CRANFIELD INSTITUTE OF TECHNOLOGY

SCHOOL OF MECHANICAL ENGINEERING

Ph.D. THESIS

Academic Year 1973/4

W.L. MACMILLAN

Development of a Modular Type Computer Program for the Calculation of Gas Turbine Off Design Performance

Supervisor: J.R. Palmer

September, 1974

SUMMARY

This thesis presents a computer program written (in Fortran IV) to simulate the steady state off design point performance of virtually any gas turbine configuration (dealing primarily with aerospace applications). The performance is calculated using component characteristic maps for compressors, combustion chambers, turbines (both compressor turbines and free turbines) and a map giving the velocity coefficient for exhaust nozzles. The off design point performance is determined by using a modified Newton-Raphson method.

With this program, there are no limitations as to the number of shafts, exhaust nozzles, air inlets, or engine configurations which may be simulated. The program has been evaluated on several different engine types and some examples are given. In addition, it has been used to simulate some variable cycle (or variable bypass) engine configurations. The raison d'être of a variable cycle engine is reviewed and the series/parallel concept is investigated.

Simulations on series/parallel engines have shown the choice of design point component parameters and airflow distribution to be a critical point of the engine design if a successful transition between cycles is to occur. The flight conditions at transition, compressor or fan pressure ratio, and internal air flow distribution are shown to be totally interconnected.

Finally, the program is used to compare the performance of some variable cycle engines with conventional fixed cycle engines at typical operating conditions.

ACKNOWLEDGEMENTS

The advice, guidance, and encouragement of the supervisor, Mr. J.R. Palmer is gratefully acknowledged. The advice given by other members of the staff of the School of Mechanical Engineering is also appreciated.

In addition, the interest shown and advice given by Mr. R. Denning, Chief Engineer of Advanced Projects at Rolls Royce, Bristol concerning performance and application of variable cycle gas turbine engines is also acknowledged.

CONTENTS

FIGURES

Notation

1. The notation for most quantities is identical, both in the written work and in the computer listing. The reason for this is to simplify reference to the listing.

2. Thermodynamic and Other Related Quantities

The notation for these quantities follows an orderly sequence, consisting of two letters (or groups of letters in some cases).

First Letter Group

AM (or M)	Flow Mach number
CS	Sonic velocity
F	Fuel/air mass ratio
FS	Stoichometric fuel/air mass ratio
H	Enthalpy
P	Total pressure
PS	Static pressure
R	Universal gas constant
T	Total temperature
TS	Static temperature
S	Entropy
W	Mass flow rate
ϕ (PHI)	Defined as:

$$\phi = \frac{1}{R} \int_{T_o}^{T} \frac{C_p}{T}\, dT, \quad \frac{\Delta S}{R} = \Delta\phi - \ln \left(\frac{PS_2}{PS_1}\right)$$

C_p (SPHT)	Specific heat
γ (GAM)	Ratio of specific heats

Second Letter Group

A	Component inlet
B	Component outlet
S	Quantity for an isentropic process

Examples

PSA = Component inlet static pressure
TBS = Component outlet total pressure assuming an isentropic process

3. Component and Engine Quantities

A	Area (gas flow area, not geometric area)
AIRIN	Engine inlet air mass flow rate
AT	Throat area, convergent/divergent nozzle

AUXWK	Auxilary power extracted from a compressor turbine
CN	Shaft corrected speed as a percent of design corrected speed,
	$CN = \left(\frac{N}{\sqrt{TA}}\right) \div \left(\frac{N}{\sqrt{TA}}\right)_{DESIGN} \times 100\%$
CNSF	Scaling factor for CN when using turbine maps
COEF	Nozzle velocity coefficient
COMWK	Power required to drive compressor
CPR	Cycle Pressure Ratio
DH	Turbine work function,
	$DH = (HA-HB)/TA$
DHSF	Scaling factor for DH when using turbine maps
ETAR	Pressure recovery for inlet (ram recovery)
ETASF	Scaling factor for efficiency when using compressor, combustion chamber, and turbine maps
HP	High pressure
LP	Low pressure
N	Shaft rotational speed
PCN	Shaft rotational speed as a percent of design,
	$PCN = \frac{N}{(N)_{DESIGN}} \times 100\%$
PR	Pressure ratio
PRSF	Scaling factor for PR when using compressor maps
SFC	Specific fuel consumption
SXN	Specific (net) thrust
TET	Turbine entry temperature
TF	Turbine flow function,
	$TF = WA \times \sqrt{TA}/PA$
TFSF	Scaling factor for TF when using turbine maps
WAC	Compressor corrected airflow
	$WAC = \frac{WA \times \sqrt{TA/TSLS}}{PA/PSLS}$ (SLS=sea level static)
WACSF	Scaling factor for WAC when using compressor maps
XG	Gross thrust

XN	Net thrust
XR	Momentum drag
η(ETA)	Component efficency (isentropic unless stated otherwise)

4. Other Notation

B	Brick
BD	Brick data
CW	Codeword
DP	Design Point
EV	Engine Vector
ODP	Off Design Point
SV	Station Vector
μ	Bypass Ratio

5. Subscripts

ACTUAL	Value of quantity obtained from actual component conditions as opposed to a MAP value
D,DESIGN	Design point conditions
MAP	Value of parameter obtained from component map

SECTION 1

INTRODUCTION

1.1 General Introduction

Since the introduction of the digital computer, several schemes have been written to apply the computer's talents to the field of gas turbine performance prediction. The purpose of the work described in this thesis was to develop an extremely flexible computer program with the capability of calculating the steady state performance at both the design point (DP) and off design point (ODP) conditions of practically any configuration of gas turbine engine.

While being primarily concerned with engines having an aircraft application, with the addition of suitable subroutines (representing intercoolers, heat exchangers, etc), other types of gas turbine engines could be simulated (for example; marine or industrial units).

The high degree of flexibility of this program has enabled studies to be made of possible future variable cycle engines. Such engines would have completely different configurations with different internal mass flow distribution at various flight conditions. Such studies would be extremely difficult (if not impossible) with most standard "general purpose" ODP programs. The variable cycle performance predictions obtained during the course of this work, while giving perhaps the ultimate test to a program's flexibility, have also produced an insight into the performance and design problems of these engines.

1.2 Uses of Computers in Gas Turbine Work

A computer simulation of a gas turbine engine can find many different applications(1,2)'. The use of engine performance computer programs begins in the engine concept stage, continues throughout the development program, and is even useful for engines in service.

The initial requirement of performance programs is for DP cycle studies of alternate engine configurations to select those worthy of more detailed evaluation. Once a particular cycle has been chosen, performance programs are subsequently used to predict the steady state performance of these engines over the complete operating range. Also at this time, programs using analogue (3), digital (4), or hybrid (5) computers may be employed to check the transient responses of the engine.

Even after an engine is constructed, a steady state performance program is still extremely useful. References 1 and 2 provide a good description of the usefulness of steady state programs at this stage. The uses can be divided into two broad categories: monitor present (in use) engines, and application studies of engines.

' figures in brackets refer to references listed at back.

Engine Monitoring. The following uses are found in this category:

a. Determine sensitive or critical areas in near-future engines under development. This is accomplished by comparing the computer predicted performance with the actual test cell performance. If the two are not in agreement, the computer program can then be used to try and isolate the reason for the poor performance and then investigate any proposed solution.

b. Determine the effects of manufacturing tolerances.

c. Diagnosing the effects of component performance deterioration on engine performance.

d. Evaluate the effect of modifications on an engine's performance.

Application Studies. This category encompasses the following uses:

a. Explore the advantages and disadvantages of proposed engines for future aircraft.

b. Provide an economical means of obtaining engine data for aircraft mission analysis and specifying the performance in new environments.

c. Intelligence gathering. If a few of the basic parameters are known about an otherwise totally unknown engine, it is possible to reasonably estimate the engine's performance over the entire operating range.

SECTION 2

2.1 Historical Development

Until the early 1960's, most engine cycle calculations were done manually, although a few computer programs were available for specific engines. Gradually, as the use of computers became more widespread, programs for gas turbine engines began to be developed, first to do the DP calculations, and later, the ODP calculations.

Basically, the evolution of these programs has been that of increasing flexibility, ie: the ability to analyse several different engine types or configurations using the same general program. The primary reason for this was to free the gas turbine engineer from writing computer programs so that he could devote full time to putting the programs to use.

Initially, a program was written to apply to just one engine. This was later extended to be suitable for all engines of a similar configuration. The next stage was to write a program for the most complicated engine. This program could then be used for simpler engines by "short-circuiting" a component. For example, if the program was written for a two shaft engine, a one shaft engine could be simulated by making the pressure ratio of one of the compressors equal to 1. The last stage of flexibility has been to write a program in which the subroutinesrepresenting components are called up only as they are required and the engine is "pieced together" by the user.

In a parallel path to these steady state programs, there has also been an evolution of transient condition programs. The most widely published work in this field is by Saravanamuttoo and Fawke (3,4,5). The advantages of determining an engine's transient performance in the design stage are obvious. However, this is not to say that steady state programs do not have a role to play since the fexibility of present transient programs does not approach that of existing steady state programs. The method of these dynamic behaviour programs is indeed general and flexible, but the whole program must be rewritten if another engine configuration or cycle is to be studied.

2.2 Design Point Programs

DP programs are the obvious starting point in any review of previous work.. The earliest programs were written to describe only one particular cycle and could only be used to study the affect of different engine parameters (such as compressor pressure ratio, bypass ratio, turbine inlet temperature, etc.) on that particular engine configuration.

2.2.1 National Research Council of Canada DP Program

Two of the earlier programs which tried to build in more flexibility are those described in references 6 and 7, published by the National Research Council (NRC) of Canada in 1964 and 1965. These two companion reports dealt with turbojets and turbofans, and turboprops and turboshafts respectively. They were capable of doing DP studies on engines of up to two shafts, but had certain restrictions in that the engine configuration had to follow the standard pattern of engine layout. For example, it was not possible to simulate on cross compound cycle engine.

In both of these programs, a three level building block approach was taken (see figure 1). A master control program calls various engine component subroutines, which in turn refer to the general thermodynamic data routines. For both programs, the bottom two levels are practically identical, but the master controlling segments are completely different. Obviously this restricts flexibility since two separate programs are required to simulate engines which use only the standard configuration of components. This type of program soon evolved into a more general program with increased flexibility (and, as was mentioned previously, the over-riding aim in the development of these programs has been to increase flexibility and ease of use).

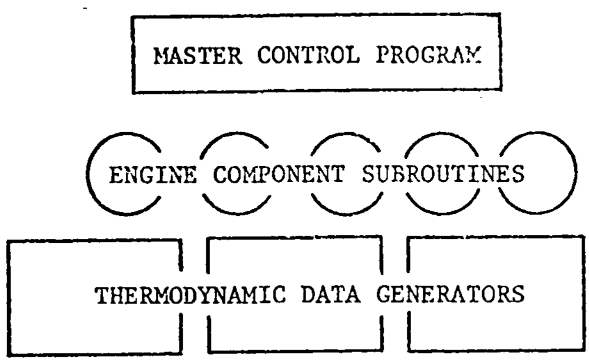


Figure 1 Three Level Program Structure

2.2.2 TURBOCODE

The next stage in flexibility was that afforded by the TURBOCODE scheme as described in references 8 and 9. The scheme has been described extensively in these references and there is no need to go into any details here. Suffice to say that a user, by calling up subroutines representing the various thermodynamic processes and inputing basic engine and component data such as efficiencies, nozzle coefficients, etc., is able to simulate virtually any engine configuration.

Like the NRC programs, TURBOCODE may be thought of as a three level program, and in fact, the same threee levels exist. The difference lies in the basic philosophy of the scheme. This results in a master segment with a completely different purpose.

The NRC master segment calls the component subroutines in a pre-determined sequence. This sequence is built into the program and cannot be altered except by complete reprogramming. If a component is not used in a simulation, it is merely bypassed and control follows to the next component in the sequence. With, TURBOCODE however, the user specifies the sequence of operations and control passes back and forth between the master segment and the component routines in the desired sequence. By permitting the user to specify the sequence, the result is a much more flexible program.

One additional useful feature TURBOCODE possessed was its ability to "loop" through the engine, incrementing some engine or component parameter each time. In this way, it was possible to study the effect of different bypass ratios, pressure ratios, etc., without having to re-imput all the data. This program has been used for approximately seven years now at the Cranfield Institute of Technology (formely College of Aeronautics) by post graduate students when doing engine design work.

It was considered desireable to have a program with the flexibility and ease of use of TURBOCODE, but capable of performing off design calculations. This is what has been accomplished with the program described in this thesis.

2.3 Off Design Point Programs

For a long time, effort was directed towards producing a program capable of accurately predicting an engine's ODP performance. To calculate an engine's performance at an ODP inevitably involves some iterative technique since at each engine operating point, the performance of the components varies as a result of the various interacting relationships established by the continuity requirements, power balances, etc.

2.3.1 The ODP Problem

In order to have steady state operation, several constraints must be satisfied:

a. The power produced by a compressor turbine must equal the power required by the compressor.

b. The sum of the mass flows in and out of any component must be equal.

c. The static pressure of both streams entering a mixing region must be equal in the mixing plane.

d. The pressure at the entrance of a nozzle must be such that the nozzle is capable of operating.

e. The rotational speeds of components on the same shaft system must be equal, or (if geared), proportional.

For each steady state operating point, there will be a unique set of component operating conditions such that all the above constraints are satisfied. The problem then is to find this unique set of conditions in the most efficient manner.

An engine performance calculation consists basically of the solution of a set of simultaneous non-linear equations in several unknowns. These equations are expressed in the form of the characteristics of the components rather than in an algebraic form. The shape of these characteristics demonstrates the non linearity of the equations. Because of the nature of the equations, a trial and error approach is required in their solution.

For this reason, the earlier efforts were directed towards finding an efficient method of adjusting the various component operating conditions to produce a "balanced" engine. It was not until an iterative technique was developed which accurately and efficiently predicted an engine's behaviour that it became possible to work towards the aim of developing an ODP program with the flexibility of a DP program such as TURBOCODE.

2.3.2 Basic Iteration Methods

Obviously, to calculate the performance at an ODP, initial guesses have to be made for such quantitites as turbine inlet temperature, compressor pressure ratio, bypass ratio, mass flow, etc. To work through the engine using the most basic iteration technique (merely an extension of the manual calculations described in any text book such as references 10 and 11), while it may be acceptable for the most simple engine, does not lend itself for efficient calculations concerning more complex engines. As the complexity of the engine is increased, the number of iterative loops that are set up increase at an even faster rate (although not necessarily by any power law). Some names given to this approach are the nested loop, concentric iteration or crossover iteration methods.

To illustrate the difference between the concentric and crossover iteration methods, consider a one spool turbojet operating at an off design shaft speed. Referring to figure 2, the following descriptions apply:

guess 1 -compressor pressure ratio

guess 2 -turbine inlet temperature

check 1 -continuity check. Compressor mass flow + fuel flow equals turbine mass flow check 2 -nozzle area required to pass turbine mass flow equals design point exit area

calculation 1 -calculate compressor work per unit mass

calculation 2 -calculate turbine pressure ratio required to produce work of calculation 1

calculation 3 -using component characteristic maps (figure 3) calculate compressor and turbine mass flows. Fuel flow may be determined from a combustion temperature rise chart (figure 3c).

calculation 4 -calculate pressure ratio across jet nozzle from the pressure ratios across inlet, compressor, combustor, and turbine

calculation 5 -calculate area of jet nozzle required to pass turbine mass flow of calculation 3.

Figure 2 is an example of the above calucations illustrating concentric iteration. Crossover iteration is similar except that check 1 is used to change guess 1.

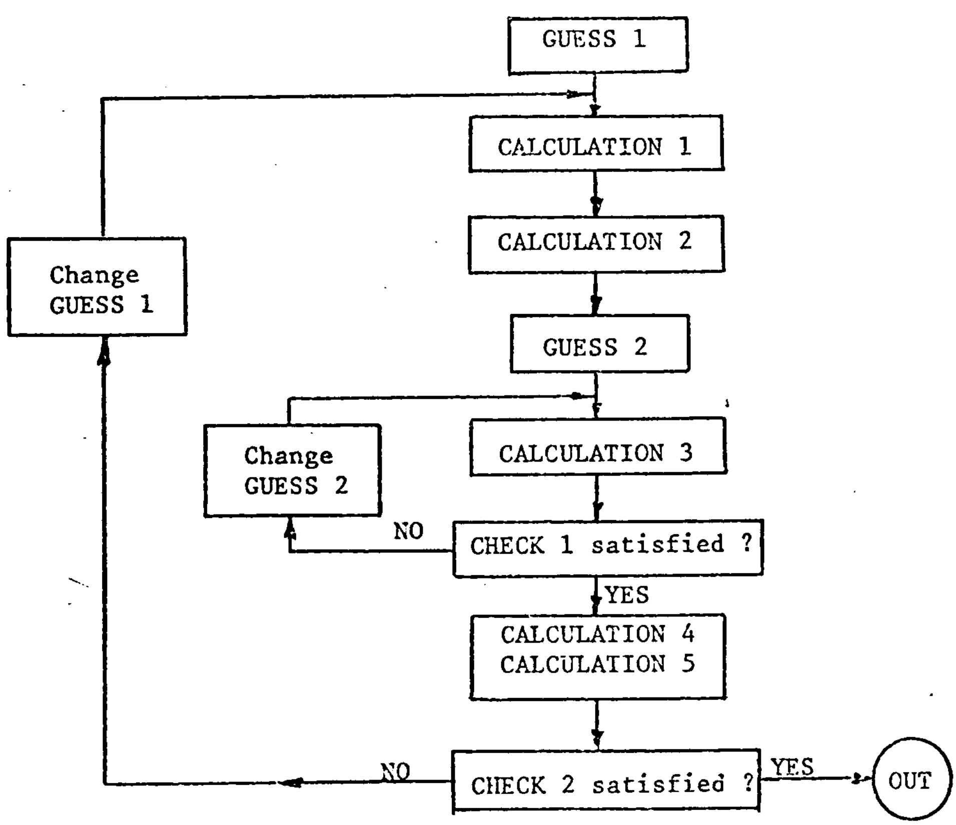
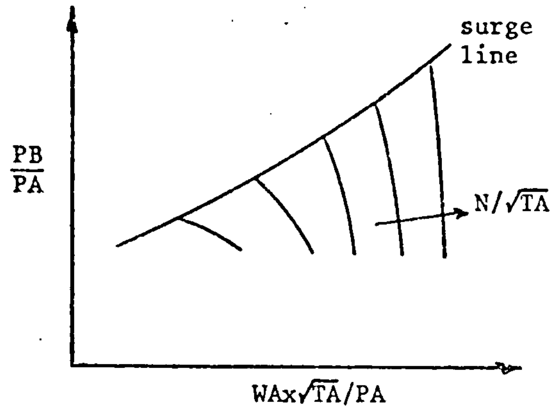
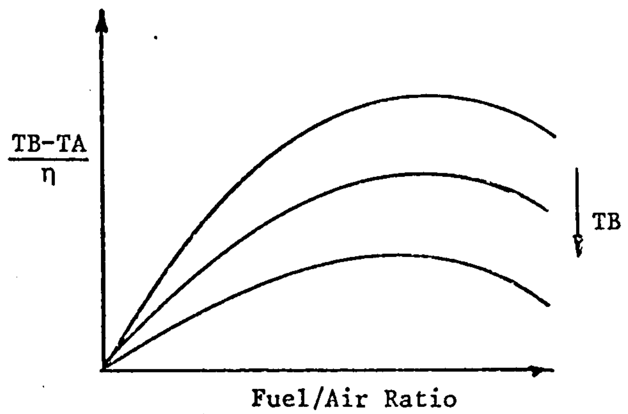


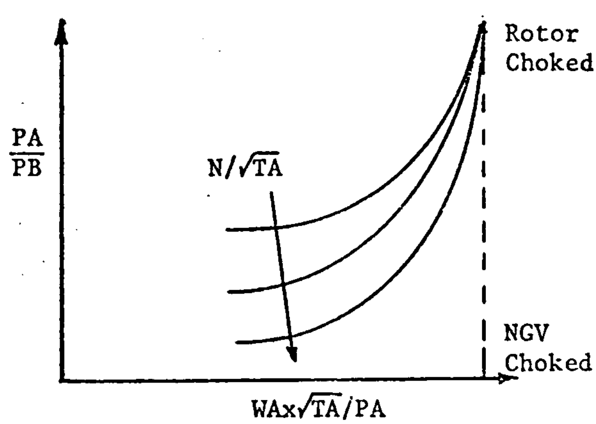
Figure 2 Basic Iteration Technique



a. Compressor Characteristic Map



b. Combustion Temperature Rise Chart



c. Turbine Characteristic Map

Figure 3 Typical Component Characteristic Maps and Combustion Temperature Rise Chart

As mentioned previously, this approach is arduous enough for simple cases and is certainly not recommended for more involved examples. The major drawback is that each check only alters one guess. Since all the guesses and checks are inter-related, the best approach would be one in which all the checks collectively alter all the guesses. This in fact is what occurs in the following more advanced iteration methods.

2.3.3 Previous Work at Cranfield

As his M.Sc. thesis at the College of Aeronautics in 1967, Annand (12) attempted to modify TURBOCODE to perform ODP calculations. While his iteration method was an improvement over the basic iteration approach, he was largely unsuccessful for several reasons. Lack of time and insufficient computer space caused his work never to be completed. His program also had several disadvantages which would have limited its use. One of these was that the user was expected to know some computer programming in order to complete the program (as opposed to just entering data in TURBOCODE) for each different simulation. One good point of his program was that it worked with actual component maps as opposed to assuming things such as constant component efficiencies as other early ODP programs did.

Annand's iteration method was the "hill climbing" technique described by (among others) Rosenbrock in reference 24. This approach should have given an improvement over the simple iteration methods since all the checks do collectively alter all the guesses. Unfortunately, his program was too large to fit on the computers at his disposal. Eventually, he had reached the stage where the iteration method could be applied to a set of simultaneous algebraic equations. However, his work was stopped before it could be tried on an actual engine simulation.

Also at Cranfield, in 1972, Finney (13) was involved with off design performance prediction. He was chiefly interested in finding the optimum engine cycle for a given aircraft mission, so the ODP predictions were necessary, but did not constitute a major part of his work.

Finney used a Newton-Raphson method as the convergence technique. His ODP calculations were greatly simplified since he only dealt with certain engine cycles and made many simplifying assumptions such as constant component efficiencies, etc.

Because Finney's work was not relevant, and Annand's was so incomplete, it was decided not to base the new program on either of them (although the Newton-Raphson method is used in the new program). Thus the program developed for this thesis is not derived from these works and bears no resemblance to either of them.

2.3.4 National Research Council of Canada (NRC) Programs

In 1967, Cockshutt at NRC published a report (14) entitled "Gas Turbine Cycle Calculations: Differential Methods in the Analysis of Equilibrium Operation". In it, as the title implies, was described a method of determining the equilibrium ODP of an engine. The essence of the method was to linearize the operating equations for a turbojet engine by expressing them in logarithmetic differential form. In other words, the object was to establish a set of linear algebraic equations correlating changes in temperatures, pressures, flow rates, etc. within the engine. It was hoped that ODP performance could be examined qualitatively without the detail of a component-characteristic map type of analysis.

The accuracy of the method was only as high as the accuracy of the component defining equations which would be written to define some change (such as change in inlet mass flow brought about by a change in engine rotational speed). It is considered that it would be very difficult to write a general program using this sort of approach, largely because the equations would need to be written to account for different initial changes.

Cockshutt then expanded the DP program of reference 6 to produce an ODP program (15) using the above method of reference 14. Unfortunately, reference 15 is unpublished and few details have been revealed. This ODP program was designed to provide some degree of flexibility, since it was capable of dealing with one or two shaft jet or fan jet engines, with or without reheat. The only information available on this program comes form reference 16, "Experimental Verification of Off Design Point Predictions for a Two-Spool Turboject with Various Air Bleeds", which was published by NRC in 1971.

The experimental verification was termed partially successful, but NRC was not satisfied with the trial overall. The prediction of major cycle parameters for part throttle operation was considered acceptable; but the prediction of responses of these parameters to HP bleed air extraction, changes in propelling nozzle area, and combinations of both perturbations was not considered accurate enough. The program described in this thesis was also evaluated using the experimental results given in reference 16. Generally, the predicted results gave a better agreement than the NRC results. Full details of this test are given in Appendix D.

The NRC ODP program suffered several disadvantages. It did not use component characteristic maps and had to assume that:

a. component (compressor, combustion chamber, turbine, jet pipe nozzle) efficiencies remain constant at the DP values

b. corrected engine intake mass flow rate varies in direct proportion to the corrected low pressure compressor rotational speed.

The program was also incapable of handling HP cooling air bleed.

In summation, from what has been published (or more significantly, what hasn't been published), one may assume that the NRC proposed method of solving the set of simultaneous engine operating equations has not been a total success in that accuracy has not met expectations, and the resulting programs do not offer a great deal of flexibility.

2.3.5 NASA Programs

In November of 1967, a report of yet another ODP program was published by the USAF Aero Propulstion Lab, entitled "Simulation of Turbofan Engine" (SMOTE) (17,18). The methods used in this program have proven to be an accurate and reasonably efficient way of performing ODP engine calculations. Its usefulness was reported in reference 2 and it has lead to more general flexible programs such as GENENG (described later) and the program which is the subject of this thesis.

SMOTE (and the later related programs) solves the ODP engine calculations by employing a modified Newton-Raphson iteration technique (described in greater detail in section 3.3.2). This program uses component maps, but significantly, these maps do not necessarily have to be maps of actual components; they may be those of an imaginary compressor, burbine, etc. Internally generated scaling factors alter the component map to agree with the given DP performance. In this way, SMOTE can deal with theoretical or hypothetical engines. The limitations with SMOTE concerned its flexibility as it was only capable of calculating turbofan (two spool) performance. Another computer program called GENENG was developed by the NASA Lewis Research Center in 1971 - 1972 to make SMOTE a more flexible tool.

GENENG and GENENG II (19,20) are in fact quite similar to SMOTE. The means of using the component maps and the balancing techniques are the same. GENENG increases the flexibility of SMOTE by being capable of simulating one and two spool turbojets and turbofans (with or without reheat, duct burning, mixed or unmixed flows, convergent or convergent/divergent nozzles). GENENG II extends the flexibility to two and three spool turbofans with as many nozzles. By the very fact of requiring two separate programs such as GENENG and GENENG II, it is obvious that the ultimate in flexibility still has not been reached. Also, they are only capable of simulating engines of the standard configuration (although it is possible to simulate an aft fan with GENENG II). Nevertheless, by proper use of these two programs, it is possible to simulate a wide range of engine types and indeed, they are very useful. However, the absolute flexibility afforded by the TURBOCODE approach is not inherent.

2.3.6 Other ODP Programs

As has been mentioned, Finney's program (13), SMOTE (17, 18) and the two GENENG schemes (19,20) all employed the Newton-Raphson technique to solve the simultaneous equations and converge upon a balanced engine. One other program which used this approach was Dennison's in his PhD thesis of 1968 (University of London), "General Non Digital Simulation Methods Applied to Gas Turbine Dynamic Behavior" (25). While the methods were applied to the transient phenomona, it was possible to do steady state calculations. No component maps were used, but regression equations representing the compressor and turbine characteristics were input. Efficiencies, however, were assumed constant. Dennison's approach does bear some similarity to SMOTE and GENENG in that the Newton-Raphson method is applied in the same fashion. That however is where the similarity ends since Dennison's program is very specific and has not attempted to be general. Indeed, a separate program is required for one or two shaft engines. What is interesting is that the Newton-Raphson method appears to have become universally accepted as a means to solving the set of simultaneous equations which define the engine's operation.

The Newton-Raphson approach is described in detail in section 3.3.2. It is seen to be an efficient iteration technique for the ODP gas turbine problem since, like hill climbing, all the checks collectively alter all the guesses. Because of the success experienced by the previous users of this method and the large amount of literature available concerning this particular application, it was decided to employ the Newton-Raphson method in the Cranfield ODP program.

2.3.7 Cranfield ODP Program

The last program in the evolutionary process of increasing flexibility for an efficient ODP program is described in this thesis. Like the two GENENG's, it too borrows from SMOTE the means of using and scaling the component characteristic maps. In addition, the modified Newton-Raphson method of arriving at a balanced engine has been retained. However, this new program contains all the flexibility of TURBOCODE.

As DP aid, it is possible to "loop" the engine by incrementing up to five engine or component parameters. As an ODP simulation, it is possible to calculate the performance of any engine configuration since it is not restricted to a certain number of shafts or components. The user has the power to specify a change in virtually any engine or component parameter to define the ODP condition. It has proven particularly useful in analysing future variable cycle engines, such as those described in references 21 and 22 where the complete internal airflow of the engine is altered at certain ODP conditions.

Two important constraints which were placed on this program were:

a. It must be a piece of software such that the user would not have to be concerned about the internal mechanics of it. In other words, the user need not know anything about computer programming - he merely inputs data representing the engine configuration as well as the basic engine and component parameters.

b. To enable quick turn arounds of the program, it must be efficient in terms of computer storage and operating time. To save operating time, the use of overlay was to be avoided. In fact, the whole program occupies 25K of storage and the running time to converge on a two spool mixed or unmixed engine ODP is less than one half a minute.

SECTION 3

3. Details of Cranfield ODP Program

3.1 Introduction

For preliminary as well as in depth studies, it is often necessary to study a broad range of engines operating at both DP and ODP conditions in order to find an efficient engine/airframe combination. The spectrum of flight conditions through which an engine must operate will strongly affect the optimum design parameters for that engine (19).

The computer program described here is a combination of GENENG and TURBOCODE, incorporating the most desireable features of each. It is capable of performing ODP studies of engine configurations which are literally limited only merely by the user's imagination. The program has proven to be reliable, accurate, and extremely versatile. Without making any change to the program itself, a user can simulate virtually an engine configuration merely by changing the data cards. It is expected that a principal use of this program will be by the M.Sc students at the Cranfield Institute of Technology in the design of engines and in the study of advanced engine designs.

3.1.1 Language

GENENG was written in Fortran while TURBOCODE used Algol. It was therefore necessary to select one of these languages for the new program. Fortran was chosen for the following reasons:

a. The majority of students at Cranfield are now taught basic Fortran as opposed to Algol.

b. It would seem that more scientific reports are written in Fortran, thus making it a more widely accepted language.

c. The COMMON feature of Fortran makes it easier to handle large blocks of data such as component maps.

d. The author was more familiar with Fortran, and thus the time required to learn a new language was saved.

3.1.2 Units

This program is designed to accept data and produce answers in either British or SI units. However, not all units are those conventionally used with the British or SI systems. The units used with this program are given below:

Quantity	"British"	"SI"
Temperature	Degrees Kelvin	Kelvins
Pressure	Atmospheres	Atmospheres
Area	Square Feet	Square Meters
Mass	Pounds Mass	Kilograms
Mass Flow	Pounds mass/sec	Kilograms/sec
Velocity	feet/sec	meters/sec
Force	Pounds force	Newtons
Energy	CHU	Joules
Power	CHU/sec	Watts

The desired system of units is specified by an entry on the first data card (see section 3.6.1). Whatever system is chosen, the units for all data, including component maps, must be consistent.

A close inspection of the program listing will reveal that the program works internally in the "British" units. Input and output are merely automatically multiplied by constants to convert to the system chosen.

3.1.3 Fuels

The program, as written, can consider only one type of fuel, "Topps Standard Kersene" with a lower heating value of 10300 CHU/lbm at 15°C. This fuel has the advantage that its molecular weight is equal to that of air. Hence the gas constant of its combustion products is independant of the fuel/air ratio. Experience has proven that this fuel may be used in computations without significant loss of accuracy for all common aircraft fuels (26).

However, the change to another fuel would not be difficult. The fuel properties of molecular weight, stoichometric fuel/air ratio, and lower heating value are located in the BLOCK DATA segment, COMMON/FUEL. Fuel dependant coefficients used in the segments dealing with thermodynaic quantities, function TRM and subroutine AIRTAB (see 3.2.3) are given in the BLOCK DATA segment COMMON/AIRCONST. Thus, to specify another fuel, it would only be necessary to change the data in these two BLOCK DATA segments. The units for these quantities must be consistent with the "British" system of units used in the program.

3.2 Basic Concepts

The basic concepts of program description follow from those used in TURBOCODE. In order that a user may transfer from TURBOCODE to this program, many of the definitions and operational concepts are identical. For this reason, it may be desirable for a reader to refer to references 8 and 9 for a fuller description of these concepts.

3.2.1 Definitions

Reference will be made to the following terms and hence these are defined as follows:

3.2.1.1 Bricks

The complete program is composed of Bricks which are either subroutines or sections of the program describing either a thermodynamic process (and hence may represent an individual component), or else an operation such as arithmetic or output of results. It is convenient to think of a thermodynamic Brick as a kind of operator which, given the gas state at inlet, calculates the gas state at outlet. By standardizing the layout of the gas state information, a common interface approach becomes possible. Thus, Bricks may be "plugged into" one another to build up an engine simulation. The program then becomes one of a modular structure, with the user at liberty to arrange the modules (ie: Bricks) in any manner desired.

3.2.1.2 Station Vectors

The interfaces between Bricks are referred to as Station Vectors (SV). For the products of combustion of a given fuel in air, the complete gas state is defined by five quantities:

1. fuel/air mass ratio
2. total or static pressure
3. total or static temperature

4 and 5. Any two of: gas mass flow, velocity, area

The minimal SV would consist of just five elements. However, the particular set of five that is convenient or necessary is not the same in all contexts; consequently, a redundant system of eight elements has been adopted as follows:

SV element	quantity
1.	fuel/air mass ratio
2.	gas mass flow
3.	static pressure
4.	total pressure
5.	static temperature
6.	total temperature
7.	velocity
8.	area

SV's are (usually) numbered consecutively from the inlet and are stored in a two dimensional array, SV (M,N) where M = SV number and N = SV element. All SV's are initially set to -1. and remain at that value unless changed in the program. Because any actual values of quantities in an engine would be non negative, if -1. appears in the output, it represents an unknown quantity.

Note: all areas used in this program are the gas flow areas as distinct from actual geometric areas.

3.2.1.3 Brick Data

The SV's alone seldom completely define the action of the Brick. Normally, additional information is needed such as efficiencies, shaft speeds, requirement for the Brick to generate an error, etc. These items constitute what is called Brick Data (BD) and are stored in a one dimensional array BD(N) where N = BD number (these values are loaded in with other input data by the user).

3.2.1.4 Engine Vectors

Similarly, Bricks produce certain results which are not SV items. Examples are thrust, work done on the air by the compressor (hence, required power input to the compressor), fuel burned during combustion, etc. These are known as Engine Vectors (EV) and are internally stored in an array, EV(N) where N represents the EV number, and is numbered consecutively from the start of the program. The user must keep track of these quantities as they are produced by the Bricks since later Bricks may eventually require them as data.

3.2.1.5 Codewords

Codewords (CW) are the means by which a user writes his "program". Each CW contains the following information:

1. What Brick is to be used
2. What SV's the Brick is operating between
3. What BD is required
4. What EV's are required
5. What (if any) BD or SV elements will be regarded as Variables (see section 3.3.1.2 for definition of Variable)

3.2.2 Bricks

As has been stated previously, the primary purpose of this program is to simulate aircraft gas turbine engines. For this reason, it was considered necessary to develop the following Bricks (the detailed thermodynamics and internal workings are to be found in Appendix A).

Note: the symbols are shown as they appear in the computer listing in order to make reference to the listing easier.

3.2.2.1 B1, Inlet

This subroutine calculates the conditions at the inlet of an engine for the given flight conditions. The International

Standard Atmosphere is assumed throughout, with the static temperature and pressure being calculated at the given altitude and ISA temperture deviation. Knowing the flight Mach number (input as BD) allows the total pressure, temperature, and velocity to be calculated.

Ram recovery, defined as PB/PA, is calculated by MIL-E-5008B (USAF) specification, or alternatively (if known) may be input as BD. In addition, B1 must have the inlet mass flow input as data (SV data). With this quantity and the flight velocity known, the momentum drag is calculated.

3.2.2.2 B2, Compression

This Brick simulates the process of compression and hence represents a compressor or fan. To do this, it performs two functions. First, it scales a general compressor map to agree with the DP data, and secondly, it performs the compression thermodynamic calculations to define the outlet SV.

Initially, the following quantities are known from the inlet SV and from the parameters obtained from the map:

FA (and hence FB = FA), WA, TA, PA, PR, η

Knowing FA and TA allows ϕA and HA to be calculated using function TRM which is described in section 3.2.3. ϕBS is then determined by

$$\phi BS = \phi A + \ln(PR)$$

TBS is now calculated using subroutine AIRTAB (described in 3.2.3) and knowing ϕBS and FB. Once TBS is determined, TRM is again used to find HBS. HB can now be calculated by

$$HB = HA + (HBS - HA)/\eta$$

Again, AIRTAB calculates TB from HB and FB. To complete the outlet SV,

$$PB = PA \times PR$$

$$WB = WA$$

Finally, the power required to drive the compressor (COMWK) is easily arrived at from:

$$COMWK = (HB - HA) \times WA$$

3.2.2.3 B3, Combustion

This Brick calculates the outlet SV of the combustion process knowing either the amount of fuel burned or else the final temperature. A combustion chamber map is used to find the combustion efficiency as a function of inlet pressure and temperature increase.

Pressure loss during combustion is accounted for by assuming that the loss is proportional to the kinetic head at the component inlet. Combustion products are calculated by knowing the combustion temperature rise. If TB has been given as input, the fuel flow rate (WFB) necessary to create that temperature is calculated. However, if WFB is given and TB is not, then initially TB is guessed and a new fuel flow rate is calculated. If the new fuel flow rate does not equal the given one, a correction is made to TB and the process repeated until the calculated WFB equals the given value (within a tolerance).

3.2.2.4 B4, Compressor Turbine

This Brick simulates a turbine which is producing power for a compressor. In addition, provision is made for possible auxilary power extraction. In doing this, it performs three functions. First, a general turbine map is scaled to agree with the DP data. Secondly, it performs the calculations associated with the thermodynamics of expansion to define the outlet SV, and thirdly, it calculates errors which are used in solving the ODP engine cycle calculations.

Initially, the following quantities are known from the inlet SV, from parameters determined from the turbine map, from the BD, and as EV's which are the result of a previous calculation:

FA(and hence FB = FA), WA,TA,PA, Auxiliary Power produced (AUXWK), Power required by compressor (COMWK), η, and DH (where $DH = \frac{HA - HB}{TA}$)

First, the shaft speed is matched to that of a compressor which is specified by the operator as BD. Knowing FA and TA allows HA and ϕA to be found by TRM. DH is a quantity taken from the turbine map, and with this value, HB is determined by

$$HB = HA + DH \times TA$$

HBS, the enthalpy assuming on isentropic expansion, is calculated by

$$HBS = HA - (HB - HA)/\eta$$

With HBS and FB known, AIRTAB is used to find TBS and with this value, TRM calculates ϕBS. PB is then found by

$$PB = PA \times e^{(\phi BS - \phi A)}$$

Finally, WB = WA, and the outlet SV is complete.

In an ODP situation, an error will exist for an unbalanced engine since the enthalpy drop calculated from the map value of DH will not equal the enthalpy drop necessary to produce the required output power (AUXWK + COMWK). Another error is produced, that of lack of continuity of mass flow in that the mass flow required by the turbine map will not equal the actual mass flow.

Direct Coupled Turbo Prop

With this Brick it is also possible to simulate a turbo prop engine in which the propellor is driven on the same shaft as a compressor and turbine. In this situation, the auxilary power extraction is considered to be the power delivered to the propellor.

3.2.2.5 B5, Convergent Nozzle

This Brick simulates a convergent nozzle. In the DP case, it calculates exit area assuming isentropic expansion to atmosphere. This area is later used in the ODP situation. In this case, the brick produces an error for an unbalanced engine. Provision is made to account for a variable area (floating) nozzle in which the exit area is varied to satisfy continuity at any ODP condition. With the nozzle floating, no error is generated. B5 also uses subroutine NOZCO to calculate the nozzle coefficient and then calculates the gross thrust produced.

The calculations proceed as follows:

1. the sonic velocity for the given nozzle inlet conditions is calculated

2. the thermodynamic properties assuming an isentropic expansion to atmosphere are calculated

3. If a DP cycle, or else ODP with a floating area nozzle, the exit area required to satisfy continuity is calculated.

4. If an ODP cycle with a fixed nozzle, the calculations proceed with the exit area fixed at the DP value or else at a new value specified as an ODP condition. This time an error will be produced if the engine is not balanced since the actual back pressure will be different from that required.

5. The subroutine NOZCO calculates the nozzle coefficient given the pressure ratio PB/PATM and the area ratio, which for a convergent nozzle is 1.

6. With the nozzle coefficient known, the gross thrust produced by the nozzle may be calculated by

$$XG = \left[WA \times (VB) \times (COEF)/g_c\right] + (PSB - PATM) \times AB$$

3.2.2.6 B6, Engine Output Parameters

This is not a separate subroutine, but is in fact contained within the program MASTER segment as it is simple and straightforward. From the EV values of total gross thrust from all nozzles, total momentum drag, total fuel flow, and total air mass entering the engine, the net thrust, specific fuel consumption, and specific thrust of the engine are found. Use of this Brick also causes the complete SV's of the engine to be output.

3.2.2.7 B7, Division/Addition of Mass Flow and/or Total Pressure

This subroutine solves the equations

$$WB = \lambda \times WA + \Delta W$$

$$PB = \lambda \times PA + \Delta P$$

Used with mass flow, it allows for the division of a gas flow into two streams, and hence is used when simulating air bleeds and the bypass of a fan. With regard to total pressures, it may simulate the loss of pressure in a duct or other component. However, pressure loss can better be expressed by using the pressure loss term possible within other bricks such as B3, combustion, or B11, duct or jet pipe.

3.2.2.8 B8, Simple Mixing

This Brick performs mixing calculations where one of the flows is small in relation to the other. The mixing process is simplified because it is then assumed that there is no change in total pressure as a result of the mixing. The only effects are a change in fuel/air ratio and total temperature. This Brick is especially suited for the mixing of cooling bleed flows where the cooling flow is usually a small proportion of the core flow.

3.2.2.9 B9, Arithmetic

This Brick is capable of performing any of the four arithmetic functions (+, −, x, ÷) on any two SV, BD, or EV quantities, with the result being placed in any SV, BD, or EV store.

3.2.2.10 B10, Complete Mixing

B10 performs mixing calculations more complex than B8 since pressure changes as a result of the mixing are accounted for. The calculations are involved and complicated and employ several iterative processes. There are situations where mixing will not occur, and these are to be accounted for.

For instance, of the static pressure of one stream is greater than the total pressure of the other, mixing is impossible. Usually in this case, the pressure ratio of the fan or compressor providing the bypass airflow must be increased to compensate for this. In the DP situation, the program will stop and the user must make changes to the DP pressure ratio. In an ODP case, the program internally selects a new compressor operating point to account for this.

To calculate complete mixing, some factors in addition to the usual two inlet SV elements must be known. Either the Mach number or the static pressure of one inlet stream must be input in order for the calculations to proceed.

In the DP case, if one of the inlet Mach numbers is given, the corresponding inlet static temperature is found by an iterative method. Knowing the static temperature allows the static pressure to be calculated. Since both inlet static pressures must be equal, the other inlet static pressure and temperature may be determined. Since inlet total properties of both streams and their mass flows were already known, and the static properties are known, it is a straightforward matter to calculate the flow velocities, densities, and areas.

