GT. 0.0) THEN
          DF = CER(J,5) - (CER(J,7) + 1)
*   W/OUT INTERCEPT TERM
        ELSE
         DF = CER(J,5) - CER(J,7)
        ENDIF
*  USE NORMAL DIST. FOR DF >= 30
        IF (DF .GE. 30) THEN
```

```
         COST = CHAT + (RN(K)*SD)
*  USE STUDENT-T DIST. FOR DF < 30
      ELSE
        CALL RNCHI(1,DF,RX)
        COST = CHAT + (SD * RN(K) * SQRT(DF/RX))
      ENDIF
*  ADJUST COST DEPENDING ON THE BASE USED
*****************************************
*  FOR LOG BASE e (NATURAL LOG)
      IF (CER(J,13) .EQ. 2.0) THEN
        COST = EXP(COST)
*  FOR LOG BASE 10
      ELSEIF (CER(J,13) .EQ. 3.0) THEN
        COST = 10**(COST)
      ENDIF
*  MULTIPLY COST TIMES SCALAR VALUE
***********************************
*  FOR FIXED SCALAR VALUE
      IF (CER(J,9) .EQ. 3.0) THEN
        COST = CER(J,12) * COST
*  FOR BETA DIST. SCALAR
      ELSEIF (CER(J,9) .EQ. 2.0) THEN
        CALL BETA(CER(J,10),CER(J,11),CER(J,12),RR)
        COST = RR * COST
*  FOR TRIANG. DIST. SCALAR
      ELSEIF (CER(J,9) .EQ. 1.0) THEN
```

```
          A = CER(J,10)
          B = CER(J,11)
          D = CER(J,12)

          IF (RU(K) .LE. (D-A)/(B-A)) THEN

            COST = (A + SQRT((D-A) * (B-A) * RU(K))) * COST

          ELSE

            COST = (B - SQRT((B-D) * (B-A) * (1-RU(K)))) * COST

          ENDIF

      ENDIF

*  DISTRIBUTION OF PAYMENT
**************************

*  FOR TRAPEZOIDAL DIST. OF PAYMENTS

      IF (CER(J,14) .EQ. 2.0) THEN

       CALL TRAP(CER(J,1),CER(J,2),CER(J,3),CER(J,4),COST,DSC,
     ^          TRPCST)

       CERDSC = CERDSC + TRPCST

*  FOR DISTRIBUTION OF PAYMENTS BASED ON PERCENTAGE DISTRIBUTION

      ELSEIF (CER(J,14) .EQ. 1.0) THEN

           DO 300 I = 1, CERNP(J)

             YRCST = CERPR(J,I)/100 * COST

             CERDSC = CERDSC + YRCST / (1 +
     ^             (DSC/100))**(CERYR(J,I))

300        CONTINUE

      ENDIF

500   CONTINUE
```

```
*  COLLECT THE SUM OF THE DISCOUNTED COST ELEMENTS' VALUE
*********************************************************

        SUMCST(K) = CCEDSC + SCEDSC + CERDSC

* END OF COLLECTING ONE APPROXIMATION
************************************

600   CONTINUE

*  SORT THE COLLECTION OF SUMCOST

      CALL SORT

*  OBTAIN THE 85,90,95 & 99% CONFIDENCE INTERVAL OF PRESENT VALUE
*  OF THE SYSTEM'S COST

      PVCST(1) = SUMCST(NINT(NRV*.85))
      PVCST(2) = SUMCST(NINT(NRV*.90))
      PVCST(3) = SUMCST(NINT(NRV*.95))
      PVCST(4) = SUMCST(NINT(NRV*.99))
      SMLCST   = SUMCST(1)
      LRGCST   = SUMCST(NRV)

*  CALCULATE THE ANNUAL PAYMENT BASED ON DISC RATE AND SYSTEM LIFE

      DO 605, J = 1, NRV
      SUMCST(J) =   SUMCST(J) * ((DSC/100)*(1+DSC/100)**NYRS)/
     ^              (((1+DSC/100)**NYRS)-1)

605   CONTINUE

        J = INDEX (ACCOUNT,' ')
        J=J-1

      FILEOUT = ACCOUNT ( :J)//'_LC.OUT'
      FILEOUT2 = ACCOUNT( :J)//'_SAS.DAT'

      OPEN (1, FILE=FILEOUT, STATUS='NEW')
      OPEN (2, FILE=FILEOUT2, STATUS='NEW')

*  WRITE IN THE OUT FILE 5 ACROSS ALL OF THE SUMCST ARRAY

      DO 610, J = 1, NRV, 5

        WRITE (2,'(5(X,F15.0))')(SUMCST(K), K = J, J+4)

610   CONTINUE
```

```
*  WRITE IN THE OUT-FILE THE INDICATED TOP PERCENTAGES OF SUMCST
*  ANNUAL COST OF THE SYSTEM, TOTAL COST, MIN AND MAX COST

