AD-A100 761   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH   SCHOO--ETC   F/G 9/3
              INTERPOLATING SUBROUTINE FOR HIGH POWER SYSTEM DESIGN.(U)
              DEC 80   B D CLARK
UNCLASSIFIED AFIT/GE/EE/80D-14   NL

| OF |
AD
A100761

INTERPOLATING SUBROUTINE FOR
HIGH POWER SYSTEM DESIGN

Thesis

Gerald D. Clark
AFIT/GE/EE/80D   Captain   USAF

DTIC
ELECTE
JUL 1   1981

F

14

INTERPOLATING SUBROUTINE FOR

HIGH POWER SYSTEM DESIGN.

Master's 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

by

Gerald D. /Clark

Captain   USAF

Graduate Electrical Engineering

December 1980

Preface

My sincere gratitude is extended to Captain Frederick C. Brockhurst, my thesis advisor, for proposing this study and for his guidance and suggestions. I would also like to thank Dr. David A. Lee, Chairman of the AFIT Mathematics Department, for his suggestions and encouragement. Finally, a special thanks goes to my wife, Barbara, for her patience and help during this study.

Gerald D. Clark

ii

Contents

List of Figures

List of Tables

AFIT/GE/EE/80D-14

Abstract

The Aero Propulsion Laboratory is currently developing
a computer-aided design program for high power airborne
systems. An important part of this design program will be
the feasibility study which was to be based on summary algor-
ithms. These algorithms were to relate the weight and volume
of each system component to the contributing operating para-
meters.

This study first centers on the requirements for an
interpolation scheme to form the summary algorithms. The scheme
will need to work in at least four dimensions and produce accu-
rate results over wide ranges. The advantages of the two
possible interpolation approaches (algorithm development and
direct interpolation) are described. The results of this com-
parison show clear advantages in the direct interpolation
approach using stored data.

The remainder of this study is a description and evalu-
ation of the subroutine which was developed, INTERP. It has
the flexibility to interpolate a data array with two or more
independent variables and will output the values of any number
of dependent variables. The routine also compensates for mis-
sing values in the known data array and issues an error message
to the user when a test point beyond the data range is input.

INTERPOLATING SUBROUTINE FOR

HIGH POWER SYSTEM DESIGN

I. Introduction

Background

The Aero Propulsion Laboratory has responsibility for
the development of multi-megawatt, airborne power systems.
These high-power systems will be required for directed energy
weapons being developed at the Air Force Weapons Laboratory.
The components required in these power systems will include
turbine, generator, transformer, rectifier, and pulse-
forming network. The three steps in the system development
are (1) the system feasibility study, (2) the detailed com-
ponent design, and (3) the dynamic system simulation
(Figure 1).

The first step, system feasibility study, will use
summary algorithms which relate the weight and volume of each
component to the contributing operating parameters. These
algorithms will then be applied over broad parameter ranges
to determine optimal combinations of components. Operating
parameters determined in this step will then be converted to
component design specifications. Computer programs which can
produce detailed component designs will then be used in step
two to specify the detailed design for each component.

The final step will be the dynamic computer simulation
of the inter-connected components. This simulation will
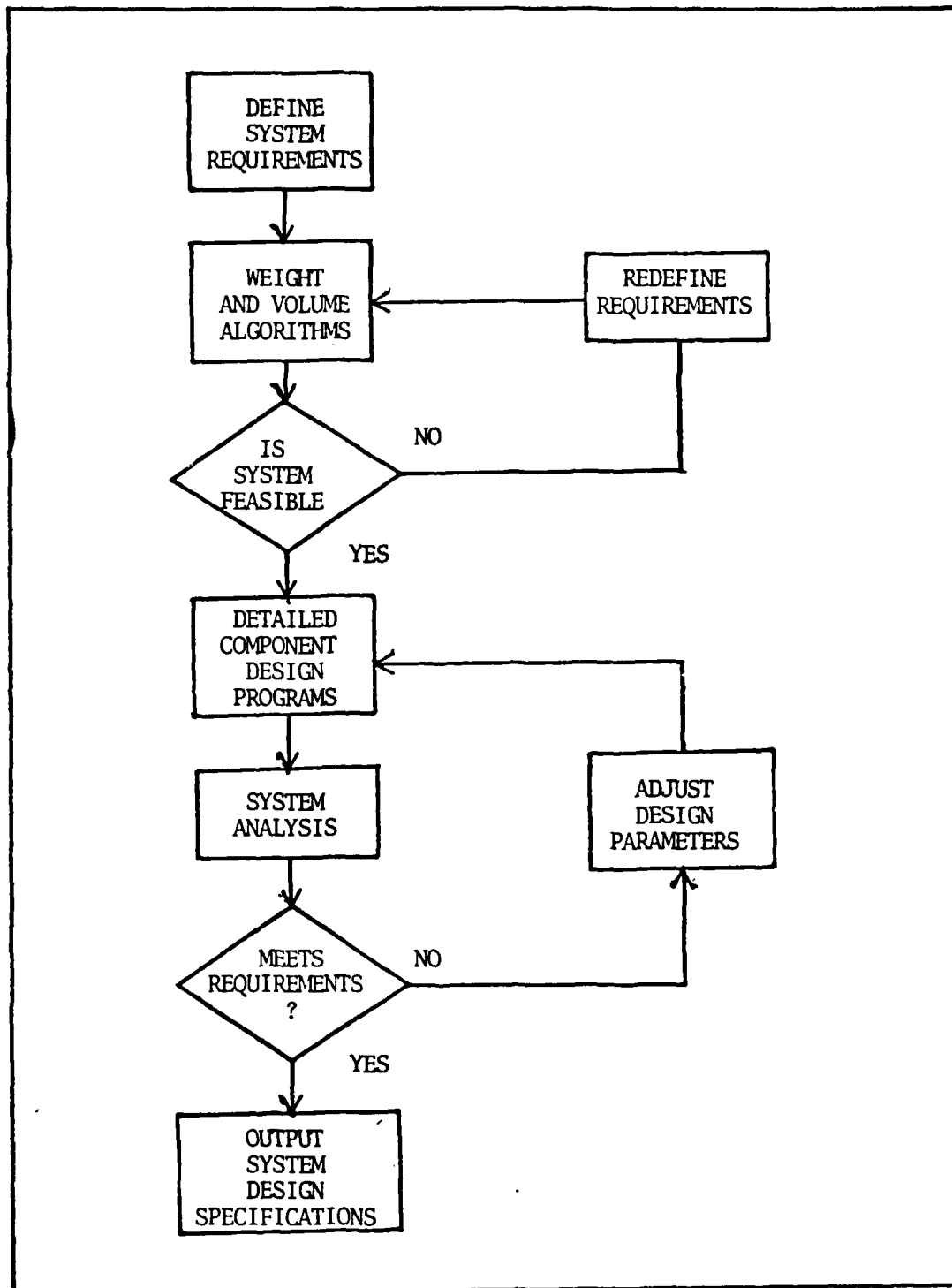
Fig 1. Flowchart of Computer Aided Design Concept

evaluate electrical and thermal performance of the entire system. The electrical simulation will permit evaluation of voltage and current transients and aid in the refinement of the control system. Thermal simulation will provide the data needed to specify the cooling system requirements.

The system study is still in the first phase, the development of summary algorithms. A summary algorithm describes the weight or volume of a component as a function of operating parameters. Examples of operating parameters are input voltage, frequency, and output power.

Currently, the Aero Propulsion Laboratory is using civilian contractors to develop design programs and point designs, which cover the expected range of each of the parameters. One contract is with the Raytheon Corporation to provide 120 minimum weight and volume designs for rectifiers. The contract also includes 110 inverter designs and 65 designs for pulse-forming networks. These designs will be delivered during 1981 and 1982.

Problem

The current problem is the need for a computer program which can use the data provided in these designs to form the summary algorithms needed for system feasibility analyses.

The first objective of this study was to determine the best format for these algorithms. The format will be determined by the choice of a multi-dimensional interpolation scheme.

Scope

Considerations in this choice will include (1) compatibility of the algorithm format with the next step of the system analysis, (2) compatibility of the scheme implementation, and (3) complexity, range, and accuracy of the scheme. The actual writing of the computer program to form these algorithms will be the second part of this study.

Approach

The first step in this investigation was a review of the High Power Studies (a set of technical reports prepared for the Aero Propulsion Laboratory) and direct consultation with project engineers in the laboratory to consider the question of compatibility with the analysis program. The other area of investigation was a literature search and consultation with Dr. Lee, Chairman of the AFIT Mathematics Department, to compare interpolation schemes.

Sequence of Presentation

A more detailed analysis of the requirements which affect the choice of the interpolation scheme is presented in Chapter II. Chapter III describes the two possible approaches to the interpolation problem and the advantages of each. The development of an interpolating subroutine based on the direct interpolation scheme is also proposed.

Chapter IV describes the subroutine developed, INTERP. In Chapter V the accuracy and effectiveness of INTERP is evaluated using the program CGEN and stored data based on conventional

generator designs. Conclusions and recommendations for future
study are presented in the final chapter.

There are also two appendices. Appendix A includes a
"User Guide" to INTERP along with a sample program and a
listing of the subroutine. Appendix B contains a listing of
CGEN as well as some of the inputs, outputs, data, and graphs
used in the evaluation of INTERP.

II. Investigation of Requirements

As stated earlier, the primary application of the sum-
mary algorithms will be in the computer program used to deter-
mine the optimal configuration of system components. This
optimal design will include some, but not necessarily all, of
the possible system components indicated below with their
associated parameters:

1) Turbine (including gas generator and fuel supply)--
speed, power, run-time, number of starts

2) Generator (three different types will be considered)
--speed, voltage, power

3) Transformer -- voltage in, voltage out, power,
frequency

4) Rectifiers -- power out, frequency, voltage out

5) Inverters -- voltage in, voltage out, power out,
frequency

6) Pulse-forming networks -- power, pulse width,
pulse repetition time.

Figure 2 depicts a possible system design.