The outlet properties are another straightforward calculation. Fuel/air ratio, total temperature, and mass flow are determined by simple continuity and enthalpy balances. The outlet area is assumed to be equal to the sum of the two inlet areas. The outlet velocity is then found since the outlet mass flow is also know. By an iterative process, the outlet static temperature is calculated. Assuming an ideal gas, the static pressure is then determined. This leads to the calculation of the final unknown property, total pressure.

For the ODP calculations, the areas calculated in the DP case are used to calculate the two inlet static pressures and Mach numbers. Unless the engine is balanced, an error will be produced since the two inlet static pressures will not equal (within a given tolerance).

3.2.2.11 B11, Duct or Jet Pipe, With or Without Burning

With this Brick it is possible to simulate a simple duct or jet pipe with a pressure loss or else a duct or pipe, with pressure loss, but also with burning. In either case, pressure loss is calculated by assuming that it is proportional to the kinetic head at the duct or pipe inlet (as in B3).

If there is no burning, exit is made from the subroutine at this point. Otherwise, TB, the final temperature must be specified, as well as the combustion efficiency. Knowing these two factors, the mass of fuel required to produce the temperature, and the outlet SV is calculated as in B3, combustion.

When burning is occuring, the final nozzle area must be allowed to float in order to satisfy the nozzle continuity requirements.

3.2.2.12 B12, Convergent/Divergent Nozzle

This Brick simulates a convergent/divergent nozzle. In the DP case, it calculates the throat area such that sonic flow occurs in the throat. The exit area is calculated assuming an isentropic expansion to the ambient static pressure.

In the ODP situation, the throat and exit areas are fixed and the Brick produces an error similar to that of B5 if the engine is not balanced. Note that the areas do not have to be fixed at the DP values. By use of BD and changing the exit area SV, new areas may be specified in the ODP conditions. Also, there is a facility for "floating" the nozzle such that it can handle the increased non dimensional mass flow during reheat operation. In this situation, the exit area is calculated to given an isentropic expansion as in the DP case.

The output for this type of nozzle is quite comprehensive. The areas, velocities and mach numbers at both throat and exit planes are presented as well as the nozzle coefficient and gross thrust produced. In addition, in an ODP situation, a statement is made if the nozzle is over or underexpanded and the location of the shock (inside or outside the nozzle) is also given.

3.2.2.13 B13, Incrementation of a DP Parameter

This Brick is used in DP studies when it is desired to study the effect of changing one of the component parameters on the engine outputs. Provision is made in the MASTER program segment for incrementing up to five such parameters (which may be either BD or SV quantities). An example of its use would be to study the effects of various compressor pressure ratios and turbine inlet temperatures on the final engine results.

3.2.2.14 B14, Free Turbine

This Brick simulates the action of a free turbine (not connected to a compressor) from which shaft power is extracted. Instead of a separate subroutine for this Brick, B4 (Compressor Turbine) subroutine is internally called, with the compressor work term, COMWK, set equal to zero, and the shaft speed input as BD. From this point, the calculations proceed as in B4.

3.2.3 Additional Subroutines and Functions

In addition to the above Bricks, the following subroutines and functions are used in this program. They are called internally and the user does not need to know their operation.

TRM

This is a function which calculates thermodynamic properties and accounts for varying temperature and fuel/air ratio. The properties calculated are either enthalpy, specific heat, or ϕ where ϕ is defined by:

$$\phi = \frac{1}{R} \int_{T_o}^{T} \frac{C_p}{T} dT, \quad \frac{\Delta S}{R} = \Delta\phi - \ln\left(\frac{P_2}{P_1}\right), \text{ and } \quad R = \frac{\overline{R}}{M}$$

The polynomials and numerical constants required to perform these calculations and the ones described below in AIRTAB have been taken from references 23 and 7. The numerical constants, which are valid for air and Topp's standard kerosene, are stored in the BLOCK DATA segment, COMMON/AIRCONST. These constants are broken down to cover two temperature ranges, below 800°K and 800°K to 2000°K.

AIRTAB

is a subroutine which performs inverse calculations of TRM. That is, for a given enthalpy, specific heat, or ϕ, and fuel/air ratio, the corresponding temperature is calculated by an iterative method.

SEARCH (19)

is a subroutine which is used to obtain values from a component map if two of the map values are known (input). Basically, it is an interpolation routine which obtains intermediate values by a similar triangle approach. If this fails to produce a value the subroutine AFQUIR is called to obtain the value.

AFQUIR

This subroutine name stands for "Air Force Quadratic Interpolation Routine" which was developed by the USAF and used in references 17 and 19. It is used by SEARCH and at other points in the program to interpolate towards a solution; for instance, to interpolate towards the final temperature of combustion if the fuel flow rate is given.

NOZCO (19)

This subroutine contains data which describes a convergent/divergent nozzle map (see section 3.5.4). The subroutine uses this map to obtain a nozzle coefficient as a function of nozzle pressure ratio and area expansion ratio. B5, convergent nozzle, and B12 convergent /divergent nozzle both call this subroutine. In the case of B5, the area ratio is set equal to 1.

3.3 Balancing Technique

As has been mentioned previously, ODP engine cycle calculations require satisfying various matching constraints (turbine and compressor rotational speed and power balance, pressure balances, continuity requirements, etc.) The program internally searches for component operating points which will satisfy all the constraints. The procedure chosen for this program is the Modified Newton-Raphson Iteration Technique as described in references 17 and 19. The Newton-Raphson technique has proven itself to be well suited to solving the ODP problem in that it is easy to apply to the gas turbine engine and results in an efficient convergence.

In order to explain how this iteration technique is applied to the program, the following descriptions of terms are given. Note that "error" and "variable" respectively correspond to the "checks" and "guesses" mentioned in section 2.3.2.

3.3.1 Description of Terms

3.3.1.1 Errors

In an ODP situation before the engine is balanced, the constraints mentioned above will not be met. This leads to certain Bricks producing errors. These errors are the result of:

a. Continuity requirements for compressors and turbines not being satisfied. The mass flow as calculated by the component characteristics does not equal the actual component inlet mass flow.

b. A power unbalance between a compressor and turbine.

c. Difference between the actual pressure and the required pressure for a nozzle inlet.

d. Static pressures at the inlets of a mixing duct not being equal.

The Bricks which generate these errors follow. Note that "ACTUAL" values are those determined from the component inlet conditions while the "MAP" values are those obtained from the component maps.

a. B2, Compressor. Only if the user selects it, the following error of continuity is produced:

1. $$\frac{WA_{ACTUAL} - WA_{MAP}}{WA_{MAP}}$$

b. B4 and B14, Turbine. B4 and B14 both call the same subroutine and produce two errors. The first is one of lack of continuity and the second is the result of a power imbalance.

1. $$\frac{TF_{ACTUAL} - TF_{MAP}}{TF_{MAP}}$$

2. $$\frac{DH_{ACTUAL} - DH_{MAP}}{DH_{MAP}}$$

c. B5, Convergent Nozzle. This Brick produces one error:

1. $$\frac{PA_{REQUIRED} - PA_{ACTUAL}}{PA_{ACTUAL}}$$

d. B10, Mixing. This Brick produces one error if the static pressures of the two inlet streams are not equal

1. $$\frac{PS_{STREAM_2} - PS_{STREAM_1}}{PS_{STREAM_2}}$$

e. B12, Convergent/Divergent Nozzle. This Brick will produce one error:

1. $$\frac{PA_{REQUIRED} - PA_{ACTUAL}}{PA_{REQUIRED}}$$

3.3.1.2 Variables

To employ the Newton Raphson technique, certain BD or SV elements are considered to be independent variables. These variables are the "guesses" mentioned previously and are altered until the engine balances and zero errors are produced (that is, until the errors approach some acceptable tolerance). The variables may be any BD or SV element, but in fact, the variables are usually selected from the following items:

a. Shaft rotational speed.

b. Distance along a constant speed line of a compressor characteristic map. Along with (a), this effectively alters the compressor mass flow, pressure ratio, and efficiency.

c. Turbine inlet temperature. Note that if this is chosen as a variable, then one of the shaft speeds must be specified and may not be a variable.

d. Bypass ratio.

e. Turbine flow function parameter (TF). This effectively alters the operating point of the turbine.

Most other engine parameters will be changed as a result of these changes in the independant variables (as described in (b) and (e)). However, any parameter which is not affected in this manner remains at its value immediately prior to the ODP run.

As the user is free to select what variables and errors are to be employed in a particular run, it is possible to simulate very complex configuration engines and not be restricted to standard ones as has been the case with previous ODP engine simulation programs.

3.3.2 Newton Raphson Technique

In employing the Newton Raphson technique in this program, the sequence of operations is as follows. Initially, the program performs a DP simulation of the engine in order to calculate the component map scaling factors and also calculate the areas of mixing ducts and nozzles. Then, after the ODP conditions are input, a run is made through the engine with all variables at their DP values. Since the engine is not balanced, errors will be produced which are referred to as Base Errors. During successive runs, a slight increment is made to each variable until the number of variables changed equals the number of errors produced per run (only one variable is incremented in each run; the remaining variables are always reset to their base values). Naturally, by slightly changing a variable, some errors will reflect a slight change. In this manner, the change in each error produced by a change in each variable is calculated.

Thus the partial derivative of an error (E) with respect to a variable (V) is calculated. This may be expressed as

$\dfrac{\partial E_i}{\partial V_j}$ where this represents the change in error E_i caused by a change in variable V_j

Since we can assume that the errors may be represented as some function of the variables,

$$E = f(V)$$

then the set of partial differential equations for this function is

$$dE_i = \sum_{j=1}^{jmax} \frac{\partial E_i}{\partial V_j} dV_j$$

for i going from 1 to jmax where jmax is the number of errors produced for a particular engine configuration.

The assumption of a small change in the variable results in the following approximations

$$dE = E - EB \qquad \text{where B represents base value}$$
$$dV = V - VB$$

and $\frac{\partial E}{\partial V} = \frac{\Delta E}{\Delta V}$

If $\frac{\Delta E_i}{\Delta V_j} = \varepsilon_{ij}$,

then with these approximations and the fact that E should equal zero for a balanced engine, the set of partial differential equations reduces to

$$E_i - EB_i = \sum_{j=1}^{jmax} \varepsilon_{ij}\, dV_j$$

After each run through the engine with an incremented variable, values of ε_{ij} (for i going from 1 to jmax) are calculated. After a number of passes equal to jmax + 1 (base value, plus incrementation of each variable) the following set of differential equations is now complete.

$$\begin{array}{l} \varepsilon_{11}\, dV_1 + \varepsilon_{12}\, dV_2 + \ldots\ldots \varepsilon_{1j}\, dV_j = EB_1 \\ \vdots \\ \varepsilon_{j1}\, dV_1 + \varepsilon_{j2}\, dV_2 + \ldots\ldots \varepsilon_{jj}\, dV_j = EB_j \end{array}$$

At this point, the terms of these equations are loaded into the subroutine MATRIX which solves this set of linear equations for dV_j (j = 1 to jmax). The variables are then given the new values,

$$V_j = VB_j + dV_j$$

If the set of differential equations, E = f(V), were linear, the engine would be balanced with the new values of the variables. This is not often the case, however. These new values of the variables become the new base values and the errors they produce are the new base errors. It is then necessary to repeat the whole process of incrementing each variable for each pass.

When all the errors are zero (actually, when all the errors described in section 3.3.1.1 are within some allowable limit, taken as ± .005 for the development of this program), the values of the independant variables are the correct values and all component matching constraints are satisfied.

3.4 Program MASTER Segment

3.4.1 General Description

This Fortran program is divided into four types of segments

1. MASTER segment
2. Functions
3. Subroutine
4. Block Data

It is the MASTER segment that directs the sequence of operation of all other segments. Control is constantly being passed back and forth between the Brick subroutines and the MASTER segment.

In describing this main part of the program, it may be helpful to include now a block diagram showing the major portions of this segment. A complete block diagram and flow chart is given in Appendix B.

Note that for the purposes of this description, the following terms have these definitions:

a. Design Study. By saying that the program is performing a design study, it is meant that no ODP conditions will be input at a later time.

b. Design Conditions. In doing any run, even an ODP study, the first pass through the engine is always made using the design conditions. This allows the component maps to be scaled and duct or nozzle areas to be calculated. After this first pass is completed, then runs are made using the off design conditions, if desired.

3.4.2 Sequence of Operations

In the MASTER segment, much switching is involved in order to give the scheme its flexibility. To try and describe each switch or branch in the logic sequence would take much space and would be difficult to follow. For this reason, only a brief description of the major parts of the segment is given here. By referring to the complete flow diagram in Appendix B, a reader should be able to trace all the logic paths.

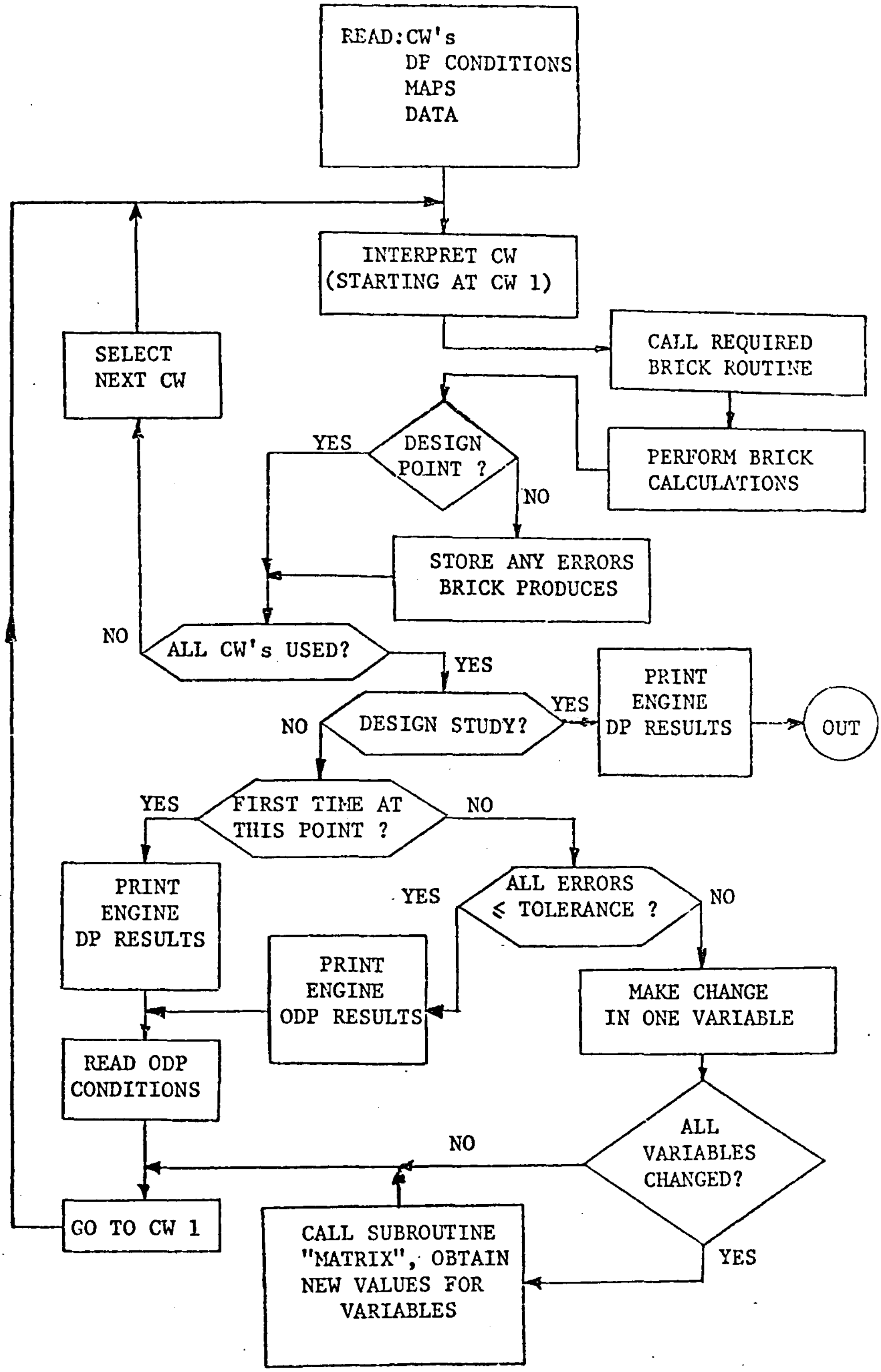


Figure 4 MASTER SEGMENT FLOW DIAGRAM

As in TURBOCODE, a user writes a series of CW's and data, all of which is merely input data for the main program. These CW's provide instructions as to which parts of the program are to be employed and in what sequence. Making reference to figure 4, the following is a brief description of the operations within the program.

a. The particular component maps, CW's, BD, and any required SV elements are read in.

b. Starting with the first CW, the program performs all the operations required by the Brick specified in this CW. All CW's are then operated on in the sequence which they are input. The values for BD and SV elements input are the DP values, even if the study is for an ODP simulation. This is done to generate scaling factors for the component maps and also to calculate the areas for nozzles and mixing ducts, which are later used in the ODP cases.

c. If a DP study, after all CW's have been operated on, the engine parameters and complete SV's are printed out. If the simulation has been designed to use B13, the incrementation Brick, the program returns to the point where this incrementation loop starts. This is repeated until all the incrementation loops have reached their final values and the program itself finishes.

d. If an ODP study, again, after all the CW's have been operated on, the engine parameters, SV's etc. are printed out. Now, at this point, the program reads more data which changes the values of one or more BD or SV elements and thus defines the ODP conditions.

e. The program returns to CW number 1 and repeats. Some Bricks will now produce errors since the engine will not be balanced under the new ODP conditions. The errors produced at this stage are the "Base Errors" and are stored in a one dimensional array, BERR. If all these Base Errors are less than a specified tolerance, a jump is made to step j.

f. Small perturbations are then made in a number of variables equal to the number of errors produced. The program then returns to CW number 1 and repeats, with one variable at its changed value and the other variables kept at the values used in step e. In this way, the errors reflect what effect the change in one variable has on the various errors.

These errors are loaded in a two dimensional array, EROR. The next variable is then set to its new perturbed value, the previous one reset to its original value, and the process repeated until all variables have been changed.

g. At this stage all the terms for the set of simultaneous equations which describe the errors as a function of the variables (as mentioned in section 3.3.2) are known. These terms are now loaded into the subroutine MATRIX which uses a matrix solution to solve the equations and obtain new changes for the variables.

h. With the variables now reset to new values (calculated in g), control passes back to step e. The steps e to h repeat until the errors are within a given tolerance, in which case control passes to step j.

j. Control now passes to a section of the program which prints out the ODP engine parameters, SV's, etc. which are the results from the balanced engine. Now the program reads more ODP data and repeats from step e. In this way it is possible to perform several ODP simulations without having to "reprogram" the engine. It should be noted that there is no real exit point from the program for an ODP study. The program continues to read ODP conditions until it runs out of data, when the computer operating system shuts down the program.

3.5 Component Characteristic Maps

As mentioned previously, component characteristc maps are an integral part of this engine simulation scheme. Two maps are contained within the body of the program. These are the combustion chamber and nozzle coefficient maps. Other component maps must be entered as data by the user. This enables the user to select particular maps and arrange them in the order required. Provision is made for up to three compressor and three turbine maps to be used in the same simulation. There is nothing that dictates that the number of compressors and turbines must be equal (for example, a free turbine engine). Additional numbers of maps could be added by including additional COMMON statements in the MASTER segment and B2 or B4 subroutines, READ statements in the MASTER segment, and increasing the size of the DIMENSION statement for scaling factors in B2 or B4. In additicn, extra calls for the SEARCH subroutine in B2 and B4 would have to be incorporated.

The program as written, is able to deal only with the four types of maps mentioned: compressor, turbine, combustion chamber, and nozzle. If required, it would be possible to alter the program to enable it to deal with afterburner or intake maps. However, this would require the reworking of the Bricks which would use these maps (B11 and B1 respectively).

The maps used to test this program have come from GENENG and SMOTE (17, 20). (They have been altered to conform with the units used in this program). The facility exists for the user to enter his own maps and the rules for inputing these are given in the following sections. One significant difference between the maps used in this program and that of GENENG is one of size. Because of computer core size limitations, maps composed of far fewer points are used here. For example, a GENENG turbine map consists of 690 points whereas a similar map used with this program consists of only 190 points. Such a plot is naturally not as "smooth" as the GENENG map, but in program testing, it has not caused any difficulty. If desired, the number of points in a map could be expanded quite easily merely by changing the appropriate DIMENSION, COMMON, and READ statements mentioned previously. The call statements for the SEARCH subroutine would also have to reflect the larger maps.

3.5.1 Compressor

1. The compressor characteristic maps are standard maps representing pressure ratio, "corrected" mass flow, WAC,

$$WAC = \frac{WA \times \sqrt{TA/T_{SL}}}{PA/P_{SL}}$$ where SL = static sea level ISA values

non dimensional speed as a percent of design non dimensional speed (see below), and isentropic efficiency.

While the maps are general and can be used in many examples by the scaling procedure, they are naturally more suited to particular cases. For example, a low pressure ratio fan map for a high bypass ratio engine of a subsonic aircraft would not properly simulate the high pressure ratio compressure of a low bypass ratio engine for a supersonic aircraft. Five different maps from references 17 and 20 have been used in developing this program. While they represent a broad spectrum of compressor types, they are by no means a complete selection. For this reason, compressor maps are entered as data at the input stage, and the user is free to select a map most closely resembling his application.

If a user wishes to use his own characteristics as data, these are the rules by which it must be prepared:

a. 10 values of "corrected shaft speed", CN, which is defined as

$$CN = \frac{N}{\sqrt{TA}} \div \left(\frac{N}{\sqrt{TA}}\right)_{DESIGN}$$

b. 5 values each of pressure ratio (PR), "corrected" airflow, and isentropic efficiency corresponding to one value of CN

c. All numerical data to be separated by at least one space (free format is used in reading the component maps)

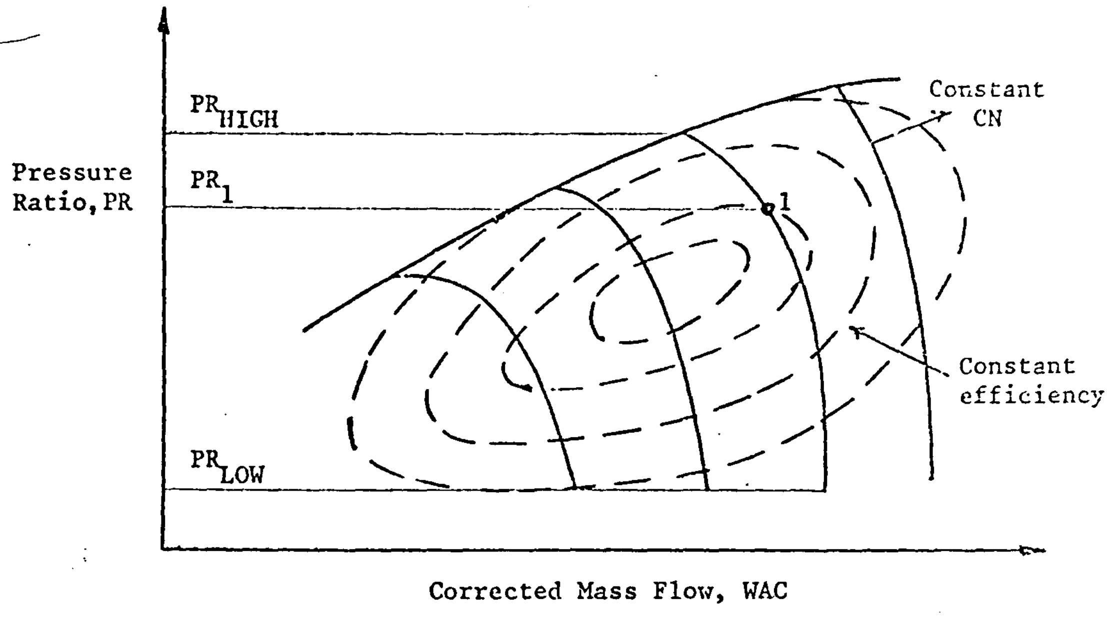


Figure 5 Example of Compressor Characteristic Map

2. Search Method

A term, Z, is defined as the ratio of pressure ratios along a constant speed line. Referring to figure 5, for point 1,

$$Z = \frac{PR_1 - PR_{LOW}}{PR_{HIGH} - PR_{LOW}}$$

(Note that this definition of Z also is often referred to as β by certain engine manufacturers.)

In effect, Z is a measure of the proximity to the surge line. As Z → 1.0, the compressor is approaching surge. (Internally, the program prevents Z from exceeding .999) In ODP studies, Z is frequently used as a variable. When entering a map, CN and Z are known and the ocrresponding PR, corrected airflow and efficiency are determined.

3. Scaling Equations

The equations used to obtain the scaling factors for the compressor are:

a. Pressure Ratio: $PRSF = \dfrac{PR_D - 1}{PR_{MAP,D} - 1}$

where the subscript MAP,D is that value obtained by entering the map with design values of CN and Z.

Subscript D is the design value specified as BD.

b. Corrected Mass Flow:

$$WASF = WAC_{ACTUAL}/WAC_{MAP,D}$$

c. Efficiency:

$$ETASF = \eta_D/\eta_{MAP,D}$$

3.5.2 Turbine

1. The turbine characteristic maps are of the form

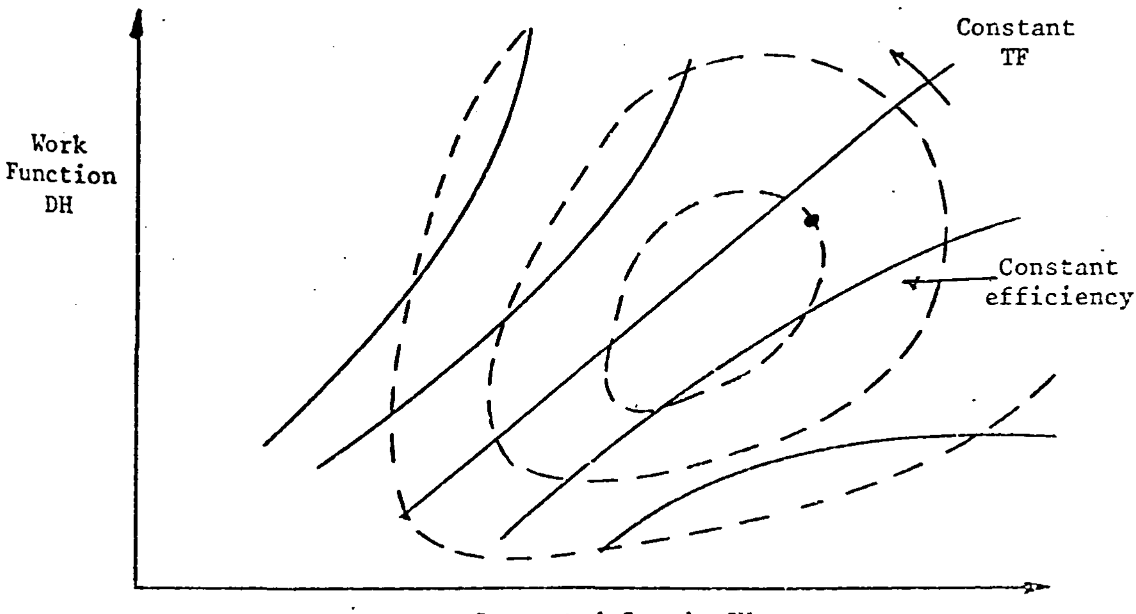


Figure 6 Example of Turbine Characteristic Map

where: TF, the turbine flow function (sometimes called the "mass flow parameter" or "swallowing coefficient") is defined as:

$$TF = WA \times \sqrt{TA}/PA$$

work function, $DH = (HA - HB)/TA$

η = isentropic efficiency

and $CN = PCN/\sqrt{TA}$

where PCN = shaft speed as a % of design speed. For a free turbine, this value is input as BD. For a turbine driving a compressor, PCN is obtained automatically when the user specifies in the BD the compressor to which it is attached.

As with compressor maps, these maps are entered as data during the input sequence. If a user wishes to enter his own turbine characteristic maps, the following rules govern the preparation of the data:

a. 10 values of TF

b. 6 values each of CN, DH, and efficiency corresponding to one value of TF

c. all numerical data to be separated by at least one space.

2. Scaling Equations

The scaling factors are defined as:

a. Turbine Flow Function:

$$TFSF = \frac{TF_D}{TF_{ACTUAL}}$$

b. Shaft Speed

$$CNSF = \frac{(CN)_D \times \sqrt{TA}}{PCN}$$

c. Work Function

$$DHSF = \frac{(DH)_{\substack{\text{TO DRIVE COMPRESSOR AT DESIGN} \\ \text{OR INPUT AS BD}}} \text{ (FREE TURBINE)}}{(DH)_{MAP,D}}$$

d. Efficiency

$$ETASF = \frac{(\eta)_D}{(\eta)_{MAP,D}}$$

3. Search Method

The search technique for the turbine maps is slightly different than that for compressors. TF and CN are entered as DP BD. They merely dictate where on the map the design point is to be located. Initially on entry to the turbine subroutine, TF' and CN' are calculated from the component inlet conditions

$$TF' = TF_{ACTUAL} = WA \times \sqrt{TA}/PA$$

$$CN' = PCN/\sqrt{TA}$$

The scaling factors, TFSF and CNSF, are determined as shown previously. Then, to enter the map TF and CN are calculated by

$$TF = TF' \times (TFSF)$$

$$CN = CN' \times (CNSF)$$

Entering the map with TF and CN, the interpolation routines (SEARCH, which calls AFQUIR) find the corresponding values of DH and η.

3.5.3 Combustion Chamber

The combustion chamber map is a plot of combustion efficiency versus temperature rise for constant input pressure. Entry to the map is through temperature rise and input pressure with efficiency being the output.

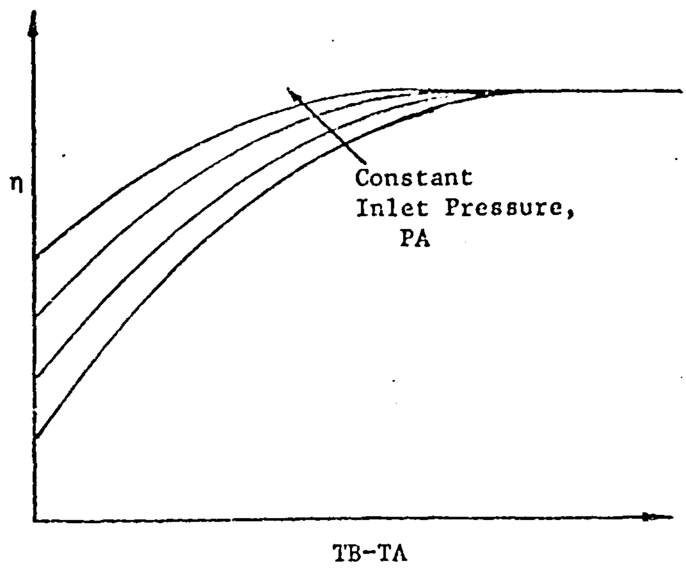


Figure 7 Combustion Chamber Map

If desired, the combustion chamber map may be replaced by a map corresponding to a known combustor. In this case, it is entered as BLOCK DATA in the segment COMMON/CUMB and the following parameters must be given:

a. input pressure, atmospheres, 10 values

b. temperature rise, ^{o}K, 8 values

c. combustion efficiency, 80 values, each corresponding to an input pressure and temperature rise.

note: combustion efficiency is defined as:

$$\eta = \frac{\text{ideal amount of fuel burned}}{\text{actual amount of fuel burned}}$$

3.5.4 Nozzle Coefficients

As in GENENG, this program uses a convergent/divergent nozzle velocity coefficient which is located in map form in the subroutine NOZCO. The map is of the form

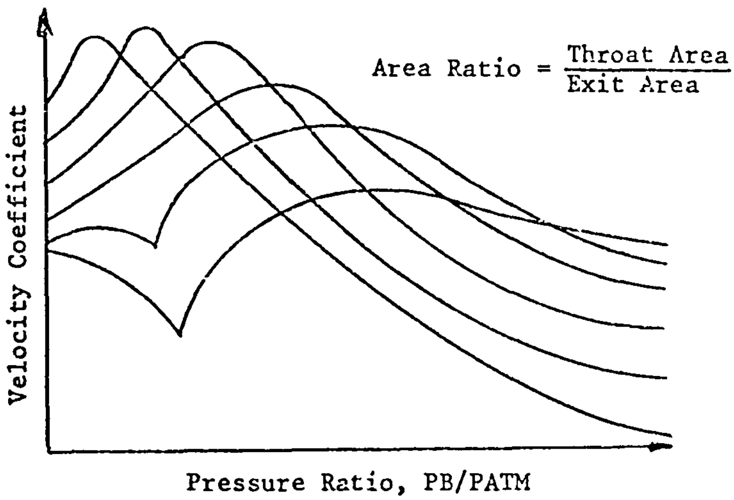


Figure 8 Nozzle Velocity Coefficient Map

The nozzle coefficient is given as a function of total pressure ratio and area ratio. Convergent nozzles may use the map with an area ratio equal to 1.

3.6 Input Procedure

As stated previously a user writes a "program" which controls the sequence of operation of the MASTER segment. In fact, his "program" is nothing more than numerical data for the main program. With the scheme in operation at Cranfield, cards are used as the input medium. By slight reprogramming (altering READ statements), any other input medium could be utilized.

It may be seen that the input procedure resembles the earlier versions of TURBOCODE. However, it is more involved than TURBOCODE, largely because more specification of variables, errors, new conditions, etc. are necessary.

Note that in the input instruction, reference is made to "integer" and "real" numbers. These follow the Fortran definition in that an "integer" number is one written without a decimal point while a "real" number must contain a decimal point.

3.6.1 Initial Cards

The initial data is input on three cards.

First Card Contains three integers separated by at least one space

First number = +1 if DP study
= -1 if ODP study

Second number = number of errors to be used in the program (meaningful only for ODP study)
= 0 if DP study

Third number Selects system of units to be used
= +1 for British units
= +2 for SI units

Second Card Contains three integers separated by at least one space. These numbers represent the number of CW's, BD, and SV elements which will be input

First number = number of CW's to be input

Second number = number of BD to be input as data

Third number = number of SV elements to be input as data

Third Card — This card contains two integers separated by at least one space. These numbers represent the number of compressor and turbine maps that will be input.

First number = number of compressor maps to be input

Second number = number of turbine maps to be input

3.6.2 Component Maps

In this location are placed the card decks which contain the compressor and turbine component characteristics maps. The compressor maps are read first and are automatically numbered in the order which the CW's operate on them. Turbine maps, which are entered next, are similarly automatically numbered.

3.6.3 Codeword Cards

Since these cards control the entire engine simulation, it is vital that they be completed correctly. The numbers must be entered in the exact columns as specified. Each card contains one codeword. Any items which do not require information for a particular Brick must be filled in with "-1". The only exception to this is if the remaining columns of a card would contain only "-1", then they may be omitted.

(1) Columns 1 - 3, Brick number of CW in three figures

e.g. 007, 014, etc.

(2) Columns 4 - 9, SV's on which the Brick operates

column 4 - 5, inlet SV
column 6 - 7, outlet SV
column 8 - 9, extra SV if required

(3) Columns 10 - 13, BD addresses

column 10 - 11, start of BD address list used by Brick

column 12 - 13, end of BD address list used by Brick

The BD for a Brick must come from consecutive numbers in the address list. For example, B4 requires five items of BD. If in a "program" the next unused BD address is number 10, the BD of B4 would occupy BD addresses 10, 11, 12, 13, 14 and columns 10 - 13 would be: 1014

(4) Columns 14 - 19, EV addresses required

column 14 - 15, first EV address required

column 16 - 17, second EV address required

column 18 - 19, third EV address required

The remaining columns are used only if one of the BD or SV elements is to be a variable in an ODP study. In all other situations, the remainder of the card may be left blank.

(5) Columns 20 - 23, Variable address list

column 20 - 21, first variable address

column 22 - 23, second variable address (there is a maximum of two variables for each CW)

(6) Columns 24 - 29, specifications for first variable

column 24 - 25 = 01 if variable is a SV quantity

= 02 if variable is a BD quantity

column 26 - 27 = SV number if variable is a SV quantity

= BD number if variable is a BD quantity

column 28 - 29 = SV element number if variable is a SV quantity

= -1 if variable is a BD quantity

(7) Columns 30 - 35, specifications for second variable. Entered in same format as first variable (above).

3.6.4 Data Input Cards

The next series of cards contains BD and SV elements which are required by the Bricks specified in the codewords.

Brick Data

These cards contain two numbers separated by a space. One card is required for each BD item.

First number, integer = BD number in address list
Second number, real = value of BD number.

SV Input Data

These cards contain three numbers all separated by one space. One card is required for each SV item being input.

First number, integer = SV number
Second number, integer = SV element number (see below)
Third number, real = value of SV element

SV element numbers:

1 = fuel/air mass ratio	5 = static temperature
2 = gas mass flow	6 = total temperature
3 = static pressure	7 = velocity
4 = total pressure	8 = area

3.6.5 ODP Condition Cards

The remaining cards are only used in ODP studies. Their purpose is to cause a change to be made in the previous data and hence define the ODP conditions. These values, once changed, will remain at their new values until changed again in a similar manner. (Note that it is meaningless to specify one of the variables of a CW as an ODP condition. This is the only restriction on the choice of a changed value).

a. First Card Contains an integer equal to the number of changes to be made to the data. As the program is presently written, there is a limit of five changes at one time. This could be overcome by expanding a DIMENSION statement in the MASTER segment.

b. Remaining Cards These cards are "mini" codewords and specify what item of input data is to be changed, and what its new value will be. One card is required for each "mini" codeword.

columns 1 - 2 = 01 if change is a SV quantity
= 02 if change is a BD quantity

columns 3 - 4 = SV number if change is a SV quantity
= BD number if change is a BD quantity

columns 5 - 6 = SV element number if change is a SV quantity

= -1 if change is a BD quantity

column 7 = left blank

column 8 onwards = any real number representing the changed value described above.

Steps a and b may be repeated any number of times as the program will continue to read new ODP conditions until all data has been used.

3.6.6 Final Note on Program Inputs

As these inputs control the whole sequence of events of the program, great care should be used in forming the data and in the punching of cards. If the input data is incorrect, either the program will stop running while it is attempting to read data, will stop running during the execution of the program, or worse, will continue to run, but results will be incorrect. While safeguards are built into the scheme (see section 3.8.3 on Error Tracing), naturally it is impossible to have planned for every possible user "programming" error.

Also, because of the Fortran compiler, it is necessary to follow the rules as to the decimal point (or lack of it) in the data. The Fortran compiler on the ICL 1900 series computer accepts data in "free format". Free format numerical data need not have the field width or number of decimal places in real numbers specified. Instead, all data is separated by a space, and the computer continues to read data until sufficient numbers have been input. As may be inferred by the above input instructions, this program uses free format frequently. If the scheme is to be run on another computer without this free format facility, the input procedures and the format statements in the MASTER segment would have to be altered.

3.7 Program Outputs - Presentation of Results

A comprehensive picture of the engine operating point is made each time a DP engine cycle is simulated and each time an ODP cycle converges on a balanced engine. In addition, after all the data cards have been read in, the input "program" consisting of this data and the definitions of the system of units to be used in the simulation are also printed.

The engine output consists of three different sections. These are:

1. component description
2. printing of complete SV's of the engine
3. overall engine outputs

3.7.1 Component Description Output

The parameters of each major component are printed by the Brick concerned every time any output is produced (a DP cycle or balanced engine for ODP conditions). If the component deals with maps and scaling factors however, these are only printed in the run at DP conditions. The various component outputs are as follows:

a. Ambient and Inlet Conditons

1) Altitude (ALT)
2) ISA Temperature deviation (ISA DEV)
3) Flight Mach number (MACH NO)
4) Ram recovery ratio (ETAR)

b. Compressors

DP only; scaling factors
1) PRSF
2) ETASF
3) WASF

DP and ODP;
1) Z factor
2) Pressure ratio (PR)
3) Percent Speed (PCN)
4) CN
5) Isentropic efficiency (ETA)
6) Input power required (COMWK)

c. Combustion Chamber

DP only; scaling factor, ETASF

DP and ODP
1) Combustion efficiency (ETA)
2) Total pressure drop (DLP)
3) Fuel burned (WFB)

d. Turbines

DP only; scaling factors,
1) CNSF
2) ETASF
3) TFSF
4) DHSF

DP and ODP;

1) Turbine flow function (TF)
2) Isentropic efficiency (ETA)
3) CN
4) Auxilary Power produced (AUXWK)

e. <u>Free Turbines</u>

As above in (d), plus the following additional terms

1) Percent Speed SPEED)
2) Shaft power produced (POWER)
3) Shaft power specific fuel consumption (SFC)

f. <u>Duct/After Burning</u>

1) Combustion efficiency (ETA)
2) Total pressure drop (DLP)
3) Fuel burned (WFB)

g. <u>Convergent Nozzle</u>

1) Exit Area
2) Exit velocity
3) Gross thrust produced
4) Nozzle coefficient

h. <u>Convergent/Divergent Nozzle</u>

1) Exit area
2) Exit velocity
3) Throat area
4) Throat velocity
5) Throat mach number
6) Gross thrust produced
7) Nozzle coefficient
8) For an ODP nozzle, the location of the shock is given, as well as a statement if it is over or under expanded.

3.7.2 <u>SV Output</u>

Each time output is produced, a complete listing of all SV's is made. Recall that any SV with a value of -1. represents a quantity which has not been calculated.

The following information is printed:

Station vector number with the corresponding:

1) fuel/air mass ratio
2) gas mass flow
3) static pressure
4) total pressure

5) static temperature
6) total temperature
7) gas velocity
8) flow area

3.7.3 Overall Engine Parameters

Finally, the following parameters which are really the desired outputs of the engine are printed:

1) Total gross thrust of all nozzles.
2) Total momentum drag of all inlets
4) Net thrust
4) Total fuel burned
5) Specific fuel consumption
6) Specific thrust

Note that the specific fuel consumption only considers the net thrust of the engine. Of the engine is a turbojet or fan, this term is satisfactory.

However, if the engine employs a free turbine for shaft output or to drive a propellor, this SFC will only represent that of the net thrust produced. To arrive at the true SFC for a turboprop, a manual calculation will be required.

Examples of the program outputs may be seen in the appendices (Appendix C).

3.8 Additional Program Information

3.8.1 Program Running at the Cranfield Institute of Technology

The computer used to develop this program was an ICL 1905 with 32K words of 2 microsecond store; actual program size was restricted to 24.7K. This size limitation has led to the restrictions, mentioned in this thesis, on the number of compressors and turbines, the number of points in component maps, and to the limits on the following quantities:

maximum number of codewords	= 45
maximum number of Brick Data	= 90
maximum number of Station Vectors	= 25
maximum number of Errors	= 10
maximum number of Engine Vectors	= 20

With these restrictions, the program was capable of running on the ICL 1905. However, at the time of writing (July 1974), a new ICL 1903 computer with 64K of store has just been installed at Cranfield. With this larger core, the above limits no longer need apply. Information has been given throughout this thesis which would enable the program to be enlarged. Probably space would now permit the incorporation of a routine such that any hydrocarbon could be used as a fuel merely by stating the hydrogen/carbon ratio. Reference 26 provides the necessary information.

Running times using a binary dump of the program on magnetic tape are not excessive. With the ICL 1905, it was possible to simulate a standard configuration two spool turbojet and perform calculations for thirteen ODP conditons using 4 min, 15 sec. of machine time. Early results with the ICL 1903 indicate that times will be reduced by approximately two-thirds.