      WRITE (1, 630) NYRS, DSC, SUMCST(NINT(NRV*.85)),PVCST(1),
     ^ SUMCST(NINT(NRV*.90)), PVCST(2), SUMCST(NINT(NRV*.95)),
     ^ PVCST(3), SUMCST(NINT(NRV*.99)), PVCST(4)

630   FORMAT (//,3X,I3,' YEARS @ ',F5.2, '% DISCOUNT RATE.',//,
     ^                3X,'% CONFIDENCE',5X,'ANNUAL COST',14X,'TOTAL
COST',/,
     ^           3X,12('-'),5X,11('-'),14X,10('-'),//,
     ^           8X,'85',6X,F15.0,6X,F18.0,/,
     ^           8X,'90',6X,F15.0,6X,F18.0,/,
     ^           8X,'95',6X,F15.0,6X,F18.0,/,
     ^           8X,'99',6X,F15.0,6X,F18.0)


      WRITE (1, 640) SMLCST, LRGCST
640   FORMAT (/,3X,'MINIMUM POSSIBLE',18X,F18.0,/,
     ^          3X,'MAXIMUM POSSIBLE',18X,F18.0)

*  CLOSE THE OUT-FILE

      CLOSE (1, STATUS='KEEP')
      CLOSE (2, STATUS='KEEP')

      STATUS = LIB$SPAWN('CLS')

      WRITE (OUT, 650) FILEOUT
650   FORMAT (/,3X,'!!! THE OUT-FILE NAME IS: ',A15,' !!!')

      WRITE (OUT, 655) FILEOUT2
655   FORMAT (//,3X,'!!! THE SAS-FILE NAME IS: ',A16,' !!!')


*  WRITE TO SCREEN END-DATA

      WRITE (OUT,630) NYRS, DSC, SUMCST(NINT(NRV*.85)),PVCST(1),
     ^ SUMCST(NINT(NRV*.90)), PVCST(2), SUMCST(NINT(NRV*.95)),
     ^ PVCST(3), SUMCST(NINT(NRV*.99)), PVCST(4)

      WRITE (OUT, 640) SMLCST, LRGCST
      WRITE (OUT, 660)
660   FORMAT (//,3X,'[ENTER RETURN TO CONTINUE]: ',$)
      READ (INN,*)

      STATUS = LIB$SPAWN('CLS')
      RETURN
      END
```

```
*******************************************************************
*    SUBROUTINE BETA: RETURNS A RANDOM NUMBER FROM A BETA DIST.   *
*******************************************************************

      SUBROUTINE BETA (A,B,D,X)

      REAL   A,B,D,X,PIN,QIN,R

      EXTERNAL  RNBET

            IF (D .EQ. 1.0) THEN
             PIN = 2.5
             QIN = 1.5
            ELSEIF (D .EQ. 2.0) THEN
             PIN = 2.35
             QIN = 2.35
            ELSEIF (D .EQ. 3.0) THEN
             PIN = 1.5
             QIN = 2.5
            ELSEIF (D .EQ. 4.0) THEN
             PIN = 4.0
             QIN = 2.0
            ELSEIF (D .EQ. 5.0) THEN
             PIN = 3.75
             QIN = 3.75
            ELSEIF (D .EQ. 6.0) THEN
             PIN = 2.0
             QIN = 4.0
            ELSEIF (D .EQ. 7.0) THEN
             PIN = 5.5
             QIN = 2.5
            ELSEIF (D .EQ. 8.0) THEN
             PIN = 5.0
             QIN = 5.0
            ELSEIF (D .EQ. 9.0) THEN
             PIN = 2.5
             QIN = 5.5
            ENDIF

            CALL RNBET(1,PIN,QIN,R)

            X = A + R * ( B - A)

            RETURN
            END
```

```
*********************************************************************
*   SUBROUTINE TRAP: RETURNS THE DISCOUNTED COST BASED ON A        *
*   TRAPEZOIDAL CASH FLOW                                           *
*********************************************************************

        SUBROUTINE TRAP(A, B, C, D, CST, DSC, X)

        REAL  A, B, C, D, CST, DSC, X, H, YRCST
        INTEGER  I

*   CALCULATE THE HEIGHT OF THE TRAPEZOID

           H = 2 / (A + (2 * B) + C)

           X = 0.0

*   INCREASING PHASE

           DO 70, I = 1, INT(A)

             YRCST = CST * H * (2*I - 1) / ( 2 * A)
             X = X + YRCST / ((1 + (DSC/100))**(I + D))

70         CONTINUE


*   CONSTANT PHASE

           DO 80, I = 1, INT(B)

             YRCST = CST * H
             X = X + YRCST / (1 + (DSC/100))**(I + D + A)

80         CONTINUE


*   DECREASING PHASE (PHASING-OUT)

           DO 90, I = 1, INT(C)