Summary algorithms have already been developed for
transformers, using careful analysis of a large number of
smoothly varying design points. Some algorithms have been
developed for generators. Due to discontinuities, they are
valid only over a limited range. Below is an example of one

6
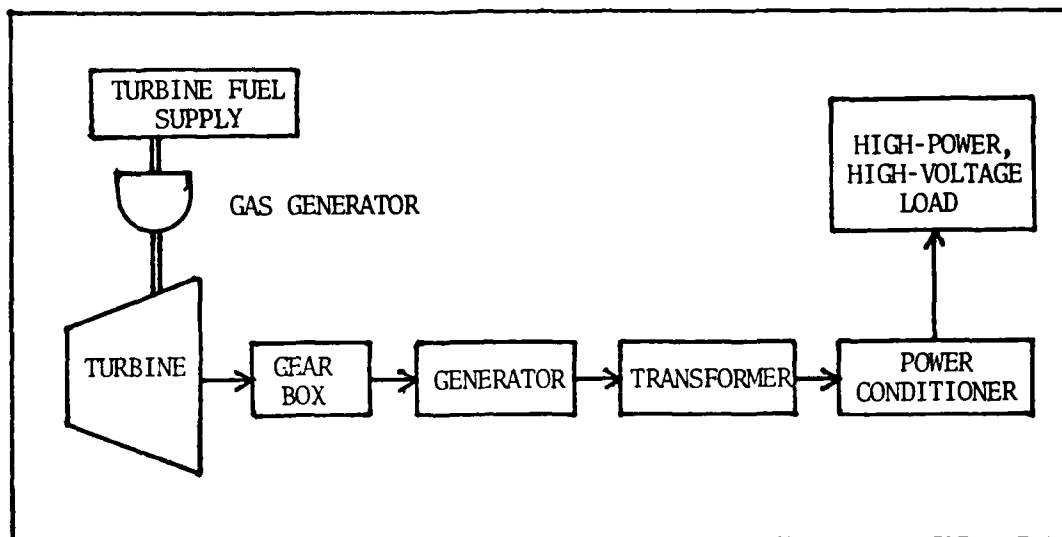
Fig 2. Generalized High-Power Airborne System

of these algorithms (Ref 2:4).

$$\text{Specific Weight (lbs/kW)} = .157 \left[1.28 - .28 \left(\frac{P}{5}\right)^{.449}\right]$$
$$\cdot \left[-.06 + 1.06\left(\frac{RPM}{1400}\right)^{-.6205}\right] \cdot \left[.8567 + .1433\left(\frac{V}{3}\right)\right] \quad (1)$$

where

P = generator output power (megawatts)

RPM = rotational speed

V = generator output voltage (volts, line-neutral)

Algorithms developed so far for power conditioning components are based on preliminary designs and give unacceptable results in most cases.

New design points, now being developed by Raytheon, will probably provide the most severe test of the interpolating scheme. Large data sets over wide ranges will be developed.

7

Inverter designs based on four independent parameters, with some discontinuities, will be included. This will require an interpolation scheme, in at least four dimensions, which can produce reasonable accuracy over wide ranges.

III. Theoretical Approaches

There are two possible approaches to an interpolating problem. The first is to use the available data to generate an algorithm, similar to Eq (1), which can then be used independently of the data. The alternative approach is to retain the data in an array or file. When a value of the function is needed, it is necessary to find the nearest known points in the array and estimate the value of the function at the new point. The next step is to compare these two approaches and determine which is best suited to this application.

Algorithm Approach

First to be considered is the algorithm approach. This approach has been used to date by engineers working on the high power systems under consideration. It has the obvious advantage of succinctness. When incorporated into the overall design feasibility program, the algorithm approach allows faster execution and requires much less core memory. However, writing a computer program to reduce a multi-dimensional array of data to a compact algorithm proves to be a very complex task.

One way to design such a program would be to use the multi-dimensional spline technique discussed by de Boor in Reference 3. At first this approach appears reasonable, though quite involved. A problem appears near the end of the development where a system of perhaps several thousand

9

simultaneous equations must be solved. Even though the system
has considerable structure, so that highly efficient solution
schemes could be used, this is a difficult problem. An alter-
native to the spline approach is to use an application of the
multi-dimensional Taylor series expansion (Eq 2).

$$F(x_1, x_2, \ldots x_n) = F(x_1', x_2', \ldots x_n')$$
$$+ \frac{\partial F}{\partial x_1} (x_1 - x_1') + \frac{\partial F}{\partial x_2} (x_2 - x_2')$$
$$+ \ldots \frac{\partial F}{\partial x_n} (x_n - x_n') + \text{higher order terms} \qquad (2)$$

where

$$(x_1', x_2', \ldots x_n') = \text{point where the value of the}$$
$$\text{function is known}$$

The key consideration in this approach is the "higher order
terms." For a nonlinear function there is no limit to the
number of higher order terms, each of which becomes more com-
plicated. The functions being considered will definitely not
all be linear. However, a function can sometimes be made
linear using a change in variables. For instance, if a
function F varies as $x^3 + 5$, we could substitute $x' = x^3 + 5$;
and F would be linear with respect to x'. This is the approach
which has been used on several data sets in the past. For data
which we can examine and manipulate by hand with the help of
human intuition, we have a reasonable likelihood of discover-
ing variations like $ax^2 + b$, $ax^c + b$, $e^x + a$, or $a^x + b$.
Physics may also suggest these groups. However, to program
a computer to uncover all of the possible variations would be

a gigantic, if not impossible, task. An additional problem is that these algorithms are based on the behavior of the function near the midpoint of the data sets and often produce serious inaccuracies when applied to points near the edges of the data array.

Direct Interpolation
Approach

After considering the complications associated with the algorithm approach, a careful look at the direct interpolation approach is definitely warranted. An advantage to this approach is that output is limited to the range between the neighboring data points. This avoids the possibly extreme errors of a modeling approach. If more accuracy is required, a spline or other more powerful interpolation technique can be used.

As mentioned previously, this approach has the disadvantage of requiring more time and memory when being executed in the system feasibility program. The objective of the feasibility program, however, is not to run in minimum time or with minimum memory. Whether it requires ten minutes and 15K of memory or six hours and 150K, in either case, if the program is successful, it will save weeks or months of research and greatly reduce the acquisition time of the final system.

Another disadvantage to be considered is the requirement to store and maintain the data sets. This is normally done on a tape at the computer center and requires careful documentation to ensure that the tapes are retained and that

11

the data on them is readily accessible. The other advantages to be gained certainly warrant the small additional effort required to document these data sets.

Conclusion

An objective of the program developed, using this interpolating scheme, will be its applicability to operate on data sets still under development. To meet this requirement and to ensure reasonable accuracy from the scheme, it is concluded that the direct interpolation approach, using stored data, is the best for this application.

The next step in the study is the development of a general subroutine to accommodate at least five variables. This subroutine should be general enough to be used on a data set from any of the high power system components. Inputs to this routine would include the number of independent variables, the number of data points for each variable, the values of the variables at the known data points, an array containing the function values at the data points, and the point at which the function is to be evaluated. An advantage to this scheme is that three values (weight, volume, and efficiency) can be stored in the same data array, and all three can be determined with one pass through the subroutine.

IV.   Description of SUBROUTINE INTERP

SUBROUTINE INTERP was developed by the author to meet
the requirements discussed in the previous chapter. The sub-
routine can accommodate two or more independent variables and
is not limited to five. It can store any number of dependent
values at each point in the array. This allows storage and
interpolation of weight, volume, efficiency, and other data.

The routine includes an error code output which sig-
nals the user when he has requested data from a point outside
the stored array bounds. If data are unavailable for some
points in the array, the user fills these locations with zeros
and the routine automatically searches until non-zero values
are found on which to base the interpolated values. Since
these values normally represent values of weight, volume, or
efficiency, a value of zero should not otherwise appear.

In the following section of this chapter is a dis-
cussion of the inputs to the routine and how the data arrays
are stored. Subsequent sections give an overview of the
routine and then describe each of the three main sections in
more detail. A complete listing of INTERP, a user guide,
and a sample program are included in Appendix A.

Arguments

The first input to the routine, NDIM, is an integer
which specifies the number of independent variables or

13

dimensions to be handled. Allowable values are two or greater. The second input, LV, is a vector (length equal to NDIM) which contains integers corresponding to the number of data points in each dimension. These points do not need to be evenly spaced. The order of these points establishes the order in which the dimensions will be considered throughout the routine.

LVT is the next input and is the sum of the elements in the length vector, LV. LDI is the number of dependent values stored at each point in the array. Any positive integer may be used. LDJ is the product of the elements in vector LV. LVT, LDI, and LDJ are used along with NDIM to dimension the remaining arrays.

The next input is the vector VAR with length equal to LVT. The values of the independent variables at the known data points are stored in this vector. The values of the first variable are stored first in increasing order, followed immediately by the values of the next variable stored the same way, and likewise through the independent variables. These variables must be stored in the same order as the length values stored in LV. The input WVE is the array containing the dependent variables (typically weight, volume, and efficiency). The array has dimensions LDI by LDJ. The first index of the array specifies which dependent variable to consider for a given set of independent variables. The second index walks through the possible data locations by first incrementing through all values of the first independent variable, then through all possible combinations of the first and second

14

independent variables, and so on through all the possible combinations of independent variables. Zeros should be stored at any indices where the data are not available. The routine will check for zero values and search for the next non-zero value.

The input XV is a vector (length equal to NDIM) which contains the coordinates of the test point, the point in the data array where the values of the dependent variables are to be estimated. The vector LWK (length equal to NDIM) is a work space providing temporary storage locations for index values within the routine. OUT is the output vector of length LDI containing the computed values of the dependent variables.

The final argument of the subroutine, IERR, is a two-digit error message output. The least significant digit will be either zero or nine. Zero indicates that a requested point is less than the lowest point, and nine indicates a requested point greater than the highest data points. The leading digit indicates the dimension in which the out of range condition occurred.

Mathematical Basis

The mathematical basis for INTERP is the multidimensional Taylor series expansion introduced in Chapter III and repeated below in Eq (2).

$$F(x_1, x_2, \ldots x_n) = F(x_1', x_2', \ldots x_n')$$
$$+ \frac{\partial F}{\partial x_1} (x_1 - x_1') + \frac{\partial F}{\partial x_2} (x_2 - x_2')$$
$$+ \ldots \frac{\partial F}{\partial x_n} (x_n - x_n') + \text{higher order terms} \qquad (2)$$

The approach used in this routine is a first-order approach and neglects the higher order terms. The values of the independent variables, x_1, x_2, ... x_n, at the point where the function is to be evaluated are labeled XV(1), XV(2), ... XV(n) in the subroutine.

The subroutine is divided into three main sections as shown in Figure 3. The first step in applying Eq (2) is to search the data to find the indices of the dependent and independent variables at a nearby known data point to be used for $(x_1', x_2', ... x_n')$. The next section of the routine checks for missing values (zeroes) in the dependent data locations specified above and increments or decrements the indices to find the nearest, non-zero, dependent variables above and below the test point.

In the final section of the routine the partial differentials are approximated by first difference expressions which are evaluated in each dimension and then summed to find the desired output. In the remainder of this chapter, these three sections are discussed in more detail.
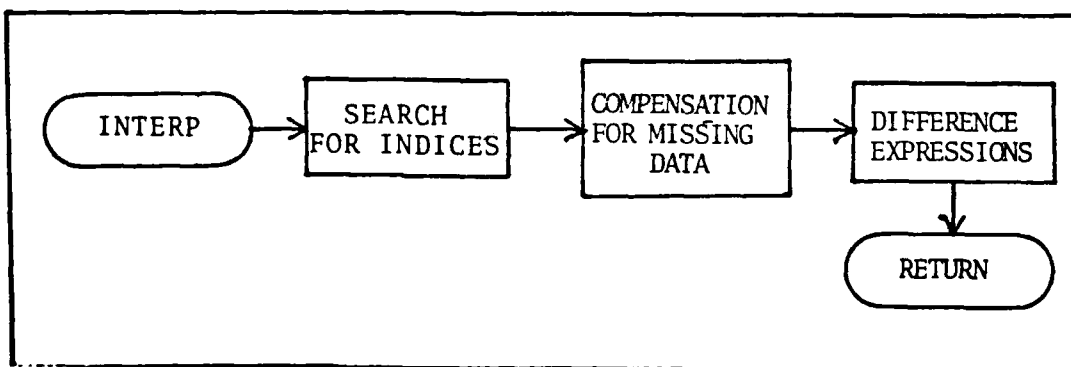
Fig 3. Flowchart of Main Sections of INTERP

16

Finding Index

The first section of the routine contains two main loops, each with an internal loop as seen in Figure 4. The first main loop searches each dimension, or string of independent variable values (VAR), to find the index of the lowest value which is greater than or equal to XV(I). When this value is found, it is stored in LWK(I). If XV(I) is less than the lowest value or greater than the highest value in the string, an error code is formed and printed, and control returns to the main program (note paths A and B in Fig 4). The second main loop of this section uses the computed values in LWK and the values in the length vector, LV, to compute INDEX. INDEX corresponds to the second index of WVE (the known dependent variable array) at a point near the desired value specified in XV.