3.8.2 Program Testing

The program has been evaluated on a wide variety of different engine types, ranging from the simplest one spool turbojet, to two shaft mixing fan engines, to completely variable cycle engines. Examples of these simulations are to be found in Appendix C.

To check for accuracy, results have been compared with those of the examples given in references 19 and 20. In most cases, the answers and engine parameters have agreed within 1%. One reason for any discrepancies can be attributed to the different accuracies in the component maps used. As mentioned in section 3.5, the GENENG maps consist of many more points than the maps used in this program.

Again, to acheck for accuracy, a simulation was made to compare results against experimental values of an engine undergoing test. The experimental values were obtained from reference 16 in which the National Research Council of Canada compared the experimental values against the predicted values of one of their ODP programs. The engine simulated was a two shaft turbojet (J-75), and the effects of reducing LP shaft speed (ie: part throttle operation), amount of HP compressor bleed off air, and nozzle area change on the engine parameters were studied.

The complete simulation is described in Appendix D. By and large, the Cranfield program gave a reasonably accurate simulation of the engine. The reduced power simulation gave very good agreement, better in fact than the NRC simulation. The effects of bleed air and nozzle area change were not as accurately represented as the part power simulation, but again, the Cranfield predictions were more accurate than those of NRC.

One possible reason for discrepancies would be the use of component maps which, while giving good overall approximations, may not have rigorously reflected the actual components. One other reason for a discrepancy could be the choice of the DP operating point on the component maps. The choice of Z for compressors and TF and CN for turbines determine where on the component map the DP is. In other words, it is an arbitrary selection which can effect component efficiencies and the errors produced. Possibly a better choice of the DP could have reduced discrepancies. Attention is drawn to Appendix D for a complete description of this test.

3.8.3 Error Tracing

In all the work done with this program so far, there has not been any case where a simulation failed due to a fault of the program. Every fault so far can be traced back to an improper set of data put in by the user (in effect, a "programming" error by the user).

If there is an error in the data, it will manifest itself in one of two ways:

1. The first way is an execution error found by the computer while executing the program which will bring an immediate end to the run. The computer will describe the error and will state approximately where it occurs. This should aid in tracing which CW has been entered with an error inherent in it. Usually this will be found to be an incorrectly formed CW.

2. The second way is an error discovered by mechanisms within the program. Certain occurrences will cause an engine run to cease, although complete exit will not take place. This is useful since it allows further ODP runs to be made which might be unaffected by the previous error. These types of errors can be caused by:

   In a ODP situation,

   a. No convergence within a specified number of loops. Possibly the ODP condition is too large a step from the previous conditon.

   b. Component operating point off the map, or SEARCH unable to find a point on the map.

   c. No convergent on temperature of combustion, given the amount of fuel burned, within a specified number of loops.

   d. Nozzle exit velocity is imaginary. The engine is obviously impossible to balance under this set of ODP conditions.

In each of these cases, a diagnostic print out of the error is made, the particular run is halted, the last complete set of SV's is printed, and control then passes to the next ODP condition.

3.8.4 Program Operation and Techniques

1. Choosing Variables

In ODP studies, it is necessary to choose as variables, those parameters which will affect positions on the component characteristic maps. For this reason, it is suggested that the following always be chosen as variables:

a. for compressors, B and PCN

b. combustion chamber, exit total temperature

Note: it is usually necessary to specify one of the shaft speeds or combustion final temperature (or fuel burned) in defining an ODP condition. Therefore all speeds and combustion temperature may not be variables simultaneously.

c. turbines, flow function, TF

d. Others. If the flow is being divided such as in a bypass engine, the value of $\lambda(W)$ used in B7 to divide the flow is required to be a variable.

2. Choosing Errors

In the more complicated engine studies, usually more variables are required than there are errors being produced. Since the number of errors must equal the number of variables, the facility is present in B2 (compressor) which generates an error only if required. This is done by a certain input of BD for B2. The error produced is one of continuity (inlet mass flow different from mass flow required by characteristic map). In this way, each compressor can give an error if necessary.

In practice, if not every compressor is required to give an error, it is best to start at the HP compressor when specifying the errors. In this way, the error so formed reflects the effects of all the variables preceeding it and so speeds up convergence. Note that if this error is not specified, it does not mean that the engine is improperly balanced since the mass flows usually "come out in the wash" after several iterations.

In program planning, it is desirable to keep the number of variables and errors to a minimum since this directly affects the number of iterations the program must make before the engine is balanced.

3. Nozzle Floating and Duct/Afterburning

Both nozzle Bricks, B5 (convergent) and B12 (convergent/divergent) have the facility to operate with either fixed or variable geometry. The choice is made by an item of BD. When using this program, it is advisable to do ODP calculations with the nozzle in the fixed position as this produces an error while the floating nozzle does not.

If afterburning (or duct burning) is desired, a run should be made through the engine at the ODP, but with no after-burning taking place (this is controlled by an input in BD for the afterburning Brick, B11). Doing this, all components are matched and the inlet mass flow of the engine is set. After this matching, a run is made with burning taking place and the nozzle area floating. The nozzle floats to satisfy continuity and so the nozzle is able to take the increased non dimensional mass flow caused by the burning. In this way, the engine remains balanced.

4. Mass Flow Inputs

Somewhere in an engine simulation, air mass flow must be input. Usually this occurs at the engine inlet, SV number one, by using the process of SV data input. However, in ODP programs, the mass flow through the engine will be varying. This occurs even though mass flow is not defined as a variable since altering the compressor operating point effectively alters the mass flow through the component. (The component inlet mass flow could be a specified variable, but this just is not necessary).

Nevertheless, there must be a means of altering the inlet mass flow to account for the new mass flow that the compressor demands. One way this is accomplished is by a device built into Brick B1. In this Brick, the inlet mass flow is automatically set equal to its outlet mass flow if the outlet mass flow is greater than 0. This works fine if compressor Brick, B2, is directly behind B1 (ie: outlet SV for B1 is inlet SV for B2). As the iterations of an ODP study proceed, the engine inlet mass flow is actually one iteration behind the compressor mass flow. This does not cause any problems, since as the errors approach zero, this discrepancy becomes irrelevant.

If B2 does not follow B1, the following approach is suggested. B9 (arithmetic Brick) is used to transfer the mass flow value of B2 (or indeed, any downstream Brick which is affecting the mass flow) back to the inlet. Note that this is merely the mass flow value and not the actual gas flow itself. An example of this type of calculation is given in Appendix E.

It is important to remember these facts when using B1 and inputting mass. This is because B1 also calculates momentum drag from the inlet mass flow. Thus, any error in its useage would affect the net thrust produced.

5. "Boosted" Compressor

By a "boosted" compressor, it is meant to be the situation whereby a fan runs on the same shaft as a compressor, or, the situation whereby some of the mass flow from the first few stages of compressor is used as bypass flow.

Such a component may be simulated by considering it to be composed of two separate compressors running on the same shaft. Since the shaft speeds must be equal, the value of PCN is carried from the first compressor to the second by the use of B9. In order for one turbine to drive the two compressors, their values of COMWK must be added together (B9) before entering the turbine Brick, B4. For speed matching purposes, the turbine may be assumed to be attached to either compressor.

An example of a "boosted" compressor is found in Appendix E.

6. Free Turbine

By the use of B14, a free turbine from which shaft power is output may be simulated. A component map of the same format as that of the compressor turbine, B4, is used. In an ODP study it is necessary to specify as a variable either the shaft rotational speed as a percent of design (PCN) or the output power produced. The quantity which is not a variable must be specified as one of the ODP conditions. Thus, in an ODP situation, at a specific gas generator condition, it is possible to calculate the output power at a particular free turbine speed, or, the free turbine speed at a required power output.

7. ODP Conditions

Any BD or SV item may be selected as an ODP condition. The exception is that it may not be a variable or an item which is found during the search of a component map (e.g. PR for compressor, efficiencies, etc). Some commonly employed ODP conditions are:

a. charge in flight condition
b. change in shaft rotational speed
c. change in turbine entry temperature
d. change in nozzle area
e. change in bleed flows.

SECTION 4

Variable Cycle Engines

4.1 Introduction

As aircraft and other aerospace vehicles begin to operate at extremes of the speed and altitude spectrum, it quickly becomes apparent that a conventional single powerplant capable of operation under all flight conditions is no longer possible, or at least feasible in terms of economy or noise generated.

Figure 9, taken from reference 27, shows typical operating limits for vehicles operating below 150,000 ft. The upper limit is a lift limit and the lower line a structural or temperature limit.

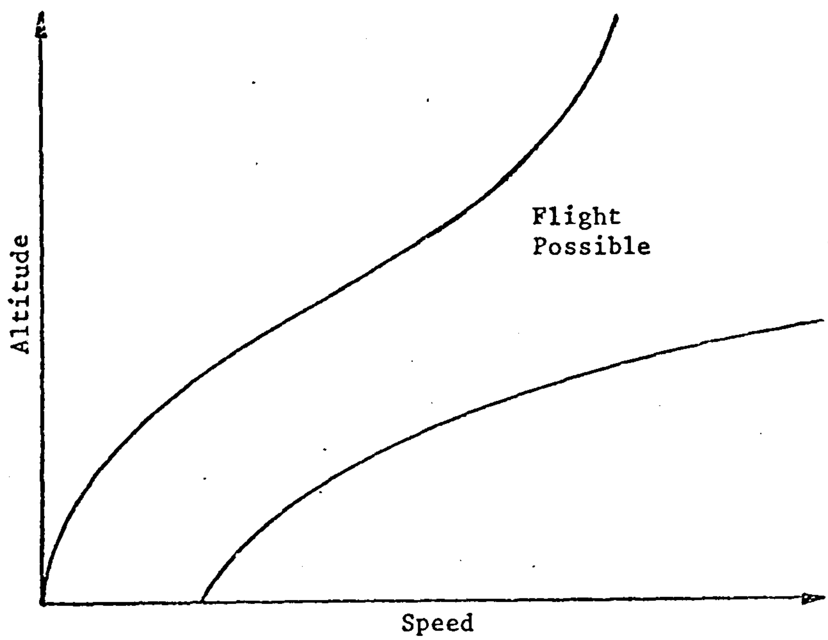


Figure 9 Vehicle Operating Corridor

Figure 10, again from reference 27, represents a typical airbreathing engine corridor (note that rockets have no speed or altitude restrictions. However, their very high specific fuel consumption at the lower end of the flight spectrum makes them unsuited for aircraft application). In figure 10, the upper limit is generally a combustion stability or ignition limit. It may or may not coincide with the vehicle upper limit. The lower limit is generally associated with structural or temperature limits similar to those imposed on the airframe.

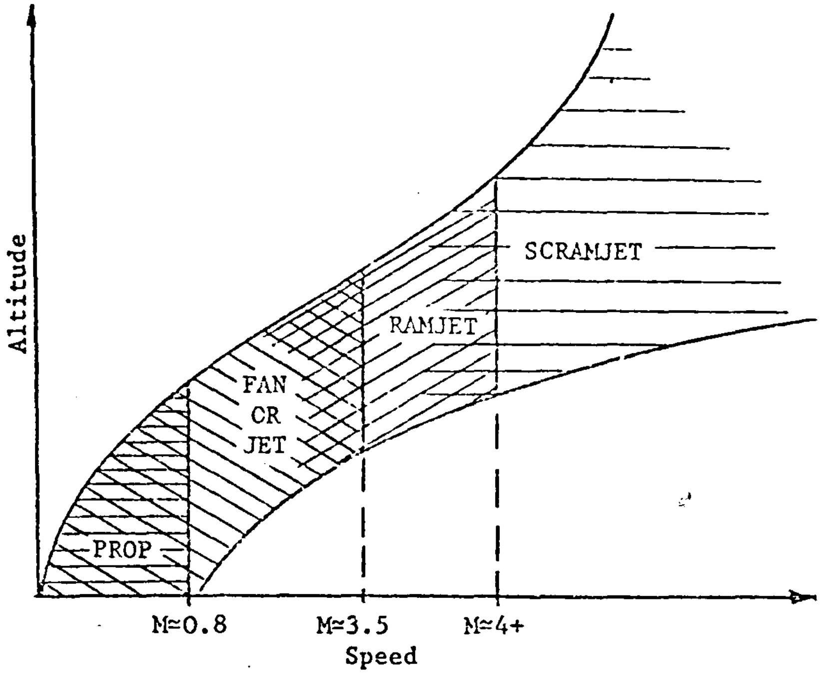


Figure 10 Airbreathing Engine Operating Corridor

As flight speed increases, the technology trend is toward engines that operate in two or more modes. To provide thrust over a wide speed and altitude range, no one system is optimum or even necessarily usable. It is required therefore, to combine the best of each cycle into an optimum propulsion system. The simplest solution is to fit different sets of engines to operate at different times. However, this solution has the obvious disadvantages of high installed weight, bulk, and associated drag.

4.1.1 Mixed and Variable Cycles

Mixed cycle engines are now being developed to combine turbojet and ramjet engines. Such an engine provides the static thrust a ramjet lacks and increases the speed capability of the turbojet. It is designed so that above a certain speed (around Mach 3), the rotating machinery stops, the airflow to the compressor is shut off, and burning occurs in a bypass duct. Other possible mixed cycle engines are ejector ramjets (combination of rocket + ramjet) and supercharged ejector ramjets (turbofan + rocket + ramjet) (27).

This leads to a definition of Mixed Cycles which reference 27 describes as:

"...single integrated engine concepts employing two or more distinct [thermodynamic] propulsion cycles. Each element is capable of operating independently or jointly."

Variable Cycle engines on the other hand are considered to contain only one distinct thermodynamic propulsion cycle and in this way are different from mixed cycles. However, like mixed cycles, they are capable of operation in more than one mode.

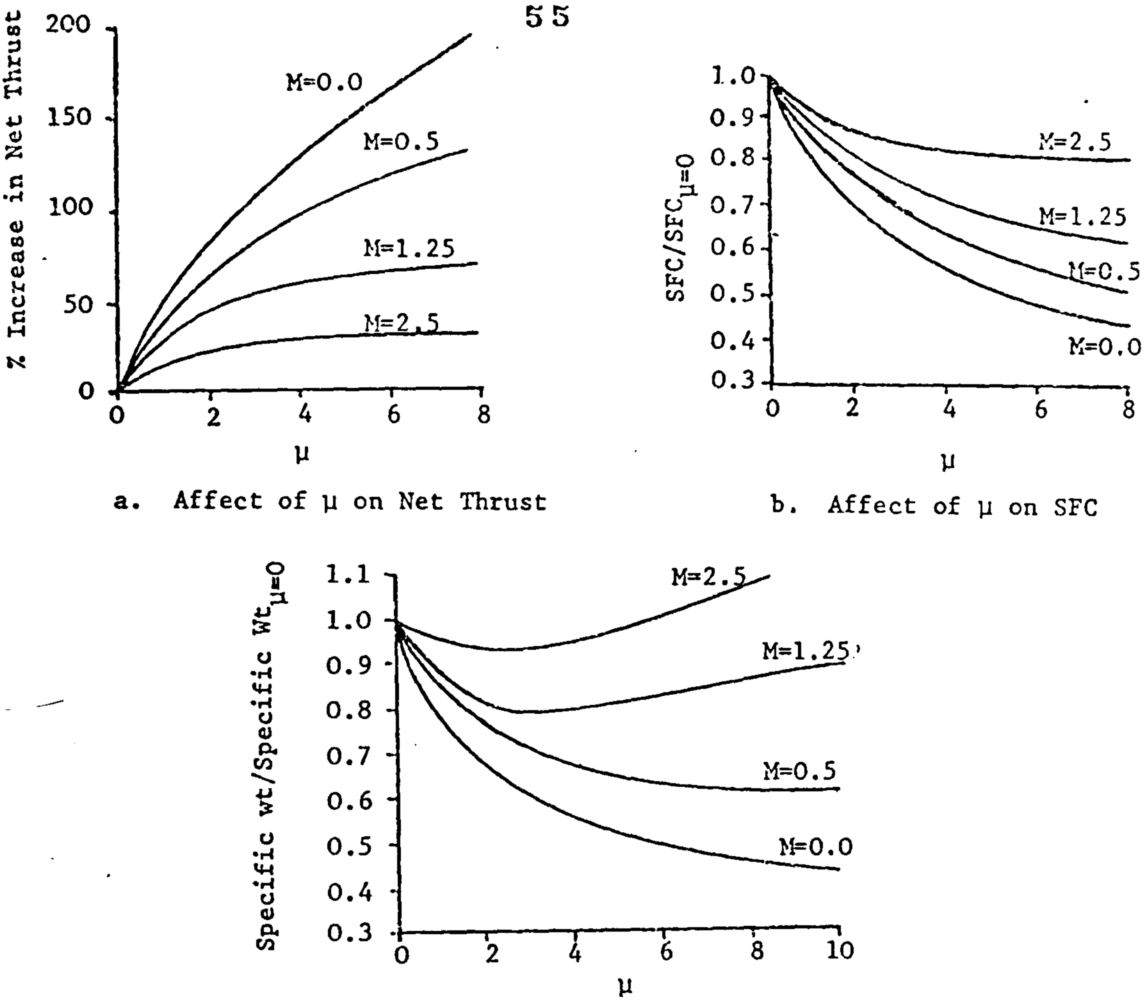
4.2 Requirement for Variable Cycles

Just as a mixed cycle propulsion system is desirable for an aircraft operating in the upper speed regions, variable cycle engines would be advantageous for an application with cruise speeds in the region of Mach 2.0 → 3.0. The principal reason for this is to improve the "total range or flexibility of a given multimission aircraft, while reducing weight and cost, to perform a prescribed mission" (28). Secondary benefits of variable cycle engines are the possible noise reduction during take off and landing, and center of lift control for V/STOL aircraft.

4.2.1 Engine Design for a Single Mission Aircraft

The engine designer has three basic engine parameters to chose from when selecting an engine for a particular application. These are bypass ratio (μ), cycle pressure ratio (CPR), and turbine entry temperature (TET).

Choice of Bypass Ratio. Figure 11a,b, and c from reference 1 compares the performance of an engine with $\mu = 0$ to an identical engine (same engine core flow and TET) with a bypass ratio, μ. These figures demonstrate clearly the effect that the choice of μ will have on an engine's performance. At low speeds, a high μ is advantageous, but as flight Mach numbers increase, a low or zero μ is necessary for good fuel economy. Not shown in these diagrams is the affect of μ on nacelle or installed drag. This would further show the advantages of low μ at higher speeds.



a. Affect of μ on Net Thrust

b. Affect of μ on SFC

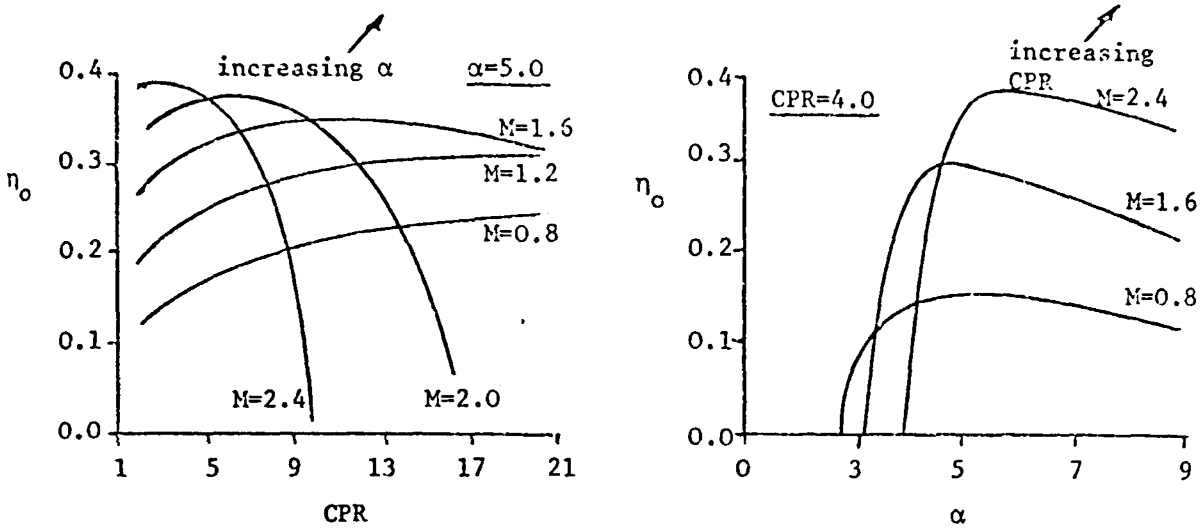
c. Affect of μ on Engine Specific Weight

Figure 11 Affect of Bypass Ratio on Various Engine Parameters

<u>Choice of TET and CPR.</u> With increasing cruise speeds, the engine designer is more and more restricted to his choice of TET and CPR. Figure 12, from reference 29, demonstrates this. Note that α, the temperature ratio, is defined as

$$\alpha = \frac{\text{total TET}}{\text{static ambient temperature}}$$

In the typical supersonic transport cruise altitude (above 36089 ft.), because of material limitations, α would currently be restricted to slightly less than 7. Materials predicted for future engines would bring this up to about 7.5.



a. Overall efficiency as a function of cycle pressure ratio for different flight Mach No's.

b. Over-all efficiency as a function of cycle temperature ratio for different flight Mach No's.

Figure 12 Over-all Efficiency as a Function of Cycle Pressure and Temperature Ratios for Different Flight Mach Numbers

The above figures show that if an engine is designed to be operated at high supersonic speeds (Mach 2.5-3.0), since TET has an upper limit, the choice of CPR to operate at best efficiency becomes rather limited, especially compared to the possible combinations that exist at the lower end of the speed spectrum where the curves are relatively flat.

With an engine designed for a single mission aircraft, selection of the proper values for the parameters will ensure the optimum performance of the engine at the design point. Traditionally, engines have been designed to operate well at one point on the aircraft performance curve. This approach, however, leads to sub-optimum performance at all other points on the curve. Nevertheless, in the past, the performance penalty for off design point operation has just been accepted as a matter of course.

However, the designer of the engines for <u>all</u> aircraft (both civil and military) designed to cruise at supersonic speeds must consider the following factors:

a) Take off thrust requirements. Frequently thrust augmentation will be required at take off if the engine is point designed for supersonic cruise.

b. Possible prolonged subsonic operation for the following reasons:

- -avoidance of sonic booms
- -possible in-flight refueling
- -holding patterns

Also, all fuel reserves are usually calculated assuming a subsonic hold or diversion.

c. For civil supersonic transports, the noise levels at take off and landing are usually a critical factor in the engine design

It quickly becomes apparent that these aircraft can no longer be considered single mission aircraft and they are in reality, multi mission. The compromise in performance previously necessary for single point designed engines is no longer acceptable. Reference 30 states that "a supersonic transport that cruises at M = 2.7 or a M = 3 supersonic bomber will use 30% to 40% of its fuel just reaching cruise speed if it is a point designed aircraft [and engine] for a given cruise speed and altitude."

4.2.2 Multi Mission Design

As was stated previously, the object of multi mission design is to improve the payload/range of the aircraft.

Design Problems. The most obvious way of increasing the payload/range of a supersonic cruise aircraft is not to obtain better SFC at the cruise condition, but rather to improve its subsonic performance. Not only would better SFC during subsonic segments of the flight profile aid in doing this, but rather because of improved subsonic SFC, the required fuel reserves would be considerably less.

However, the means of obtaining good subsonic performance when the engine is designed for supersonic cruise is the problem.

Subsonic Operation. Generally, lower thrust levels are required during subsonic operation. These lower levels can be achieved in any number of ways, the problem then is to operate subsonically most efficiently.

Simply reducing TET automatically results in lower thrust levels. However, referring to figure 12b, it will be seen that engine overall efficiency falls sharply if TET is reduced too much. This is due to the fact that as TET is reduced, the shaft rotational speeds decrease and there is a corresponding decrease in component efficiencies. This speed drop also results in lower compressor pressure ratios which figure 12a shows is detrimental to engine efficiency in the subsonic regions. It is desirable then to operate with as high a TET as possible and still maintain the desired thrust level.

If it were possible to alter μ (increase it), this would be seen to represent the most efficient means of obtaining the desired lower thrust levels during subsonic flight. Figure 11 demonstrates this clearly. TET's could remain high and by its definition, the propulsive efficiency of the engine would improve. In addition to being a more efficient subsonic cycle, there would be a spin off of other improvements to SFC.

The greatest other improvement would be to better nacelle performance. Swann, in reference 28, states that current engines designed for supersonic operation have inlets and exhaust nozzles designed to operate most efficiently at the cruise condition. For subsonic operation, very complex variable geometry intakes and nozzles are necessary to match the airflow requirements for the fixed cycle engine. Some of these variable devices are:

- translating centerbodies
- throat doors
- bypass doors
- takeoff doors
- secondary flow values

One result of these devices has been to increase installed engine weight and operational complexity. Another result has been to generate a significant amount of spillage and boattail drag during subsonic flight.

At M = .9, a subsonic pod has an installed SFC 50% lower than a supersonic pod containing a fixed cycle engine (28). Anything that could be done to make the intake and nozzle flow full (to reduce the spillage and boattail drag) would improve the subsonic SFC of a supersonic aircraft. Naturally, by increasing μ, and hence increasing the mass flow at the same thrust level, these sources of drag would be reduced.

<u>Take off Thrust Augmentation and Noise Reduction</u>. The problems of take off thrust augmentation and noise reduction are indirectly related to the principal aim of improving payload/range. For a civil SST, the noise problem would probably be the most significant one in the whole engine design.

Frequently, the two problems of fuel economy and noise reduction will be related. If the engines were designed to give optimum supersonic cruise performance, it may be found that they were underpowered at take off (31). Thrust conventionally is augmented by the use of afterburning, but with the increased jet velocities, the jet noise is also increased. Another way of meeting the take off thrust requirements would be to enlarge the engine to increase the mass flow. This approach would result in the penalties of increased engine weight, increased nacelle drag and the lower overall efficiencies at cruise (both subsonic and supersonic) since the engine would be overpowered at cruise and would need to be throttled back under these conditions.

Again, an increased bypass ratio for take off is seen as a means of both thrust augmentation, and by providing the thrust at a lower specific thrust, jet noise is reduced.

4.2.3 The Case for Variable Bypass Engines

Essentially the variable bypass engine concept is considered desirable for a supersonic cruise aircraft since its engine has conflicting requirements during different phases of the flight profile. Since it will be necessary to operate at optimum SFC at all times, the following contradictions arise:

a. at supersonic cruise, for low SFC, μ must be low and hence specific thrust will be high

b. for low SFC during subsonic flight and for low take off noise, specific thrust must be low and hence high μ is required.

In a variable bypass design, low supersonic SFC is obtained by proper choice of the design point engine parameters. Low subsonic SFC is obtained mainly through the more efficient higher bypass ratio engine. However, improvements in installed drag also aid in the low subsonic SFC.

It should be remembered that the primary goal of any variable cycle concept is to improve the aircraft's payload/range. This is done by reducing the powerplant system (including bare engine, installation, and fuel weights) weight proportion.

Weight Savings. The following data was given by Swann in reference 28. The figures pertain to the Boeing 2707-300 SST, a 750000 lb. aircraft with a payload of 61000 lb/300 passengers. The numbers represent the weight that could be saved on this SST for a typical flight with a variable cycle engine.

Item	Equivalent Weight Saving
Reduced fuel reserves because of better off design SFC	15000 lb
Better SFC on 400 N. mile subsonic legs	14000
Eliminate inlet complexity	7000
Weight saved on nozzle	5000
Weight saved on sound suppressor	9000
Drag saved on sound suppressor	6000
Off design intake drag saved	2000
Off design boattail drag saved	3000
	61000 lb = 300 passengers

Naturally, a variable cycle bare engine would weigh more and be more bulky than a fixed cycle bare engine. However, the weight saved on inlet and nozzle complexity should more than compensate for this.

Range Effects. Improved subsonic SFC and reduced installed weight for a variable cycle engine manifests itself in the graph shown as figure 13 (reference 28).

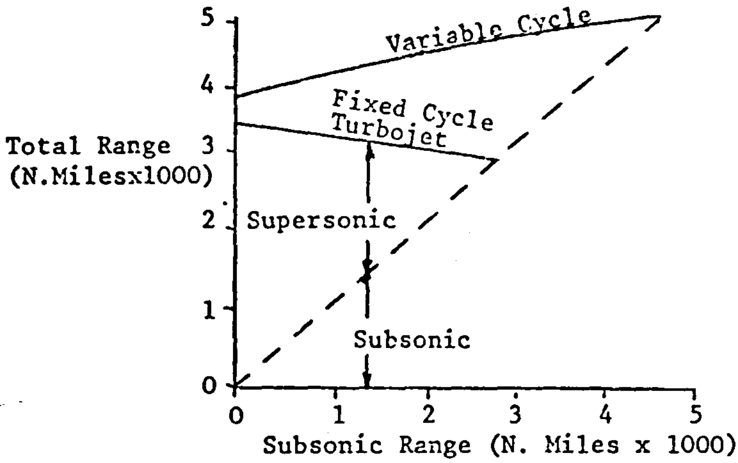


Figure 13 Effect of Subsonic Range of Total Range of an SST

Other Variable Cycle Applications. In addition to providing an economical engine for all sections of the flight profile of a supersonic cruise aircraft, an application would also be found in any mixed-mission aircraft where the thrust requirements and flight conditions vary considerably such as in a VTOL fighter. Reference 30 states that "such an engine will lend itself admirably to any application where more thrust modulation than can be achieved by throttle settings is desired or required."

Summary. This then is the case for a variable cycle engine which will vary the bypass ratio to enable efficient engine operation through all flight regimes of a supersonic cruise aircraft. Current literature and discussions with advanced planning engineers of a major engine manufacturer have convinced the author that such an engine will be a requirement for future high supersonic cruise speed aircraft. Reference 30 states:

> "The whole aerospace community of propulsion and airframe companies and agencies is convinced that the development of a variable cycle jet engine is the best and biggest step that can be taken now"

Because engine designers have just begun to take these engines seriously, very little information on their performance has been published (at least in the public sector). What this thesis proposes to do is to demonstrate that the ODP computer program developed during the course of this thesis and described in section 3 is capable of performing variable cycle engine calculations. The predicted performance of certain variable cycle engine configurations is also given.

4.3 Proposed Variable Cycle Engine Types

Once it is seen that an engine with a variable bypass ratio is advantageous, the problem then becomes one of how to create the variable bypass feature. What follows here are some ideas currently being proposed for such an engine.

4.3.1 Variable Geometry Aft Fans

This concept of a variable cycle engine is in reality an "add-on" approach. It was proposed for the GE/4 engine (late Boeing SST engine) when it was found that the basic engine take off thrust was insufficient and required augmentation. The use of afterburning to obtain the extra thrust produced excess noise levels, so an aft fan, which was only to be used at take off and subsonic speeds, was also considered. Reference 31 describes the concept and this particular application.

The configuration consisted of a basic GE/4 engine with an additional free turbine with a tip fan. Air entered the fan by blow-in doors or was ducted from the main inlet. At high flight speeds, the fan inlet was closed and the free turbine stopped by variable stators. It was calculated that this configuration was capable of saving 20000 lb. of fuel for a "normal" SST flight. Even more could be saved if a good deal of subsonic flight was necessary to avoid sonic booms over built up areas.

It is not known why this concept was not advanced further, but perhaps the designers chose to obtain the additional thrust by more conventional means such as afterburning plus noise suppressors. In any event, the variable geometry aft fan for the GE/4 died a natural death with the demise of the Boeing SST.

Such an engine may be simulated with the computer program described in this thesis in a manner similar to the Series/Parallel engines described in the next section.

4.3.2 Series/Parallel Components

The engine configuration described here represents the true variable cycle engine in that there is internal redistribution of the airflow. By valving and duct arrangements, air from an upstream compressor or fan may either enter another compressor (and ultimately the engine core), or may be exhausted separately.

A working model of such a valve/duct arrangement now exists. Reference 32 is a United States patent granted to G.W. Klees concerning an "Annulus Inverting Value" which was designed to enable series/parallel operation.

The series/parallel concept was described by Nichols in reference 33. Figure 14 demonstrates this concept applied to two compressors. Note that it could similarly be applied to two fans (one HP and one LP) or even two fans where one is an aft fan.

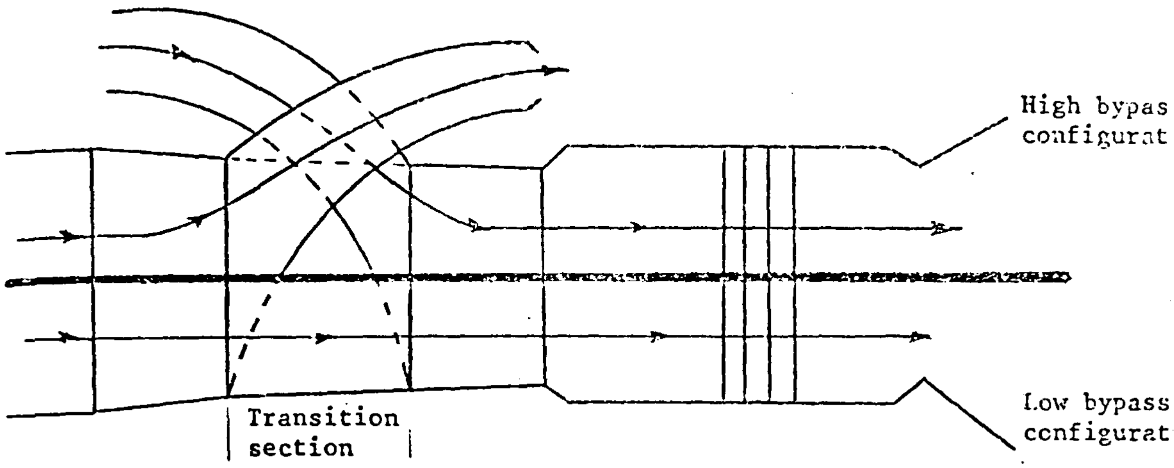


Figure 14 Engine With Series/Parallel Compressors

This variable bypass arrangement may be easily simulated using the computer program which has been developed. No new Bricks or subroutines are required and this method has proven itself in many trials. The portion of the engine which would be most difficult to simulate is, of course, the transition section. A schematic of this is shown in figure 15 which demonstrates a series/parallel fan arrangement.

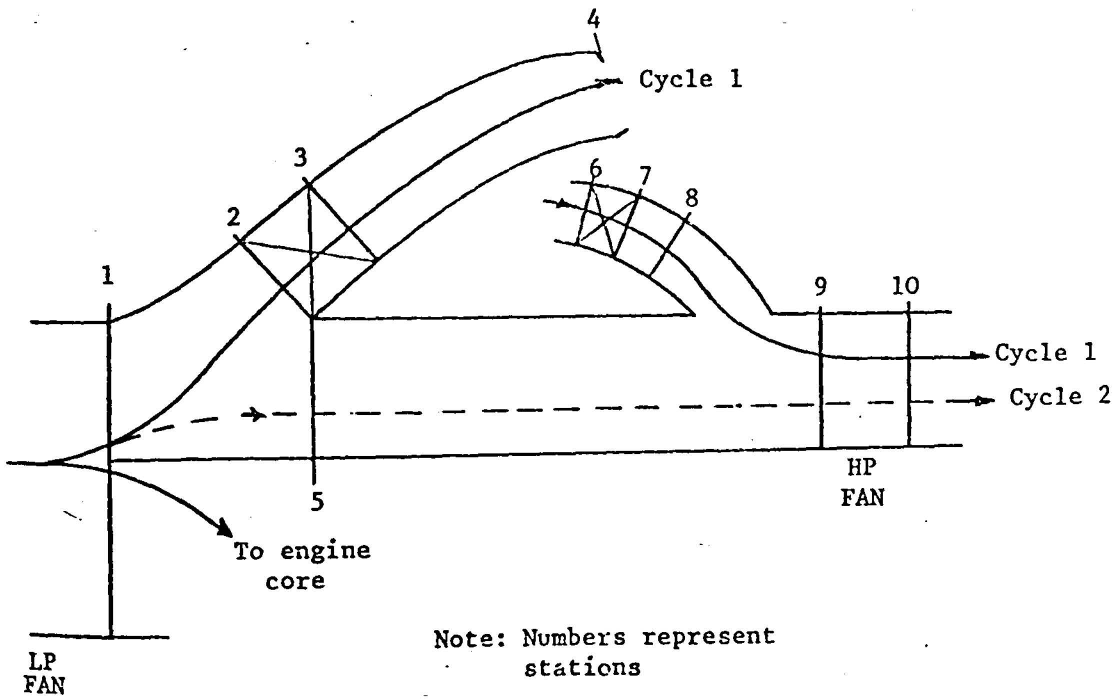


Figure 15 Schematic of Transition Section For Series/Parallel Fans

In the discussion that follows, cycle 1 represents the high bypass ratio and cycle 2, the low bypass ratio configuration. The flow at SV1 is assumed to be the outlet of a fan. Part of the fan exit air enters the engine core (lower arrow) while a portion is bypassed. This bypass air either is exhausted directly (cycle 1), or is compressed still further in cycle 2 by a fan operating between station 9 and 10. This downstream fan receives its air directly from atmosphere in cycle 1 operation.

The sequence of Bricks operating between the stations is given below.

Station	Operation	Brick	Result	Remarks
1→2	Flow split	7	W2 = λxW1	Provides bypass flow
2→3	On/off valve	7	W3 = λ_1xW2	λ_1 = 1.0 for cycle 1 λ_1 = 0.0 for cycle 2
3→4	Jet pipe	11		
1→5	Arithmetic	9	W5 = W1-W3	
6→7	On/off valve	7	W7 = λ_1xW6	λ_1 as above
7→8	Intake	1		
8+5→9	Y junction	8	W9 = W5+W8	Either W5 or W8 = 0.0
9→10	Compressor	2		

As can be seen, the "transition section" may be perfectly simulated in this mannner. In cycle 1, air from the first fan is exhausted through a nozzle and the second fan receives its air from the ambient conditions. By changing one BD value (λ_1) from 1.0 to 0.0, the section moves into cycle 2 operation. Now, no air flows past SV2 and so SV5 is equivalent to SV2. Again, no air flows past SV6 so that when SV's 5 and 8 are mixed, there is no flow at SV8 and hence SV9 is equivalent to SV5. Therefore the conditions are those at the outlet of the first compressor.

This technique may be used to represent any transistion section, whether separating two compressor, fans, or an aft fan.

4.3.3 Satellite Engine

The satellite engine proposals (references 21 and 33) are extensions of the series/parallel concept, the difference being that it is two engines which operate either in series or parallel. The front (upstream) engine would be a low pressure ratio turbofan with one or more turbojets downstream. The main drawback to this proposal is the weight and bulk required.

4.3.4 Constantly Varying Airflows

This design, referred to as the "modulating airflow engine" in reference 21 is shown in figure 16. This particular configuration has three spools and three exhaust nozzles. By varying the exit areas of the nozzles and closing the nozzle of LP fan, the proportion of airflow from the fans entering the engine core would also be varied. In this manner, the bypass ratio is changed to give the optimum engine configuration for each flight condition.

This type of engine could also be simulated in a straightforward manner. Simply resetting the nozzle areas as an ODP condition would result in an engine with the bypass ratio altered.

However, while it would be easy to simulate, such an engine would probably prove difficult to operate without avoiding operation off the component maps. In particular, the turbines would be faced with operation at several different gas flows and work requirements. Variable geometry turbines (and perhaps fans) would prove to be necessary for this type of engine if the bypass ratio was to be altered considerably. This of course would make the engine simulation much more involved.

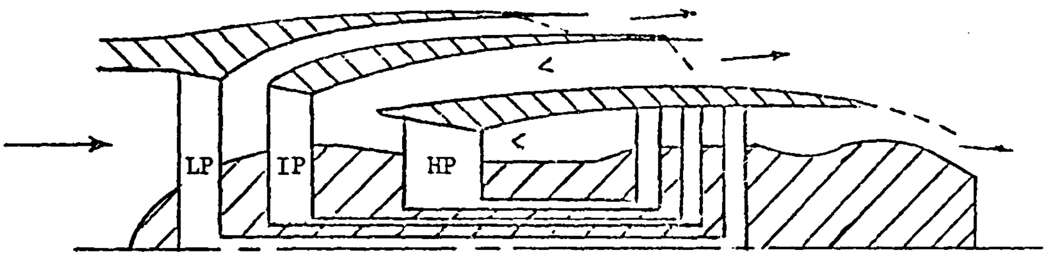


Figure 16 Three Spool Modulating Airflow Concept

4.3.5 It has been considered in the course of this work to investigate variable bypass engines employing a series/parallel concept as opposed to one of the other configurations mentioned previously for the following reasons:

1. The design and working model of a transition section (valve) means that new technology will probably be directed to engines employing this device.
2. It should be possible to design series/parallel engines without going into variable geometry components (such as variable area turbines) which would be necessary with a concept such as the "modulating airflow".
3. Because of points 1 and 2, in all likelihood, the first development of variable cycle gas turbines will be directed along the series/parallel concept.

SECTION 5

Series/Parallel Variable Cycle Engines

This section deals with the selection of design point parameters and overall engine performance of variable cycle engines employing the series/parallel concept. Compressors, fans, and aft fans all operating in series/parallel are considered.

5.1 Selection of Design Point Component Parameters

From the work done at Cranfield doing steady state simulations of these various engine concepts, it quickly became apparent that the selection of initial design point component parameters is extremely critical if the engine is to be capable of operation in the two cycles. The changes that can occur within the engine after transition from one cycle to the other are in essence what determine the engine design point parameters.

5.1.1 Transition Between Cycles

Ideal Transition. The problems of transition are a result of striving to attain what may be described as "ideal" transition. It would seem a reasonable requirement that, after the cycle change, the engine operation must not be significantly different from that before the change. This requirement manifests itself as follows:

1. shaft speeds should not change significantly
2. net thrust should not change significantly
3. components must not move into zones of impossible operation (eg. compressor surge)
4. TET limitations must not be exceeded.

It is in trying to meet these constraints (particularly compressor surge) that leads to problems in selecting the engine component design points.

5.1.2 Problems of Transition

Downstream (HP) Compressor Surge

Unless care is taken in the design point parameters, the downstream compressor (or fan) in a series/parallel configuration will surge as a result of the transition from parallel to series operation. The compressor on the outer shaft is designated "LP" and that on the inner shaft, "HP". Parallel operation is denoted "cycle 1" while series operation is "cycle 2". Figure 17 illustrates this configuration.

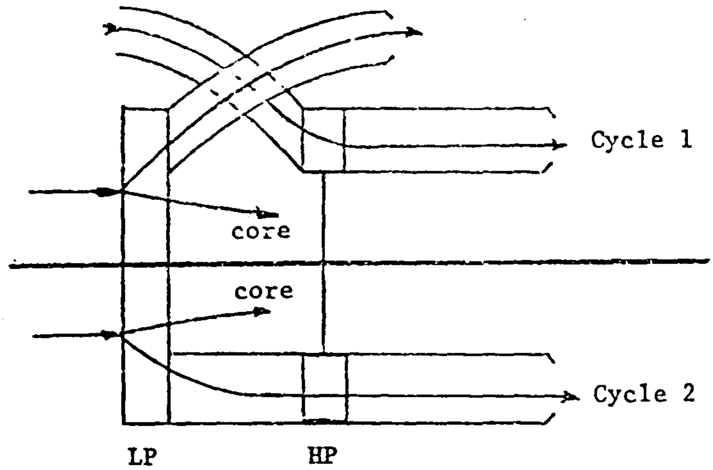


Figure 17 Series/Parallel Fans

In cycle 1 operation, both compressors receive ambient air and have their own inlet. The corresponding HP operating point is shown as point 1 on figure 18. When operating in cycle 2, the HP compressor receives compressed air from the LP fan. The fact that the HP entrance air temperature and pressure have now both increased will affect WAC, $N/\sqrt{T}$, and hence the operating point will be affected.