             YRCST = CST * H * ((2 * C + 1) - (2 * I)) /
     ^                   (2 * C)
             X = X + YRCST / (1 + (DSC/100))**(I + D + A + B)

90         CONTINUE

       RETURN
       END
```

```
**********************************************************************
*  SUBROUTINE SORT: SORTS THE UPPER 70 PERCENT OF 'SUMCST(NV)'     *
**********************************************************************

      SUBROUTINE SORT

      INTEGER    NC, NY, NV, ND
      PARAMETER  (NC=10, ND=10, NY=99, NV=800)

      INTEGER  I, LAST, NRV
      REAL SUMCST(NV), TEMP
      COMMON /SRT/ SUMCST, NRV

      LAST = NRV

* SORT THE SMALLEST COST TO POSITION 'SUMCST(1)'.

      DO 5, I = LAST, 2, -1

         IF (SUMCST(I) .LT. SUMCST(I-1)) THEN
            TEMP = SUMCST(I)
            SUMCST(I) = SUMCST(I-1)
            SUMCST(I-1) = TEMP
         ENDIF

5     CONTINUE

*   SORT THE LARGEST 20 % OF COST TO THE LARGEST POSITION OF
'SUMCST(I)'.

10    DO 20 , I = 1, LAST - 1
        IF (SUMCST(I) .GT. SUMCST(I+1)) THEN

          TEMP = SUMCST(I+1)
          SUMCST(I+1) = SUMCST(I)
          SUMCST(I) = TEMP

        ENDIF

20    CONTINUE

*
*  SORT ONLY THE LARGEST 20% OF COST (PRESENT VALUE)
*

      IF ( LAST .GT. INT(.8*NRV) ) THEN

        LAST = LAST - 1
        GO TO 10
```

```
ENDIF

RETURN
END
```

Bibliography

1. Cavinato, Dr. Joseph. "Product Costs from Cradle to Grave". Chilton's Distribution, 91, 59-62, (January 1992).

2. Department of the Air Force. Cost Estimating Procedure. AFCM 173-1. Washington: HQ AFSC, April 1972.

3. Kalos, Malvin H. Monte Carlo Methods. New York: John Wiley & Sons, Inc., 1986.

4. Office of the Assistant Secretary of Defense. Cost and Operational Effectiveness Analysis Guidelines. DoD Directive 5000.1. Washington: Government Printing Office, February 1990.

5. Seldon, Robert M. Life Cycle Costing: A Better Method of Government Procurement. Boulder, CO: Westview Press. 1979.

6. Sumner, Capt David L. An Interactive Life Cycle Cost Forecasting Tool. MS thesis, AFIT/GOR/ENS/90M-17. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 1990.

7. McCracken, Daniel D. and Salmon, WIlliam I. Computing for Engineering and Scientists with Fortran 77. Second Edition. John WIley & Sons, Inc., 1988.

Vita

Captain Nicolas M. Habash was born in Cairo, Egypt, on 10 April 1956. He Immigrated to the United States with his parents in 1967. In 1973 he graduated from John T. Hoggard High School and enlisted in the U.S.A.F. He was honorably discharged from the U.S.A.F in 1977. He graduated from California State University Northridge in 1981 with a B.S. in Business Administration (Finance). He entered the U.S.A.F in 1984 and attended Louisiana Tech University as an A.F.I.T. student. In 1986 he completed a B.S. degree in Electrical Engineering and proceeded to work for the Air Force Electronic Warfare Center, San Antonio, Texas until he entered AFIT in 1991 in pursuit of an M.S. degree in Operations Research. He graduated December 1992 and is currently in the USAF inactive reserves.

December 1992 Master's Thesis

Life Cycle Cost Forecasting Tool

Nicolas M. Habash

Air Force Institute of Technology (AFIT)
WPAFB, OH 45433-6583

AFIT/GOR/ENS/92D-01

SOCHE
2900 Acosta Street
Dayton, OH 45420

Approved for public release; distribution unlimited.

This thesis is about computing life cycle cost and culminates with a life cycle cost forecasting tool (software). The software is written in FORTRAN 77 and uses IMSL subroutines extensively for random number generation from various distributions. The software can compute life cycle cost using the following types of cost elements: constant cost element, stochastic cost elements, and cost estimating relationships (CERs). The use of a trapezoid approximation of payment allocation is allowed. Percent allocation of payment is also allowed and is recommended where the trapezoid fails to correctly represent the payment schedule. The CER algorithm can handle cost data estimated in base 10, natural logarithm, and logarithm. File management capability is an integral part of the software. The software can read, create, view and edit cost files. The output of the software includes two files. File filename_LC.OUT contains an annual cost and a total cost at 85, 90, 95, and 99 percent confidence levels. File filename_SAS.DAT contains all randomly generated annual costs. A screen output of filename_LC.OUT occurs after executing the algorithm on the data.

136

Life Cycle Cost Forecasting.

Unclassified	Unclassified	Unclassified	Unclassified