The error indicator, IERR, is initialized to zero at the beginning of the routine and is revalued when necessary by the output error code as shown in Figure 5. The most significant digit of IERR comes from I, the loop index, and indicates the dimension being searched when the out-of-range condition occurred. The least significant digit is left zero in the case of a below range condition (path A), and is changed to nine in the case of a value of XV beyond range (path B).

Compensation for Missing Data

Section 2 of the routine is actually within two do loops of section 3 (see Figs 6 and 7). The discussion of both

17
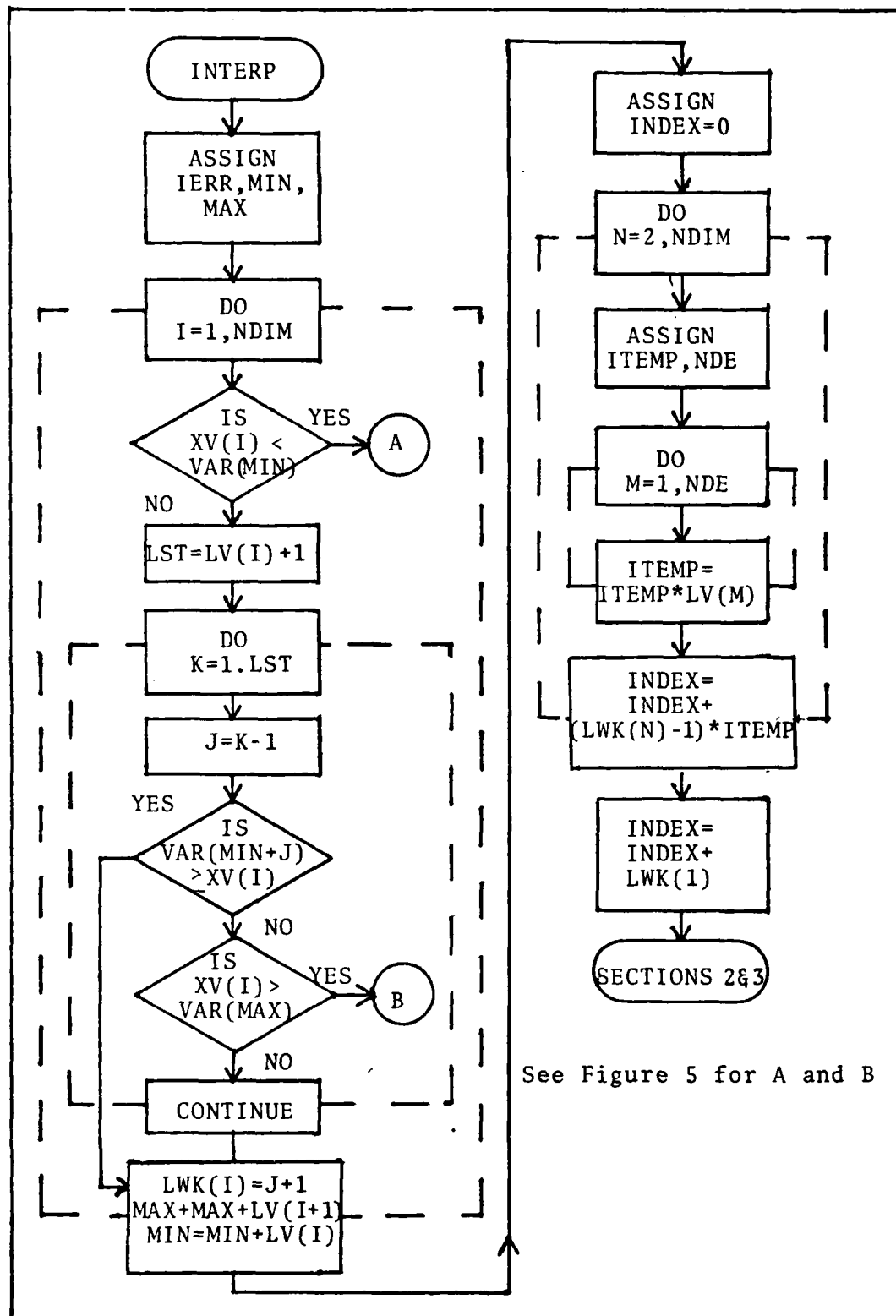
Fig 4. Flowchart of Section 1, Finding Index
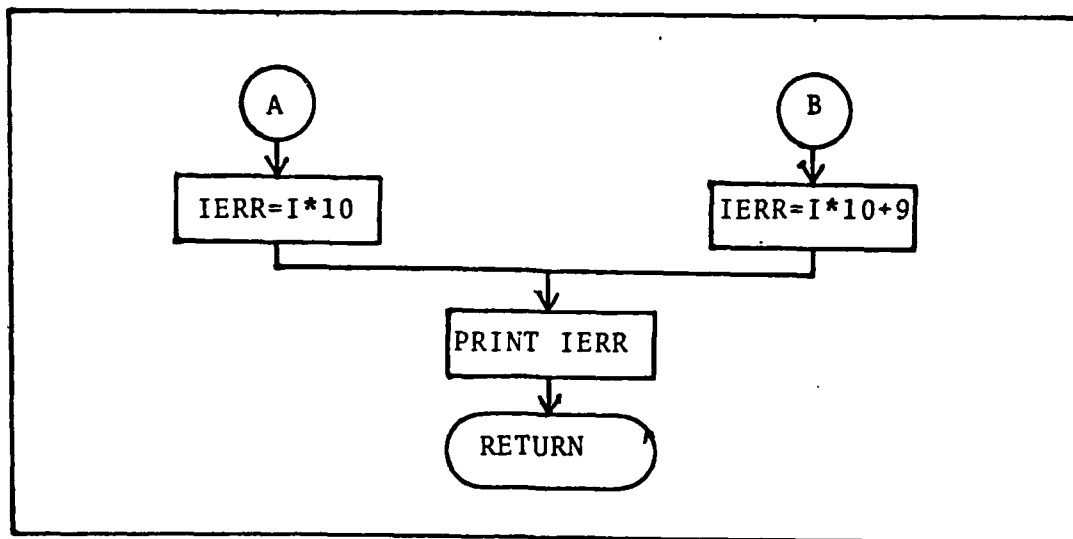
Fig 5. Flowchart of Output Error Code

sections will be clearer if section 2 is considered first.
It must be recalled that TMP2, IT1, and IT2 are already
assigned in section 3 and that section 2 is included within
loops with indices K and L.

The source of the data used as inputs to this routine
are usually the result of point designs developed by the Aero
Propulsion Laboratory or by civilian contractors. Examining
existing data reveals that in many cases data for some of the
points within the array are not available. In this routine
the missing values of WVE are filled with zeroes. Since these
values normally represent values of weight, volume or effici-
ency of a system component, a value of zero should not other-
wise appear. The final section of the program will use the
values of the dependent variables at points above and below
the test point in each dimension. This requires that many
points be checked for zero.