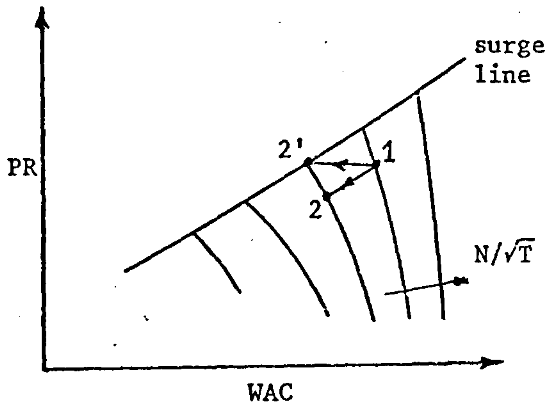


Figure 18 Compressor Map Demonstrating HP Surge

Since shaft speed is to be kept constant (in any event, because of inertia, its response would be such that it would remain constant for an initial period), $N/\sqrt{TA}$ will decrease slightly. How much will depend on pressure ratio of the LP fan.

Because the LP pressure ratio will not be large, (this will be reinforced later), the associated temperature rise will not be large either. Hence $N/\sqrt{TA}$ will not be decreased very much.

However, WAC, where

$$WAC = \frac{WA x \sqrt{TA/TSLS}}{PA/PSLS} \qquad SLS = \text{sea level static}$$

can change considerably. If it decreases too much, the compressor will surge as represented by point 2' on figure 18. It is necessary to specify the design point mass flows for both compressors, taking into account the LP pressure ratio so that the WAC value is not decreased excessively during the cycle transition.

Referring again to figure 18, assume that a cycle 2 value of HP WAC of

$$WAC_2 = 0.8xWAC_1 \qquad \begin{Bmatrix} \text{subscript } 1 = \text{cycle } 1 \\ \text{subscript } 2 = \text{cycle } 2 \end{Bmatrix}$$

will surge the compressor (point 2'), and a value of

$$WAC_2 = 0.9xWAC_1$$

is required for stable operation (point 2).

Further assume that the transition is to occur at sea level static conditions.

Therefore, $WAC_2 = 0.9xWAC_1$

$$\frac{WA_2 x \sqrt{TA_2/TSLS}}{PA_2/PSLS} = 0.9xWA_1$$

Since $PA_2/PSLS$ = LP Pressure Ratio (PR),

WA_2 = LP mass flow directed to HP,

and $TA_2/TSLS = PR^{\frac{\gamma-1}{\gamma}}$ (assuming an isentropic compression)

Therefore we require for stable operation,

$$\frac{WA_2 x PR^{\frac{\gamma-1}{2\gamma}}}{PR} = 0.9xWA_1$$

For $\gamma = 1.4$,

$$WA_1 = 1.11xWA_2xPR^{-0.857}$$

Thus, in order to ensure continuity of operation during transition, the above relationship must hold. In effect then, the design point values of airflow for both compressors will be dictated by the LP pressure ratio such that, at the transition point, the above equation remains valid.

For the above example, it may be seen that for a LP pressure ratio greater than 1.13, the cycle 1 HP mass flow must be less than the cycle 1 LP mass flow (which in cycle 2 enters the HP compressor). For efficient cycle 1 operation, it is desirable to have as large a bypass ratio as possible. This leads to the desirability of having a high bypass air flow. Since the bypass airflow is:

$$\text{bypass air flow} = WA_1 + WA_2$$

$$= WA_2 x(1+1.11xPR^{-0.857})$$

this shows the advantage to be gained by having a low LP pressure ratio.

Upstream (LP) Compressor Surge

This condition is a result of a severe compressor-turbine work imbalance and is to be found in series/parallel configurations where the entire compressors are operating in series/parallel and not just a portion of the flow (as in a fan). An example of this concept was shown in figure 14.

The previous section demonstrated why the cycle 1 HP mass flow must be less than the cycle 1 LP mass flow (in fact, the ratio of the mass flows is inversely proportional to the LP pressure ratio) in order to avoid HP surge. This mass flow requirement leads to LP surge problems for the engine configuration discussed here.

Consider such a hypothetical engine with a LP pressure ratio of 1.75 in cycle 1 just before transition and with a cycle 1 LP mass flow 1.5 times the cycle 1 HP mass flow. Thus, after transition to cycle 2, the HP (and core) mass flow has been increased 50%.

Looking at the turbines now, the HP turbine is required to do 50% more work because of the greater mass flow through the HP compressor. This is acceptable since this turbine now has 50% more hot gases to expand. However, the work load for the LP turbine has not increased, but has stayed constant. It too has 50% more hot gases to expand and immediately a work imbalance is created. Initially, the excess work being done by the turbine will cause the LP shaft to speed up and the LP compressor will surge. This is not self correcting; with an increased speed, the compressor will attempt to draw in more air, but this merely compounds the problem. Figure 19 demonstrates this type of surge.

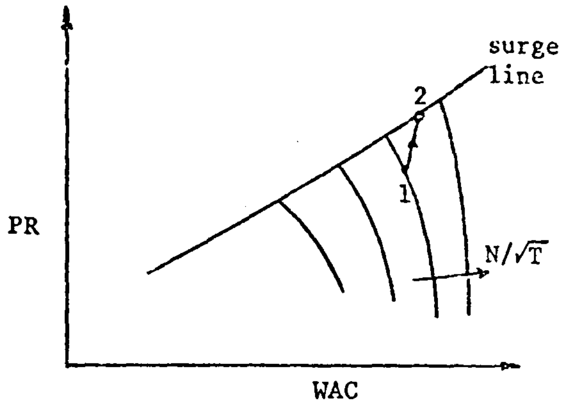


Figure 19 Compressor Map Demonstrating LP Surge

One possible means of solving this problem is a variable area LP turbine. This solution would result in a heavier and more complex engine. Until variable area turbines are developed and working satisfactorily, there is no point in designing series/parallel compressor engines. For this reason, only series/parallel fan engines are considered further in this work.

5.1.3 Point of Transition

Because of the problems that can occur as a result of a cycle change, it is obvious that the flight conditions where transition can occur satisfactorily are directly related to the values of the component parameters at that time. What is necessary, is that before transition, the LP and HP mass flows and LP pressure ratio must be such that HP compressor surge after transition is avoided. Since the mass flows, pressure ratio, and $N/\sqrt{T}$ value change as a result of flight speed and altitude, throttle setting, and nozzle area, then there will be a narrow band of flight conditions where transition is possible (for a given turbine entry temperature and nozzle area). The question then is not what is the optimum point of transition, but rather what is a possible transition point for a particular engine.

Transition would be possible over a wider range of flight conditions if the fuel flow and nozzle area were adjusted to keep the mass flows and LP pressure ratio at the necessary values. This leads of course to the requirement for a sophisticated control system to adjust the fuel flow and nozzle areas as required.

The point in an aircraft's flight profile where the change from cycle 1 to cycle 2 is desirable would be that point where there is a requirement for a higher specific thrust. Transonic acceleration is one such point, so the engine could be effectively designed to change cycle at a high subsonic speed. If this point was chosen as the engine design point, the correct design point values of the component parameters could be easily established.

5.1.4 Summary of Design Point Selection

In the forgoing discussion, it has been shown that the engine design point parameters are of necessity directly linked to the flight conditions, fuel flow, and nozzle area at which the engine is to change cycle. It simply is not possible to specify component design point parameters without considering the conditions during transition. In particular, the LP and HP mass flows and LP pressure ratio require careful selection.

If the transition point was chosen as the engine design point, the problem of choosing correct component parameters is somewhat simplified. This is really a compromise solution since the best design point (for engine efficiency considerations) should be the supersonic cruise condition. However, choosing the transition point provides a good starting place for a bit of trial and error work which would eventually produce an engine configuration capable of a cycle change and also incorporating good cruise performance.

5.2 Variable Cycle Performance Comparisons

It is the object of this section to demonstrate both the capabilities of the ODP computer program described in the thesis and the performance advantages of a series/parallel variable cycle engine for a supersonic transport application.

A comparison is made between the following engines:

Engine 1: fixed cycle, single spool turbojet
Engine 2: fixed cycle, two spool, two exhaust turbofan
Engine 3: variable cycle, two spool series/parallel fans
Engine 4: variable cycle, three spool series/parallel aft fan

5.2.1 Engine Performance Goals

The engines were designed and sized to give approximately the same cruise performance. It was then possible to comparethe performance away from the cruise condition and thus illustrate the advantages of the variable cycle concept. The SFC, specific thrust, and net thrust of the four engines was compared at the following points:

1. Take off (ISA, sea level static)
2. Altitude = 10000 ft
   Mach No. = 0.4
   Different throttle settings
3. Altitude = 36000 ft
   Mach No. = 0.9
   Different throttle settings
4. Cruise (altitude = 60000ft., Mach No. = 2.0)

The performance figures shown below were obtained from Mr. R. Denning (and staff) of the Advanced Projects Group at Rolls Royce, Bristol. These figures are representative of the required (or desired) performance of an advanced engine for a future supersonic transport aircraft.

Cruise condition: altitude = 60000 ft
Mach No. = 2.0
Net thrust at cruise ≈ 7000→7500 lbf (dry)
Compressor delivery temperature at cruise ≈ 800^{o}K
TET at cruise = 1625^{o}K
Bypass ratio at cruise ≈ 1.0 (not applicable to engine 1)
Take off thrust ≈ 35000÷40000 lbf
Maximum take off TET = 1700^{o}K

For noise considerations, the specific thrust at take off was to be less than 50 lbf/lbm/sec.

To simplify the comparisons, all engines are considered to have variable exit area convergent/divergent nozzles such that there is optimum expansion at all times. However, the throat areas are considered to be fixed. In addition, all engines are considered to operate with similar component efficiencies, pressure losses, percent of flow for cooling, etc.

5.2.2 Fixed Cycle Engines

Engine 1 is a conventional single shaft turbojet with the following basic parameters at cruise:

Compressor PR = 14.0:1
TET = 1625^{o}K

Engine 2 is a conventional two shaft turbofan with separate exhausts for both the bypass and core stream. At cruise, it has the following basic parameters:

Fan PR = 2.5:1
Compressor PR = 6.5:1
TET = 1625^{o}K
Bypass ratio = 0.915

5.2.3 Variable Cycle Engines

Schematic diagrams and codeword "programs" required for the simulation of engines 3 and 4 are given in Appendix E.

The variable cycle engines involved in this comparison are rather simple ones and hence represent engines that could be built with existing technology. There are no variable geometry turbines and the nozzle throat areas are considered fixed. Considerably more variation of bypass ratio would have been possible if both these components had variable geometry. However, the convergent/divergent nozzle exit areas are considered variable so that optimum expansion always occurs. The nozzle that only operates in the high bypass mode (cycle 1) is considered to be fixed area convergent while the other two nozzles are convergent/divergent.

It is recognized that two variable exit area convergent/ divergent nozzles would be extremely difficult (if not impossible) to design and construct. Two, two-dimensional nozzles are a possibility, but the actual construction and controls would indeed be very complicated. The approach taken here is that inferred in reference 20 which states:"...determine what a variable cycle engine exhaust nozzle can be desgned to do rather than how it can be built." Considering the exhaust system to be two independent nozzles is not an attempt to ignore the issue of a realistic working nozzle. Rather, it provides a starting point for any design. Probably the best nozzle system for these engines would be a mixing series/ parallel arrangement with a convergent/divergent nozzle designed on the ejector principle.

5.2.4 Performance Presentation

The performance comparisons are presented in two tables and two graphs. The first two tables compare the performance of the engines at the cruise and take off condition respectively. The two graphs illustrate SFC vs. net thrust curves of the engines at different TET's at two subsonic flight conditions.

The units for the quantities in the tables are:

Net trust = lbf
Specific thrust = lbf/lbm/sec
SFC = lbm/hr/lbf
TET = ^{o}K

	Cruise: 60000 ft, ISA, Mach No.=2.0			
	Engine 1	Engine 2	Engine 3	Engine 4
Net thrust	7118.0	7106.0	7131.0	7274.0
Specific thrust	62.47	31.97	32.75	33.43
SFC	1.3519	1.3070	1.3030	1.2858
Core PR	9.68	10.50	10.02	10.35
Bypass PR	-	1.99	2.13	2.08
Bypass Ratio	-	0.915	0.915	0.895
TET	1625.0	1625.0	1625.0	1625.0

	Take off: sea level, ISA, static			
	Engine 1	Engine 2	Engine 3	Engine 4
Net thrust	32355.0	41021.0	43500.0	43324.0
Specific thrust	93.77	61.64	50.63	50.97
SFC	1.0219	0.7745	0.6660	0.6592
Core PR	15.12	16.95	16.00	16.00
LP fan PR	-	2.56	2.00	2.00
HP or Aft fan PR	-	-	1.65	1.65
Bypass Ratio	-	0.795	1.340	1.345
TET	1650.0	1600.0	1500.0	1500.0

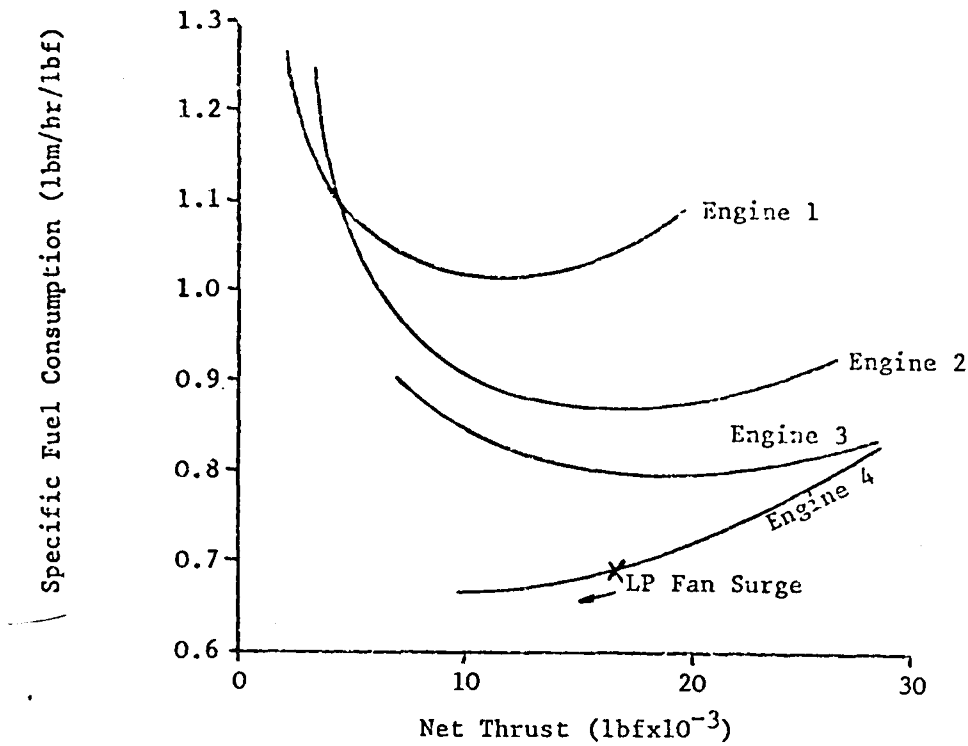


Figure 20 SFC vs. Net Thrust Curves at 10000 ft, Mach No. = 0.4

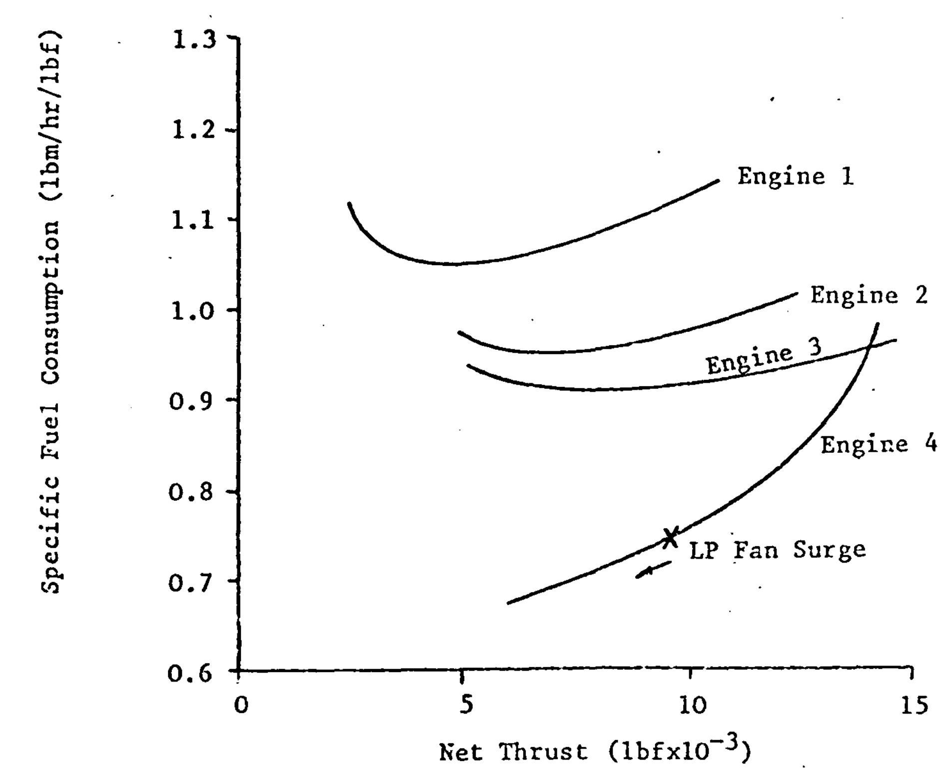


Figure 21 SFC vs. Net Thrust Curves at 36000 ft, Mach No. = 0.9

5.2.5 Remarks

While the comparison has dealt with relatively simple engines, the advantages of the variable cycle concept are clear. For equivalent cruise performance, the take off thrust levels for the variable cycle engines are achieved at lower IET, lower specific thrust (and hence lower noise levels), and lower specific fuel consumption.

Specific fuel consumption at holding and subsonic cruise flight conditions for various power settings is also seen to be better than that attainable with the two fixed cycle engines. (Note however that the aft fan engine had a tendancy to surge (the P fan) at low throttle settings. A better choice of component parameters, incorporation of an anti-surge bleed, or varying the nozzle area could have prevented this.)

It is worth repeating that the variable cycle engines considered here were merely simplified designs to demonstrate the concept. More sophisticated designs would include nozzles with variable throat areas in order to further control the bypass ratio. To vary the bypass ratio even further, variable geometry turbines would be a requirement. Also, none of these engines have been optimized for any particular mission and it is recognized that the overall performance of them may be far from ideal.

5.3 Recommended Future Work with Series/Parallel Engines

Because this thesis has not attempted to deal with variable cycle engines in detail, there remains much scope for future investigations. This computer program is seen to be able to analyse series/parallel designs relatively easily, and it is recommended that these engines be studied in greater detail.

One area for investigation would be the actual engine layout. For instance, what would be the advantages or disadvantages of series/parallel fans consisting of:

1. LP fan and aft fan (engine 4)
2. LP fan and HP fan (engine 3)
3. Two fans on the same LP shaft
4. other possible combinations

The determination of component operating lines would be required to answer these questions as the configurations are all thermodynamically similar.

Possibly the largest topic requiring investigation is that associated with nozzle area change (throat area for convergent/divergent nozzles). Merely cycling the engine with no nozzle area change reduces the possible change of bypass ratio. With area change, the engine airflow and hence bypass ratio could be more effectively controlled. However, this nozzle area change leads to many questions. Chief among these are:

1. How will the choice of DP parameters be affected?
2. How will the cycle transition point be affected?
3. How will the component operating lines be affected? Will the compressors move into surge or will temperature limitations be exceeded?
4. How much area change (and hence bypass ratio change) is possible before variable geometry turbines are required?
5. Will the area change result in sufficiently improved engine performance in order to justify it?

The suggestions are of course only a guide to a starting point. Once any investigations are begun, the requirement for more studies concerned with variable cycle engines will become obvious.

SECTION 6

DISCUSSION

1. While computer programs written to analyse the performance of a gas turbine engine have been in use for several years, those capable of simulating off design point performance have been restricted to studying only certain types of engines. Design point programs capable of simulating any engine have been in existance for a few years, but their flexibility has not spread into the off design point simulations.

2. The computer program which is the subject of this thesis combines the flexibility of design point programs with the facility of off desing point simulation. Because it is capable of dealing with an infinite number of engines and configurations, it is especially suited to institutions wishing to study several different engines without rewriting a program for each engine type. It is expected that the prime users of the program will be M.Sc students at the Cranfield Institute of Technology during their engine design studies. However, any organization or individual wishing to simulate differing engine types will find it useful.

3. The extreme flexibility of the program has proven useful in simulating variable cycle engines. Current literature (21, 22, 28, 30) and discussions with engine designers indicate that variable cycle engines will be the subject of a great deal of investigation in the near future. It is not known if any other program is capable of simulating these engines yet. For this reason, the computer program described in this thesis may prove valuable to any designer studying variable cycle engines. Many such engines have been simulated with the program, primarily series/parallel types including series/parallel aft fans.

4. Initial investigations with series/parallel engines have shown that in order to successfully transition from the parallel to series mode (or vice-versa), the choice of design point component parameters, airflow distribution, and flight conditions at transition are all interelated. In addition, certain types of series/parallel engines are not possible without the use of variable area turbines. Variable geometry turbines coupled with variable nozzles will probably be used in order to obtain maximum advantage of the bypass ratio control of a variable cycle engine.

5. Several areas of recommended study concerning variable cycle engines (primarily investigation of variable nozzle area) using this computer program are given in section 5.3.

6. Future development of the computer program may proceed along the following lines:

a. Rework of the thermodynamic data subroutines such that any hydro-carbon fuel may be considered.

b. Incorporation of a routine to deal with varying atmospheric humidity.

c. Slight rework of compressor and turbine bricks to permit variable geometry. The ability to alter the component characteristic maps' scaling factors in the middle of a run should allow some degree of variable geometry.

d. Finally, the development of heat exchangers and intercooler bricks would enable the program to simulate virtually every kind of gas turbine engine.

REFERENCES

1. SAWYER, J.W. (Editor) — Sawyers Gas Turbine Engineering Handbook. 2nd ed. Gas Turbine Publications, Stamford, Conn., 1972

2. MAY, R.J., BROWNSTEIN, B.J., PRZYBYLKO, S.J. — A Review of Current and Projected Aspects of Turbine Engine Performance. United States Air Force Aero Propulsion Lab, AFAPL-TR-71-34, 1972

3. SARAVANAMUTTOO, H.I.H. — Analogue Computer Study of the Transient Performance of the Orenda 600 HP Regenerative Gas Turbine. ASME Paper 63-AHGT-38, 1963

4. SARAVANAMUTTOO, H.I.H., FAWKE, A.J. — Simulation of Gas Turbine Dynamic Performance. ASME Paper 70-6T-23, 1970

5. SARAVANAMUTTOO, H.I.H., MacISAAC, B.D. — The Use of a Hybrid Computer in the Optimization of Gas Turbine Control Parameters. ASME Paper 73-GT-13, 1973

6. CHAPPELL, M.S., COCKSHUTT, E.P., SHARP, C.R. — Gas Turbine Cycle Calculations: Design Point Performance of Turbojet and Turbofan Engines. National Research Council of Canada, Aeronautical Report, LR-407, 1964

7. COCKSHUTT, E.P., SHARP, C.R. — Gas Turbine Cycle Calculations: Design Point Performance of Turbopropellor and Turboshaft Engines. National Research Council of Canada, Aeronautical Report, LR-448, 1965

8. PALMER, J.R. — The "Turbocode" Scheme for the Programming of Thermodynamic Cycle Calculations on an Electronic Digital Computer. College of Aeronautics Report, Aero 198, 1967

9. PALMER, J.R., ANNAND, K.P. — Description of the Algol Version of The "Turbocode" Scheme for the Programming of Thermodynamic Cycle Calculations on an Electronic Digital Computer. College of Aeronautics Report, Aero 203, 1968

10. HILL, P.G., PETERSON, C.R. — Mechanics and Thermodynamics of Propulsion. Addison-Wesley, 1965

11. COHEN, H.
ROGERS, G.F.C.
SARAVANAMUTTOO, H.I.H.
Gas Turbine Theory.
2nd ed., Longman, 1972

12. ANNAND, K.P.
Computer Programmes for the Analysis of Gas Turbine Performance.
M.Sc. Thesis, College of Aeronautics, 1967

13. FINNEY, P.
Computer Optimisation of Aircraft Gas Turbine Engines.
Cranfield Institute of Technology
M.Sc. Thesis, 1972

14. COCKSHUTT, E.P.
Gas Turbine Cycle Calculations; Differential Methods in the Analysis of Equilibrium Operation.
National Research Council of Canada, Aeronautical Report, LR-481, 1967

15. COCKSHUTT, E.P.
Gas Turbine Cycle Calculations; A General Method for the Computation of Off Design Point Performance.
National Research Council of Canada, unpublished

16. CHAPPELL, M.S.
GRABE, W.
Gas Turbine Cycle Calculations; Experimental Verification of Off Design Point Predictions for a Two-Spool Turbojet with Various Air Bleeds.
National Research Council of Canada, Aeronautical Report, LR-555, 1971

17. McKINNEY, J.S.
Simulation of Turbofan Engine, Part 1. Description of Method and Balancing Technique.
United States Air Force, Aero Propulsion Lab, AFAPL-TR-67-125-Part 1, 1967

18. McKINNEY, J.S.
Simulation of Turbofan Engine, Part 2. Users Manual and Computer Listing.
United States Air Force, Aero Propulsion Lab, AFAPL-TR-67-125-Part 2, 1967

19. KOENIG, R.W.
FISHBACH, L.W.
GENENG - A Program for Calculating Design and Off Design Performance for Turbojet and Turbofan Engines.
NASA, TND-6552, 1972

20. FISHBACH, L.W.
KOENIG, R.W.
GENENG II - A Program for Calculating Design and Off Design Performance of Two and Three-Spool Turbofans With as Many Nozzles.
NASA, TN D-6553, 1972

21. Aviation Week and Space Technology
17 September 1973, pages 73-75

22. Flight
11 October 1973, page 613

23. CHAPPELL, M.S.
COCKSHUTT, E.P.
Gas Turbine Cycle Calculations; Generation of Thermodynamic Data for Dry Air and Combustion Products.
National Research Council of Canada, Aeronautical Report LR-381, 1963

24. ROSENBROCK, H.H.
Computational Techniques for Chemical Engineers.
Pergamon Press, 1966

25. DENNISON, C.
General Non Linear Digital Simulation Methods Applied to Gas Turbine Dynamic Behavior.
Ph.D. Thesis, University of London, 1968

26. APPARAO, T.A.P.S.
Gas Turbine Cycle Calculations: The Effects of Fuel Composition and Heat of Combustion.
Canadian Aeronautics and Space Institute Transactions, Vol. 3, September 1970

27. BREMA, A.
Recent Advances in Mixed Cycle Engine Design and Application.
1st International Symposium on Air Breathing Engines, Marseille France, June 1972

28. SWANN, W.C.
Performance Problems Related to Installation of Future Engines in Both Subsonic and Supersonic Transport Aircraft.
2nd International Symposium on Air Breathing Engines, Sheffield, 1974

29. ZUCROW, M.J.
Aircraft and Missile Propulsion, Vol. II.
Wiley, 1958

30. Aviation Week and Space Technology
29 April 1974, pages 40-43

31. WEBER, R.J.
EVANS, D.G.
Variable Geometry Aft Fan for Takeoff Quieting or Thrust Augmentation of a Turbojet
NASA, TMX-67983, December, 1971

32. KLEES, G.W. — Annulus Inverting Valve
United States Patent, 3,779,282,
Granted 18 December, 1973

33. NICHOLS, M.R. — Dual Cycle Aircraft Turbine Engine
NASA Patent Application,
NASA Case-LAR-11310-1, Filed
6 September, 1973

APPENDIX A

Detailed Descriptions of Subroutines

Bricks

A.1 B1 - Engine Inlet and Atmospheric Properties

1. Required BD:

    1. Altitude
    2. ISA deviation
    3. Flight Mach number
    4. Ram recovery (if known)

2. Required SV Quantities:

    1. Inlet mass flow

3. Quantities Calculated:

    1. Inlet and outlet SV's
    2. Atmospheric properties
    3. Inlet total conditions
    4. Ram recovery (if not specified as BD(4))
    5. Momentum drag (appears as EV quantity)

4. Errors Produced: none

5. Mechanics of Calculations:

    a. The means of calculating the required quantities are quite straightforward. The International Standard Atmosphere is assumed throughout and the static temperature and pressure are calculated at the given altitude (allowance can be made for deviations in the standard atmosphere by BD(2)).

    b. Knowing the flight Mach number and the ambient temperature, the flight velocity is calculated by

$$V = M \times \sqrt{\gamma \times R \times (TSA)}$$

Having calculated V, the total pressure and temperature are calculated by

$$HA = h - V^2/2g\ J, \quad \text{hence obtain TA}$$

$$PA = PSA \times e^{(\phi(TA)\ -\ \phi(TSA))}$$

c. If the ram recovery (η) has not been input as BD, it is calculated according to MIL-E-5008B (USAF) specifications as follows:

$$\text{if } M \leq 1. \quad , \quad \eta = 1.$$

$$\text{if } 1. \leq M < 5. \quad , \quad \eta = 1. - .075\ ([M-1.]^{1.35})$$

$$\text{if } M \geq 5. \quad , \quad \eta = 800./(M^4 + 935.)$$

d. By definition of ram recovery,

$$PB = \eta \times PA$$

also, to complete the outlet SV,

$$TB = TA$$

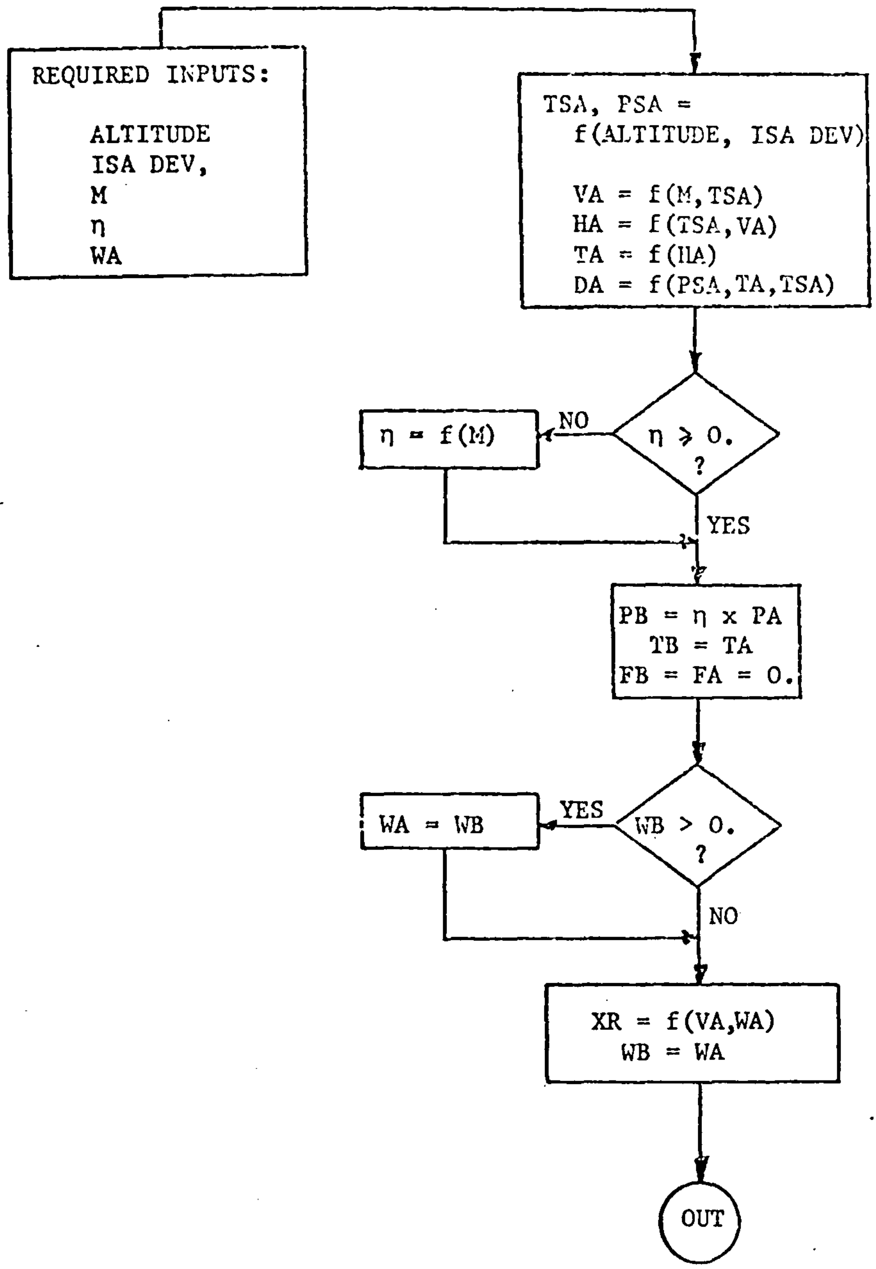
$$FB = FA = 0.$$

e. Momentum drag is calculated knowing the inlet velocity and mass flow:

$$XR = WA \times V/g$$

f. Finally, there is one other point to be mentioned about B1. As ODP cases require the engine mass flow to be varied, the engine inlet mass flow must also be varied. One way this is done is built into B1, by setting WA = WB <u>if</u> WB > 0. This method only is satisfactory if B2 (compressor) immediately follows B1. Section 3.8.4, paragraph 4 of the main body of the thesis further amplifies this point.

6. Brick 1 Flow Diagram



FLOW DIAGRAM, B1

A.2 B2 - Compression

1. Required BD:

   1. Z (ratio of pressure ratios along a constant CN line)
   2. PCN (shaft speed as a percentage of design shaft speed)
   3. Pressure Ratio
   4. isentropic efficiency
   5. Error producing switch,

   Plus a separate component map for each compressor.

2. Required SV Quantities: inlet SV

3. Quantities Calculated:

   1. outlet SV
   2. Power input required (appears as EV quantity)
   3. Scaling factors for using map in ODP cases

4. Errors Produced:

   1. If BD(5) > 0. , one error is produced:

$$\text{ERROR (1)} = \frac{WA - W_{MAP}}{W_{MAP}}$$

   If BD(5) $\leqslant$ 0. , no error is produced.

5. Mechanics of Calculations

   a. The workings of this brick consist basically of two types. One type being the mechanics of entering a component map and creating and/or using scaling factors to determine the component characteristics. The other workings are those associated with the thermodynamics associated with the process of compression.

   b. Use of Compressor Maps:

   The compressor maps consist of a plot of "corrected" airflow, WAC, where

$$WAC = \frac{WA \times \sqrt{TA/T_{SL}}}{PA/P_{SL}}$$

SL = sealevel, static conditons

versus PR for lines of constant CN over which are superimposed lines of constant isentropic efficiency, η. By specifying CN and Ƶ, where Ƶ is defined as:

Ƶ = Ratio of pressure ratios on a CN line

$$= \frac{\text{PR on speed line} - \text{low PR on speed line}}{\text{High PR on speed line} - \text{low PR on speed line}}$$

it is possible to obtain corresponding values of WAC, PR, and η.

c. Initially, the DP values of PCN, Ƶ, PR, and η are input (WAC is initially calculated from the inlet SV). PCN is converted to CN by

$$CN = \frac{PCN}{\sqrt{TA}} \div \left(\frac{100}{\sqrt{TA}}\right)_{Design}$$

Using the DP values of CN and Ƶ, the SEARCH subroutine finds the corresponding map values of WAC, PR, and η, and then calculates the map scaling factors by the following relationships:

$$PRSF = (PR_D - 1.) \; / \; (PR_{MAP,D} - 1.)$$

$$WASF = W_D \; / \; W_{MAP,D}$$

$$ETASF = \eta_D \; / \; \eta_{MAP,D}$$

where the subscript D represents the DP values of the quantities and MAP,D represents the values of the map quantity corresponding to the DP CN and Ƶ.

d. In ODP cases, CN and Ƶ are no longer at the DP values, and the values of WAC, PR, and η obtained from the maps are converted to the actual values by

$$PR = PRSF \times (PR_{MAP} - 1.) + 1.$$

$$ETA = ETASF \times ETA_{MAP}$$

$$WA = WASF \times W_{MAP}$$

where MAP represents the values of the MAP quantity corresponding to the ODP CN and Z.

e. Thermodynamics

The thermodynamics of the compression process are fairly straightforward. The thermodynamics are calculated after the map searching, and by that time the following quantities are known:

FA (and hence FB = FA), WA, TA, PA, PR, η

Knowing FA and TA allows

$\phi A = f(FA, TA)$
$HA = f(FA, TA)$ to be calculated.

The calculations then proceed as follows:

$$\phi BS = \phi A + \ln(PR)$$

$$TBS = f(FB, \phi BS)$$

$$HBS = f(FB, TBS)$$

$$HB = HA + (HBS - HA)/\eta$$

$$TB = f(FB, HB)$$

$$PB = PA \times PR$$

$$COMWK = (HB - HA) \times WA$$

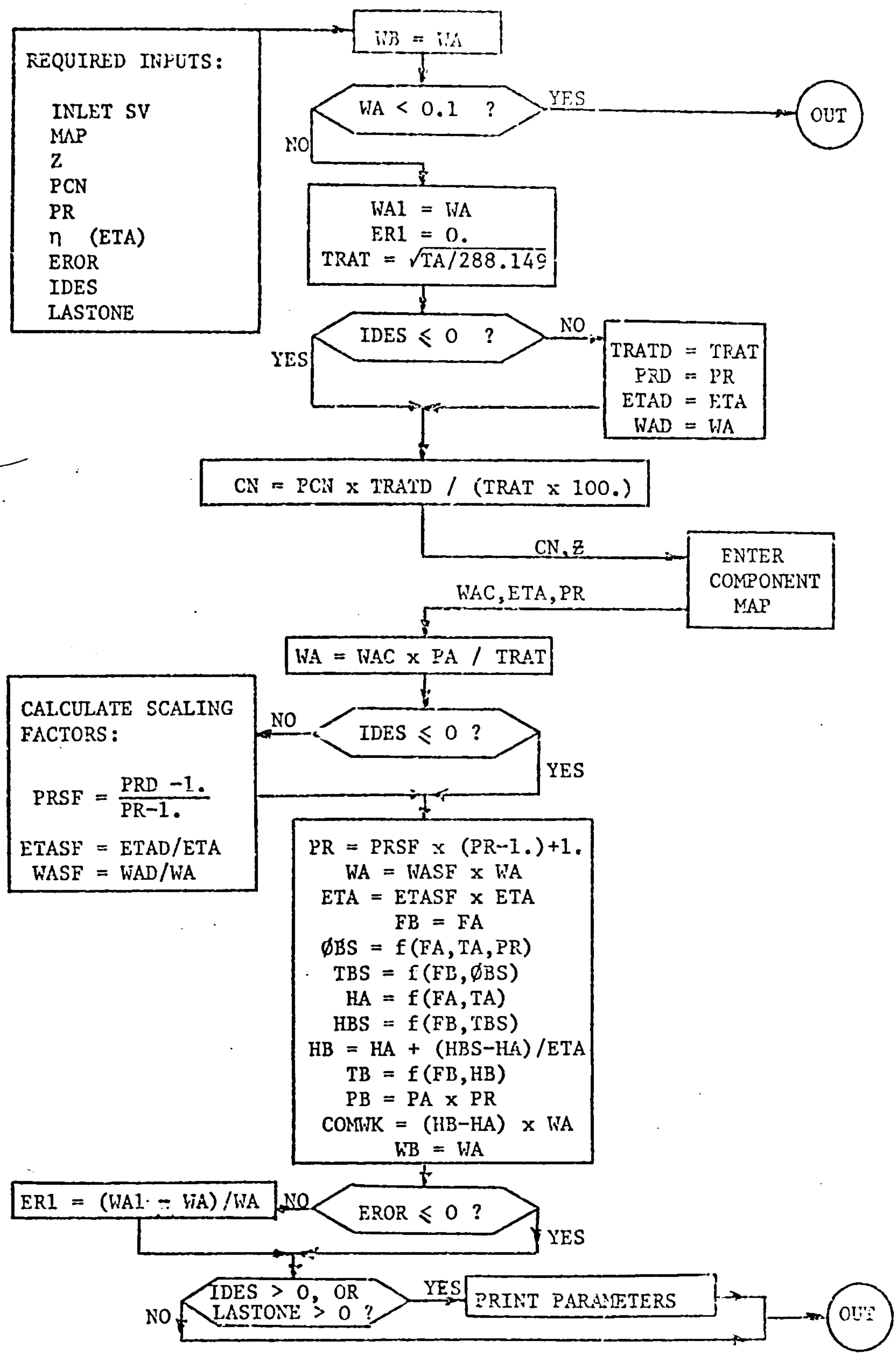
and finally, $WB = WA$

At this point, the outlet SV and the power input required by the compressor have been calculated.

f. If an error is to be produced by the brick, it is calculated now. The error is one of loss of mass continuity in that the actual inlet mass flow is different from the mass flow required by the component map.

$$ER1 = (WA - WA_{MAP}) / WA_{MAP}$$

6. Brick 2 Flow Diagram



FLOW DIAGRAM, B2

A.3 B3 Combustion Chamber

1. Required BD:

    1. (PA − PB) /PA (DELP)
    2. Combustion efficiency
    3. Fuel flow rate (WFB) only if it is to be specified

    (Combustion chamber map is contained within main body of the program)

2. Required SV Quantities:

    1. inlet SV
    2. if BD(3) $\leq$ 0. , then TB must be specified

3. Quantities Produced:

    1. outlet SV
    2. fuel flow rate (appears as EV quantity)
    3. scaling factor for using combustion chamber map in CDP cases.

4. Errors Produced: none

5. Mechanics of Calculations:

    a. This brick calculates the outlet SV of the combustion process knowing either the amount of fuel burned or else the final temperature. A combustion chamber map is used to find the combustion efficiency as a function of inlet pressure and temperature increase.

    b. The combustion process is calculated with a pressure loss (ΔP) that is assumed to be proportional to the kinetic head at the component inlet. Thus

    $$\Delta P = K \times (WA)^2 \times (TA/PA)$$

    the constant of proportionality, K, is determined by

    $$K = \left(\frac{\Delta P \times PA}{WA^2 \times TA} \right)$$ where all quantities within the parentheses are at their DP values.

c. Combustion products are calculated by knowing the combustion temperature rise. If TB has been given as input, the fuel flow rate (WFB) necessary to create that temperature is calculated. However, if WFB is input and TB is not, then initially TB is guessed and WFB' required to produce that temperature is calculated. If WFB' $\neq$ WFB within tolerance, then a new TB is chosen using the interpolation subroutine, AFQUIR. The process is then repeated until WFB' = WFB within an allowable tolerance.

d. Thermodynamics

B3 calculates combustion with FA $\geq$ 0. Thus the supporting medium may be the results of a previous combustion. Initially the increase in the enthalpy of the medium is calculated,

$$H = HB_{F=FA} - HA_{F=FA}$$

where HB_F is the enthalpy calculated at temperature TB and fuel/air ratio F.