19
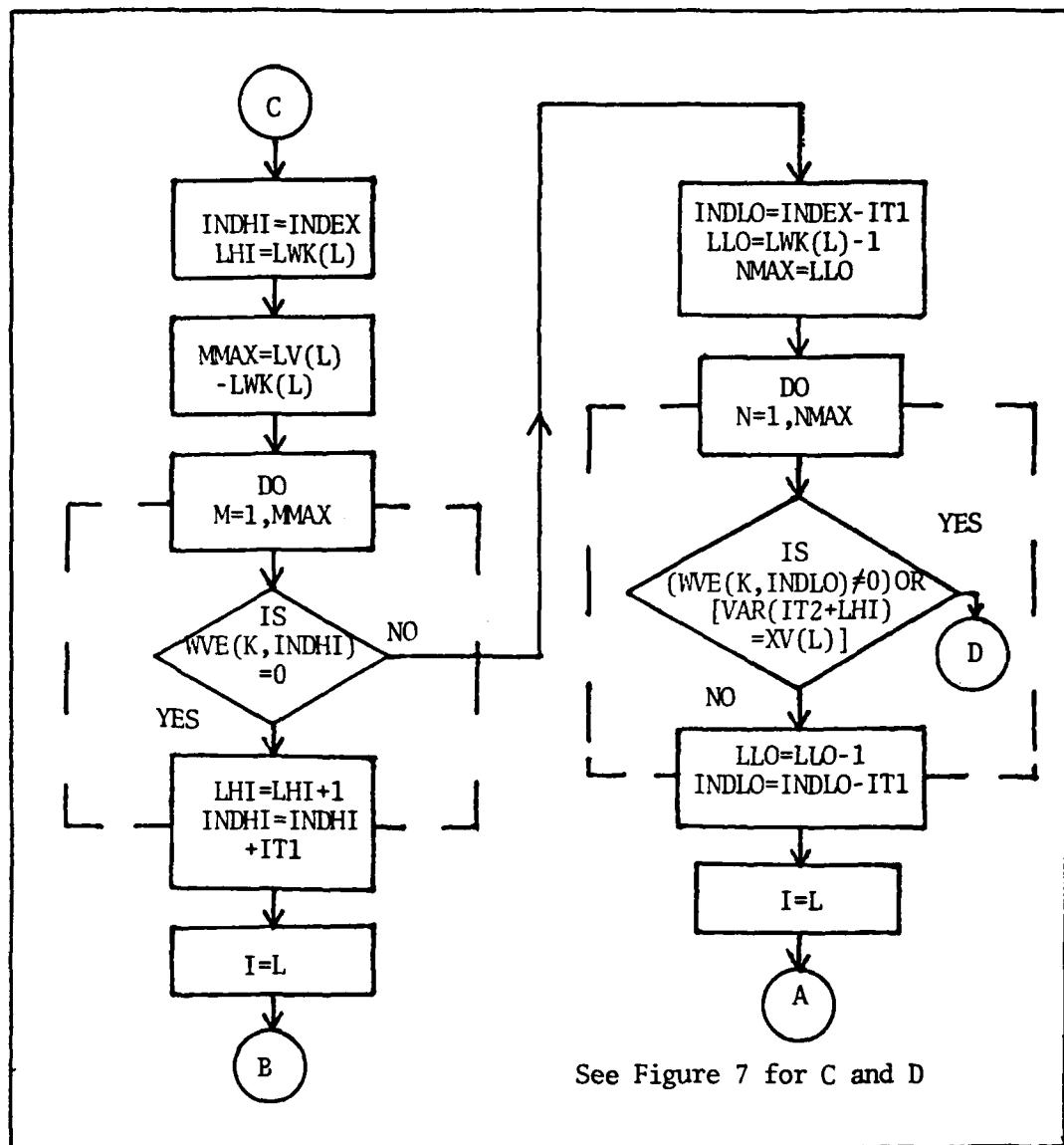
Fig 6. Flowchart of Section 2,
Compensation for Missing Data
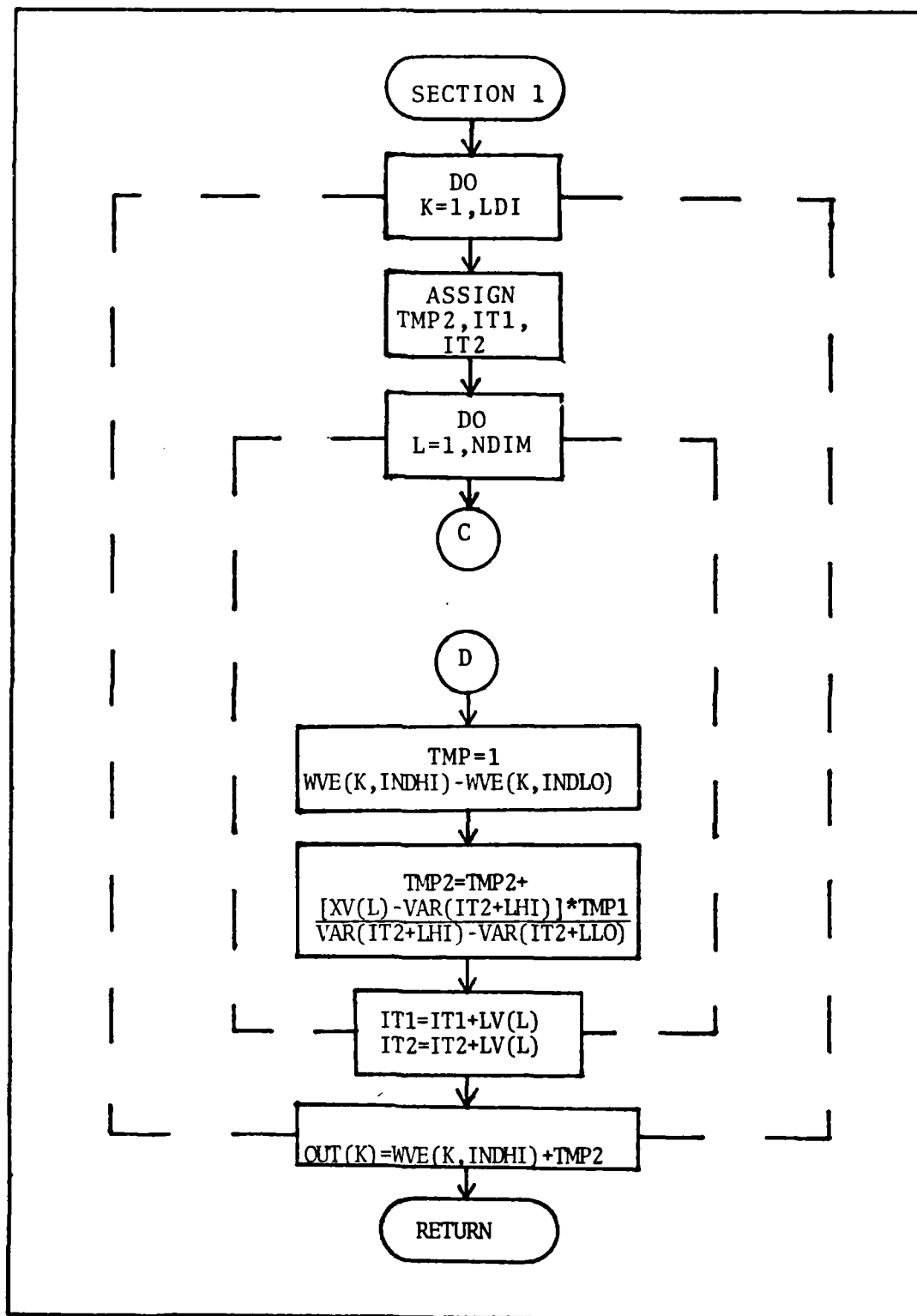
Fig 7. Flowchart of Section 3, Difference Expressions

21

In the first part of this section, the index of the point above the test point, INDHI, is initialized as index (see Fig 6). The stored value of WVE (K,INDHI) is then tested for zero. If a zero is found, INDHI is incremented and the check is made again. If the check fails after the highest value in the string has been checked, an error output is generated (path b) as in section 1, indicating that the point is beyond range and in which dimension. Note that the index of the independent variable, originally stored in LWK, is stored in LHI and is incremented along with INDHI. This value is required in the final section of the routine.

The remainder of this section repeats the process just described for the known point below the test point. In this case, the dependent variable index is stored in INDLO and the dependent variable index in LLO. Again, a route to the error code is provided (path A) when a non-zero value is not found in the string.

Difference Expressions

The objective of the final section is to determine the estimated values of the dependent variables at the test point. This is done by approximating Eq (2), using first difference equations to replace the partial differential. TMP1 divided by VAR(IT2+LHI) - VAR(IT2+LLO) replaces the partial differential and $(x_n = x_n')$ is replaced by XV(L) - VAR(IT2+LHI). Compare Figure 7 and Eq (2). The quantity WVE(K,INDHI) in the final statement corresponds to $F(x_1', x_2', \ldots x_n')$ in Eq (2).

22

The results of these approximations are discussed in the following chapter. It is shown there that the dependent variable values are output unchanged when the test point coincides with a known data point, and that the output corresponds to a straight-line approximation for points between known data points.

V.   Evaluation of SUBROUTINE INTERP

The objective of the evaluation was to choose a repre-
sentative data base and develop the program to call INTERP.
The output of this program is then studied to ensure that the
subroutine is producing reliable data. The data set selected
contains the results of a conventional generator design pro-
gram.

This data set is well suited for a test of the routine
with three independent variables and three dependent variables.
The independent variables are RPM, voltage, and power. The
dependent variables are specific weight, efficiency, and
volume. This data set has the added advantage that an algorithm
has already been developed to approximate the variation of spe-
cific weight as a function of the three independent variables.
This algorithm is the one presented earlier as a representative
case of three algorithms (Eq 1, page 7). This will provide
the opportunity to compare the subroutine output with the
algorithm results.

Program CGEN

The program developed to manipulate the conventional
generator data is named CGEN. A complete listing of the pro-
gram, along with input and output data, is located in Appendix
B. As seen in the flowchart in Figure 8, the program begins
by printing the labels for the output data. Both the dependent
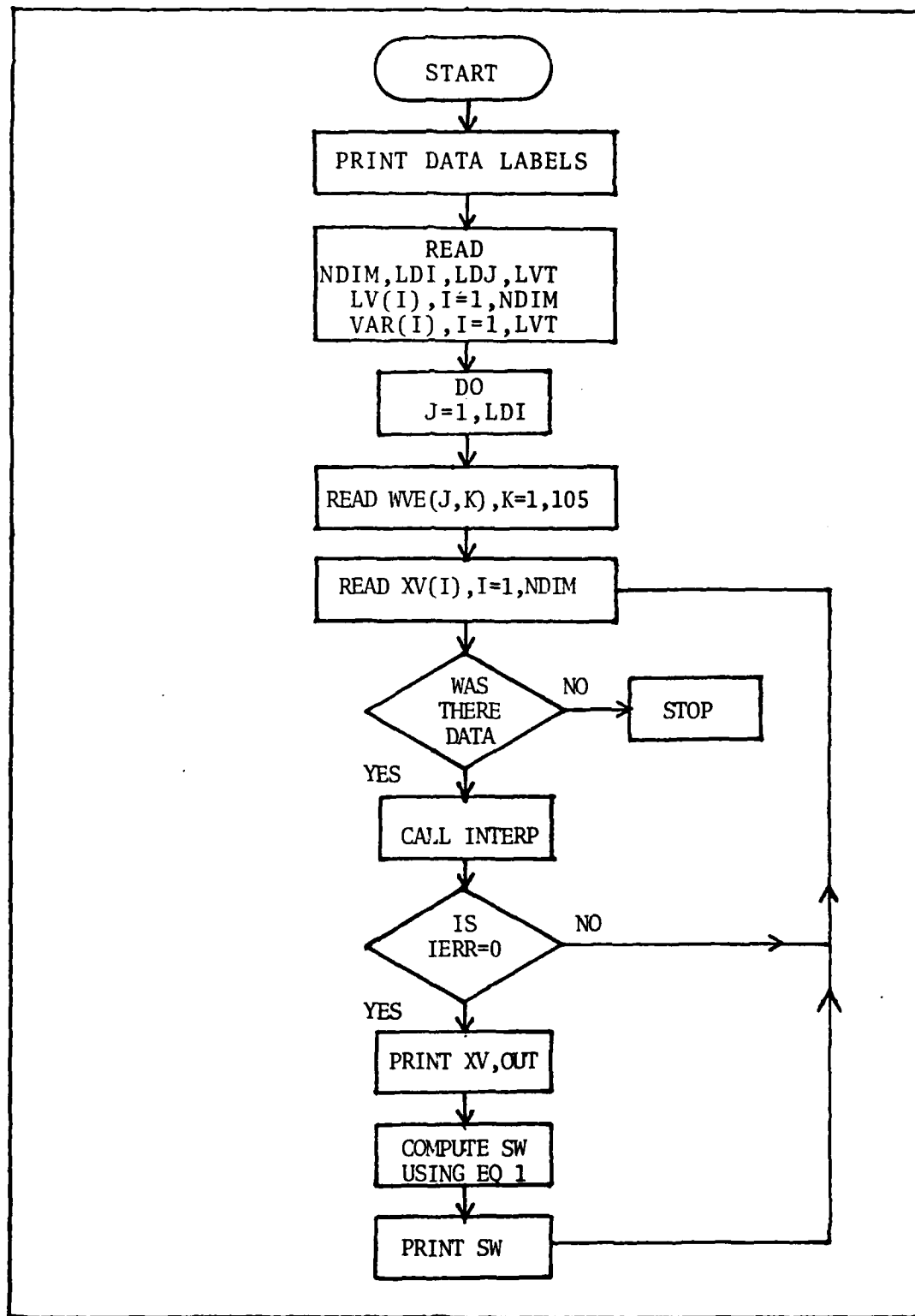
Fig 8. Flowchart of Program CGEN

and the independent variable values at the test points are included in the printed output. The dimensioning information and values of the independent and dependent variables at the known points are then read from the data file. As anticipated, there were numerous cases where the dependent variable values were not available. Zero fill was used at these positions, as previously discussed.

The remainder of the program reads a test point, XV, and calls INTERP. If no error is encountered, the output is printed. The program then computes the specific weight using the algorithm presented earlier (Eq 1) and prints that value for comparison. If an error is encountered, the message will be printed by the routine and the program goes on to the next test point. The program stops when it has used all the test points in the input data file.

Results

Many test points were run through CGEN to ensure that the routine was working properly. Several errors in the routine were discovered and corrected before it reached its present form. Test points were run at known points to ensure the routine agreed there. Points above and below the range were used to test the error code function. Points were also tried at and near locations with zero fill to ensure section 2 of the routine performs correctly.