Next, the effective calorific value (ECV) is determined by

$$ECV = CV + \frac{HB_{F=0}}{FS} - HB_{F=FS} \times \left(1 + \frac{1}{FS}\right)$$

where FS is the stoichometric fuel/air ratio and CV is the fuel calorific value

By the definition of combustion efficiency, η, FB may now be calculated

$$FB = \frac{\Delta H}{ECV} \times \frac{1}{\eta}$$

The mass of air involved in the combustion process is calculated by

$$WA_{air} = (1 - FA) \times WA$$

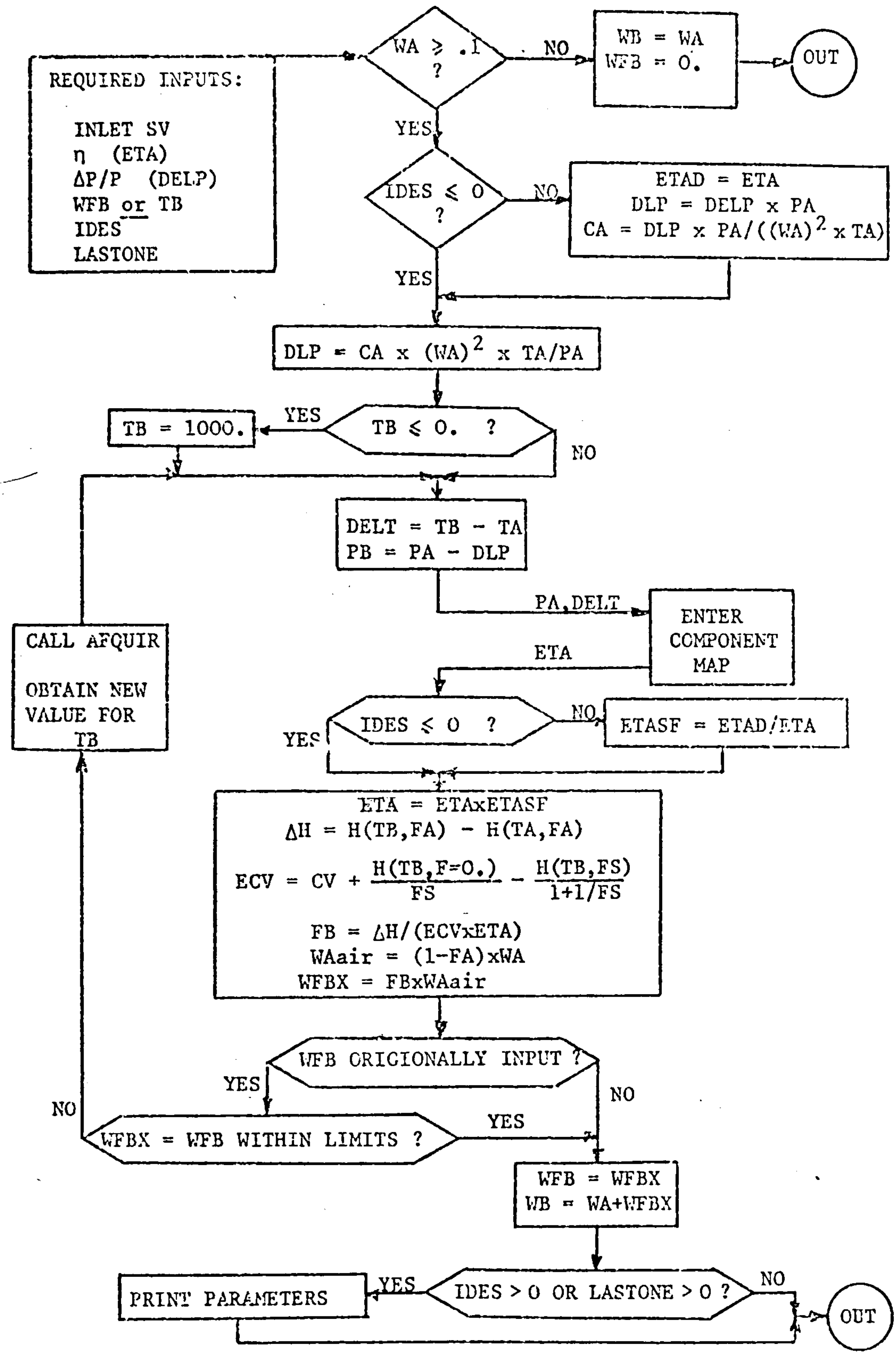
Knowing this, the fuel flow rate is found by

$$WFB = FB \times (WA_{air})$$

and

$$WB = WA + WFB$$

6. Brick 3 Flow Diagram



FLOW DIAGRAM, B3

A.4 B4 Compressor Turbine

1. Required BD:

    1. Auxilary power extracted (AUXWK)
    2. isentropic efficiency, η
    3. Turbine flow function ($TF = WA\sqrt{TA}/PA$) *
    4. CN *
    5. Number of compressor to which turbine is attached *

   Plus a separate component map for each turbine

   * see paragraph 6b for clarification

2. Required SV Quantities:

    1. Inlet SV

3. Required EV Quantities:

    1. Power requirement of compressor to which it is attached (COMWK)

4. Quantities Calculated:

    1. Outlet SV
    2. Scaling factors used in ODP cases

5. Errors Produced:

    1. $ER1 = (TF_{ACTUAL} - TF_{MAP}) / TF_{ACTUAL}$

    2. $ER2 = (\Delta H_{COMP. + AUXWK} - \Delta H_{TURB\ MAP}) / \Delta H_{COMP. + AUXWK}$

6. Mechanics of Calculations:

    a. As with, B2, compressor, the workings of this brick consist basically of two types. One type is concerned with the workings of the component map and the other type is associated with the thermodynamics involved.

    b. BD(3) and BD(4) are concerned only with the component map. They <u>are not</u> the actual values of the quantities but are the values to be used in entering the component map. Hence they merely specify where on the map one is working. BD(5) is the number of the compressor to which the turbine is attached.

In this way, the turbine rotational speed is matched to that of the compressor.

c. <u>Use of Turbine Map</u>

The turbine maps consist of a plot of temperature corrected enthalpy drops, DH

$$DH = (HB - HA) / TA$$

versus CN where

$$CN = \frac{PCN}{\sqrt{TA}}(COMPRESSOR)$$

for lines of constant "Turbine Flow Function", TF

$$TF = WA \times \sqrt{TA} / PA$$

over which are imposed lines of constant isentropic efficiency, η. By specifying TF and CN, appropriate values of DH and η are determined.

d. As with other components using maps, the scaling factors are calculated in the DP run. Initially, the turbine speed scaling factor, CNSF, is calculated by

$$CNSF = \left[\frac{CN \times \sqrt{TA}}{PCN_{COMP}} \right]_{DESIGN}$$

Next, the flow function scaling factor (TFSF) is found,

$$TFSF = \left[TF \div \left(\frac{WA \times \sqrt{TA}}{PA}\right) \right]_{DESIGN}$$

The enthalpy drop is calculated knowing the total work done (AUXWK + COMWK). HA is easily found since FA and TA are known. Thus

$$HB = HA - (AUXWK + COMWK) / WA$$

Using the DP values of TF and CN, the SEARCH subroutine finds the corresponding map values of DH and η and then B4 calculates the remaining scaling factors by the following relationships:

$$DHSF = \left[\frac{HB - HA}{TA}\right]_{DESIGN} \div DH_{MAP,D}$$

$$ETASF = \eta_{DESIGN} \div \eta_{MAP,D}$$

where $DH_{MAP,D}$ and $\eta_{MAP,D}$ were found using DP values of TF and CN.

e. In ODP cases, to enter the map, CN is determined by

$$CN = CNSF \times \left(\frac{PCN_{COMP}}{\sqrt{TA}}\right)$$

(TF is usually a variable for ODP work and does not need to be corrected for here. The scaling factor is required when an error is produced.)

The remaining parameters are derived from the map values by

$$\eta = ETASF \times \eta$$

$$DH = DHSF \times DH$$

Because of the shape of the curves on a turbine map, it is possible that an ODP point would result in values not on the turbine maps. Subroutine MAPBAC changes the map value and on independent variable (PCN_{COMP} or TA) in an attempt to rectify the situation.

f. Thermodynamics

After the map has been entered and the parameters scaled, the following quantities are known:

FA (and hence FB = FA), WA, TA, PA, η, DH

Initially, knowing TA and FA allows

$$HA = f\ (FA, TA)$$

$$\phi A = f\ (FA, TA)$$

to be determined. Then, HB is found from

$$HB = HA + DH \times (TA)$$

The enthalpy for an isentropic expansion, HBS, is calculated

$$HBS = HA - (HB - HA) / \eta$$

The calculations then proceed as follows

$$TBS = f(FB, HBS)$$

$$\phi BS = f(FB, TBS)$$

$$PB = PA \times e^{(\phi BS - \phi A)}$$

$$TB = f(FB, HB)$$

$$WB = WA$$

At this point, the outlet SV has been calculated. If an ODP case, there will also be two errors produced.

g. Errors

The first error produced in due to a lack of continuity, where the (scaled) actual inlet flow function does not equal the flow function used to enter the map.

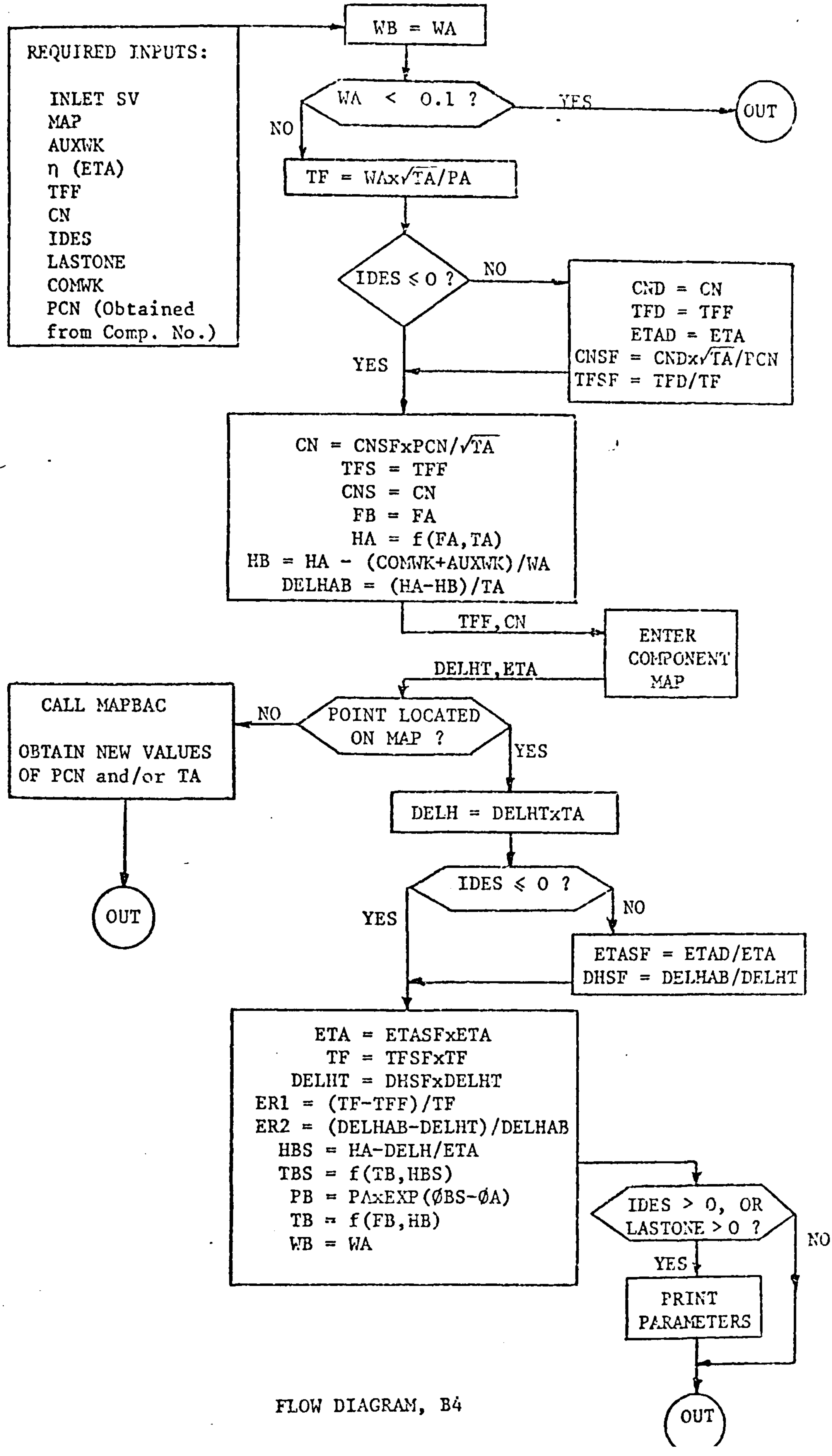
$$TF_{\substack{ACTUAL \\ (SCALED)}} = TFSF \times TF_{ACTUAL}$$

$$ER1 = (TF_{\substack{ACTUAL \\ (SCALED)}} - TF_{MAP}) / TF_{\substack{ACTUAL \\ (SCALED)}}$$

The second error is one of work imbalance. The work required by the compressor and auxilary output does not equal the work done by the turbine.

$$ER2 = (\Delta H_{\substack{COMP. \\ + \\ AUXWK}} - \Delta H_{\substack{TURB \\ MAP}}) / \Delta H_{\substack{COMP. \\ + \\ AUXWK}}$$

7. Brick 4 Flow Diagram



FLOW DIAGRAM, B4

A.5 B5, Convergent Nozzle

1. Required BD:

   1. Floating exit area switch

   if BD(1) > 0. , exit area floats to satisfy continuity

   if BD(1) < 0. , exit area is fixed

2. Required SV Quantities

   1. Inlet SV
   2. Free stream SV

3. Required EV Quantities: none

4. Quantities Calculated:

   1. Outlet SV, including area and velocity
   2. Nozzle coefficient (using subroutine NOZCO)
   3. Gross thrust (appears EV quantity).

5. Errors Produced:

   If the nozzle area is fixed, one error is produced:

   1. $ER1 = (PA_{REQUIRED} - PA_{ACTUAL}) / PA_{REQUIRED}$

6. Mechanics of Calculations

   The calculations proceed as follows:

   a. The sonic velocity for the given component inlet conditions is determined by an iterative process. First, a guess is made of TSS (Temperature, static, sonic). With this value, HSS and CSS (enthalpy and sonic velocity at TSS) may be determined. This value of HSS should equal the enthalpy found by

   $$H = HA - CSS^2 / 2g_cJ$$

   If HSS does not equal H within some allowable tolerance, a new guess of TSS is made and the process repeated until TSS is determined.

b. Assuming an isentropic expansion to PATM, the thermodynamic properties for this type of expansion are calculated, again by an iterative process. Using a first guess of TSA (static temperature at inlet) the entropy (SSA) at TSA and PATM may be determined. Also, the entropy (SA) at TA and PA may be calculated. For an isentropic expansion, SSA should equal SA. If not, a change is made to the value of TSA and the process is repeated until SA - SSA approaches an allowable minimum value.

c. The velocity at these values of TA and TSA may be determined by

$$VAS = \sqrt{2g_c J \times (HA - HSA)}$$

By comparing this velocity to CSS, it is then known whether the expansion is to PATM or else a higher value (ie: exit velocity is either sonic or subsonic). Now, the exit area and velocity may be calculated for either case if the nozzle has a floating area or else the simulation is a DP cycle.

1) If VAS < CSS, the exit velocity is subsonic. Therefore,

$$VB = VAS$$

$$TSB = TSA$$

$$PSB = PATM$$

The density, RHO, may be calculated knowing TSB and PSB, and thus, the exit area, AB is found

$$AB = WA / (RHO \times VB)$$

2) If VAS $\geq$ CSS,

$$VB = CSS$$

$$TSB = TSS$$

$$PSB = PA \times (TSB / TA)^{\gamma/\gamma-1}$$

Again, the density and exit area are found as in 1) above

3) A jump is now made to step g.

d. If an ODP cycle with a fixed area nozzle, the calculations continue with the area, AB, found in step c. This time an error is generated as the actual back pressure, PA, will be different from that required for the outlet conditions.

$$ER1 = (PREQ - PA) / PREQ$$

e. First, the area assuming sonic exit velocity (ABCRIT) is calculated. If AB > ABCRIT, then the exit velocity is subsonic. The static outlet conditions are determined by an iterative process. Since the velocity is subsonic, PSB = PATM.

With an initial guess of TSB, the density, RHO, and hence VB may be found

$$RHO = PSB / (R \times TSB)$$

$$VB = WA / (RHO \times AB)$$

The enthalpy found by

$$H = HA - VB^2/2g_cJ$$

must equal HSB (enthalpy at TSB) within some tolerance. If not, then TSB is altered and the process repeated until the two enthalpies are within the tolerance.

f. If AB < ABCRIT, then the exit velocity is sonic and

$$VB = CSS$$

In this situation, $TSB = TSS$

and $PSB = PATM \times (ABCRIT/AB)$

g. Once the static outlet conditions are known,

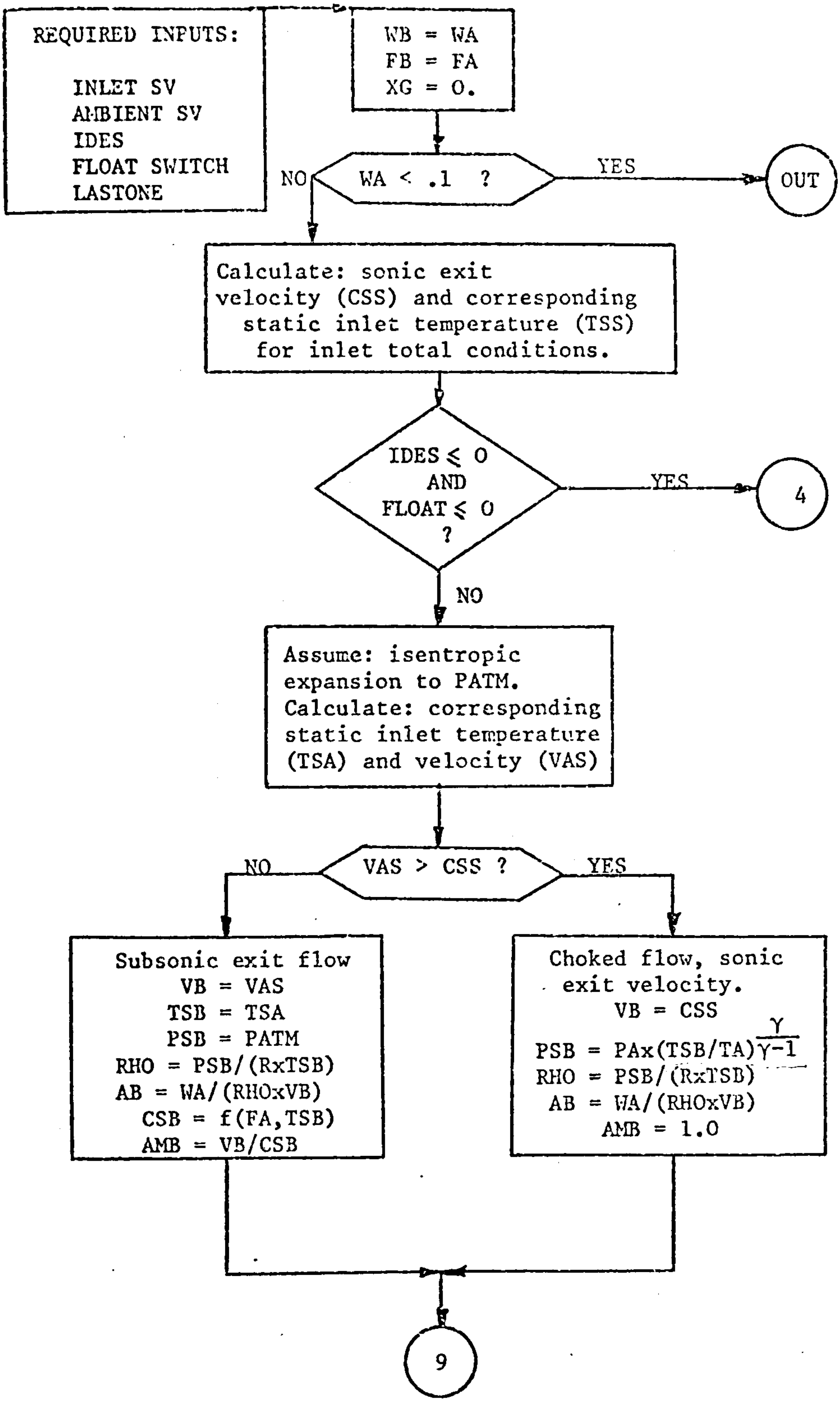
$$TB = TA$$

$$PB = PA$$

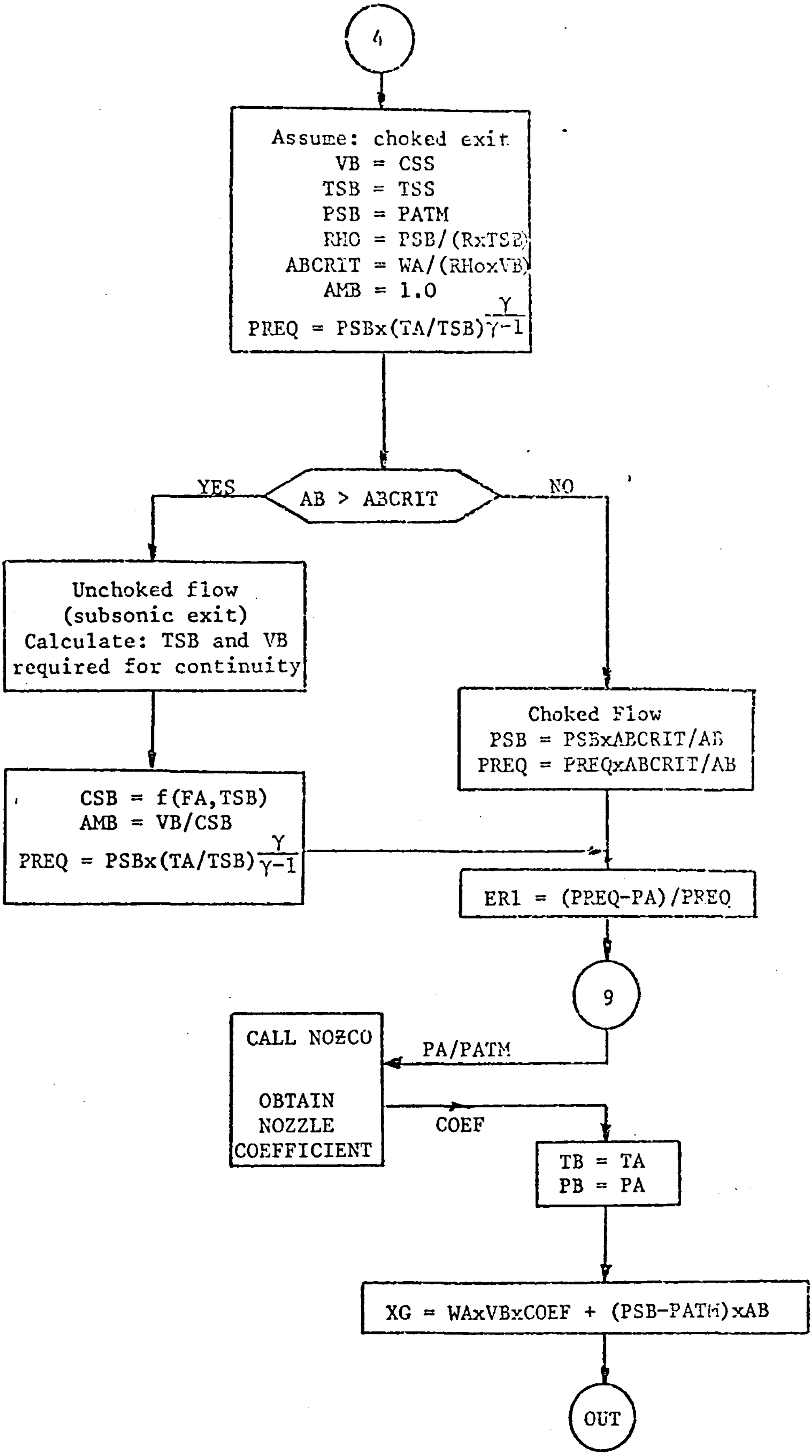
Using the pressure ratio, PA/PATM, and the fact that the area ratio for a convergent nozzle is 1.0, subroutine NOZCO calculates the nozzle coefficient (COEF). Finally, the gross thrust produced by the nozzle is calculated by

$$XG = WA \times VB \times COEF + (PSB - PATM) \times AB$$

6. Brick 5 Flow Diagram



FLOW DIAGRAM B5



FLOW DIAGRAM, B5 (CONTINUED)

A.6 B6, Calculation of Engine Performance Parameters

1. Required BD: none

2. Required SV Quantities:

   1. The engine inlet air meass flow is required in order to calculate specific thrust. This is input by putting the SV numbers of the engine air inlets in the SV positions of the CW. In this way, up to three air inlets may be accommodated. If the engine employs more than three inlets, B9, the arithmetic brick, should be used to add the airflows. In this case, the total airflow would be input as one SV.

3. Required EV Quantities:

   1. Total gross thrust produced by all nozzles (XG)
   2. Total momentum drag of all inlets (XR)
   3. Total fuel flow of the engine (WFB)

4. Quantities Calculated

   1. Net thrust (XN)
   2. Specific fuel consumption (SFC)
   3. Specific thrust (SXN)

5. Mechanics of Calculations:

   a. This brick is not a separate subroutine, but is contained within the MASTER segment as it is so simple. It does nothing but perform arithmetic calculations to transform the inputs into engine output parameters.

   b. First, the total inlet air mass flow (AIRIN) is calculated by adding the mass flows of the SV's specified in paragraph 2.

   c. Since the following quantities are now known:

      AIRIN, XG, XR, WFB;

      the remaining parameters are calculated by

      XN = XG - XR

      SFC = WFB/XN

      SXN = XN/AIRIN

A.7 **B7, Division/Addition of Mass Flow and/or Total Pressure**

The brick solves the equations

$$WB = \lambda x (WA) + \Delta W$$
$$PB = \lambda x (PA) + \Delta P$$

1. Required BD:

    1. λ (for WA)
    2. ΔW
    3. λ (for PA)
    4. ΔP

2. Required Quantities:

    1. Inlet SV

3. Quantities Calculated:

    1. Outlet SV

4. Errors Produced: none

5. Mechanics of Calculations:

    a. This is a simple, straightfoward set of calculations which solves the equations

    $$WB = \lambda \; x (WA) + \Delta W$$
    $$PB = \lambda \; x (PA) \div \Delta P \; .$$

    Thus, it is used if a portion of the mass flow is to be bled off or if a portion of the total pressure of a flow is to be lost. The "Δ" terms allow an addition or subtraction to the flow or pressure to be made.

    b. As it is so short and simple, the calculations are performed as part of the MASTER segment and are not a separate subroutine.

A.8 B8, Simple Mixing Where one Flow is Small Relative to the Other

1. Required BD: none

2. Required SV Quantities:

   1. Main inlet SV (SV1)
   2. Secondary inlet SV (SV2)

3. Quantities Calculated:

   1. Outlet SV (SV3)

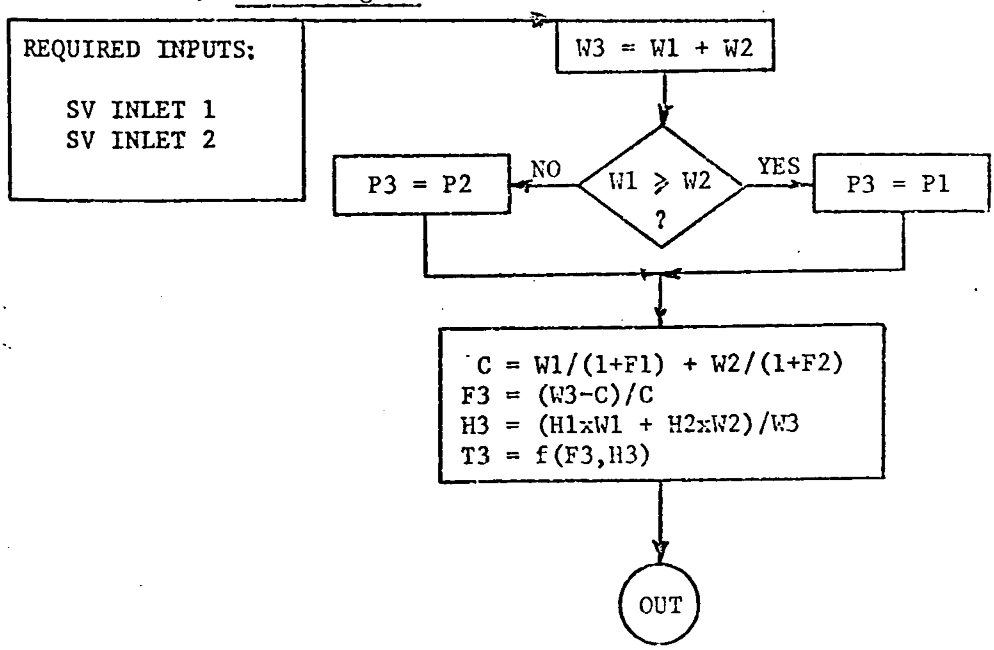
4. Errors Produced: none

5. Mechanics of Calculations:

   a. As this brick simulates simple mixing where one of the flows is small relative to the other, the simplifying assumption is that there is no change in the main stream total pressure (P1) as a result of the mixing. Thus, the only affect is a change in fuel/air ratio and total temperature.

   b. An example where this is successfully used is in cooling bleed air where the bleed flow represents a small proportion of the main flow. One other use is in the form of a "Y" junction where by proper use of B7, only one arm of the "Y" is in use at one time.

6. Flow Diagram



A.9 B9, Arithmetic Brick

1. Description:

This brick is capable of performing any of the four arithmetic functions (+, -, x, ÷) on any two SV, BD or EV quantities with the result being placed in any SV, BD, or EV store.

2. Input Requirements:

a. This brick does not require any inputs in the form of BD or SV data. Instead, all information is specified in the CW calling the brick.

b. The CW takes the usual form; using the following columns:

column 1 → 3	Brick number, 009
column 4,5	Operation number
column 6 → 9	operator 1 (OP1) description
column 10 → 13	OP2 description
column 14 → 17	OP3 description

c. The operator (OP) description is as follows:

first two columns = -1 if operator is EV quantity
= 0 if operator is BD quantity
= SV number if operator is SV quantity

second two columns = EV number if operator is EV quantity
= BD number if operator is BD quantity
= SV item number if operator is SV quantity

d. Operations

Operation number	Operation	
01	R1 = OP1 + OP2	*see para. 3 for description of R1
02	R1 = OP1 - OP2	
03	R1 = OP1 x OP2	
04	R1 = OP1 ÷ OP2	
05	OP1 = OP2 + OP3	
06	OP1 = OP2 - OP3	
07	OP1 = OP2 x OP3	
08	OP1 = OP2 ÷ OP3	

There is no restriction that OP1 can not be the same as OP2 or OP3. This leads to the possibility of equations such as

OP1 = OP1 - OP3

3. <u>Output Quantities:</u>

   a. In operations 01 to 04, the result of the operation, R1, is put in the next available EV store

   b. In the other operations, the result, OP1, takes on the value determined by the equation.

4. <u>Example:</u>

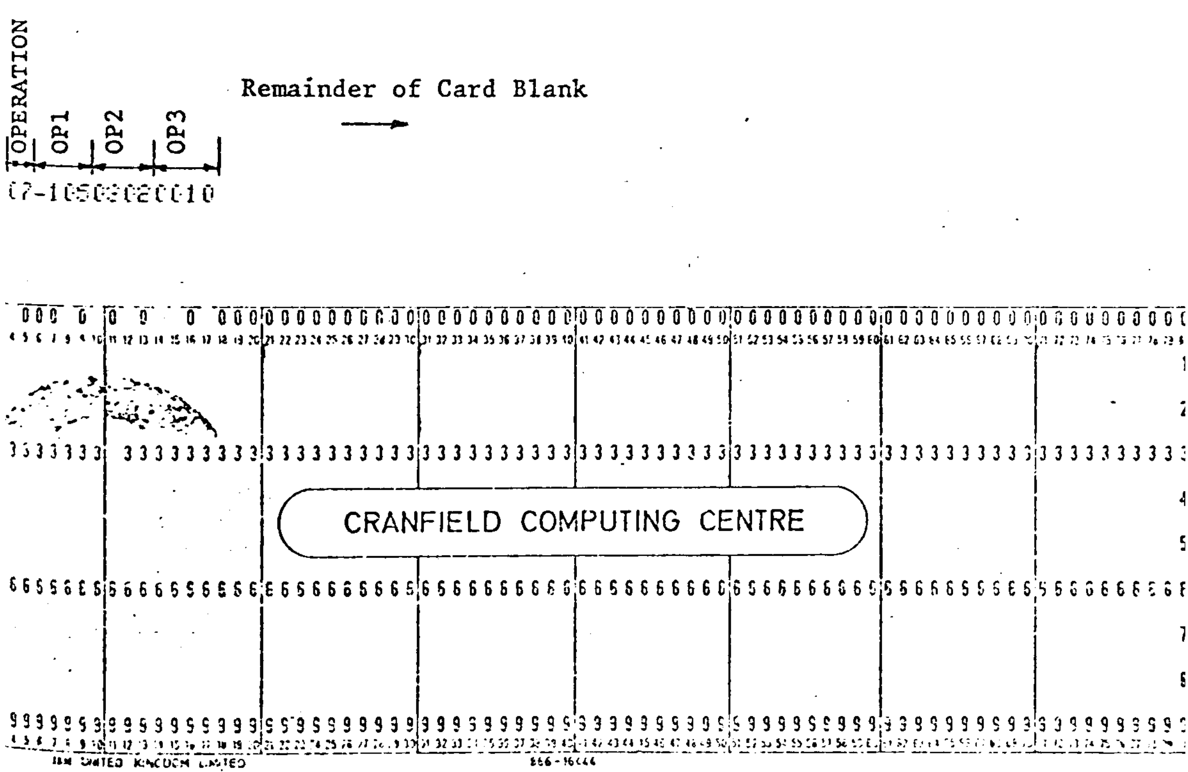
   While the instructions for using this brick may at first seem complex, they are quite easily mastered and the use of the brick will become indispensable to a serious user of the program. An example of how the CW is formed is given below.

   a. It is desired to perform the following calculation

      SV (3,2) x BD (10)

      with the result to be placed in EV(5). In other words, EV(5) will be reset to the result of this operation.

   b. A punched card showing this CW is:



A.10 B10, Complete Mixing

1. Required BD:

   1. Number of compressor providing flow for stream 2 (usually bypass stream).
   2. SWITCH
   3. VALUE

      IF SWITCH > 0. , VALUE REPRESENTS MACH NUMBER, STREAM 1

      IF SWITCH ≤ 0. , VALUE REPRESENTS STATIC PRESSURE, STREAM 1

2. Required SV Quantities:

   1. SV stream 1
   2. SV stream 2

3. Required EV Quantities: none

4. Quantities Calculated:

   1. Outlet SV, stream 3
   2. Areas, streams 1,2,3
   3. Mach numbers, streams 1,2,3
   4. Static pressures, streams 1,2,3

5. Errors Produced:

   1. Inlet static pressures not equal,

      ER1 = (PS1 - PS2) / PS1

6. Mechanics of Calculations

   a. Description:

   The mechanics of calculations associated with complete mixing are indeed quite complicated. Basically, what this brick does is, given the DP value of the static pressure or mach number of stream 1, it calculates for all streams the Mach numbers, static pressures, and flow areas. In the ODP situation, the brick uses the DP areas to calculate static pressures, Mach numbers and the outlet SV.

b. Mixing is complicated by the fact that it may actually be impossible if one of the inlet total pressures is less than the other inlet static pressure. If this situation occurs, exit is made from the brick and the run stops. Advise to the user is then given by a diagnostic print out. Also, in an ODP run, mixing may prove impossible while the engine is trying to balance. This situation is internally accounted for by automatically selecting a new compressor operating point and thus avoiding a possible trouble area. When the user specifies the number of the compressor providing the flow for stream 2 in BD(1), he is in fact detailing which compressor will obtain the new operating point, should this prove necessary.

c. <u>Calculations:</u>

Initially, the following data are known for the two inlet streams:

T, P, F, and either PS or Mach number (AM) for stream 1. Knowing T and F allows the following to be calculated using function TRM and common relationships:

$$R1 = f(F1)$$

$$R2 = f(F2)$$

$$H1 = f(F1,T1)$$

$$H2 = f(F2,T2)$$

$$\phi 1 = f(F1,T1)$$

$$\phi 2 = f(F2,T2)$$

If the run is an ODP case, a jump is made to step g.

d. First, A1 and A2 are calculated with PS2 = PS1. If PS1 is not given but AM1 is, then TS1 (static temperature, stream 1) is calculated by an iterative process. Once TS1 has been found, PS1 is calculated directly

$$PS1 = P1 \times e^{(\phi 1 - \phi S1)}$$

If PS1 is greater than P2, then mixing is impossible and exit is made from the program.

e. At this point, PS1 (and hence PS2 = PS1) will be known. If PS1 was given as data, it still remains to calculate TS1, again by an iterative process. HS1 is then found using TRM. The velocity, density (RHO) and area can now be found by

$$V1 = \sqrt{2g_c Jx(H1 - HS1)}$$

$$RHO1 = PS1 / (R1 \times TS1)$$

$$A1 = W1 / (RHO \times V1)$$

Also, knowing F1 and TS1 allows the sonic velocity, CS1 for these conditions to be calculated. Hence the mach number may be determined

$$AM1 = V1/CS1$$

f. Since this is a DP situation, PS2 = PS1 and with methods similar to step e, TS2, HS2, V2, RHO2, A2, and AM2 may be calculated. A jump is then made to step k.

g. ODP calculations follow next. What will be known at this point are the inlet SV's and the areas calculated during the DP run. Because the engine will not initially be balanced, the inlet SV's will be such that the static pressures will not be equal and therefore an error is produced.

h. First, the inlet mass flow per unit area, WQA,

$$WQA = W1/A1$$

is calculated.

Next, a guess is made of the inlet Mach number, AM1. The static temperature, TS1, corresponding to AM1 is found iteratively. With the value of TS1 and the assumed value of AM1, V1 may be found. Now, the mass per unit area corresponding to these conditions may be calculated. If the initial guess of AM1 was correct, these two values of mass flow per unit area would be equal (within a tolerance). If this is not the case, the quadratic interpolation routine, AFQUIR, is used to interpolate towards a value of AM1 such that the two mass flows per unit area are equal. Hence, TS1 is determined. Once TS1 is known, PS1 may be calculated since P1 is also known.

j. The process of step h is now repeated for inlet 2 such that TS2, PS2, P2, and AM2 are found.

k. The outlet SV (stream 3) is now calculated as follows

$$A3 = A1 \div A2 \text{ (DP ONLY)}$$

$$W3 = W1 + W2$$

$$C = \frac{W1}{1+F1} + \frac{W2}{1+F2}$$

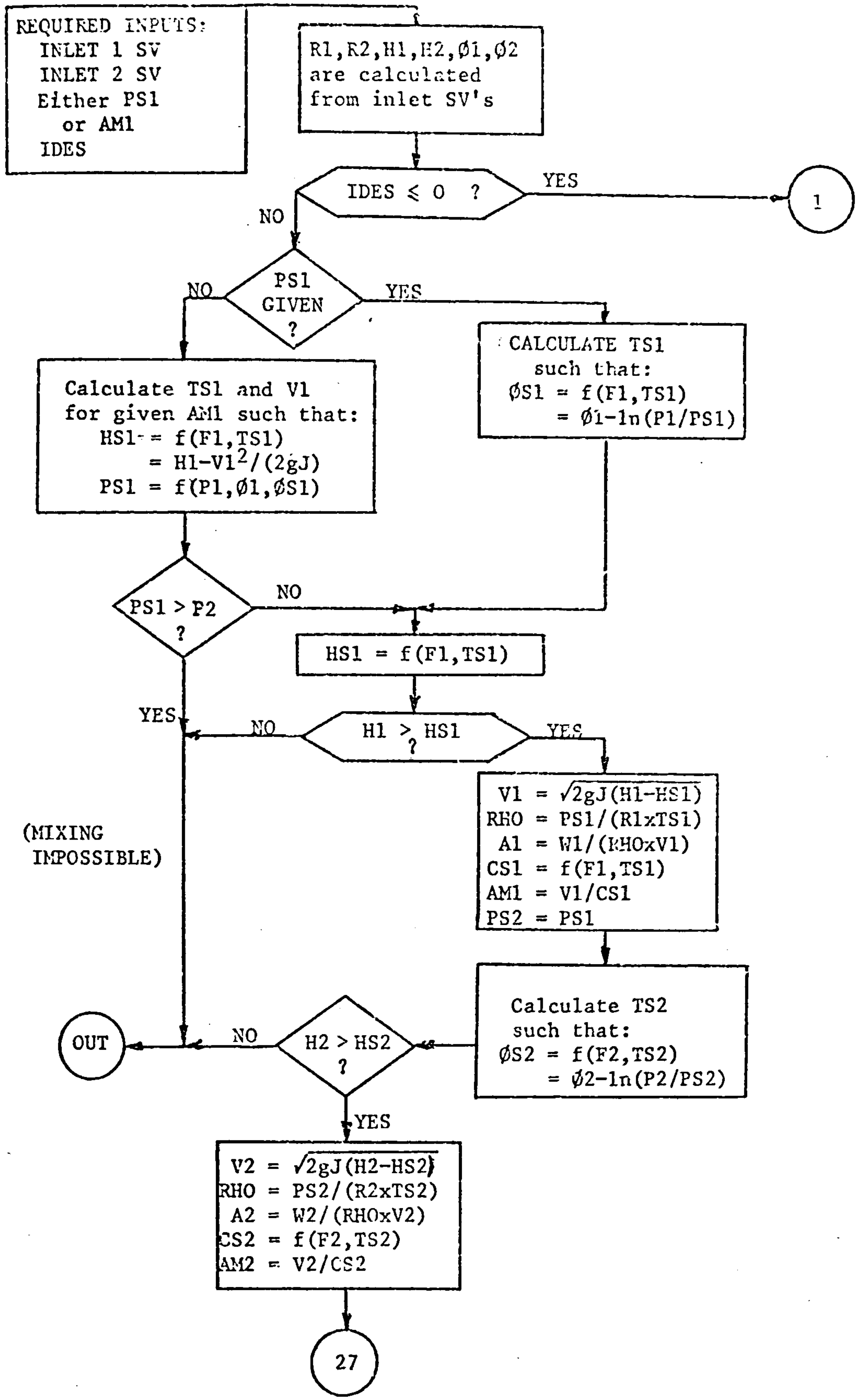
$$F3 = (W3 - C)/C$$

$$H3 = \frac{W1 \times (H1) + W2 \times (H2)}{W3}$$

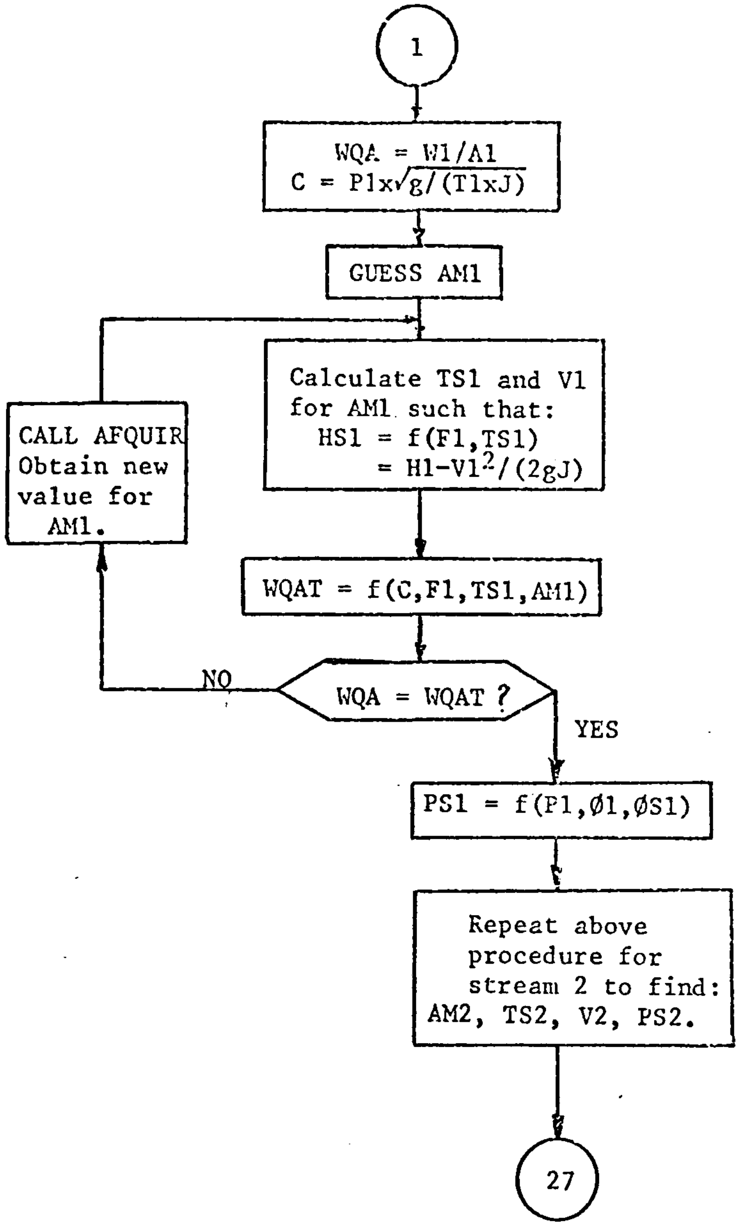
$$T3 = f(F3, H3)$$

Again, an iterative process is used to calculate TS3. Knowing TS3 allows PS3 and finally P3 to be calculated.