With all corrections made and all functions of the program and routine working correctly, the test points included
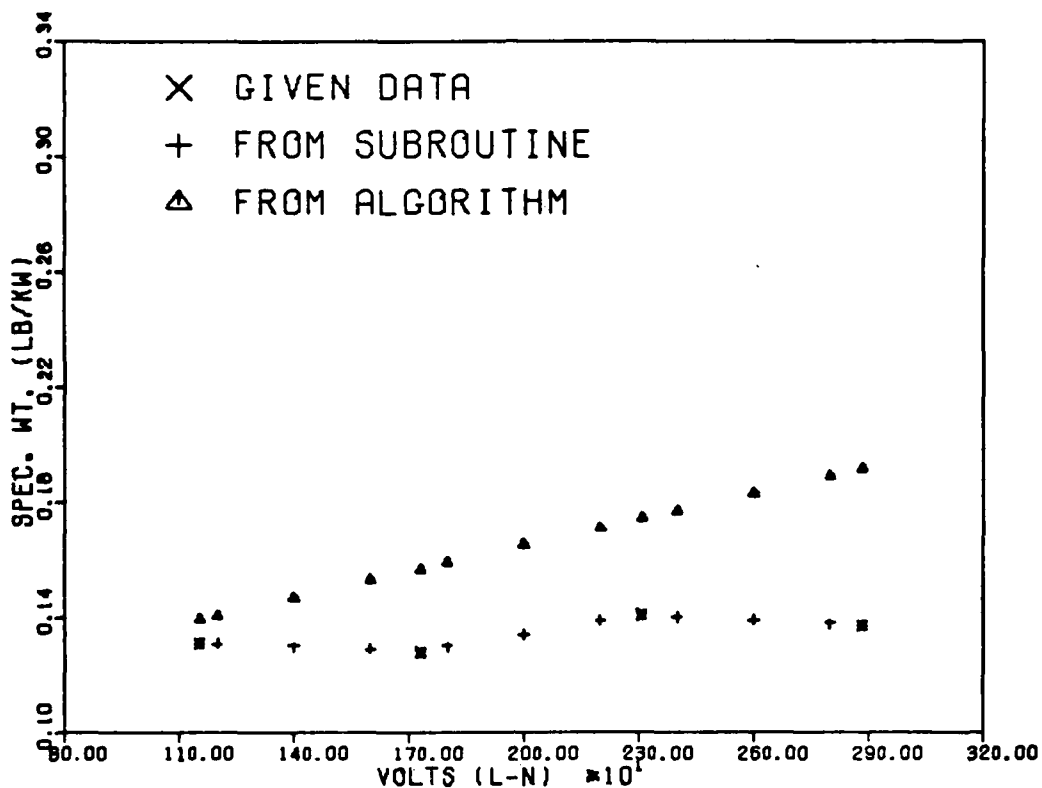
Fig 9.   Comparison of Algorithm and Subroutine
Changing Voltage, Volts .65

in Appendix B were used to produce a set of data for closer
study.   In one group of test points, the voltage was varied
while holding the RPM at 18,000 and the power level at 8.83
megawatts.   The results are plotted in Figure 9.   As can be
seen in the figure, the output from INTERP agree with the
given data at the known points and form a straight-line
approximation between known points.   The value from the algor-
ithm is close to the given data at the lowest voltage, an
error of 6.6 percent.   However, at the highest voltage the
error increases to 40 percent.

27

There are two additional groups of data presented in Appendix B. In one group the voltage is held constant at 577 volts and the power at 4.91 megawatts. The RPM is varied from 8,000 to 20,000. The subroutine output again agreed with the known data and a straight-line approximation between points. The algorithm data agreed much better in this case with the error varying between 1.2 and 12 percent.

In the other portion of the data set, the RPM was held at 18,000 and the voltage at 2885 volts, while the power level was varied. The subroutine output again agreed well with the data as expected; however, the error in the algorithm output was nearly constant at 40 percent.

Plots of all three sections of the data are included in Appendix B. Any number of data sets and plots could have been listed and plotted, but these three were chosen to show the range of possibilities and advantages of the subroutine approach over the algorithm approach used previously.

VI. Conclusions and Recommendations

The original objective of this study was the development of a computer program to produce the weight and volume algorithms required for the computer aided design of high power systems (Fig 1, page 2). These algorithms were to be based on the data contained in point designs being produced by contractors. A careful comparison of the advantages and disadvantages of an algorithm program versus a subroutine to assess and interpolate the data from these designs was made. The result of that comparison was the decision to develop an interpolating subroutine.

Conclusions

INTERP is the subroutine that was developed. It has the flexibility to interpolate a data array with two or more independent variables and will output the interpolated values of any number of dependent variables. Missing values in the stored data are compensated for by using zero fill at these data locations. Output values at known points agree exactly with the stored values, and the values between points are based on a straight-line approximation in each dimension.

The results of the evaluation made show that the routine meets all the objectives. The flexibility of the routine exceeds the original requirement to handle five independent and three dependent variables. Accuracy is as expected, and

an error code informs the user when a desired test point is outside of the data range. A comparison with a previously developed algorithm shows that INTERP can greatly reduce errors that sometimes exceeded 40 percent using the algorithm approach.

Recommendations

One of the next steps in this computer-aided design program will be the development of an executive program to implement the design tradeoff study in Figure 1. This program will iterate through large numbers of possible component configurations to find designs which optimize weight, volume, and efficiency.

Several of the component design programs have already been developed and a major task will be the interfacing of these programs, available data sets, and the subroutine INTERP. Another project that may prove beneficial is the extension of INTERP to give it the capability to also extrapolate beyond the given data range. It is not yet clear whether there will be a requirement for this capability.

Bibliography

1. Brockhurst, Frederick C. "Application of Subsystem Algorithms for High Power System Studies," 2nd IEEE International Pulsed Power Conference, Lubbock, Texas (June 1979).

2. _____. "Computerized Designs and Optimization of Self Contained Airborne High Power Electrical Systems," NAECON 1980, Dayton, Ohio (May 1980).

3. de Boor, Carl. "Splines as Linear Combinations of B-Splines," Approximation Theory II. New York: Academic Press, 1976.

4. Kaplan, Wilfred. Advanced Calculus (Second Edition). Reading, Mass: Addison-Wesley Publishing Co., 1966.

5. Lee, David A. Chairman, Mathematics Department, Air Force Institute of Technology (personal interview). Wright-Patterson AFB, Oh., April 10, 1980.

APPENDIX A

SUBROUTINE INTERP

User Guide to INTERP

This routine is used to interpolate between known data points within a data array containing two or more independent variables and any number of dependent variables. The user must supply the routine with known values of both the independent variables (VAR) and the dependent variables (WVE) at some known points. Values for the independent variables are then input (the test point) and the outputs of the routine are the dependent variable values associated with the test point.

If a value of the test point is outside the range of the known values, an error message will be output, indicating which dimension (or which of the independent variables) was out of range, and whether the test value was below or beyond the data range. This error code (IERR) is further discussed in the argument description below.

To call INTERP, the following statement is used:

CALL INTERP (NDIM, LV, LVT, LDI, LDJ, VAR, WVE, XV, LWK, OUT, IERR)

Arguments must be dimensional and valued as described below:

NDIM -- Number of dimensions or independent variables (integer, two or higher)

LV -- Length vector containing number of known data points in each dimension (integer vector, length equals NDIM)

LVT -- Sum of the elements in the vector LV (integer)

LDI -- Number of dependent variables (any positive integer)

LDJ -- Product of the elements in the vector LV (integer)

VAR -- Values of the independent variables at the known points (vector, length equals LVT)

WVE -- Values of the dependent variables at the known points (array, LDI by LDJ), use zero fill for array points with no data

XV -- Test point, coordinates where the dependent variables are to be estimated (vector, length equals NDIM)

LWK -- Working space (vector, length equals NDIM)

OUT -- Output, containing computed values of the dependent variables (vector, length equals LDI)

IERR -- Error message (output), zero indicates no error; otherwise, first digit indicates dimension, second digit indicates out of range; zero - below data range, nine - above data range.

Sample Program

Program SAMPLE is listed below as an example to the user. The known data is shown in Table I. Notice there are two independent variables (VAR1, VAR2) and one dependent variable (WVE). If there was a second dependent variable, its values would have been stored in WVE (1,7) through WVE (1,12).

34

The test points used and the resulting output are shown in Table I for comparison with Table I. If one of the dependent variable values listed in Table I had not been available, it could have been replaced by a zero. INTERP would then have ignored the point and made the interpolation based on the remaining points.

TABLE I

Input Data for Program SAMPLE

VAR1 ⟍ VAR2	5.0	10.0	20.0
20.0	1.0	2.0	3.5
50.0	3.0	4.0	7.6
	WVE		

TABLE II

Test Points and Resulting Output (SAMPLE)

VAR1	VAR2	OUT
6	25	1.53
8	30	2.27
12	40	3.35
16	45	5.47
18	55	*

*Error (IERR=29), VAR2 Beyond Range

```
            PROGRAM SAMPLE(INPUT,OUTPUT)
            DIMENSION WVE(1,6),VAR(5),XV(2),LWK(2),OUT(1),LV(2)
C
C   DIMENSION INFORMATION
C
            NDIM=2
            LV(1)=3
            LV(2)=2
            LVT=5
            LDI=1
            LDJ=6
C
C   KNOWN INDEPENDENT VARIABLE VALUES
C
            VAR(1)=5
            VAR(2)=10
            VAR(3)=20
            VAR(4)=20
            VAR(5)=50
C
C   KNOWN DEPENDENT VARIABLE VALUES
C
            WVE(1,1)=1
            WVE(1,2)=2
            WVE(1,3)=3.5
            WVE(1,4)=3
            WVE(1,5)=4
            WVE(1,6)=7.6
C
C   READ TEST POINT
   15       READ*,(XV(I),I=1,NDIM)
            IF(EOF(5LINPUT).NE.0.)STOP
            CALL   INTERP(NDIM,LV,LVT,LDI,LDJ,VAR,WVE,
          1 XV,LWK,OUT,IERR)
            PRINT*,"OUT=",OUT(1)
            GO TO 15
            END
```

Input Data	Output Data

Input Data	Output Data
6 25	OUT=1.533333333333
8 30	OUT=2.266666666667
12 40	OUT=3.353333333333
16 45	OUT=5.476666666667
18 55	IERR=29
*EOR	OUT=5.476666666667
*EOF	STOP

36

Listing of INTERP

```
      SUBROUTINE INTERP(NDIM,LV,LVT,LDI,LDJ,VAR,WVE,XV,
     1 LWK,OUT,IERR)
      DIMENSION WVE(LDI,LDJ),VAR(LVT),XV(NDIM),
     1 LWK(NDIM),OUT(LDI),LV(NDIM)
C
C   SECTION 1. FINDING INDEX
C
      IERR=0
      MIN=1
      MAX=LV(1)
C
C   FIND LWK TO FORM INDEX
C
      DO 20 I=1,NDIM
C
C   CHECK FOR VALUE BELOW RANGE: IF SO, GO TO ERROR CODE
C
      IF(XV(I).LT.VAR(MIN)) GO TO 30
      LST=LV(I)+1
C
C   LOOP TO FIND THE MINIMUM STORED VALUE, WHICH IS GREATER
C   THAN OR EQUAL TO THE TEST POINT
C
         DO 5 K=1,LST
         J=K-1
         IF(XV(I).LE.VAR(MIN+J)) GO TO 10
C
C   CHECK FOR VALUE ABOVE RANGE: IF SO, GO TO ERROR CODE
C
         IF(XV(I).GT.VAR(MAX)) GO TO 35
    5    CONTINUE
C
C   STORE THE INDEX OF THE INDEPENDENT VARIABLE FOUND IN LWK(I)
C
   10    LWK(I)=1+J
C
C   SET MAX AND MIN FOR NEXT DIMENSION
C
      MAX=MAX+LV(I+1)
   20 MIN=MIN+LV(I)
      GO TO 39
C
C   OUTPUT ERROR CODE
C
   30 IERR=I*10
   31 PRINT*,"IERR=",IERR
      RETURN
   35 IERR=I*10+9
      GO TO 31
```

```
C
C   FIND INDEX OF WVE CORRESPONDING TO LWK
C
 39      INDEX=0
            DO 50 N=2,NDIM
            ITEMP=1
            NDE=N-1
              DO 40 M=1,NDE
 40           ITEMP=ITEMP*LV(M)
 50         INDEX=(LWK(N)-1)*ITEMP+INDEX
          INDEX=INDEX+LWK(1)
C
C   SECTION 3. LOOPS TO FORM OUT(K)
C
        DO 70 K=1,LDI
        TMP2=0
        IT2=0
        IT1=1
          DO 60 L=1,NDIM
C
C   SECTION 2. COMPENSATION FOR MISSING DATA
C
C   INITIALIZE INDHI &CHECK FOR ZERO AT WVE(K,INDHI)
.C
          INDHI=INDEX
          LHI=LWK(L)
          MMAX=LV(L)-LWK(L)
            DO 55 M=1,MMAX
            IF(WVE(K,INDHI).NE.0) GO TO 56
C
C   IF ZERO, INCREMENT INDHI AND P   HECK
C
            LHI=LHI+1
 55         INDHI=INDHI+IT1
C
C   IF END OF DATA STRING IS REACHED, SEND ERROR MESSAGE
C
          I=L
          GO TO 35
C
C   INITIALIZE INDLO &CHECK FOR ZERO AT WVE(K,INDLO)
C
 56       INDLO=INDEX-IT1
          LLO=LWK(L)-1
          NMAX=LLO
            DO 57    N=1,NMAX
            IF((WVE(K,INDLO).NE.0).OR.(XV(L).EQ.VAR(IT2+LHI)))
            GO TO
C
C   IF ZERO DECREMENT INDLO
C
            LLO=LLO-1
 57         INDLO=INDLO-IT1
```

38

```
C
C   IF BEGINNING OF STRING IS REACHED, SEND ERROR MESSAGE
C
      I=L
      GO TO 30
C
C   FORM THE DIFFERENCE EXPRESSIONS AND SUM THEM TO GET OUT(K)
C
  58    TMP1=WVE(K,INDHI)-WVE(K,INDLO)
        TMP2=TMP2+((XV(L)-VAR(IT2+LHI))*TMP1/
     1   (VAR(IT2+LHI)-VAR(IT2+LLO)))
C
C   RESET TEMPORARIES IT1 AND IT2 FOR NEXT DIMENSION
C
        IT1=IT1+LV(L)
  60    IT2=IT2+LV(L)
  70    OUT(K)=WVE(K,INDHI)+TMP2
      RETURN
      END
```

APPENDIX B

PROGRAM CGEN

TABLE III

Input Data for Program CGEN

Independent Variables			Dependent Variables		
RPM (* 1000)	Voltage (volts)	Power (MW)	Sp. Wt. (1b/kW)	Efficiency (%)	Volume (cu. in * 1000)
8	577	4.91	.224	94.9	16.7
10	577	4.91	.194	95.3	10.1
12	577	4.91	.160	95.5	7.0
14	577	4.91	.140	95.8	5.5
16	577	4.91	.134	95.4	4.8
18	577	4.91	.124	95.4	4.1
20	577	4.91	.122	94.9	3.8
8	1154	4.91	.231	94.8	16.9
10	1154	4.91	.185	95.5	10.2
12	1154	4.91	.164	95.7	7.2
14	1154	4.91	.145	95.8	5.6
16	1154	4.91	.138	95.7	4.9
18	1154	4.91	.128	95.6	4.2
20	1154	4.91	.132	95.0	4.0
8	1731	4.91	.226	94.9	17.0
10	1731	4.91	.195	95.4	10.5
12	1731	4.91	.162	95.8	7.4
14	1731	4.91	.157	95.6	5.9
16	1731	4.91	.146	95.6	5.0
18	1731	4.91	.133	95.6	4.4
20	1731	4.91	*	*	*

*No Data Available

41

Independent Variables			Dependent Variables		
RPM (* 1000)	Voltage (volts)	Power (MW)	Sp. Wt. (lb/kW)	Efficiency (%)	Volume (cu. in * 1000)
8	2308	4.91	.237	94.7	17.3
10	2308	4.91	.195	95.4	10.6
12	2308	4.91	.176	95.6	7.6
14	2308	4.91	.160	95.7	6.0
16	2308	4.91	.149	95.7	5.1
18	2308	4.91	.137	95.5	4.6
20	2308	4.91	.134	95.3	4.2
8	2885	4.91	.245	94.6	17.6
10	2885	4.91	.197	95.4	10.7
12	2885	4.91	.170	95.7	7.8
14	2885	4.91	.160	95.7	6.2
16	2885	4.91	.149	95.5	5.1
18	2885	4.91	.147	95.5	4.7
20	2885	4.91	.142	95.3	4.5
8	577	6.87	.196	95.4	17.7
10	577	6.87	.172	95.8	11.3
12	577	6.87	.154	95.8	8.4
14	577	6.87	.139	95.6	6.8
16	577	6.87	.137	95.3	6.0
18	577	6.87	.121	95.0	5.3
20	577	6.87	.120	94.3	4.9

Independent Variables			Dependent Variables		
RPM (* 1000)	Voltage (volts)	Power (MW)	Sp. Wt. (lb/kW)	Efficiency (%)	Volume (cu. in. * 1000)
8	1154	6.87	.197	95.5	18.0
10	1154	6.87	.175	95.8	11.5
12	1154	6.87	.154	96.0	8.6
14	1154	6.87	.140	96.0	6.9
16	1154	6.87	.134	95.6	6.2
18	1154	6.87	.125	95.5	5.4
20	1154	6.87	*	*	*
8	1731	6.87	.201	95.6	18.2
10	1731	6.87	.188	95.7	11.8
12	1731	6.87	.156	96.0	8.7
14	1731	6.87	.149	95.8	7.2
16	1731	6.87	.138	95.7	6.3
18	1731	6.87	.130	95.5	5.7
20	1731	6.87	*	*	*
8	2308	6.87	*	*	*
10	2308	6.87	.177	95.8	12.1
12	2308	6.87	.160	96.0	8.9
14	2308	6.87	.154	95.8	7.4
16	2308	6.87	.143	95.7	6.4
18	2308	6.87	.135	95.6	5.9
20	2308	6.87	*	*	*

*No Data Available

Independent Variables			Dependent Variables		
RPM (* 1000)	Voltage (volts)	Power (MW)	Sp. Wt. (1b/kW)	Efficiency (%)	Volume (cu. in * 1000)
8	2885	6.87	.215	95.4	18.8
10	2885	6.87	.186	95.8	12.2
12	2885	6.87	.173	95.8	9.3
14	2885	6.87	.154	96.0	7.6
16	2885	6.87	.151	95.6	6.7
18	2885	6.87	.142	95.5	6.0
20	2885	6.87	*	*	*
8	1154	8.83	.188	95.7	19.1
10	1154	8.83	.168	96.0	12.8
12	1154	8.83	.153	95.9	9.9
14	1154	8.83	.136	95.9	8.1
16	1154	8.83	.138	95.4	7.5
18	1154	8.83	.131	95.1	6.8
20	1154	8.83	*	*	*
8	577	8.83	*	*	*
10	577	8.83	*	*	*
12	577	8.83	*	*	*
14	577	8.83	*	*	*
16	577	8.83	*	*	*
18	577	8.83	*	*	*
20	577	8.83	*	*	*

*No Data Available

Independent Variables			Dependent Variables		
RPM (* 1000)	Voltage (volts)	Power (MW)	Sp. Wt. (lb/kW)	Efficiency (%)	Volume (cu. in * 1000)
8	1731	8.83	.185	95.8	19.5
10	1731	8.83	.166	96.1	13.0
12	1731	8.83	.152	96.1	10.1
14	1731	8.83	.142	95.9	8.4
16	1731	8.83	.141	95.6	7.7
18	1731	8.83	.128	95.5	6.8
20	1731	8.83	*	*	*
8	2308	8.83	.189	95.9	19.7
10	2308	8.83	.171	96.0	13.3
12	2308	8.83	.156	96.1	10.3
14	2308	8.83	.151	95.8	8.8
16	2308	8.83	.145	95.6	7.9
18	2308	8.83	.141	95.2	7.3
20	2308	8.83	*	*	*
8	2885	8.83	.195	95.8	20.1
10	2885	8.83	.173	96.0	13.4
12	2885	8.83	.158	96.1	10.5
14	2885	8.83	.154	95.8	9.0
16	2885	8.83	.147	95.7	8.1
18	2885	8.83	.137	95.5	7.3
20	2885	8.83	*	*	*