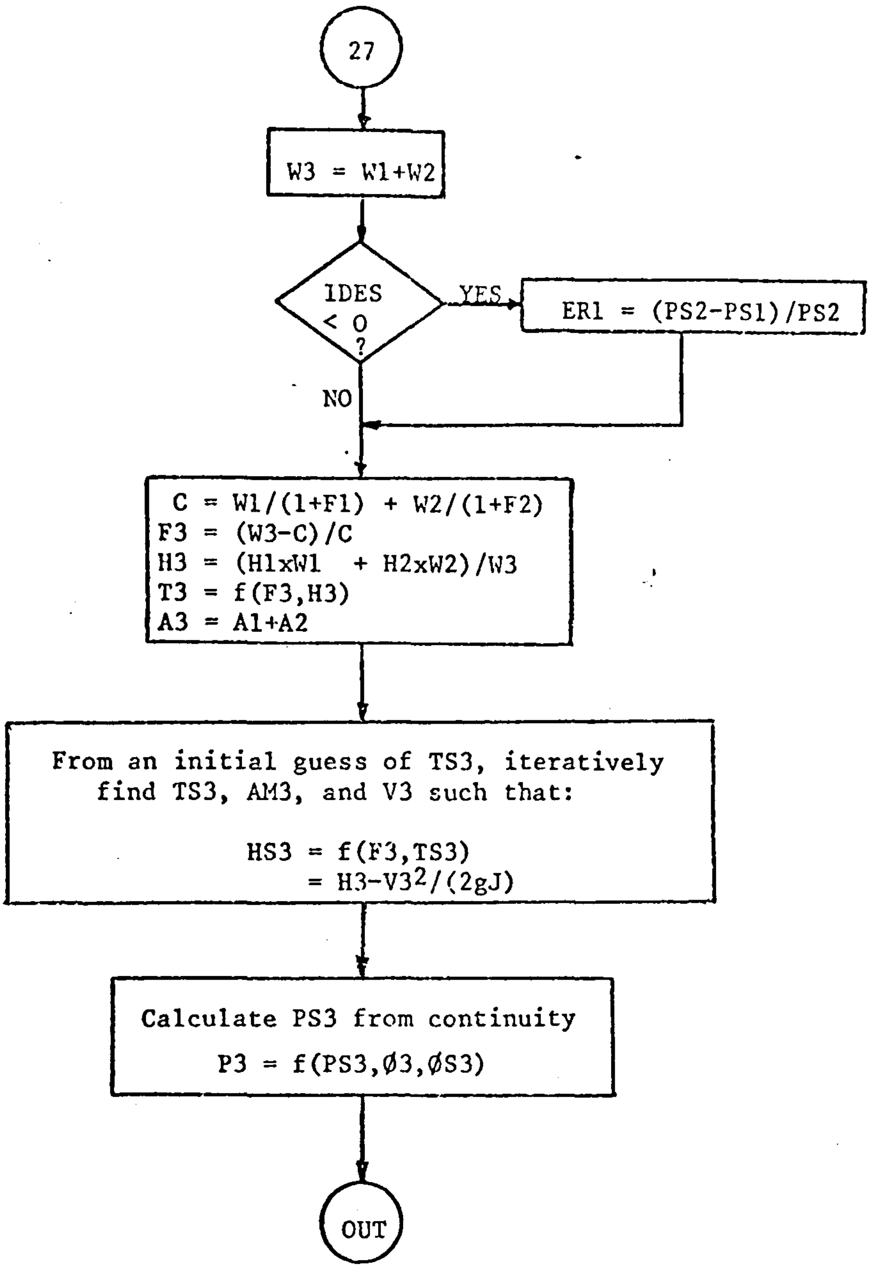
7. Brick 10 Flow Diagram



FLOW DIAGRAM, B10



FLOW DIAGRAM, B10 (CONTINUED)



FLOW DIAGRAM, B10 (CONTINUED)

A.11 B11, Duct or Pipe, With or Without Burning

1. Required BD:

   1. Switch for Burning (see below)
   2. Pressure loss, (PA - PB) / PA
   3. Combustion efficiency, η, (meaningful only if burning is occurring)

   Note: if BD(1) < 0. , brick represents only a duct, not equipped for burning

   BD(1) = 1. , brick represents a duct or jet pipe equipped for burning. However, no burning is taking place at this time.

   BD(1) > 2. , brick represents a duct or jet pipe in which burning is occurring.

2. Required SV Quantities:

   1. Inlet SV
   2. If BD(1) > 2. , then outlet total temperature, TB, must be specified.

3. Required EV Quantities: none

4. Quantities Calculated:

   1. Outlet SV
   2. If BD(1) > 1., the fuel mass flow (WFB) is calculated and appears as an EV quantity. Note that if BD(1) = 1. , WFB will be 0. (since no burning is taking place).

5. Errors Produced: none

6. Mechanics of Calculations:

   a. As may be inferred from above, this brick simulates two components. One is a duct or jet pipe with a pressure loss. The other is a duct or jet pipe, with pressure loss, but in which burning may occur.

   b. In both situations, the outlet total pressure, PB, is calculated by assuming that the pressure loss (ΔP) is proportional to the kinetic head at the duct or pipe inlet. Thus

$$\Delta P = K \times (WA)^2 \times (TA/PA)$$

where the constant of proportionality, K, is determined by

$$K = \left\{ \frac{\Delta P}{WA^2} \times \left(\frac{PA}{TA}\right) \right\}$$

where all the quantities within the parentheses are at the DP values.

c. If BD(1) < 2. , exit from the subroutine is now made. The only difference being that if BD(1) = 1. , then WFB is set to a small positive number (0.00001). If BD(1) < 1. , then WFB remains undefined. This is necessary to allow a space in the EV store <u>if</u> it is possible that burning will later occur.

d. If BD(1) > 2. , the outlet SV is calculated in a similar method to that of B3, combustion. The only difference is that in this case, no maps are used to account for varying η and it is assumed constant at the value specified in BD(3). The combustion process is calculated knowing the final temperature, TB. Initially, the increase in enthalpy of the <u>medium</u> is calculated:

$$\Delta H_{MEDIUM} = HB_{F=FA} - HA_{F=FA}$$

where $HB_{F=FA}$ represents the enthalpy at TB and fuel/air ratio FA.

Next, the effective calorific value of the fuel (ECV) is determined by

$$ECV = CV + \frac{HB_{F=0.}}{FS} - HB_{F=FS} \times \left(1 + \frac{1}{FS}\right)$$

where FS is the stoichometric fuel/air ratio and CV is the calorific value of the fuel.

By the definition of combustion efficiency, FB may be calculated by

$$FB = \frac{\Delta H}{ECV} \times \frac{1}{\eta}$$

The mass of air involved in the combustion process is calculated by

$$WA_{air} = (1 - FA) \times WA$$

Knowing this, WFB and hence WB may be calculated

$$WFB = FB \times (WA_{air})$$

$$WB = WA + WFB$$

At this point, the outlet SV is defined

e. Notes on Using Burning:

The use of duct or afterburning will usually cause some trouble with any downstream nozzles unless allowance is made for the burning. This is because with the increased temperature, the nondimensional mass flow, $W\sqrt{T}/P$, will also increase. Since the nozzles are likely to be running choked anyway, this increase will mean that the nozzle will be unable to pass the flow. To avoid this, the solution is that taken by all such engines; an increased exit nozzle area. There are two avenues open here, either complete floating in which the area is exactly matched to the flow, or a new fixed area which is large enough to take the flow.

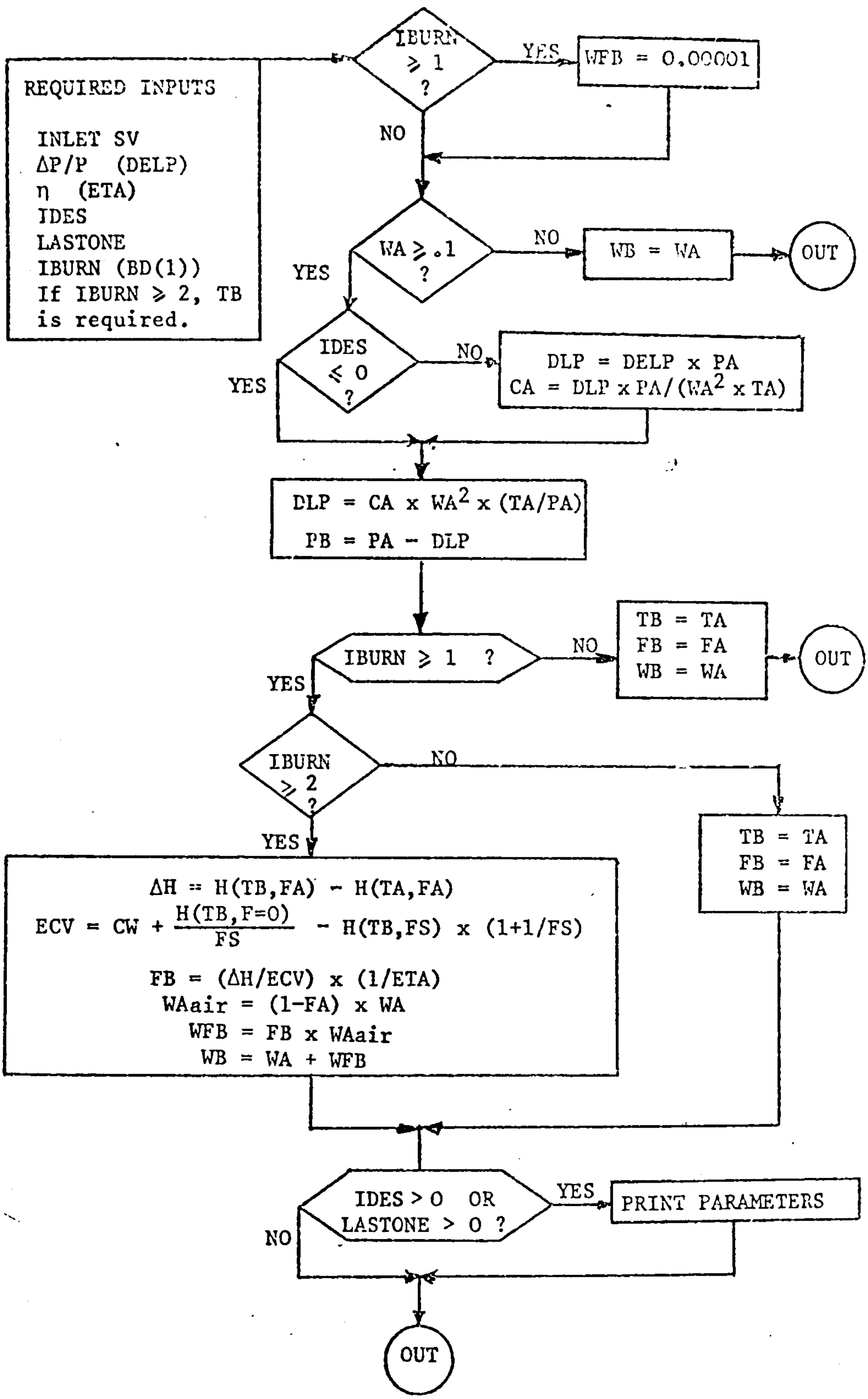
(1) Floating Nozzle

The nozzle exit area may easily be floated by BD(1) of the appropriate nozzle brick (B5 or B12). The procedure to be followed using this program is to first make an engine run at the ODP condition, but with no burning (BD(1) = 1.) and the nozzle area fixed. In this run, the mass flow is determined and the components balanced to conform to the ODP condition. Next, a run is made with the nozzle floating, burning occurring (BD(1) = 2.), and TB specified. The nozzle area will vary to match the flow and therefore no error will be produced in the nozzle brick. Also, since the components were all balanced in the previous run, all errors will be within tolerance and a balanced engine print out will occur.

(2) Larger Fixed Nozzle Area

Using this method, a larger area nozzle is specified with the other ODP data. Additionally, TB and BD(1) = 2. are input. The engine will then attempt to balance as in any ODP case. However, this method is not recommended as usually one would not know the necessary larger area and the possibility exists that the engine will prove to be difficult to balance.

7. Brick 11 Flow Diagram



Flow Diagram, B11

A.12 B12, Convergent/Divergent Nozzle

1. Required BD:

    1. Floating areas switch:
    if BD(1) > 0. , throat and exit areas float to satisfy continuity and obtain optimum expansion to the ambient static pressure

        if BD(1) < 0. , throat and exit areas are fixed

    2. Throat area. BD(2) is normally set equal to -1. If in an ODP case, it is desired to change the throat area, then BD(2) is set equal to the new value.

2. Required SV Quantities:

    1. Inlet SV
    2. Free stream SV

3. Required EV Quantities: none

4. Quantities Calculated:

    1. Outlet SV
    2. Area, mach numbers, and velocities of throat and exit
    3. Nozzle coefficient
    4. Gross thrust (appears as EV quantity).

5. Errors Produced:

    If the nozzle areas are fixed, one error is produced:

    1. $ER1 = (PA_{REQUIRED} - PA_{ACTUAL}) / PA_{REQUIRED}$

6. Mechanics of Calculations:

    a. In the DP case, this Brick calculates the throat area such that sonic flow (Mach number = 1.0) occurs at the throat. The exit area is calculated assuming an isentropic expansion to the ambient static pressure. The calculations here are similar to those for a convergent nozzle, B5.

b. In the ODP situation, the throat and exit areas are fixed and the Brick produces an error if the engine is unbalanced. In performing ODP calculations, it is first assumed that the flow at the throat is sonic (choked condition) and the corresponding exit static properties are calculated assuming an isentropic expansion to the exit area. If the exit static pressure is less than the ambient static pressure, the the assumption of sonic flow at the throat is invalid. The nozzle then is not operating choked and the throat velocity is subsonic. In this situation, the exit static pressure must equal the ambient static pressure, and the exit static temperature and velocity are found iteratively.

c. If the assumption of sonic throat velocity is valid, then further assumptions are required. The first is that isentropic expansion to the ambient static pressure occurs. The exit area required to pass such a flow is then determined. Next, the exit static temperature and mach number necessary to pass the flow through the actual exit area are calculated assuming an isentropic expansion to this actual area.

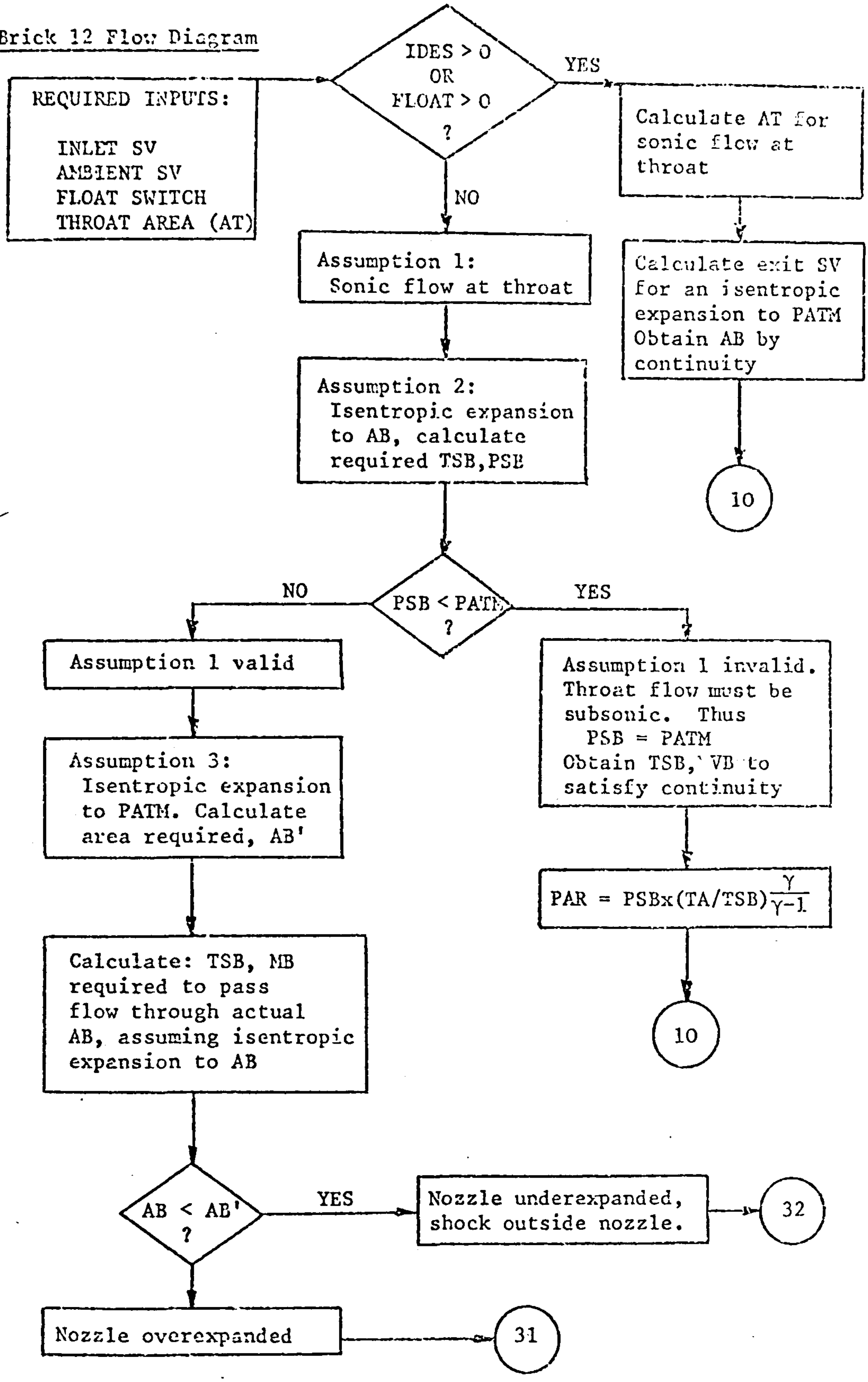
d. If the actual exit area is less than the above required area, the nozzle is underexpanded and a shock occurs outside the nozzle. However, if the actual area is greater than the required area, the nozzle is overexpanded and the shock may occur either inside or outside the nozzle.

e. If the nozzle is overexpanded, it is first assumed that the shock stands at the nozzle exit and the static pressures on both sides of the shock are calculated. If the static pressure after the shock is less than the ambient value, then the shock must occur outside the nozzle. The remainder of the solution for this possibility and that of an underexpanded nozzle are identical. The exit mach number and static temperature are those found assuming an isentropic expansion to the exit area, as in step c. Knowing these two quantities enables the exit velocity to be determined and the exit static pressure is that required to satisfy continuity.

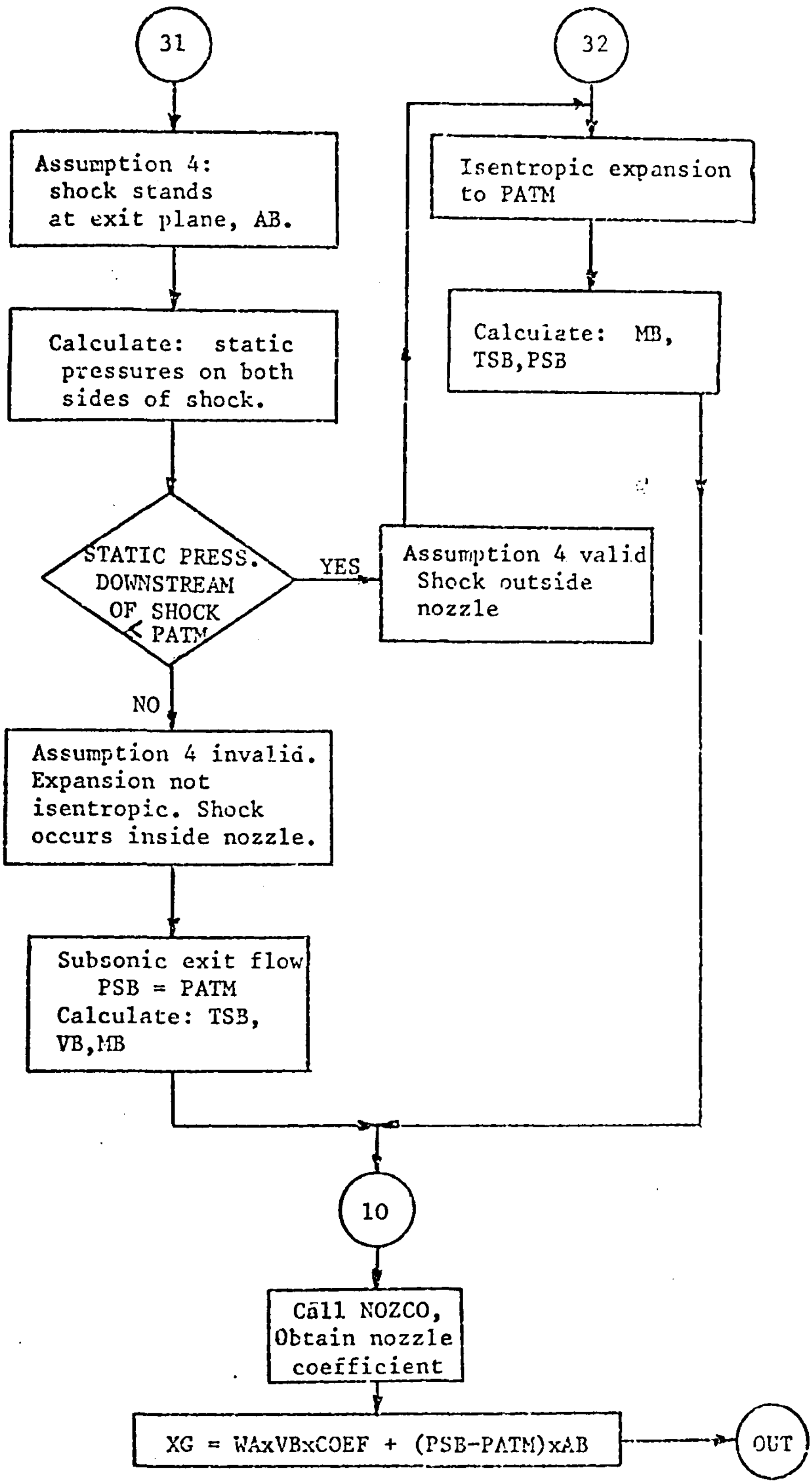
f. However, if for an overexpanded nozzle, the shock occurs inside the nozzle, then the assumption of an isentropic expansion is invalid. Since in this case the exit flow must be subsonic, the exit static pressure equals the ambient static pressure. The corresponding static temperature, mach number, and velocity are then calculated by an iterative process.

g. Once the exit conditions are known, then subroutine NOZCO is called to obtain the nozzle coefficient and the gross thrust produced by the nozzle is calculated.

Brick 12 Flow Diagram



FLOW DIAGRAM, B12



FLOW DIAGRAM, B12 (CONTINUED)

A.13 B13 Incrementation of a Design Point Parameter

1. Description:

This brick is used in DP studies when it is desired to study the effect of changing one of the component parameters on the engine results. Provision is made within the MASTER program segment for incrementing up to five such component parameters (which may be either BD or SV quantities). An example of its use would be to study the effects of various compressor pressure ratios and turbine inlet temperatures on the final engine parameters.

2. Input Requirements

a. This brick requires inputs in the form of BD, but it may be seen to resemble B9, arithmetic. As has been mentioned, up to five parameters may be incremented. The number of uses of this brick numbered consecutively from the start of the CW list is put in columns 4 and 5 of the CW card.

b. Required BD:

1. BD(1) < 0 if item to be incremented is a BD item
   BD(1) > 0 if the item is a SV item, in which case BD(1) = SV number

2. BD(2) = BD number if BD(1) < 0
   BD(2) = SV item number if BD(1) = SV number

3. Initial value of item

4. Incremental value

5. Final value of item.

3. Mechanics of Operation

a. The CW using this brick is always placed immediately preceeding the CW which will use the changed value

b. When the brick is entered, the incremental value is added to the existing value of the item specified and hence it is set to a new value. Control then passes back to the main program segment and the run through the engine continues until all CW's are used and a print out of the engine parameters occurs.

c. The MASTER segment of the program then transfers control back to the incrementation CW, where the value of the specified item is incremented again. The run continues in this manner until the incremented value eventually is equal to or greater than its specified final value.

d. If there is only one incremental loop the final exit from the program is made after the incremented value has reached its final value and the corresponding engine print out is made.

e. If there are two loops, the process is analogous to a nested do loop in FORTRAN. After the value of the item specified in the inner loop reaches its final value, it is reset to its initial value while the value of the outer loop is incremented. Finally when both items reach their final values, exit from the program is made.

4. <u>Example</u>

a. Perhaps the easiest way to demonstrate the procedure is an example. A full example is given in Appendix C, example 1.

A.14 B14 Free Turbine

1. Required BD:

    1. Shaft rotational speed as a percent of design (PCN)
    2. Design power produced
    3. Turbine flow function ($TF = WA\sqrt{TA}/PA$) *
    4. CN = "non dimensional" speed = $PCN/\sqrt{TA}$ *
    5. isentropic efficiency

    Plus a component map to represent the turbine characteristics

    * See paragraph 6b of Appendix A, section 4 for clarification

2. Required SV Quantities:

    1. Inlet SV

3. Required EV Quantities:

    1. Mass of fuel burned (WFB)

4. Quantities Calculated:

    1. Outlet SV
    2. Scaling factors for use in ODP cases
    3. Shaft power specific fuel consumption

5. Errors Produced

    1. $ER1 = (TF_{ACTUAL} - TF_{MAP})/TF_{ACTUAL}$
    2. $ER2 = (\Delta H_{\substack{POWER\\PRODUCED\\(BD(2))}} - \Delta H_{MAP})/\Delta H_{\substack{POWER\\PRODUCED}}$

6. Comments

    B14 merely calls B4, compressor turbine Brick, with the compressor work set equal to zero. All errors and variables are identical to B4, as are the mechanics of calculations.

    In ODP calculations, either BD(1) or BD(2) must be specified as a variable in addition to the usual turbine variable, TF. The item not specified as a variable <u>must</u> be specified as one of the ODP conditions. Thus, in an ODP situation, it is possible to calculate the output power at specific gas generator conditions and free turbine speed, <u>or</u> the speed is calculated for a given power output.

Summary of Outputs and Required Inputs for Bricks

BRICK	DESC.	STATION VECTORS	BRICK DATA	EV's REQUIRED	EV's PRODUCED	ERRORS	POSSIBLE VARIABLES	REMARKS
001	INLET, ATMOSPHERE PROPERTIES	INLET, OUTLET	1) ALT 2) ISA DEV 3) M 4) η	NONE	1) XR	NONE	NONE	If η is not specified, η = -1.
002	COMPRESSOR (OR FAN)	INLET, OUTLET	1) Ƶ 2) PCN 3) PR 4) η 5) ERROR?	NONE	1) COMWK	1) $\frac{WB-WA}{WA}$ *	Ƶ, PCN	* If BD(5) > 0. , error produced. If BD(5) ≤ 0. , no error produced.
003	COMBUSTION CHAMBER	INLET, OUTLET	1) ΔP/PA 2) η 3) WFB *	NONE	1) WFB	NONE	TB	* Either TB or WFB required as input. If WFB not input, BD(3) ≤ 0.
004	COMPRESSOR TURBINE	INLET, OUTLET	1) AUXWK 2) η 3) TF 4) CN 5) COMP. NUMBER	1) COMWK	NONE	1) $\frac{TF-TF_{MAP}}{TF}$ 2) $\frac{\Delta H-\Delta H_{MAP}}{\Delta H}$	TF	BD(5) = number of compressor to which turbine is coupled.
005	CONVERGENT NOZZLE	INLET, OUTLET, AMBIENT	1) FLOAT ?	NONE	1) XG	1) $\frac{PAR-PA}{PAR}$ *	NONE	If BD(1) > 0. , exit area floats. If BD(1) ≤ 0. , exit area fixed. * PAR=PA required to pass mass flow

Outputs and Required Inputs for Bricks (Continued)

BRICK	DESC.	STATION VECTORS	BRICK DATA	EV's REQUIRED	EV's PRODUCED	ERRORS	POSSIBLE VARIABLES	REMARKS
006	PERFORMANCE CALCS.	SEE REMARKS *	NONE	1) XG 2) XR 3) WFB	1) XN 2) SFC 3) SP. THRUST	NONE	NONE	* for SV's, put SV numbers of engine air inlets
007	MASS/PRES. CHANGE	INLET, OUTLET	1) λ1 2) ΔW 3) λ2 4) ΔP	NONE	NONE	NONE	λ1 FOR BYPASS ENGINE	Solves equations WB=λ1xWA + ΔW PB=λ2xPA + ΔP
008	SIMPLE MIXING	SA=INLET A SB=INLET B SC=OUTLET	NONE	NONE	NONE	NONE	NONE	Performs simple mixing. Pressure of main stream remains unchanged
009	ARITHMETIC	SA=OPERATION SB=-1 IF OP1 IS EV =0 IF OP1 IS BD =SV No. IF OP1 IS A SV SC= EV No. = BD No. or= SV ELEMENT No.	1)=-1 IF OP2 IS EV =0 IF OP2 IS BD =SV No. IF OP2 IS A SV 2)= EV No. = BD No. or= SV ELEMENT No. 3)= -1	1)=-1 IF OP3 IS EV =0 IF OP3 IS BD =SV No. IF OP3 IS A SV 2)= EV No. = BD No. or= SV ELEMENT No. 3)= -1	1)POSSIBLY R1(RESULT)	NONE	NONE	OPERATION 01 R1=OP1+OP2 02 R1=OP1-OP2 03 R1=OP1xOP2 04 R1=OP1/OP2 05 OP1=OP2+OP3 06 OP1=OP2-OP3 07 OP1=OP2xOP3 08 OP1=OP2/OP3

Outputs and Required Inputs for Bricks (Continued)

BRICK	DESC.	STATION VECTORS	BRICK DATA	EV's REQUIRED	EV's PRODUCED	ERRORS	POSSIBLE VARIABLES	REMARKS
010	COMPLETE MIXING	SA=INLET A SB=INLET B SC=OUTLET	1) FAN No. 2) SWITCH 3) VALUE	NONE	NONE	1) $\frac{PSA-PSB}{PSA}$	NONE	BD(1) = number of compressor, providing bypass flow. If BD(2) > 0. , BD(3) = MA If BD(2) ≤ 0. , BD(3) = PSA
011	DUCT; WITH OR WITHOUT BURNING	INLET, OUTLET	1) SWITCH 2) $\Delta P/PA$ 3) η	NONE	1) WFB, ONLY IF BD(1) ≥ 1.	NONE	NONE	If BD(1)=0. , no burning ever. If BD(1) = 1. , burning later. If BD(1) > 1. , burning now, TB required.
012	CONVERGENT DIVERGENT NOZZLE	INLET, OUTLET, AMBIENT	1) FLOAT? 2) THROAT AREA	NONE	1) XG	1) $\frac{PAR-PA}{PAR}$ *	NONE	If BD(1) > 0. , areas float. If BD(1) < 0. , areas are fixed. BD(2) > 0. only to specify area for ODP study. *PAR=PA required to pass mass flow

Outputs and Required Inputs for Bricks (Continued)

BRICK	DESC.	STATION	BRICK DATA	EV's REQUIRED	EV's PRODUCED	ERRORS	POSSIBLE VARIABLES	REMARKS
013	LOOPING BRICK	SA = No. of uses of Brick from start of CW sequence	1) $\leq$ 0. IF ITEM IS BD $>$ 0, = SV No. 2) = BD No. IF BD(1) $\leq$ 0. = SV ELEMENT No. IF BD(1) $>$ 0. 3) = INITIAL ITEM VALUE 4) = VALUE OF INCREMENT 5) = FINAL VALUE	NONE	NONE	NONE	NONE	This Brick must be placed immediately in front of the CW which uses the incremented value. The use of this Brick causes successive engine runs to be made with the specified item being changed until the final value is reached. It is only applicable to Design Point work.
014	FREE TURBINE	INLET, OUTLET	1) PCN 2) POWER 3) TF 4) CN 5) η	1) WFB	NONE	1) $\frac{TF-TF_{MAP}}{TF}$ 2) $\frac{\Delta H-\Delta H_{MAP}}{\Delta H}$	TF, PCN,	Either PCN or POWER is to be a variable, but not both.

APPENDIX B

Appendix B, Program MASTER Segment

1. As the name implies, the MASTER segement is the central controlling portion of the program. It is this segment which directs the reading of data, sequence of all operations, engine balancing calculations, and output of engine results.

2. Switching Terms

As may be seen from the following flow diagram, the switching arrangements are quite involved. To simplify interpretation, the various switching devices are given below:

```
     DES > 0, DP case  } = IDES in subroutine
         ≤ 0, ODP case }

FINISHED = 0 initially                          } = LASTONE in
         > 0 engine run has finished            }   subroutines
           (balanced engine in ODP case), and   }
           printout will occur                  }

    IDES > 0 DP study
         ≤ 0 ODP study

  IKOUNT = -1 initially
         > 0 if BD and SV elements are to be set
           to the value of an appropriate variable

   KOUNT = 0 initially.  If ODP study, ODP conditions
                         are read when KOUNT = 0
         = 1 errors produced by bricks are stored
           as BASE ERRORS in array BERR(K)
         = 2 errors are stored in array  EROR (L,K)
         = 3 after new values of variables have been
           calculated in "error solving process".
           If KOUNT = 3, errors are stored in BERR(K).

  KUNITS = 1 British units to be used
         = 2 SI units to be used

 LOOPING = -1 initially
         > 0 if B13(incrementation brick) is used

  MISMAT = 0 initially
         > 0, counter used in "error solving process"

   NOMIS = 0 initially
         > 0, used in "error solving process"

     NOW = -1 initially (most common value)
         > 0, used in conjunction with IKOUNT > 0 to
           set BD to new value or reset it to previous value
```

NUMAP = 0 initially
> 0, is a counter for number of times REMAP > 0 in an attempt to redefine a new component map

REMAP = 0 initially
> 0, operating point is off a component map (occurs during use of B4 or B10). Selects a new shaft speed or turbine inlet temperature to try and correct the situation. Only has meaning in ODP cases.

NCODE = 7, error has occurred in a subroutine. Program stops, last complete SV's are printed, and a diagnostic printing also usually occurs. If NCODE $\neq$ 7, no errors have arisen.

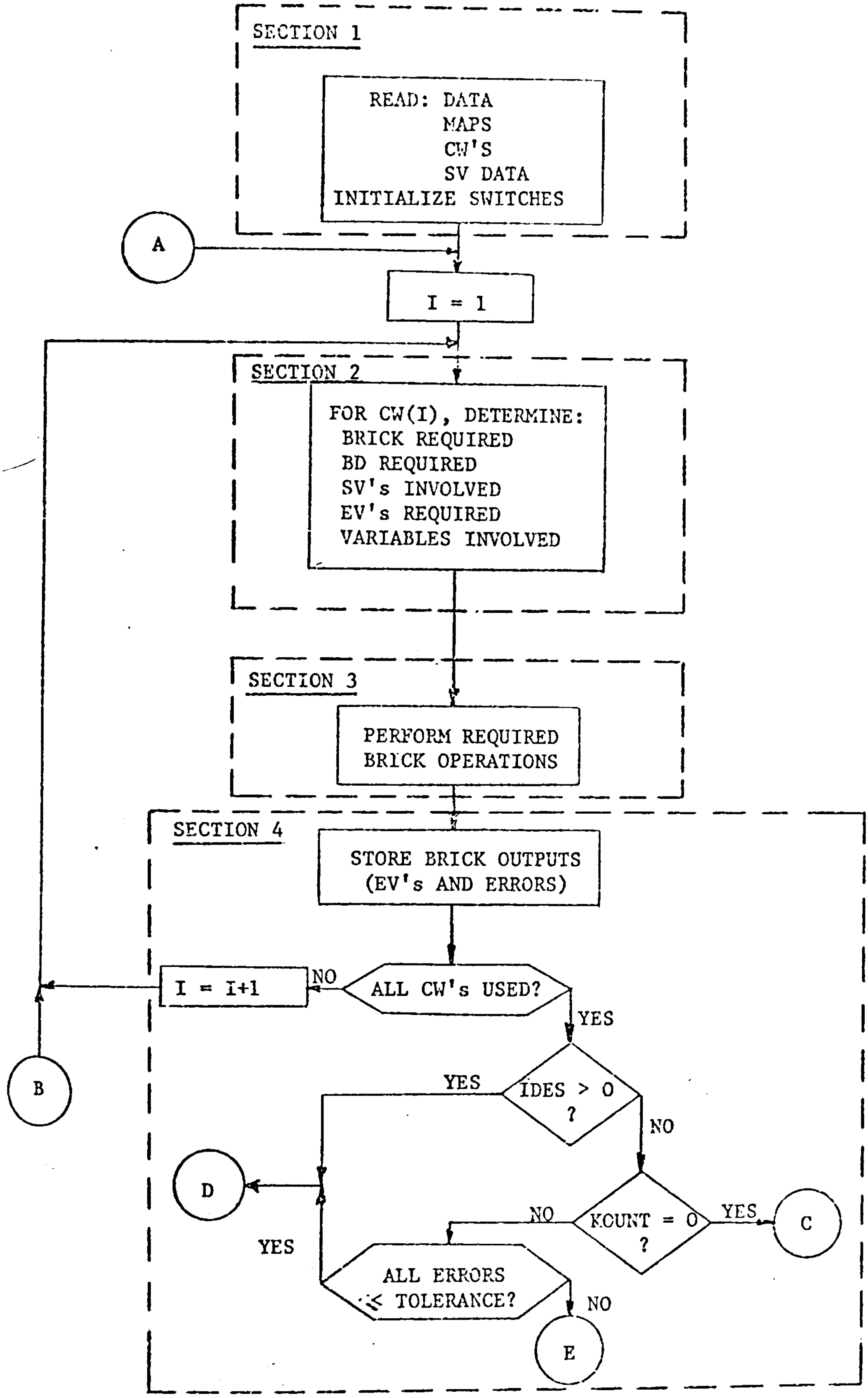
3. Other symbols which may be of use in following the program logic of the MASTER segment are:

NEXT = counter used in EV array
NCOMP = counter for number of compressor
NTURB = counter for number of turbine
NDUCT = counter for number of duct
NBURN = counter for number of duct with burning
NCONDI = counter for number of convergent/divergent nozzle
NCONVRG = counter for number of convergent nozzle
NLOOP = counter for number of passes through an engine before a balance occurs in an ODP case.

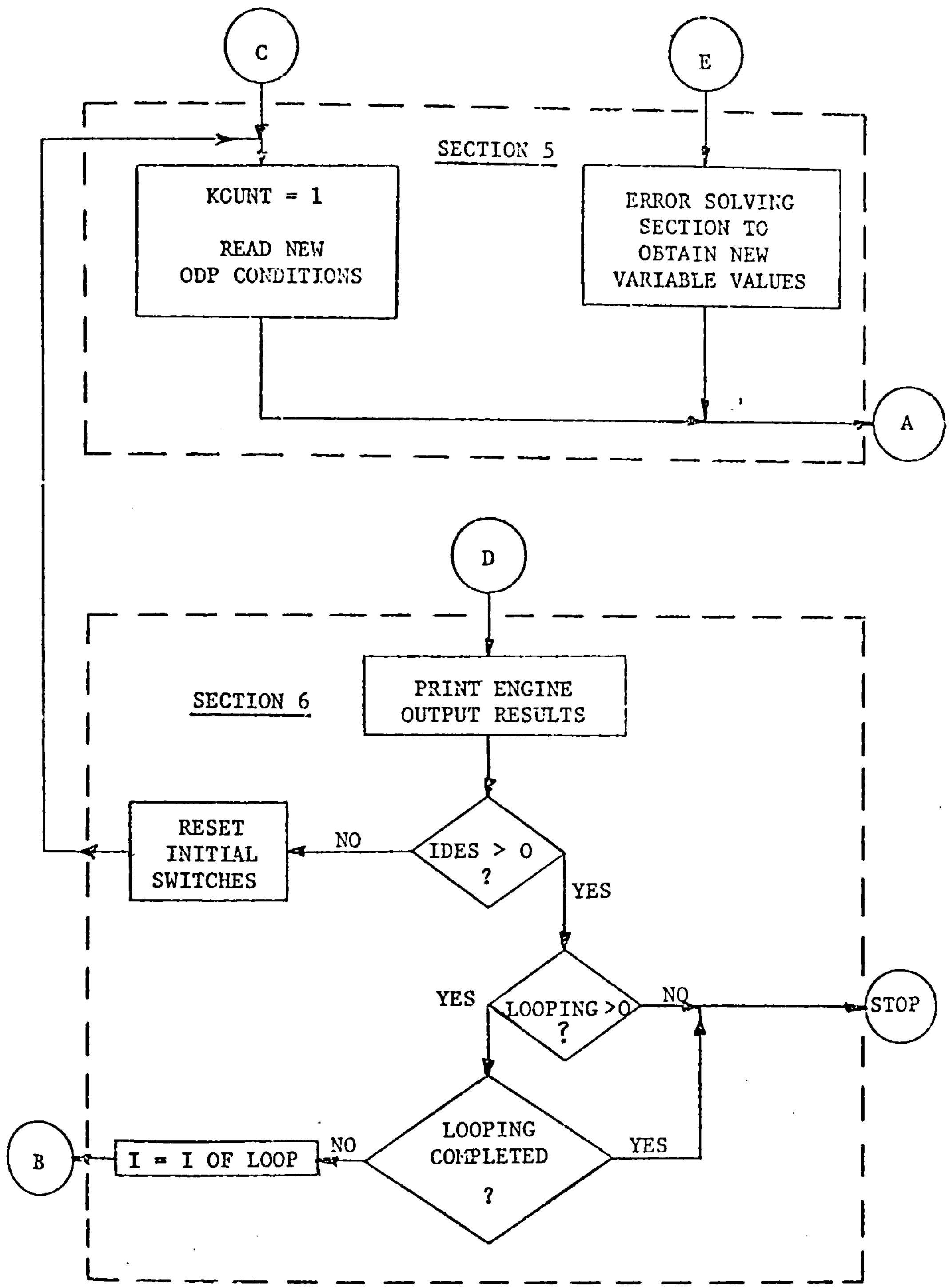
KN = number of variables to be used (equals number of errors)
NI = number of CW's to be input
ND = number of BD to be input
ISV = number of SV elements input as data
R1,R2,R3,R4 = brick results which are to be stored as EV
NER = number of errors brick produces
ER1,ER2 = errors produced by brick
NCMAP = number of compressor maps to be input
NTMAP = number of turbine maps to be input

4. To simplify the flow chart representation of the MASTER segment, it has been divided into six sections. The relationship of these sections is shown by the diagrams on the next two pages. Basically, the content of each section is as follows:

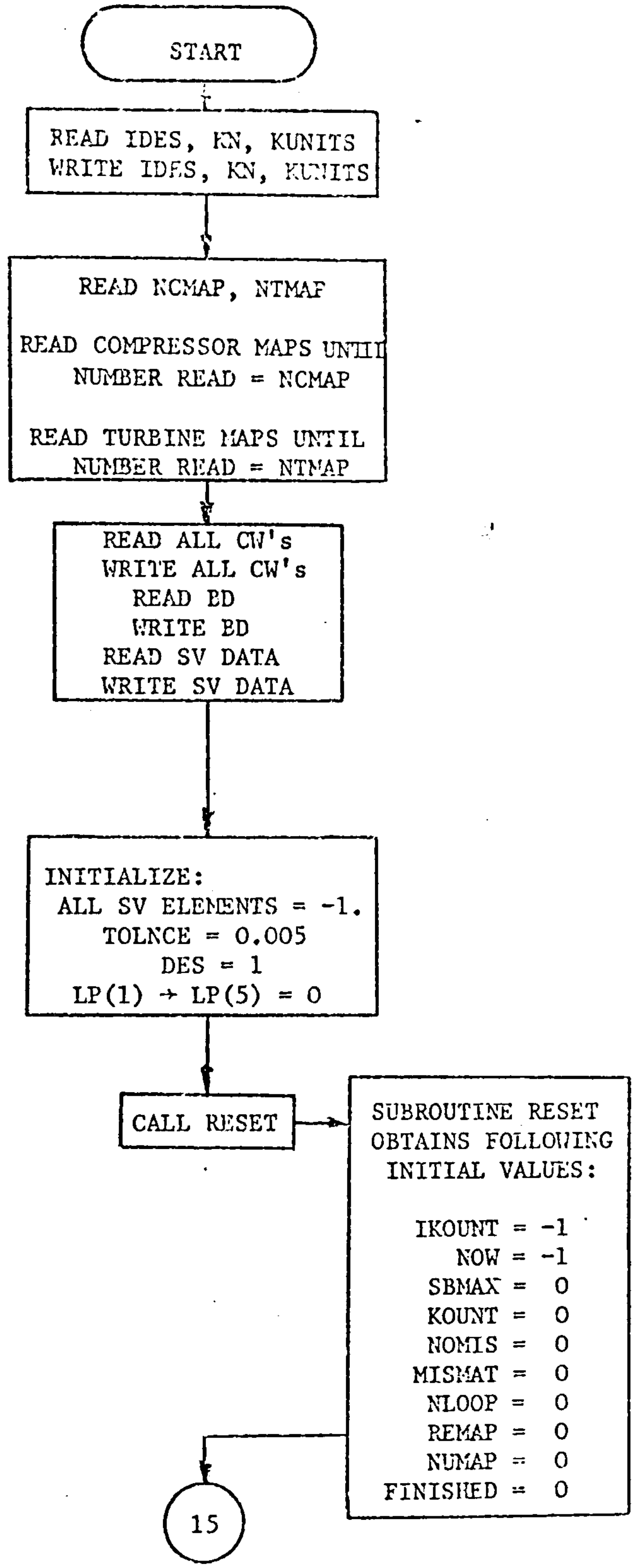
Section 1; program data input
Section 2; interpretation of codeword
Section 3; selection and employment of Brick subroutine
Section 4; analysis and storage of Brick outputs
Section 5; error solving section (to obtain new values for variables)
Section 6; engine outputs



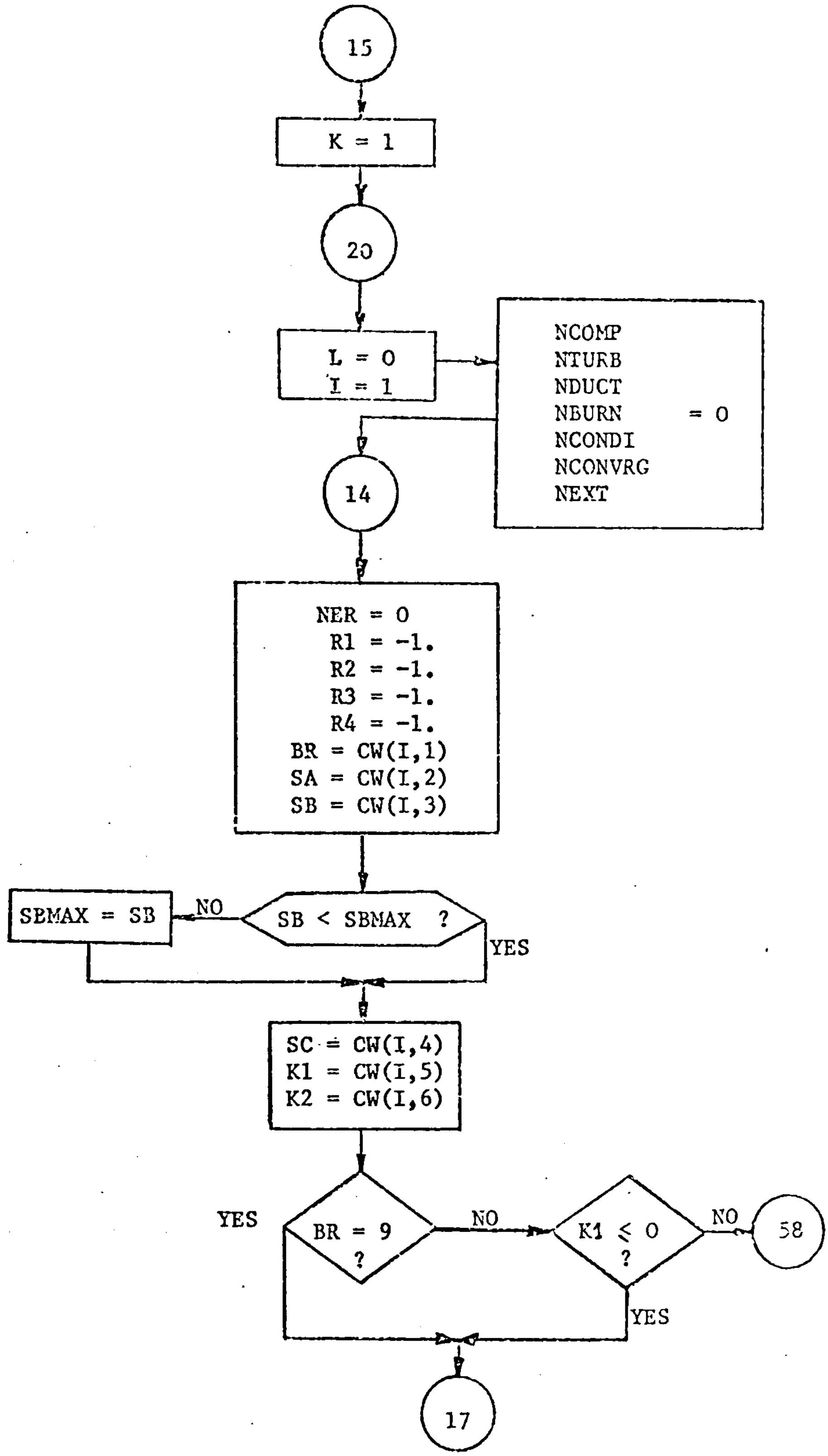
MASTER SEGMENT, SHOWING SECTIONS



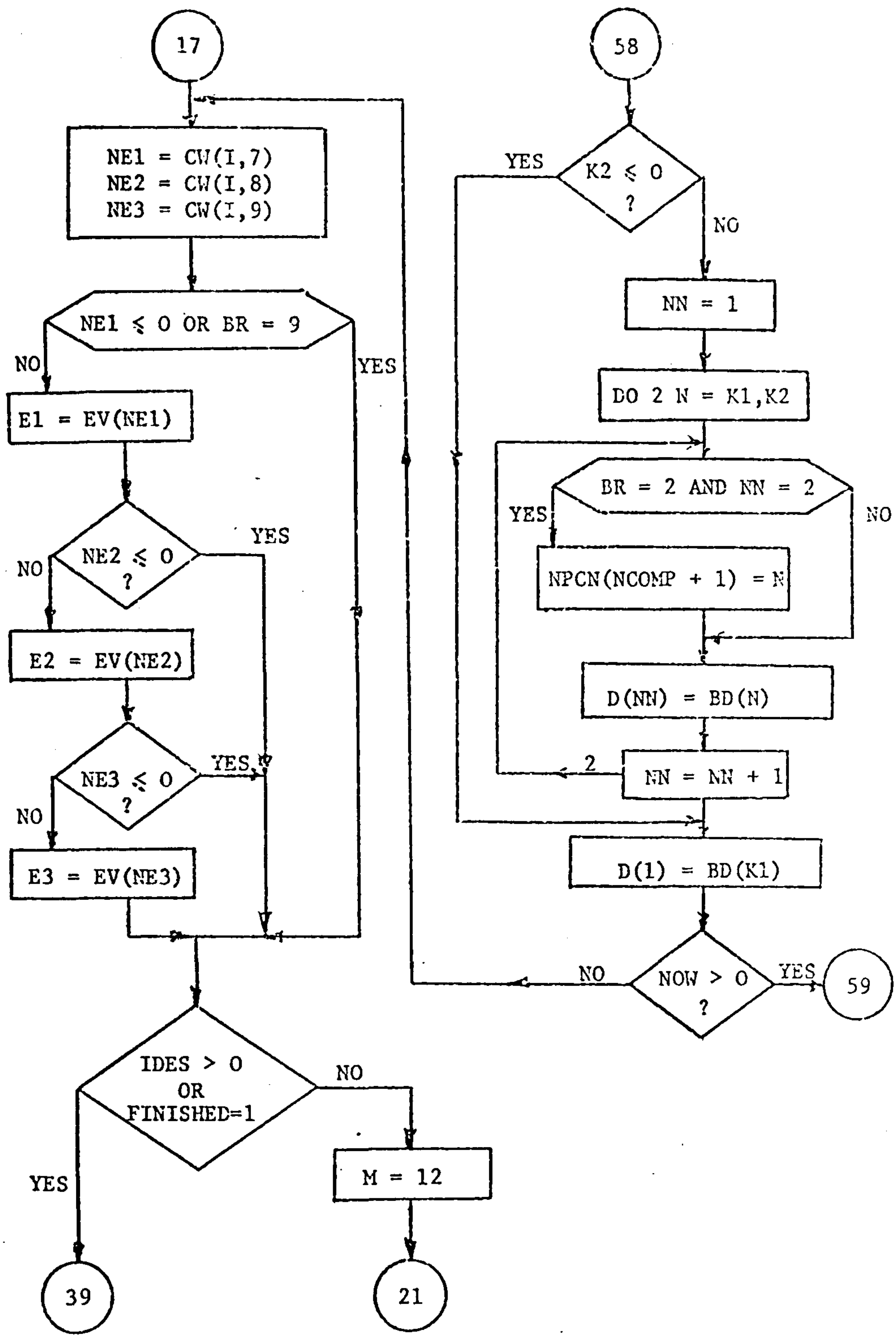
MASTER SEGMENT, SHOWING SECTIONS (CONTINUED)



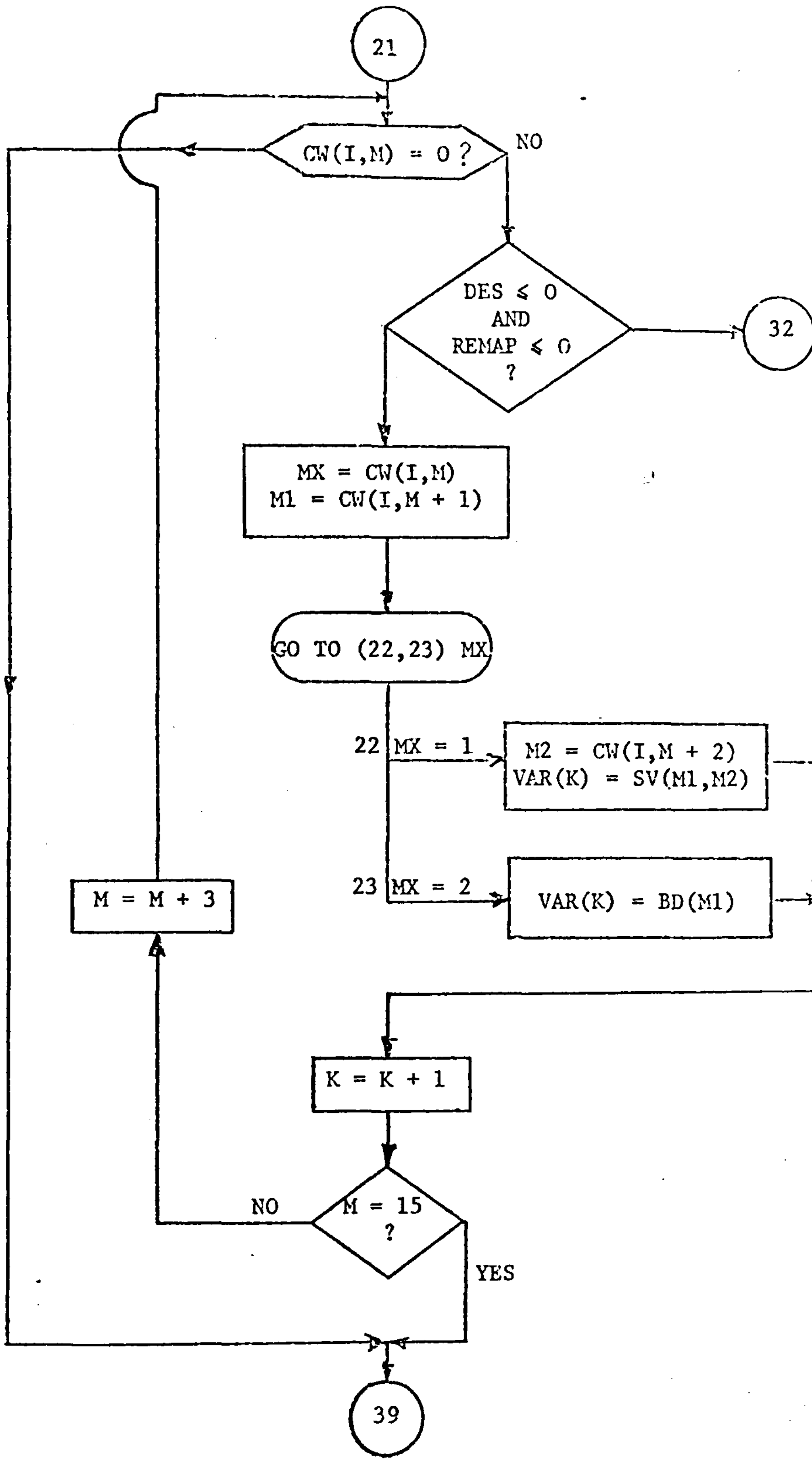
MASTER SEGMENT, SECTION 1



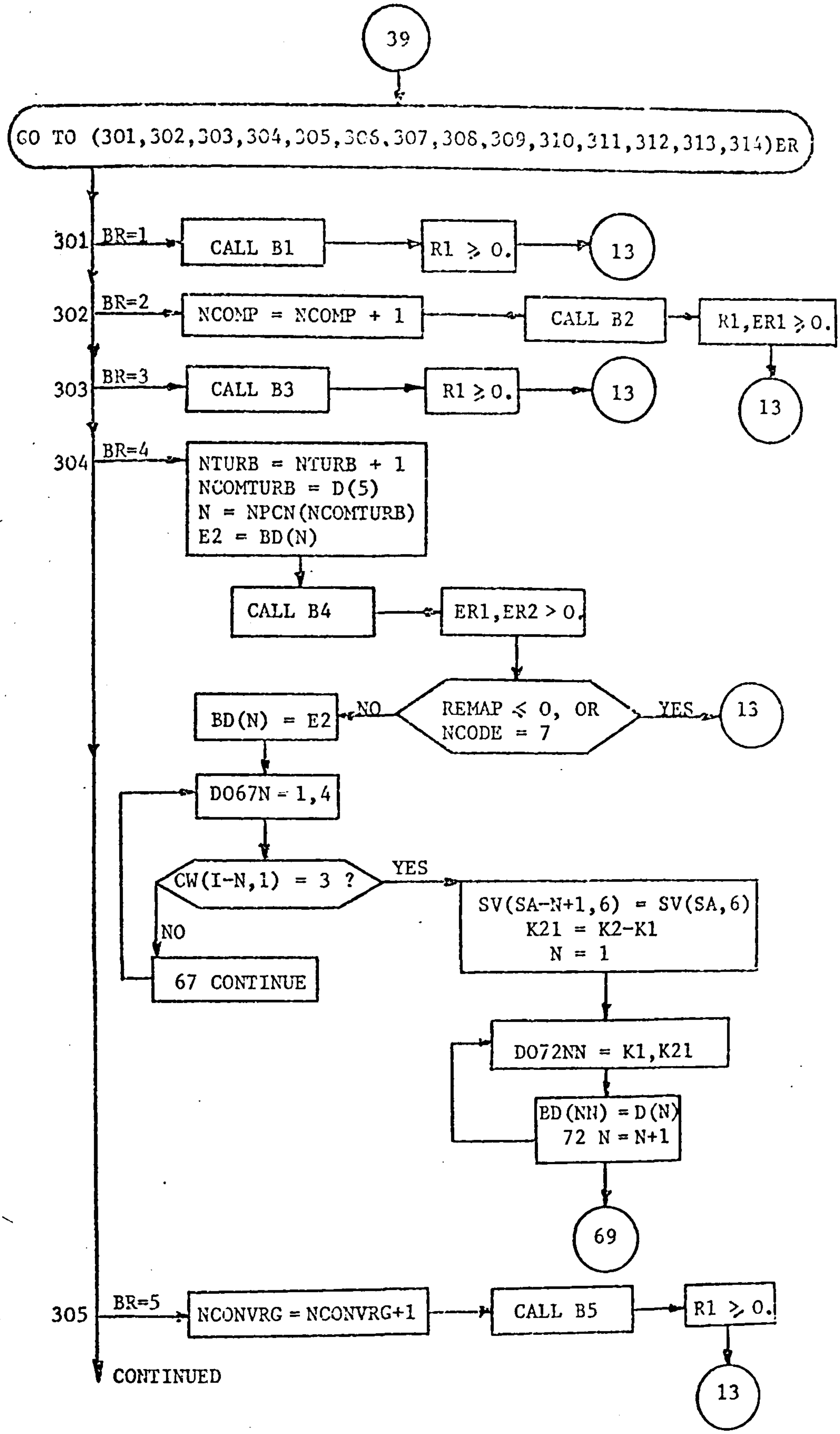
MASTER SEGMENT, SECTION 2



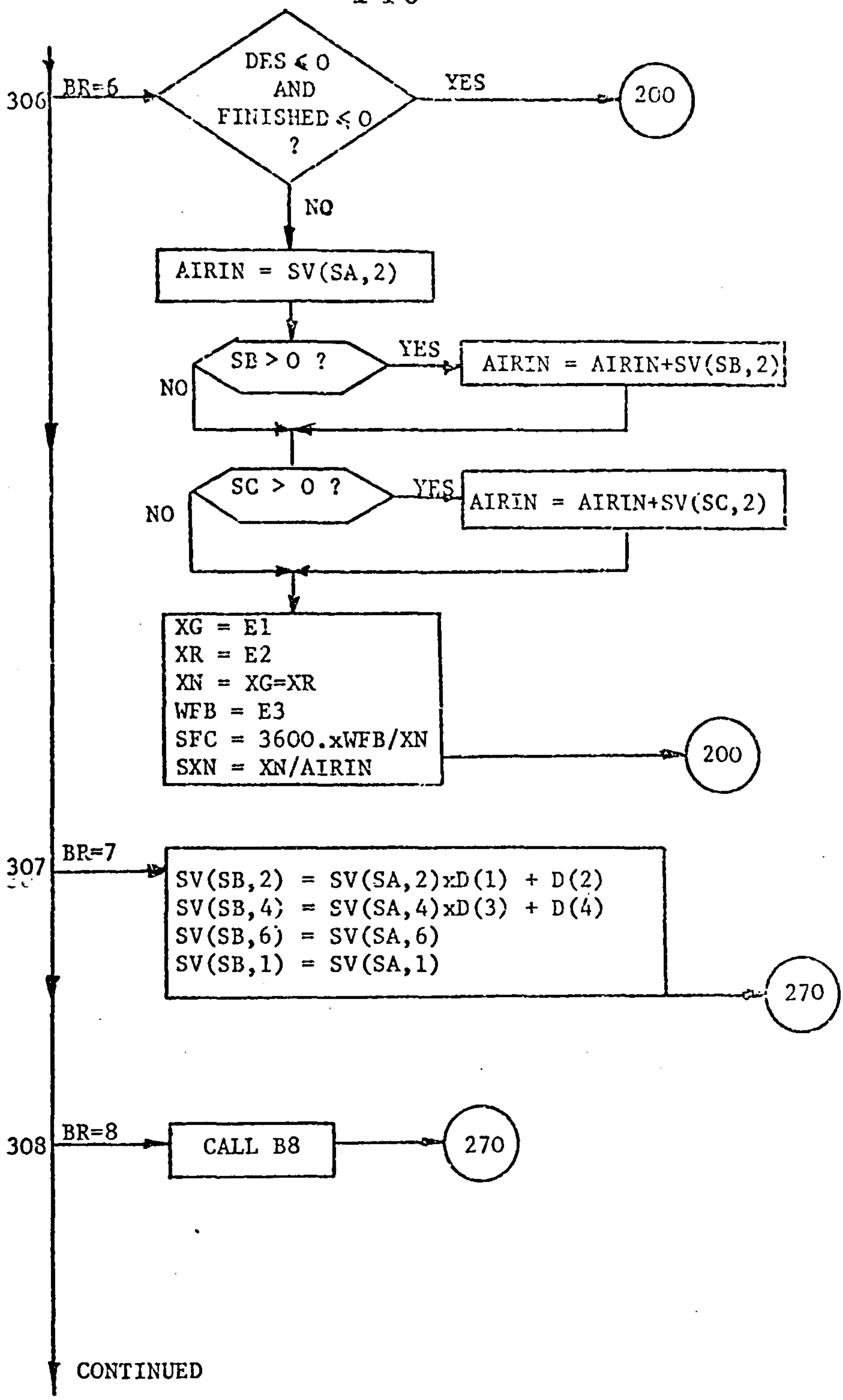
MASTER SEGMENT, SECTION 2 (CONTINUED)



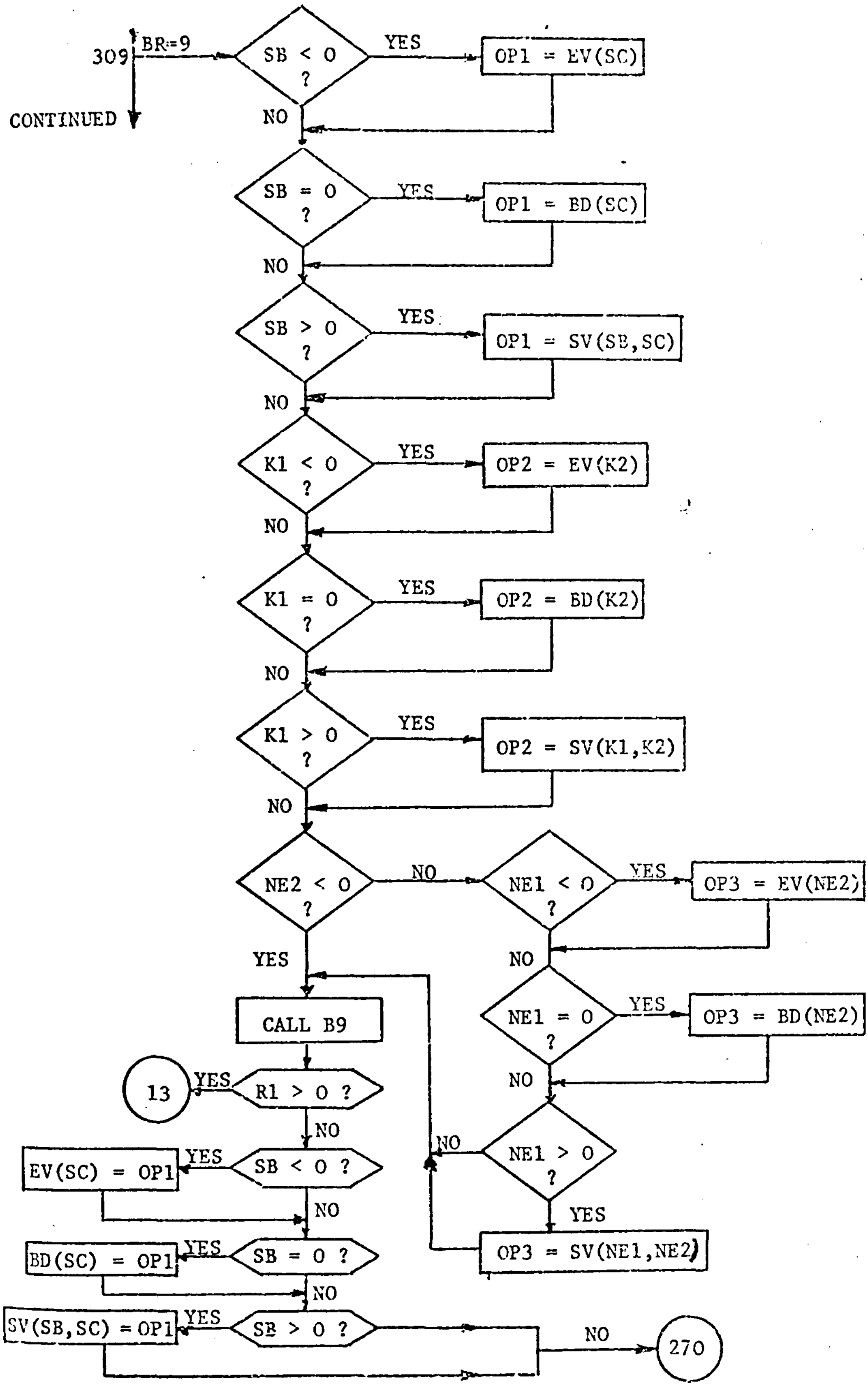
MASTER SEGMENT, SECTION 2 (CONTINUED)



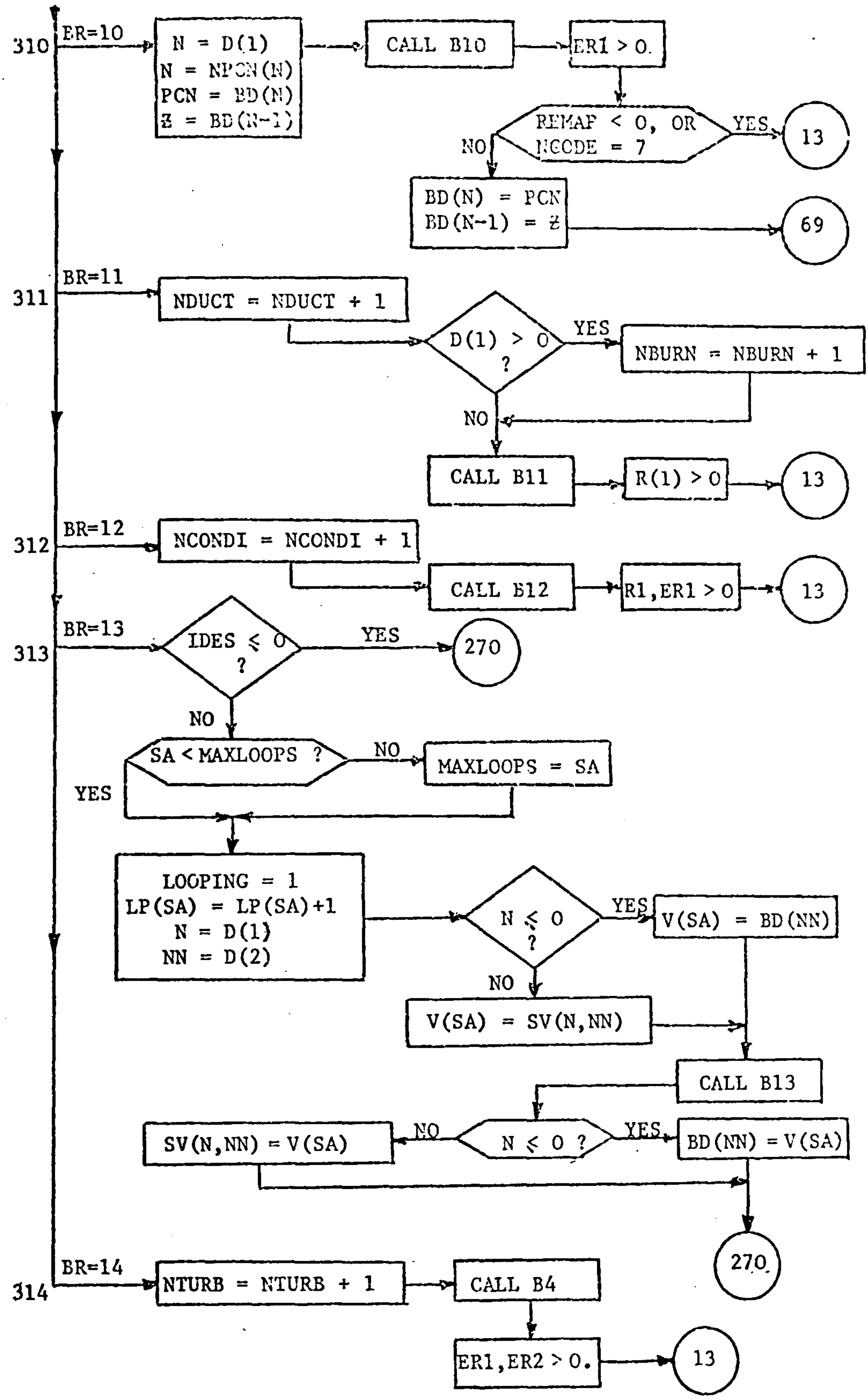
MASTER SEGMENT, SECTION 3



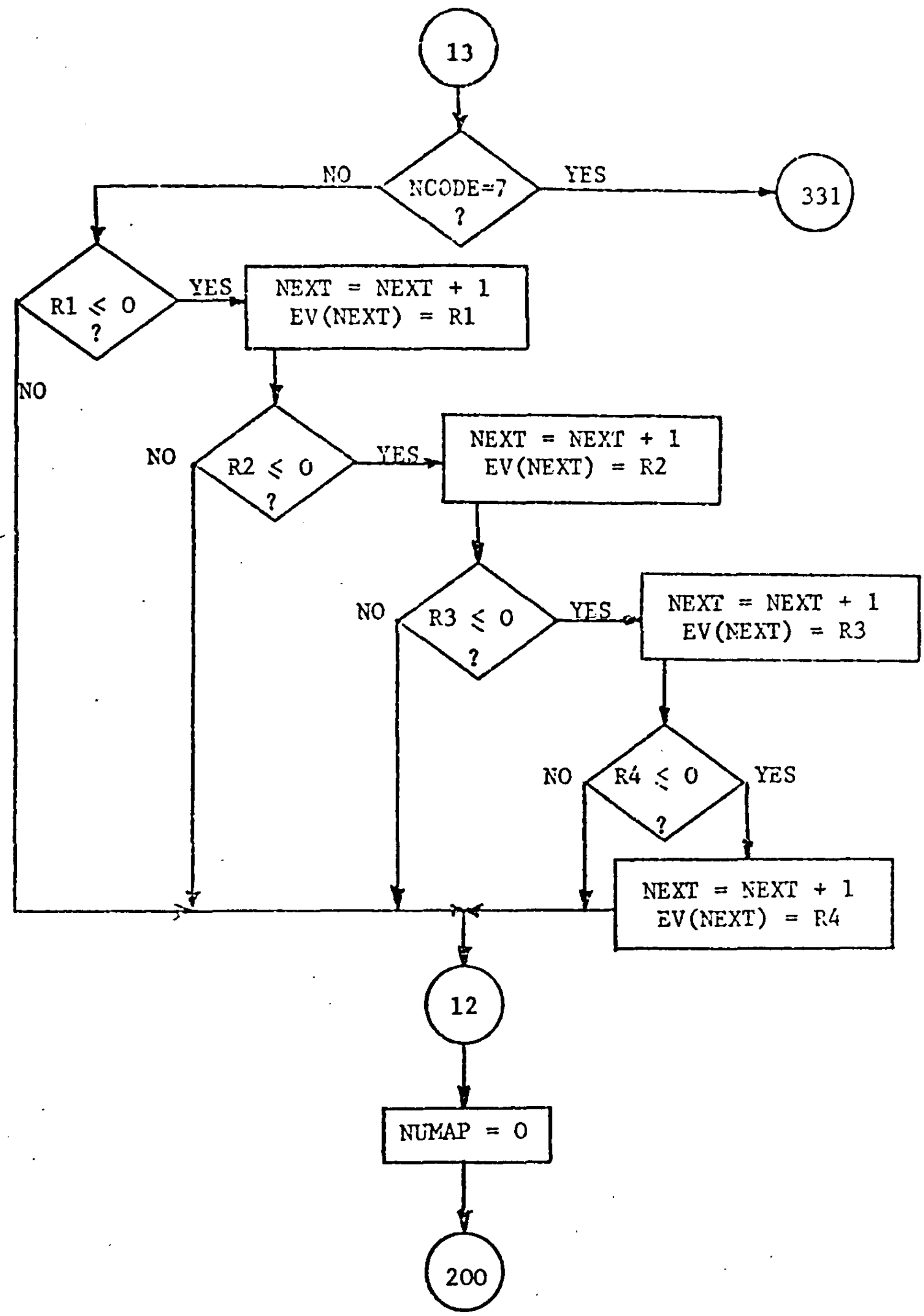
MASTER SEGMENT, SECTION 3 (CONTINUED)



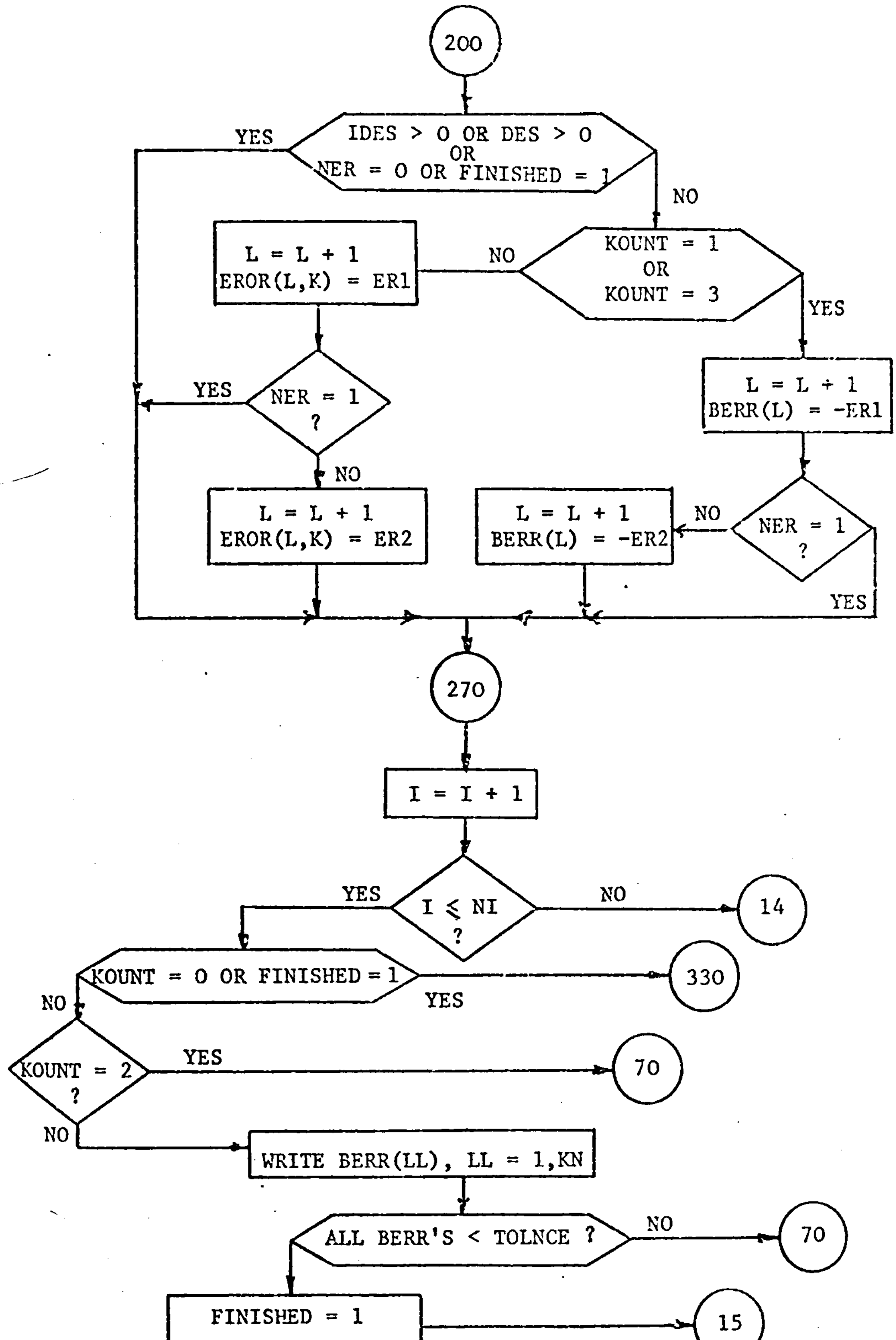
MASTER SEGMENT, SECTION 3 (CONTINUED)



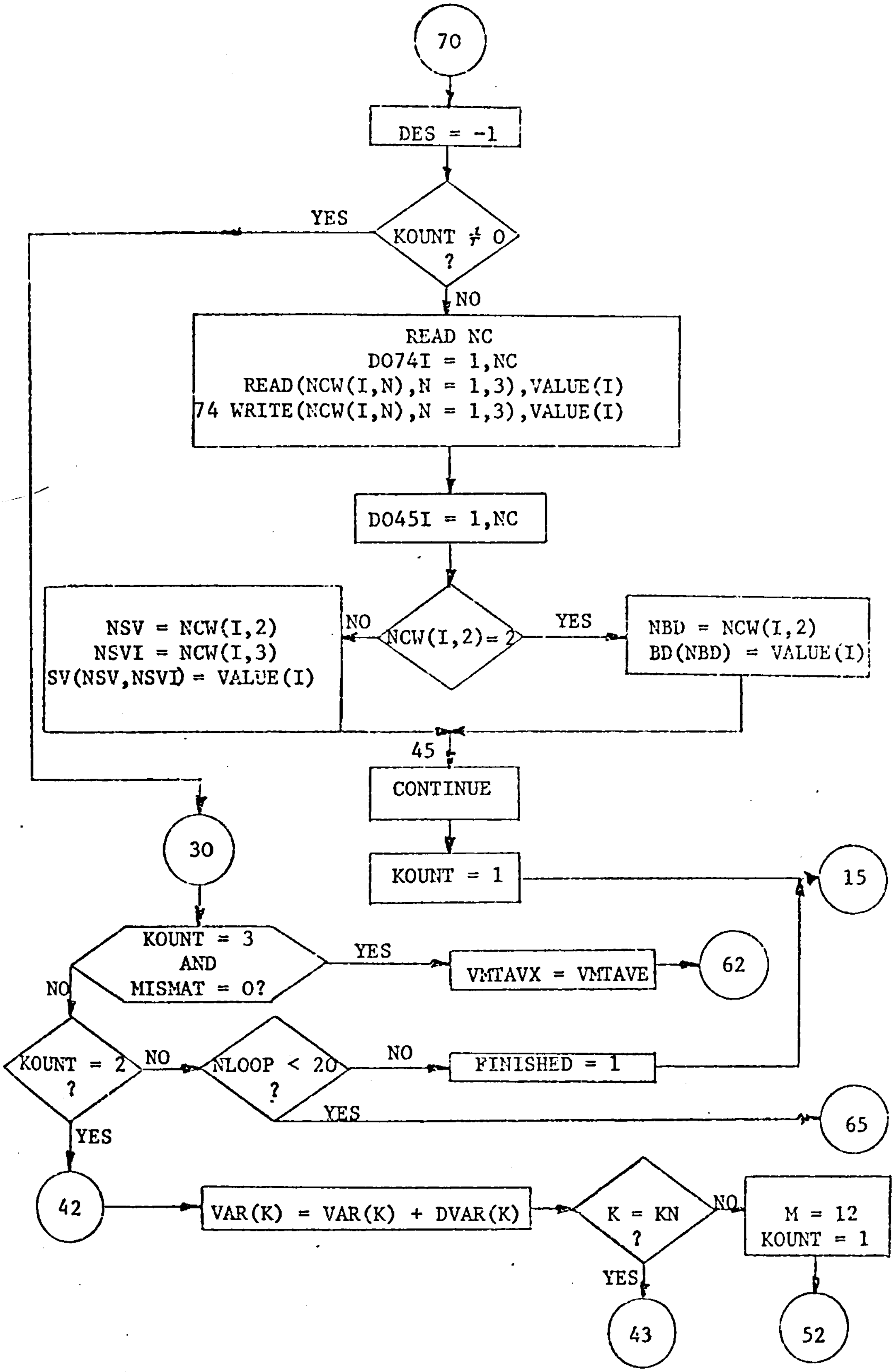
MASTER SEGMENT, SECTION 3 (CONTINUED)