*No Data Available

Listing of Program CGEN

```fortran
      PROGRAM CGEN(INPUT,OUTPUT)
      DIMENSION WVE(3,105),VAR(15),XV(3),LWK(3),OUT(3),LV(3)
C
C  PRINT COLUMN HEADINGS FOR OUTPUT DATA
C
  2   PRINT*,"RPM *K     V L-N       PWR (MW)   SPWT LB/KW    EFF     V
     1IN"
  4   FORMAT(2(F5.0,5X),4(F6.3,6X))
C
C  READ INFORMAYION NEEDED TO DIMENSION ARRAYS
C
      READ*, NDIM,LDI,LDJ,LVT
      READ*, (LV(I),I=1,NDIM)
C
C  READ THE INDEPENDENT VARIABLE VALUES
C
      READ*, (VAR(I),I=1,LVT)
C
C  READ THE DEPENDENT VARIABLE VALUES
C
      DO 10 J=1,LDI
 10   READ*, (WVE(J,K),K=1,105)
C
C  READ THE TEST POINT
C
 15   READ*, (XV(I),I=1,NDIM)
      IF(EOF(5LINPUT).NE.0.)STOP "END OF RUN"
      CALL  INTERP(NDIM,LV,LVT,LDI,LDJ,VAR,WVE,
     1 XV,LWK,OUT,IERR)
C
C  IF THERE WAS AN ERROR, GO TO THE NEXT POINT
C
      IF(IERR.NE.0) GO TO 15
C
C  PRINT THE INDEPENDENT AND DEPENDENT VALUES AT THE TEST POINT
C
      PRINT 4, ((XV(I),I=1,3),(OUT(J),J=1,3))
C
C  COMPUTE THE SPEC. WT. USING THE ALGORITHM, FOR COMPARISON
C
      SW=.157*(1.28-.28*((XV(3)/5)**.449))*(-.06+1.06*((XV(1)/14)
     **(-.62 05)))*(.8567+.1433*(XV(2)*SQRT(3.)/1000))
C
C PRINT THE SPEC. WT. COMPUTED
C
      PRINT*,"SP WT (FROM ALG)=",SW
      GO TO 15
      END
```

```
3.3.105.15
7 5 3
8.10,12.14.16.18,20.577,1154,1731,2308,2885,4.91,6.87,8.83
.224 .194 .160 .140 .134 .124 .122 .231 .185 .164 .145 .138
.128 .132 .226 .195 .162 .157 .146 .133 .000 .237 .195 .176
.160 .149 .137 .134 .245 .197 .170 .160 .149 .147 .142 .196
.172 .154 .139 .137 .121 .120 .197 .175 .154 .140 .134 .125
.000 .201 .188 .156 .149 .138 .130 .000 .000 .177 .160 .154
.143 .135 .000 .215 .186 .173 .154 .151 .142 .000 .000 .000
.000 .000 .000 .000 .000 .188 .168
.153 .136 .138 .131 .000 .185 .166 .152 .142 .141 .123 .000
.189 .171 .156 .151 .145 .141 .000 .195 .173 .158 .154 .147
.137 .000
94.9 95.3 95.5 95.8 95.4 95.4 94.9 94.8 95.5 95.7 95.8 95.7
95.6 95.0 94.9 95.4 95.8 95.6 95.6 95.6 00.0 94.7 95.4 95.6
95.7 95.7 95.5 95.3 94.6 95.4 95.7 95.7 95:7 95.5 95.3 95.4
95.8 95.8 95.6 95.3 95.0 94.3 95.5 95.8 96.0 96.0 95.6 95.5
00.0 95.6 95.7 96.0 95.8 95.7
95.5 00.0 00.0 95.8 96.0 95.8 95.7 95.6 00.0 95.4 95.8 95.8
96.0 95.6 95.5 00.0 00.0 00.0 00.0 00.0 00.0 00.0 00.0
95.7 96.0 95.9 95.9 95.4 95.1 00.0 95.8
96.1 96.1 95.9 95.6 95.5 00.0 95.9 96.0 96.1 95.8 95.6 95.2
00.0 95.8 96.0 96.1 95.8 95.7 95.5 00.0
16.7 10.1 7.0 5.5 4.8 4.1 3.8 16.9 10.2 7.2 5.6 4.9 4.2 4.0
17.0 10.5 7.4 5.9 5.0 4.4 0.0 17.3 10.6 7.6 6.0 5.1 4.6 4.2
17.6 10.7 7.8 6.2 5.1 4.7 4.5 17.7 11.3 8.4 6.8 6.0 5.3 4.9
18.0 11.5 8.6 6.9 6.2 5.4 0.0 18.2 11.8 8.7 7.2 6.3 5.7 0.0
0.0 12.1 8.9 7.4 6.4 5.9 0.0 18.8 12.2 9.3 7.6 6.7 6.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0
19.1 12.8 9.9 8.1 7.5 6.8 0.0 19.5 13.0 10.1 8.4 7.7 6.8 0.0
19.7 13.3 10.3 8.8 7.9 7.3 0.0 20.1 13.4 10.5 9.0 8.1 7.3 0.0
8 577 4.91
9 577 4.91
10 577 4.91
11 577 4.91
12 577 4.91
13 577 4.91
14 577 4.91
15 577 4.91
16 577 4.91          18 1154 8.83
17 577 4.91          18 1200 8.83
18 577 4.91          18 1400 8.83
19 577 4.91          18 1600 8.83
20 577 4.91          18 1731 8.83
18 2885 4.91         18 1800 8.83
18 2885 5.           18 2000 8.83
18 2885 5.5          18 2200 8.83
18 2885 6.           18 2308 8.83
18 2885 6.5          18 2400 8.83
18 2885 6.87         18 2600 8.83
18 2885 7.           18 2800 8.83
18 2885 7.5          18 2885 8.83
18 2885 8.           ◆EOR
18 2885 8.5          ◆EOF
18 2885 8.83
```

Output from CGEN

RPM •K	V L-N	PWR (MW)	SPWT LB/KW	EFF	VOL KCU IN
8.	577.	4.910	.224	94.900	16.700
SP WT (FROM ALG)=.2265851858366					
9.	577.	4.910	.209	95.100	13.400
SP WT (FROM ALG)=.2099506545953					
10.	577.	4.910	.194	95.300	10.100
SP WT (FROM ALG)=.19606479942					
11.	577.	4.910	.177	95.400	8.550
SP WT (FROM ALG)=.1842652009665					
12.	577.	4.910	.160	95.500	7.000
SP WT (FROM ALG)=.1740842274673					
13.	577.	4.910	.150	95.650	6.250
SP WT (FROM ALG)=.1651918460602					
14.	577.	4.910	.140	95.800	5.500
SP WT (FROM ALG)=.1573433828831					
15.	577.	4.910	.137	95.600	5.150
SP WT (FROM ALG)=.1503540225909					
16.	577.	4.910	.134	95.400	4.800
SP WT (FROM ALG)=.1440813001396					
17.	577.	4.910	.129	95.400	4.450
SP WT (FROM ALG)=.1384134517209					
18.	577.	4.910	.124	95.400	4.100
SP WT (FROM ALG)=.1332614447906					
19.	577.	4.910	.123	95.150	3.950
SP WT (FROM ALG)=.1285533911983					
20.	577.	4.910	.122	94.900	3.800
SP WT (FROM ALG)=.1242305464016					
18.	2885.	4.910	.147	95.500	4.700
SP WT (FROM ALG)=.2096072064393					
18.	2885.	5.000	.147	95.500	4.760
SP WT (FROM ALG)=.2091315750777					
18.	2885.	5.500	.145	95.500	5.091
SP WT (FROM ALG)=.2065712856301					
18.	2885.	6.000	.144	95.500	5.423
SP WT (FROM ALG)=.2041362995148					
18.	2885.	6.500	.143	95.500	5.755
SP WT (FROM ALG)=.2018107499669					
18.	2885.	6.870	.142	95.500	6.000
SP WT (FROM ALG)=.2001526730145					
18.	2885.	7.000	.142	95.500	6.086
SP WT (FROM ALG)=.1995816274473					
18.	2885.	7.500	.140	95.500	6.418
SP WT (FROM ALG)=.1974390167889					
18.	2885.	8.000	.139	95.500	6.749
SP WT (FROM ALG)=.1953735623996					
18.	2885.	8.500	.138	95.500	7.081
SP WT (FROM ALG)=.1933780834534					
18.	2885.	8.830	.137	95.500	7.300
SP WT (FROM ALG)=.1920963068465					

RPM *K	V L-M	PWF (MW)	SPWT LB/KW	EFF	VOL KCU IN
18.	1154.	8.830	.131	95.100	6.800

SP WT (FROM ALG)=.1396205175665

| 18. | 1200. | 8.830 | .131 | 95.132 | 6.800 |

SP WT (FROM ALG)=.1410150215183

| 18. | 1400. | 8.830 | .130 | 95.271 | 6.800 |

SP WT (FROM ALG)=.1470780820914

| 18. | 1600. | 8.830 | .129 | 95.409 | 6.800 |

SP WT (FROM ALG)=.1531411426645

| 18. | 1731. | 8.830 | .128 | 95.500 | 6.800 |

SP WT (FROM ALG)=.1571124473398

| 18. | 1800. | 8.830 | .130 | 95.464 | 6.860 |

SP WT (FROM ALG)=.1592042032375

| 18. | 2000. | 8.830 | .134 | 95.360 | 7.033 |

SP WT (FROM ALG)=.1652672638106

| 18. | 2200. | 8.830 | .139 | 95.256 | 7.206 |

SP WT (FROM ALG)=.1713303243837

| 18. | 2308. | 8.830 | .141 | 95.200 | 7.300 |

SP WT (FROM ALG)=.1746043770932

| 18. | 2400. | 8.830 | .140 | 95.248 | 7.300 |

SP WT (FROM ALG)=.1773933849568

| 18. | 2600. | 8.830 | .139 | 95.352 | 7.300 |

SP WT (FROM ALG)=.1834564455299

| 18. | 2800. | 8.830 | .138 | 95.456 | 7.300 |

SP WT (FROM ALG)=.189519506103

| 18. | 2885. | 8.830 | .137 | 95.500 | 7.300 |

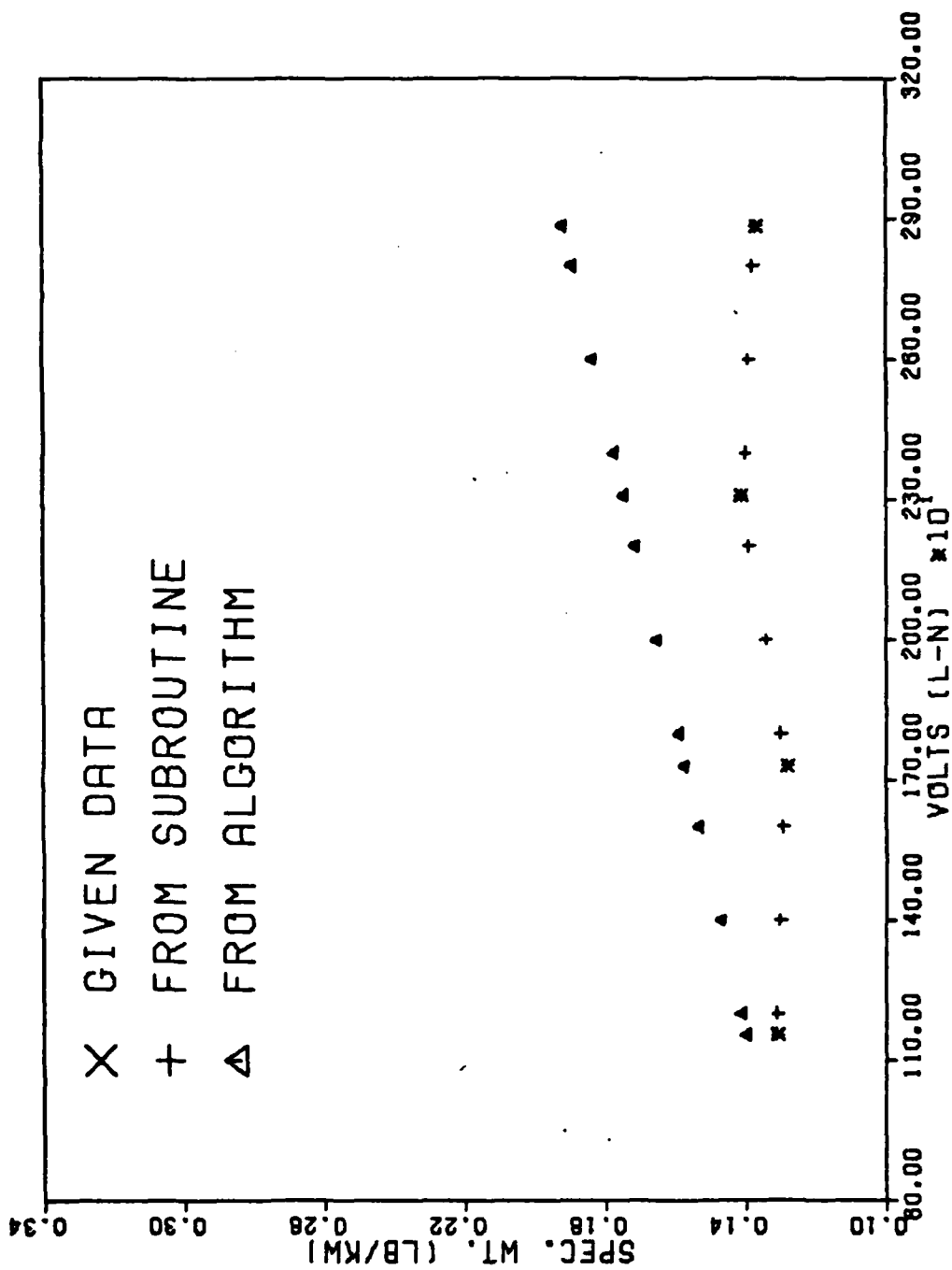SP WT (FROM ALG)=.1920963068465

      STOP END OF RUN

Fig 10. Comparison of Algorithm and Subroutine, Changing Voltage

X CIVEN DATA

+ FROM SUBROUTINE

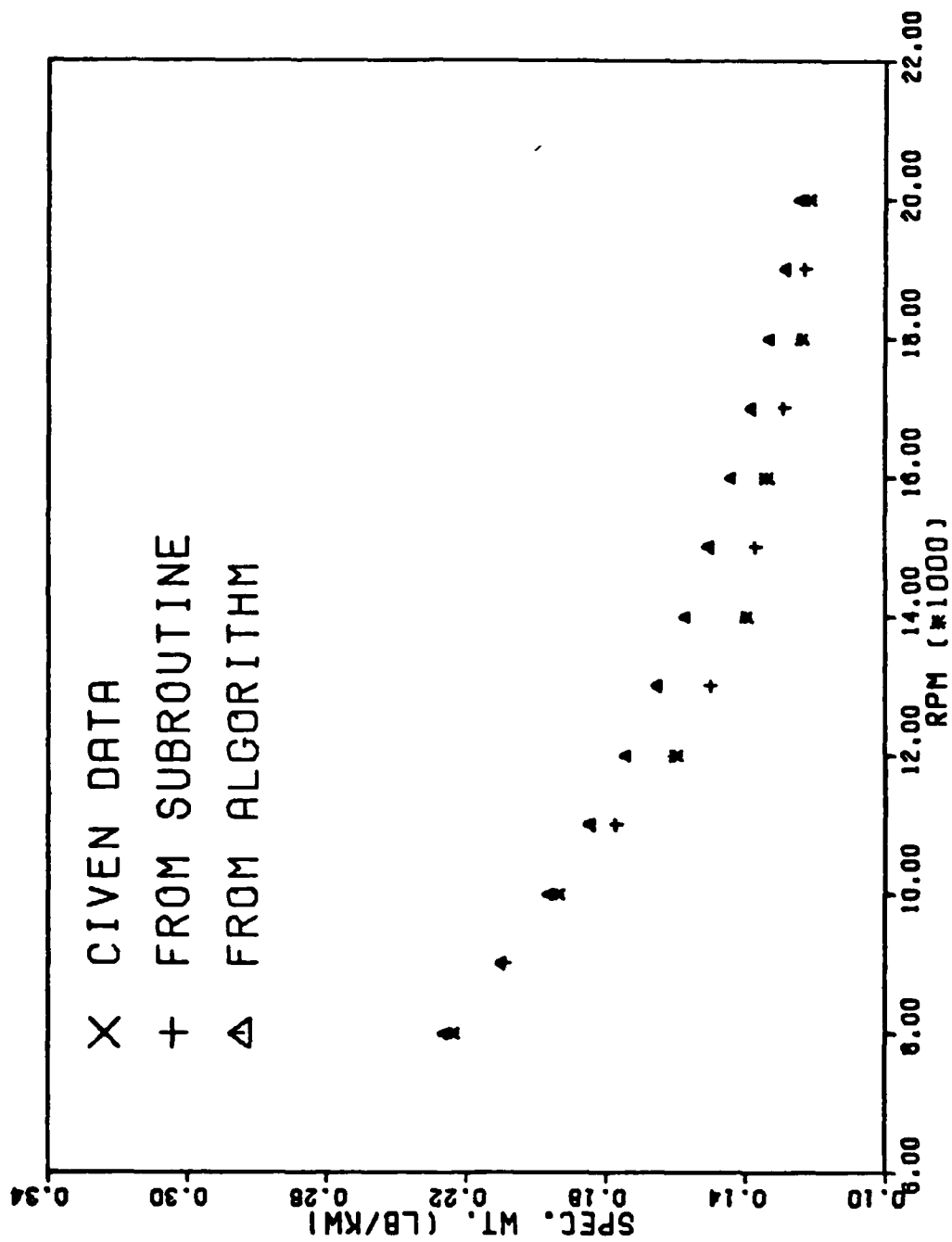△ FROM ALGORITHM

SPEC. WT. (LB/KW)

RPM (*1000)

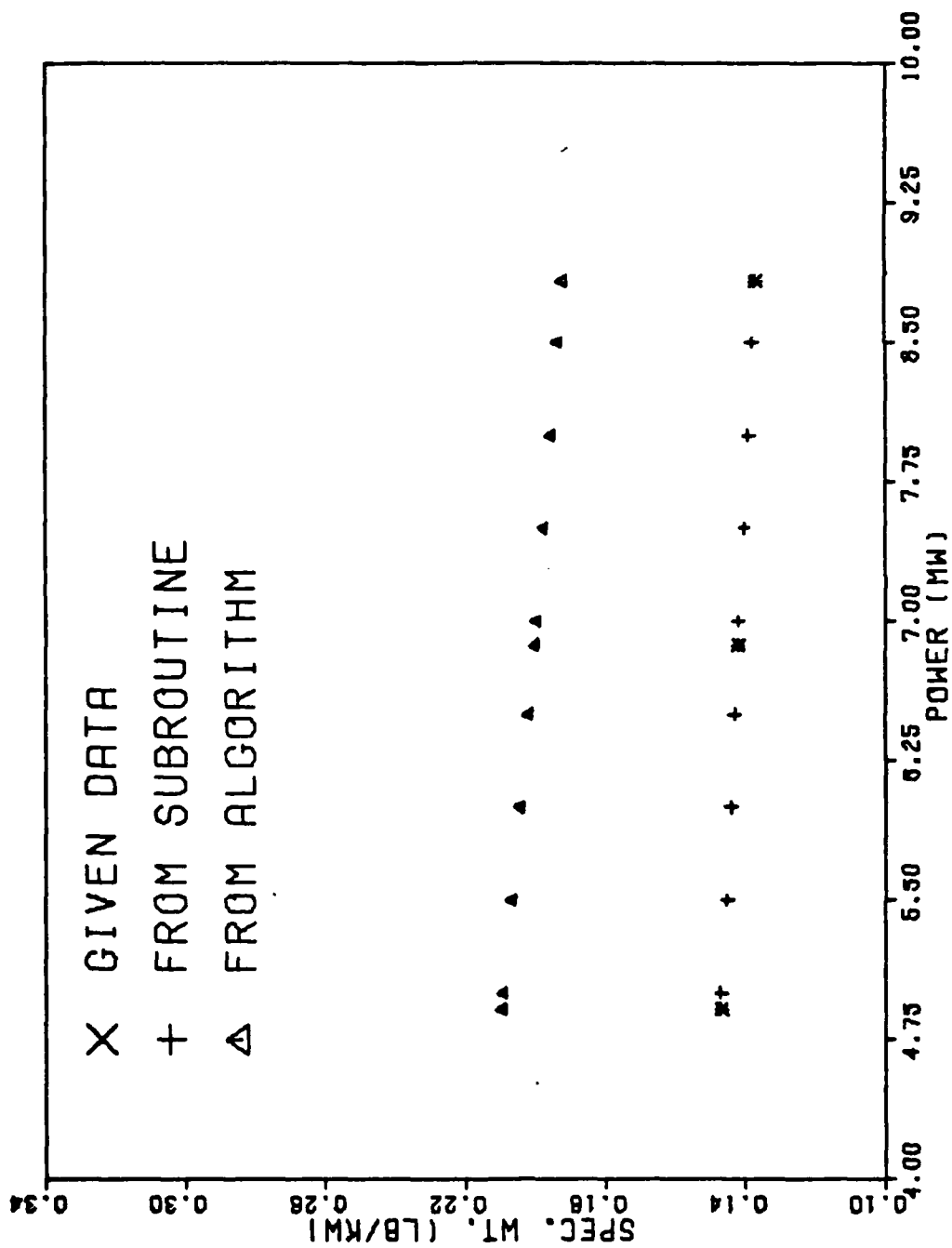Fig 11. Comparison of Algorithm and Subroutine, Changing RPM

51

Fig 12. Comparison of Algorithm and Subroutine, Changing Power

Vita

Gerald D. Clark was born on February 23, 1950 in Sellersville, Pennsylvania. He graduated from Pennridge High School, Perkasie, Pennsylvania in 1968 and entered Lehigh University in Bethlehem, Pennsylvania. He graduated in June 1972 with a Bachelor of Science degree in Electrical Engineering and received his commission in the USAF through the ROTC program. He completed pilot training in October 1973 at Moody AFB, Georgia. He served as a T-37 instructor pilot at Vance AFB, Oklahoma and at Randolph AFB, Texas until entering the School of Engineering, Air Force Institute of Technology, in June 1979.

Permanent Address:

713 Allentown Road

Sellersville, Pa. 18960

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/80D-14 ✓	2. GOVT ACCESSION NO. AD-A100 762	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* INTERPOLATING SUBROUTINE FOR HIGH POWER SYSTEM DESIGN		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gerald D. Clark Capt USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Aero Propulsion Laboratory Air Force Wright Aeronautical Laboratories Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 60
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, If different from Report)*

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

Fredric C. Lynch
FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs   16 JUN 1981

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Multi-dimension Interpolation
Data Fitting
Computer Aided Design
High Power Electrical Systems

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The Aero Propulsion Laboratory is currently developing a computer-aided design program for high power airborne systems. An important part of this design program will be the feasibility study which was to be based on summary algorithms. These algorithms were to relate the weight and volume of each system component to the contributing operating parameters.

DD ₁ FORM 73 **1473**   EDITION OF 1 NOV 65 IS OBSOLETE   UNCLASSIFIED

BLOCK 20:

This study first centers on the requirements for an inter-
polation scheme to form the summary algorithms. The scheme will
need to work in at least four dimensions and produce accurate
results over wide ranges. The advantages of the two possible
interpolation approaches (algorithm development and direct
interpolation) are described. The results of this comparison
show clear advantages in the direct interpolation approach
using stored data.

The remainder of this study is a description and evaluation
of the subroutine which was developed, INTERP. It has the flexi-
bility to interpolate a data array with two or more independent
variables and will output the values of any number of dependent
variables. The routine also compensates for missing values in
the known data array and issues an error message to the user
when a test point beyond the data range is input.

END

DATE
FILMED

7-81

DTIC