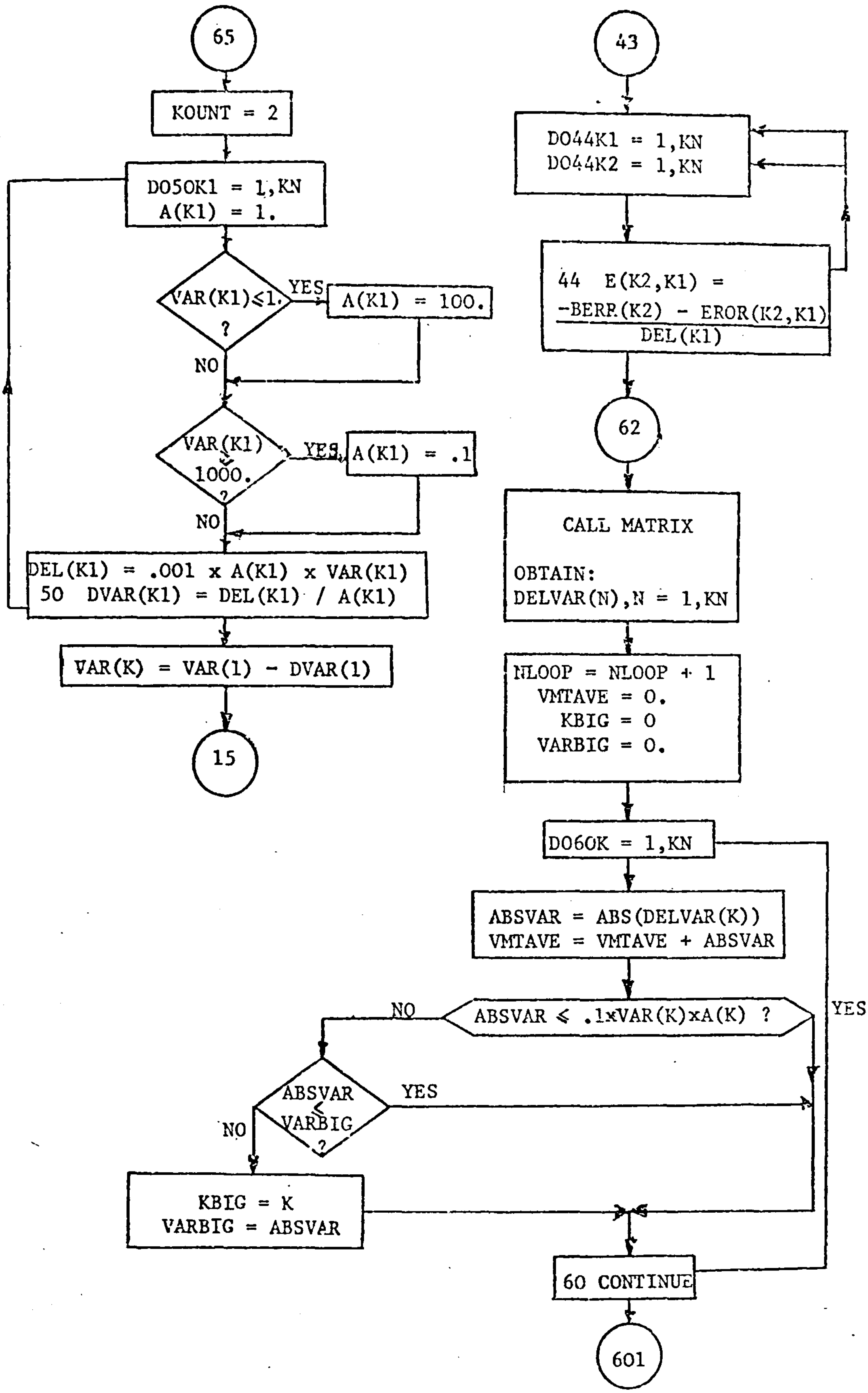
MASTER SEGMENT, SECTION 4



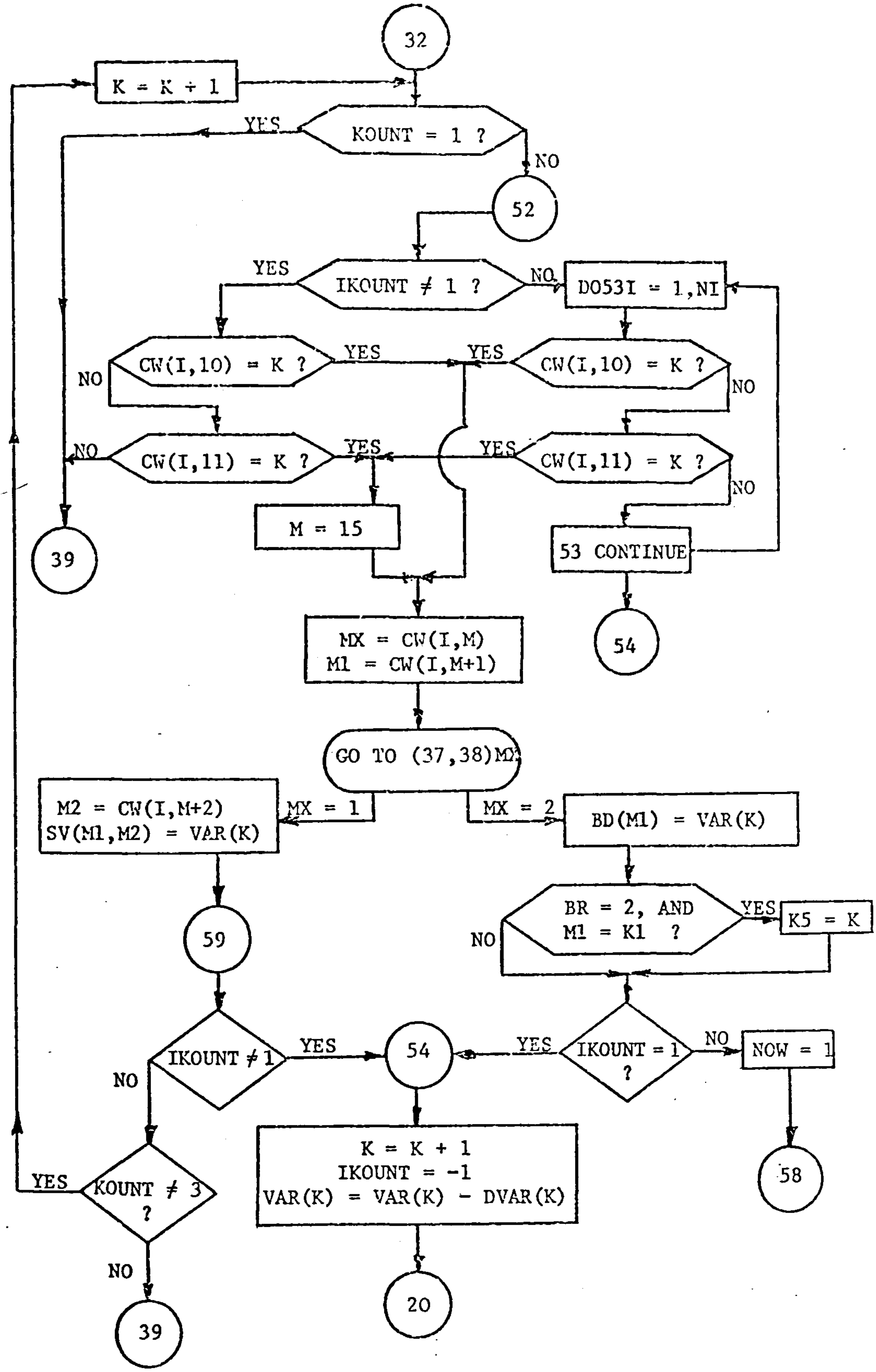
MASTER SEGMENT, SECTION 4 (CONTINUED)



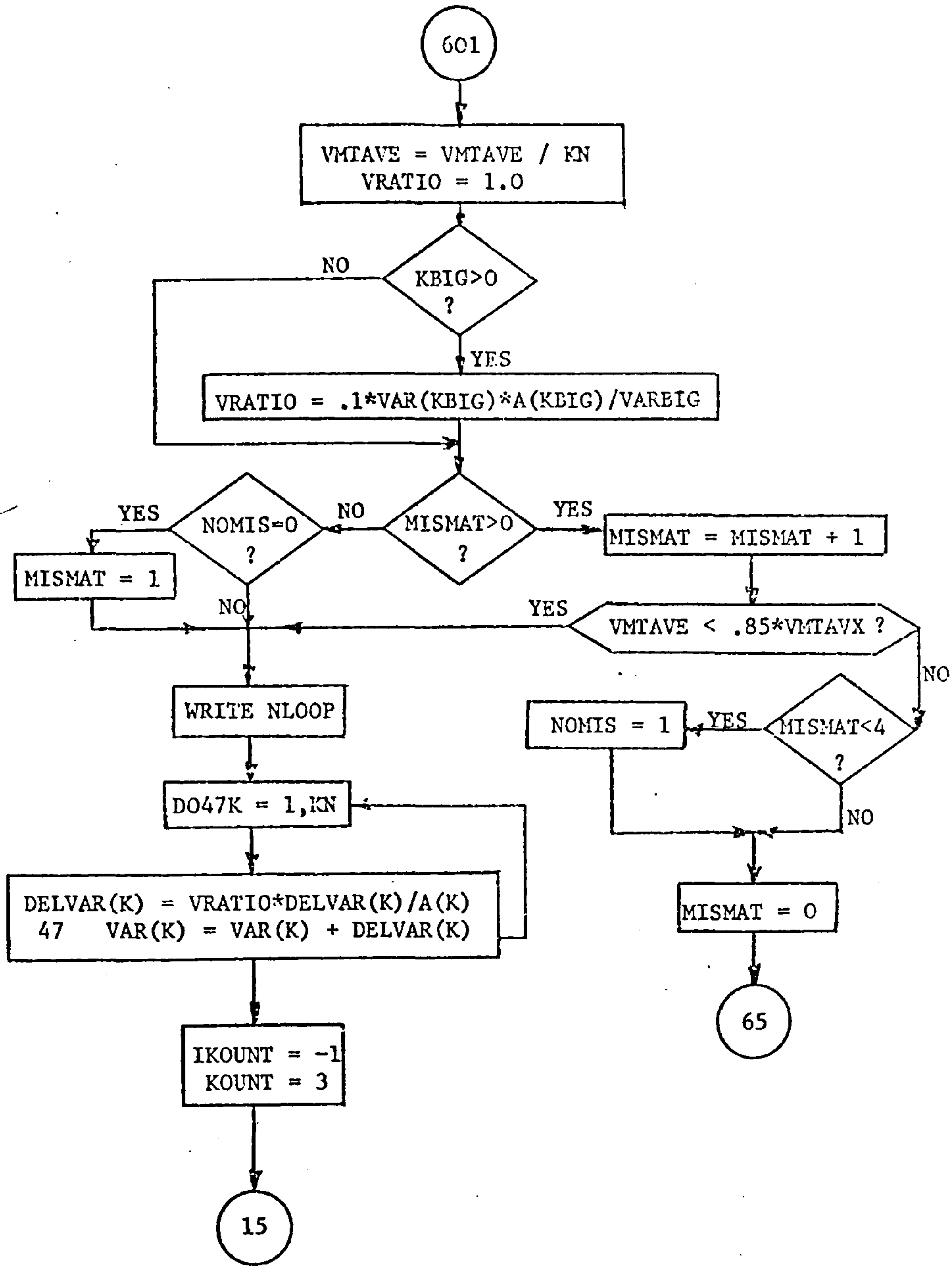
MASTER SEGMENT, SECTION 5



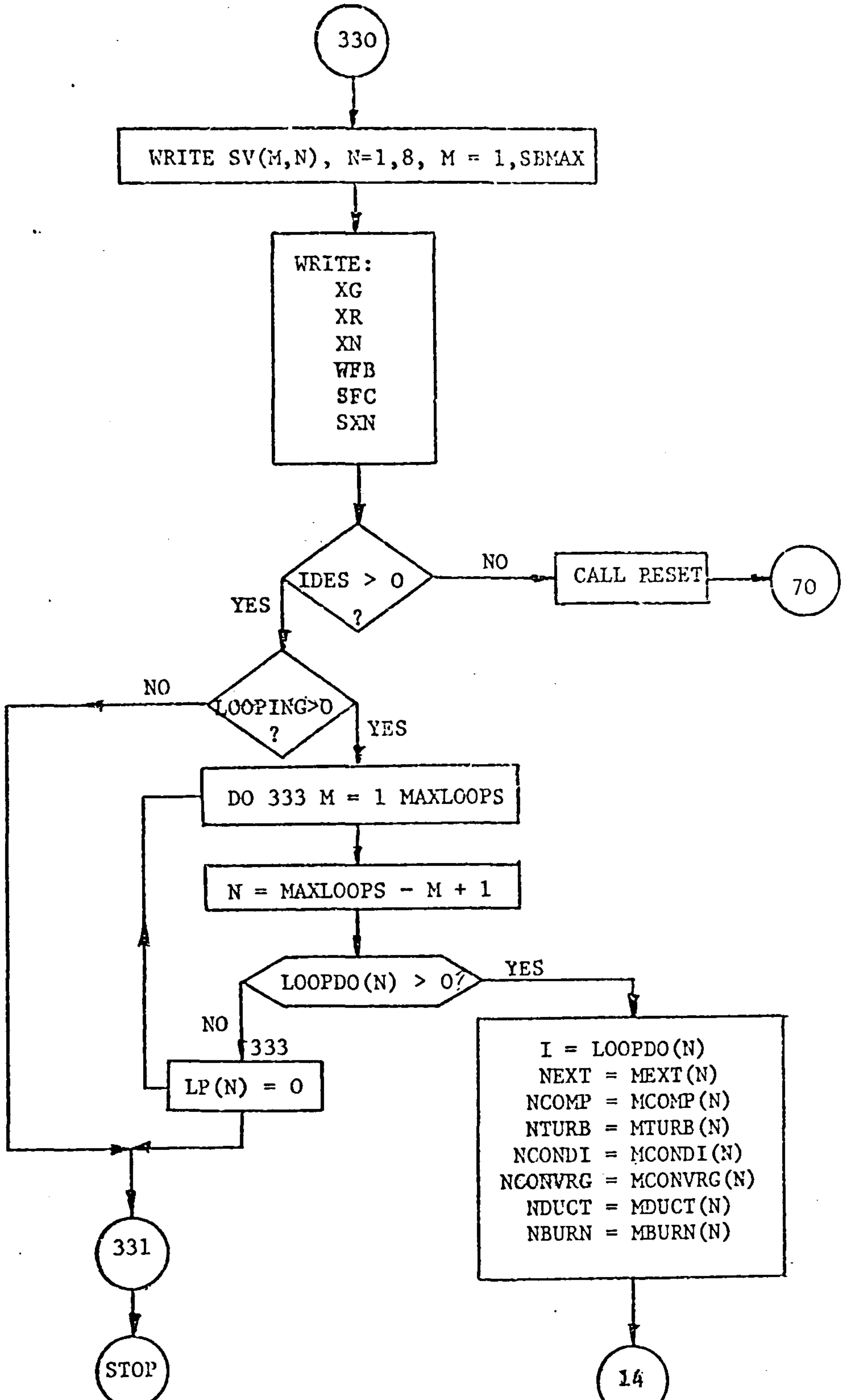
MASTER SEGMENT, SECTION 5 (CONTINUED)



MASTER SEGMENT, SECTION 5 (CONTINUED)



MASTER SEGMENT, SECTION 5 (CONTINUED)



MASTER SEGMENT, SECTION 6

APPENDIX C

Examples of Engine Cycle Simulations

C.1 Example 1 Single Spool Turbojet, Design Point

For the first example, the most simple simulation will be shown. This is a design point simulation of a one spool turbojet engine. This example also shows how two of the DP parameters may be looped to help in determining the optimum parameters for a particular cycle.

Also, the first example will explain in detail how the "program" is written. Further examples will not do this.

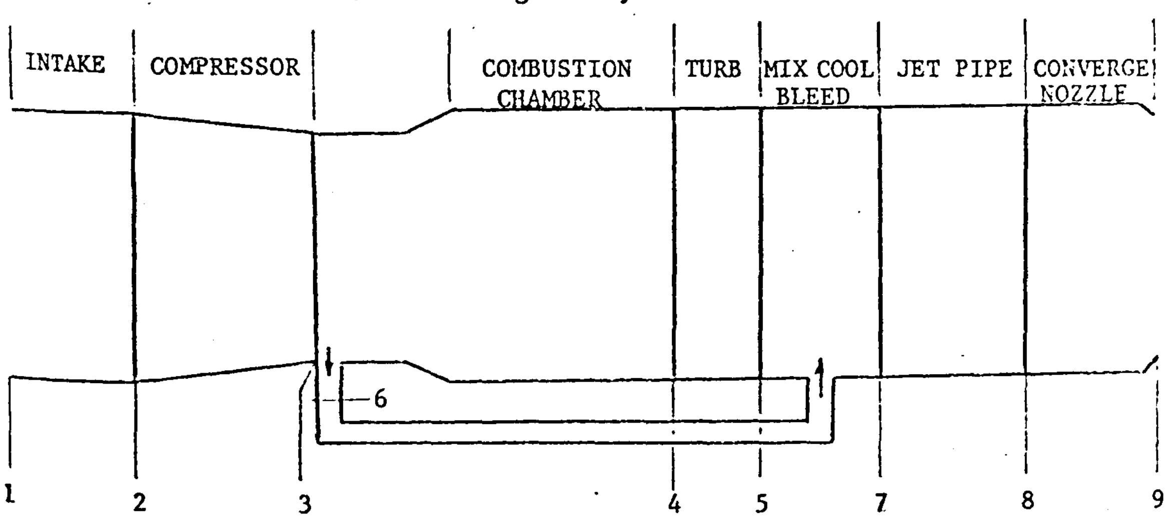
Engine: Single spool turbojet

Design Point Data:

altitude = sea level
flight speed = 0. (static)
intake pressure recovery = 98%
compressor pressure ratio, to be varied
compressor isentropic efficiency = 88%
combustion efficiency = 98%
combustion chamber $\Delta P/P$ = 0.7
turbine entry temperature, to be varied
cooling air bleed = 5% of compressor mass flow
jet pipe pressure loss = 3% of turbine outlet total pressure

It is desired to vary the compressor pressure ratio from 9.0:1 to 10.5:1 in a step of 1.5:1 and vary the turbine inlet temperature from 1400°K to 1600°K in a step of 200°K to study the effect of these parameters on the engine outputs.

A schematic of the engine layout is as follows:



NOTE: Numbers represent Stations.

The codewords which describe the engine are as follows:

BR	SV's	BD	EV's REQ.	REMARKS
001	0102-1	0104		Brick number 1, operating between SV's 1 and 2, using BD 1 to 4. Produces EV1 (XR=momentum drag)
013	01-1-1	0509		Brick 13, the first incrementation Brick
002	0203-1	1014		Produces EV2 (COMWK)
007	0306-1	1518		By these two Bricks, the mass flow is divided so that some goes through SV6, while the mass flow at SV3 is reset to W3 = W3-W6
009	060302	0302	0602	
013	02--1-1	1923		Second incrementation Brick
003	0304-1	2426		Produces EV3(WFB)
004	0405-1	2731	02-1-1	
008	050607			Mixes cooling bleed (SV6) with core stream (SV5)
011	0708-1			Provides jet pipe ΔP/P
005	080901	35		Produces EV4 (XG)
006	01-1-1	-1-1	040103	

The following Brick Data is required (one BD value per card):

BD(No)	Value	Item
1	0.	Alt.
2	0.	ISA.Dev.
3	0.	M
4	0.98	η
5	-1.	BD Qty.
6	12.	BD(No)
7	9.	Initial Value
8	1.5	Delta Value
9	10.5	Final Value
10	0.85	z
11	100.	PCN
12	9.	PR
13	0.88	η
14	-1.	ERROR?
15	0.05	λ(W)
16	0.	ΔW
17	1.	λ(P)
18	0.	ΔP

BD(No)	Value	Item
19	4.	SV No.
20	6.	SV Item No.
21	1400.	Initial Value
22	200.	Delta Value
23	1600.	Final Value
24	0.07	ΔP/P
25	0.98	η
26	-1.	WFB
27	0.	AUXWK
28	0.90	η
29	51.5	TF
30	2.4	CN
31	1.	COMP No.
32	0.	No Burn
33	0.03	ΔP/P
34	-1.	η
35	-1.	No Float

One SV quantity is required to be input, the inlet mass flow. This takes the form

```
01 02 200.          (SV(1,2) = mass flow SV1 = 200. lbm/sec)
```

The codewords have now all been formed, and all input quantities are recognized. There are 12 codewords, 35 BD quantities, and 1 SV input. One compressor and one turbine map are also required.

The deck of cards to simulate this engine would be as follows:

```
+1 -1 +1            (design study, no errors, British units)
12 35 1             (12 CW's,35 BD, 1 SV data)
01 01               (one compressor map, one turbine map)
COMPRESSOR MAP
TURBINE MAP
  CODEWORDS
  BRICK DATA
  SV DATA
```

The outputs for this simulation are found on the next 4 pages.

Example 1; CFR=9.0:1, TET=1400^{o}K

```
***** AMBIENT AND INLET PARAMETERS *****
ALT =        0.0             ISA DEV =     0.000           MACH NO =  0.00          ETAR = 0.9800

***** COMPRESSOR  1 PARAMETERS *****
PRSF =  0.12184E 01        ETASF =  0.10994E 01        WASF =  0.46420E 01
Z = 0.85000                PR =    9.000               ETA = 0.88000
PCN = 100.0000             CN = 1.00000                COMWK =  0.15951E 05

***** COMBUSTION CHAMBER PARAMETERS *****
ETASF =  0.98000E 00
ETA = 0.98000       DLP = 0.6174      WFB =    5.3918

***** TURBINE  1 PARAMETERS *****
CNSF =  0.12048E 01        ETASF =  0.10006E 01        TFSF =  0.54541E 00     DHSF =  0.14295E 00
TF =  51.500               ETA = 0.90000               CN =  2.400
AUXWK =  0.00000E 00

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  3.0079   EXIT VELOCITY = 2001.50   GROSS THRUST =  19245.19
NOZZLE COEF =  0.97041E 00
```

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	233.00	1.000	1.000	288.0	288.0	0.0	-1.0000
2	0.0000	233.00	-1.000	0.980	-1.0	288.0	-1.0	-1.0000
3	0.0000	221.35	-1.000	8.820	-1.0	569.3	-1.0	-1.0000
4	0.0244	226.74	-1.000	8.203	-1.0	1400.0	-1.0	-1.0000
5	0.0244	226.74	-1.000	3.327	-1.0	1161.9	-1.0	-1.0000
6	0.0000	11.65	-1.000	8.820	-1.0	569.3	-1.0	-1.0000
7	0.0231	238.39	-1.000	3.327	-1.0	1135.4	-1.0	-1.0000
8	0.0231	238.39	-1.000	3.227	-1.0	1135.4	-1.0	-1.0000
9	0.0231	238.39	1.763	3.227	979.9	1135.4	2001.5	3.0079

```
 GROSS THRUST = 19245.19
MOMENTUM DRAG =      0.00
   NET THRUST = 19245.19
  FUEL BURNED =     5.392
          SFC = 1.008590
    SP THRUST =    82.597
```

Example 1; CPR=9.0:1, TET=1600^{o}K

```
*****************************************************************************
***** COMBUSTION CHAMBER PARAMETERS *****
ETASF =  0.98000E 00
ETA = 0.98000      DLP = 0.6174     WFB =   6.9048

***** TURBINE  1 PARAMETERS *****
CNSF =  0.12880E 01      ETASF =  0.10006E 01      TFSF =  0.50681E 00      DHSF =  0.12425E 00
TF =  51.500             ETA = 0.90000             CN =  2.400
AUXWK =  0.00000E 00

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  2.8944   EXIT VELOCITY = 2165.61   GROSS THRUST =  21868.04
NOZZLE COEF =  0.96752E 00
```

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	233.00	1.000	1.000	288.0	288.0	0.0	-1.0000
2	0.0000	233.00	-1.000	0.980	-1.0	288.0	-1.0	-1.0000
3	0.0000	221.35	-1.000	8.820	-1.0	569.3	-1.0	-1.0000
4	0.0312	228.25	-1.000	8.203	-1.0	1600.0	-1.0	-1.0000
5	0.0312	228.25	-1.000	3.794	-1.0	1371.4	-1.0	-1.0000
6	0.0000	11.65	-1.000	8.820	-1.0	569.3	-1.0	-1.0000
7	0.0296	239.90	-1.000	3.794	-1.0	1336.5	-1.0	-1.0000
8	0.0296	239.90	-1.000	3.680	-1.0	1336.5	-1.0	-1.0000
9	0.0296	239.90	2.020	3.680	1161.4	1336.5	2165.6	2.8944

```
 GROSS THRUST = 21868.04
MOMENTUM DRAG =     0.00
   NET THRUST = 21868.04
  FUEL BURNED =    6.905
          SFC = 1.136689
    SP THRUST =   93.854
```

Example 1; CPR=10.5:1, TET=1400°K

```
******************************************************************************
***** COMPRESSOR  1 PARAMETERS *****
PRSF =  0.14468E 01        ETASF =  0.10994E 01       WASF =  0.46420E 01
Z = 0.85000                PR =  10.500               ETA = 0.88000
PCN = 100.0000             CN = 1.00000               COMWK =  0.17484E 05

***** COMBUSTION CHAMBER PARAMETERS *****
ETASF =  0.98000E 00
ETA = 0.98000      DLP = 0.7203      WFB =   5.2339

***** TURBINE  1 PARAMETERS *****
CNSF =  0.12048E 01        ETASF =  0.10006E 01       TFSF =  0.63676E 00      DHSF =  0.15679E 00
TF =  51.500               ETA = 0.90000              CN =  2.400
AUXWK =  0.00000E 00

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  2.8144   EXIT VELOCITY = 1982.95   GROSS THRUST =  19371.00
NOZZLE COEF =  0.96954E 00
```

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	233.00	1.000	1.000	288.0	288.0	0.0	-1.0000
2	0.0000	233.00	-1.000	0.980	-1.0	288.0	-1.0	-1.0000
3	0.0000	221.35	-1.000	10.290	-1.0	595.6	-1.0	-1.0000
4	0.0236	226.58	-1.000	9.570	-1.0	1400.0	-1.0	-1.0000
5	0.0236	226.58	-1.000	3.517	-1.0	1138.1	-1.0	-1.0000
6	0.0000	11.65	-1.000	10.290	-1.0	595.6	-1.0	-1.0000
7	0.0225	238.23	-1.000	3.517	-1.0	1113.7	-1.0	-1.0000
8	0.0225	238.23	-1.000	3.411	-1.0	1113.7	-1.0	-1.0000
9	0.0225	238.23	1.862	3.411	960.4	1113.7	1982.9	2.8144

```
 GROSS THRUST = 19371.00
MOMENTUM DRAG =     0.00
   NET THRUST = 19371.00
  FUEL BURNED =    5.234
          SFC = 0.972700
    SP THRUST =   83.137
```

Example 1; CPR=10.5:1, TET=1600°K

```
*******************************************************************************
***** COMBUSTION CHAMBER PARAMETERS *****
ETASF =  0.98000E 00
ETA = 0.98000      DLP = 0.7203      WFB =   6.7440

***** TURBINE  1 PARAMETERS *****
CNSF =  0.12880E 01      ETASF =  0.10000E 01      TFSF =  0.59169F 00      DHSF =  0.13629E 00
TF =  51.500             ETA = 0.90000             CN =  2.400
AUXWK =  0.00000E 00

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  2.6699   EXIT VELOCITY = 2149.27   GROSS THRUST =  22034.08
NOZZLE COEF =  0.96348F 00
```

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	233.00	1.000	1.000	288.0	288.0	0.0	-1.0000
2	0.0000	233.00	-1.000	0.980	-1.0	288.0	-1.0	-1.0000
3	0.0000	221.35	-1.000	10.290	-1.0	595.6	-1.0	-1.0000
4	0.0305	228.09	-1.000	9.570	-1.0	1600.0	-1.0	-1.0000
5	0.0305	228.09	-1.000	4.076	-1.0	1348.7	-1.0	-1.0000
6	0.0000	11.65	-1.000	10.290	-1.0	595.6	-1.0	-1.0000
7	0.0289	239.74	-1.000	4.076	-1.0	1315.7	-1.0	-1.0000
8	0.0289	239.74	-1.000	3.954	-1.0	1315.7	-1.0	-1.0000
9	0.0289	239.74	2.169	3.954	1142.6	1315.7	2149.3	2.6699

```
  GROSS THRUST = 22034.08
MOMENTUM DRAG =      0.00
    NET THRUST = 22034.08
   FUEL BURNED =     6.744
           SFC = 1.101856
     SP THRUST =    94.567
```

C.2 Example 2. Single Spool Turbojet, Off Design Point Study

This example is the logical extension of example 1. From the DP studies, a selection is made from the possible engine parameters to describe what the designer feels is the optimum cycle. Now, ODP performance predictions are required.

For this example, the engine configuration is identical to that of example 1, only the compressor pressure ratio has been fixed at 10:1 and the turbine entry temperature set at 1500°K. It is desired to check the engine performance at the following points.

Case 1. DP situation, sea level, static.

Case 2. Altitude = 10000 ft.
Flight Mach number = 0.4
Turbine entry temperature = 1300°K

Case 3. Altitude = 36089 ft.
Flight Mach number = 0.9
Turbine entry temperature = 1450°K

Case 4. As in Case 3, only afterburner on
Afterburner η = 98%
Afterburner final temperature = 1500°K
Nozzle exit area floating

Note that the ODP condition is created by specifying the turbine entry temperature. The shaft rotational speed will be varied to correspond to this temperature.

As this is an ODP study, the previous "program" will need rewriting to include "variables". Also, the incrementation Bricks (B13) will be removed. Note in the codewords that 3 errors will be produced, therefore 3 variables must be specified.

The codewords become:

BR	SV's	BD	EV's REQ.	VAR's	VAR 1	VAR 2	REMARKS
001	0102-1	0104					
002	0203-1	0509	-1-1-1	0102	0205-1	0206-1	Variables are Z and PCN
007	0306-1	1013					
009	060302	0302	0602-1				
003	0304-1	1416					
004	0405-1	1721	02-1-1	03-1	0219-1		Variable is TF. Produces 2 errors
008	050607						
011	0708-1	2224					
005	080901	25					Produces 1 error
009	05-103	-103	-103-1				Adds WFB from combustion and afterburning
006	01-1-1	-1-1	050103				

The required Brick Data is:

BD(No)	Value	BD(No)	Value
1	0.	14	0.07
2	0.	15	0.98
3	0.	16	-1.
4	0.98	17	0.
5	0.85	18	0.9
6	100.	19	51.5
7	10.	20	2.4
8	0.88	21	1.
9	-1.	22	1.
10	0.05	23	0.03
11	0.	24	-1.
12	1.	25	-1.
13	0.		

Two SV quantities are required to be input. These are the inlet mass flow and final temperature of combustion. The SV input data is thus:

01 02 200. (SV(1,2) = inlet mass flow = 200 lbm/sec)
04 06 1500. (SV(4,6) = T4 = 1500°K)

It can be seen that in this simulation,

Number of Codewords = 11
Number of Brick Data = 25
Number of SV Data = 2

Next, the "mini" codewords required for the ODP cases are:

<u>Case 2</u>

3	(Number of changes to be made)
0201-1 10000.	(Altitude = 10000 ft)
0203-1 .4	(Flight Mach No. = 0.4)
010406 1300.	(T4 = 1300°K)

<u>Case 3</u>

3	
0201-1 36089.	(Altitude = 36089 ft)
0203-1 .9	(Flight Mach No. = 0.9)
010406 1450.	(T4 = 1450°K)

<u>Case 4</u>

4	
0222-1 2.	(Afterburner on)
0224-1 0.98	(Afterburner η 98%)
010806 1500.	(T8 = 1500°K)
0225-1 1.	(nozzle exit area floats)

The deck of cards to simulate the engine would then be as follows:

-1 03 +1	(off design study, 3 errors, British units)
11 25 2	(11 CW's, 25 BD, 2 SV data)
01 01	(one compressor map, one turbine map)

COMPRESSOR MAP

TURBINE MAP

CODEWORDS

BRICK DATA

SV DATA

MINI CODEWORDS FOR EACH ODP CASE

The outputs for this engine at its DP and the 3 ODP cases are given in the next 4 pages.

Example 2, Case 1 (Design Point), Outputs

```
***** AMBIENT AND INLET PARAMETERS *****
ALT =     0.0          ISA DEV =     0.000          MACH NO =   0.00          ETAR = 0.9800

***** COMPRESSOR  1 PARAMETERS *****
PRSF =  0.13706E 01        ETASF =  0.10994E 01       WASF =  0.46420E 01
Z = 0.85000                PR =  10.000               ETA = 0.88000
PCN = 100.0000             CN = 1.00000               COMWK =  0.16992E 05

***** COMBUSTION CHAMBER PARAMETERS *****
ETASF =  0.98000E 00
ETA = 0.98000       DLP = 0.6860      WFB =    6.0297

***** TURBINE  1 PARAMETERS *****
CNSF =  0.12471E 01        ETASF =  0.10006E 01       TFSF =  0.58382E 00      DHSF =  0.14172E 00
TF =  51.500               ETA = 0.90000              CN =  2.400
AUXWK =  0.00000E 00

***** DUCT/AFTER BURNING 1 PARAMETERS *****
ETA = -1.0000              DLP = 0.1120               WFB =   0.0000

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  2.7939    EXIT VELOCITY = 2073.60    GROSS THRUST =  20729.08
NOZZLE COEF =  0.96855E 00
```

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	233.00	1.000	1.000	288.0	288.0	0.0	-1.0000
2	0.0000	233.00	-1.000	0.960	-1.0	288.0	-1.0	-1.0000
3	0.0000	221.35	-1.000	9.800	-1.0	587.1	-1.0	-1.0000
4	0.0272	227.38	-1.000	9.114	-1.0	1500.0	-1.0	-1.0000
5	0.0272	227.38	-1.000	3.734	-1.0	1251.1	-1.0	-1.0000
6	0.0000	11.65	-1.000	9.800	-1.0	587.1	-1.0	-1.0000
7	0.0259	239.03	-1.000	3.734	-1.0	1221.7	-1.0	-1.0000
8	0.0259	239.03	-1.000	3.622	-1.0	1221.7	-1.0	-1.0000
9	0.0259	239.03	1.982	3.622	1057.7	1221.7	2073.6	2.7939

```
 GROSS THRUST = 20729.08
MOMENTUM DRAG =     0.00
   NET THRUST = 20729.08
  FUEL BURNED =    6.030
          SFC = 1.047175
    SP THRUST =   88.966
```

Example 2, Case 2, Outputs

***** OFF DESIGN ENGINE CALCULATIONS. CONVERGED AFTER 3 LOOPS *****

***** AMBIENT AND INLET PARAMETERS *****
ALT = 10000.0 ISA DEV = 0.000 MACH NO = 0.40 ETAR = 0.9800

***** COMPRESSOR 1 PARAMETERS *****
Z = 0.80107 PR = 8.566 ETA = 0.89050
PCN = 91.8157 CN = 0.93656 COMWK = 0.10442E 05

***** COMBUSTION CHAMBER PARAMETERS *****
ETA = 0.98000 DLP = 0.4809 WFB = 3.4498

***** TURBINE 1 PARAMETERS *****
TF = 51.752 ETA = 0.90055 CN = 2.367
AUXWK = 0.00000E 00

***** DUCT/AFTER BURNING 1 PARAMETERS *****
ETA = -1.0000 DLP = 0.0740 WFB = 0.0000

***** CONVERGENT NOZZLE 1 PARAMETERS *****
AREA = 2.7939 EXIT VELOCITY = 1929.13 GROSS THRUST = 13372.21
NOZZLE COEF = 0.96958E 00

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	165.46	0.688	0.768	268.2	276.8	431.1	-1.0000
2	0.0000	165.46	-1.000	0.752	-1.0	276.8	-1.0	-1.0000
3	0.0000	157.19	-1.000	6.446	-1.0	536.9	-1.0	-1.0000
4	0.0219	160.64	-1.000	5.965	-1.0	1300.0	-1.0	-1.0000
5	0.0219	160.64	-1.000	2.429	-1.0	1076.4	-1.0	-1.0000
6	0.0000	8.27	-1.000	6.446	-1.0	536.9	-1.0	-1.0000
7	0.0208	168.91	-1.000	2.429	-1.0	1052.1	-1.0	-1.0000
8	0.0208	168.91	-1.000	2.355	-1.0	1052.1	-1.0	-1.0000
9	0.0208	168.91	1.288	2.355	905.1	1052.1	1929.1	2.7939

GROSS THRUST = 13372.21
MOMENTUM DRAG = 2216.93
NET THRUST = 11155.28
FUEL BURNED = 3.450
SFC = 1.113501
SP THRUST = 67.418

***** OFF DESIGN ENGINE CALCULATIONS. CONVERGED

```
***** AMBIENT AND INLET PARAMETERS *****
ALT =  36089.0             ISA DEV =     0.000          MACH NO =  0.90            ETAR = 0.9800

***** COMPRESSOR  1 PARAMETERS *****
Z = 0.85747                PR =  10.253                 ETA = 0.87133
PCN =  94.5364             CN = 1.01121                 COMWK =  0.62568E 04

***** COMBUSTION CHAMBER PARAMETERS *****
ETA = 0.98000        DLP = 0.2653      WFB =    2.1781

***** TURBINE  1 PARAMETERS *****
TF =  51.605               ETA = 0.90020                CN =  2.392
AUXWK =  0.00000E 00

***** DUCT/AFTER BURNING 1 PARAMETERS *****
ETA = -1.0000              DLP = 0.0434                 WFB =   0.0000

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  2.7939   EXIT VELOCITY = 1965.88   GROSS THRUST =   8776.18
NOZZLE COEF =  0.93460E 00
```

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	95.63	0.223	0.378	216.5	251.7	871.6	-1.0000
2	0.0000	95.63	-1.000	0.370	-1.0	251.7	-1.0	-1.0000
3	0.0000	90.65	-1.000	3.796	-1.0	522.0	-1.0	-1.0000
4	0.0240	93.03	-1.000	3.530	-1.0	1350.0	-1.0	-1.0000
5	0.0240	93.03	-1.000	1.444	-1.0	1120.9	-1.0	-1.0000
6	0.0000	4.78	-1.000	3.796	-1.0	522.0	-1.0	-1.0000
7	0.0228	97.81	-1.000	1.444	-1.0	1094.1	-1.0	-1.0000
8	0.0228	97.81	-1.000	1.400	-1.0	1094.1	-1.0	-1.0000
9	0.0228	97.81	0.763	1.400	943.1	1094.1	1965.9	2.7939

```
 GROSS THRUST =  8776.18
MOMENTUM DRAG =  2590.58
   NET THRUST =  6185.60
  FUEL BURNED =     2.178
          SFC = 1.267065
    SP THRUST =    64.680
```

Example 2, Case 3, Outputs

***** OFF DESIGN ENGINE CALCULATIONS, CONVERGED AFTER 8 LOOPS *****

***** AMBIENT AND INLET PARAMETERS *****
ALT = 36089.0 ISA DEV = 0.000 MACH NO = 0.90 ETAR = 0.9800

***** COMPRESSOR 1 PARAMETERS *****
Z = 0.85747 PR = 10.253 ETA = 0.87133
PCN = 94.5364 CN = 1.01121 COMWK = 0.62568E 04

***** COMBUSTION CHAMBER PARAMETERS *****
ETA = 0.98000 DLP = 0.2653 WFB = 2.1781

***** TURBINE 1 PARAMETERS *****
TF = 51.605 ETA = 0.90020 CN = 2.392
AUXWK = 0.00000E 00

***** DUCT/AFTER BURNING 1 PARAMETERS *****
ETA = 0.9800 DLP = 0.0434 WFB = 1.2815

***** CONVERGENT NOZZLE 1 PARAMETERS *****
AREA = 3.3423 EXIT VELOCITY = 2289.65 GROSS THRUST = 10464.79
NOZZLE COEF = 0.93460E 00

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	95.63	0.223	0.378	216.5	251.7	871.6	-1.0000
2	0.0000	95.63	-1.000	0.370	-1.0	251.7	-1.0	-1.0000
3	0.0000	90.85	-1.000	3.796	-1.0	522.0	-1.0	-1.0000
4	0.0240	93.03	-1.000	3.530	-1.0	1350.0	-1.0	-1.0000
5	0.0240	93.03	-1.000	1.444	-1.0	1120.9	-1.0	-1.0000
6	0.0000	4.78	-1.000	3.796	-1.0	522.0	-1.0	-1.0000
7	0.0228	97.81	-1.000	1.444	-1.0	1094.1	-1.0	-1.0000
8	0.0367	99.09	-1.000	1.400	-1.0	1500.0	-1.0	-1.0000
9	0.0367	99.09	0.771	1.400	1310.2	1500.0	2289.7	3.3423

GROSS THRUST = 10464.79
MOMENTUM DRAG = 2590.58
NET THRUST = 7874.21
FUEL BURNED = 3.460
SFC = 1.581707
SP THRUST = 82.338

C.3 Example 3. Two Spool Mixed Flow Turbofan, ODP Study

This final example demonstrates a complex simulation; a two spool turbofan with a mixed exhaust nozzle. The design point pressure ratios have been chosen such that mixing is possible.

Two points to note in this simulation are:

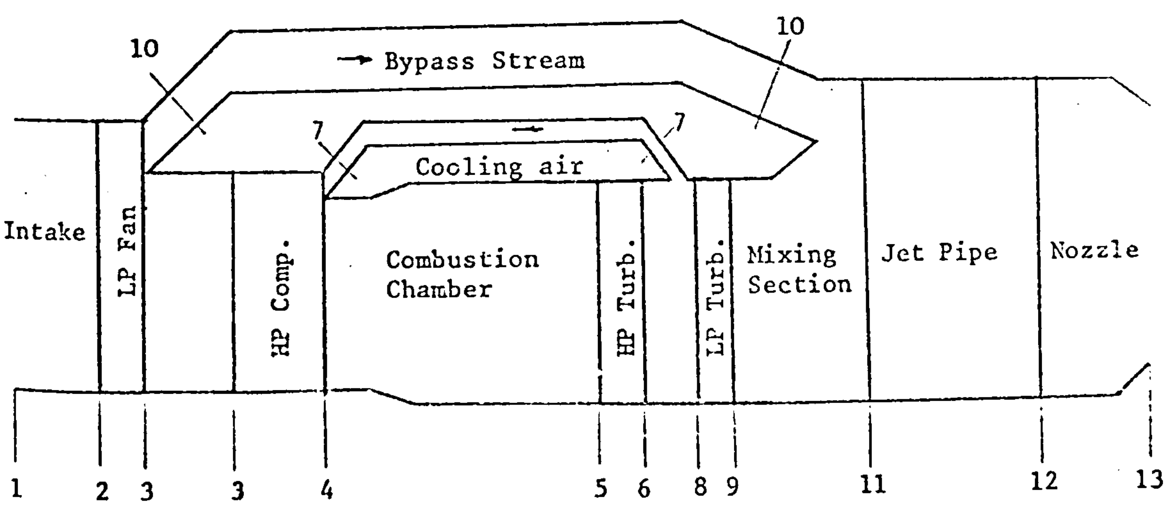
1) TET is not entered as data, rather the fuel flow rate is specified

2) 7 variables are required to properly simulate this engine. However, only 6 errors would normally be generated; therefore by specifying BD(18)>0., the HP compressor produces the extra necessary error.

Engine Design Point Data:

flight condition = sea level, static
intake pressure recovery = 98%
fan pressure ratio = 4.25
fan isentropic efficiency = 0.829
compressor pressure ratio = 5.75
compressor isentropic efficiency = 0.85
engine bypass ratio = 0.5
combustion efficiency = 0.985
fuel flow rate = 9.51 lbm/sec
combustion chamber ΔP/P = 0.056
HP cooling bleed = 9.7%
turbine efficiencies = 0.90
core Mach No. before mixing = 0.24
jet pipe pressure loss = 5%

Engine Schematic Showing Stations



The codewords which describe this simulation follow:

BR	SV's	BD	EV's REQ.	VAR's	VAR 1	VAR 2	REMARKS
001	0102-1	0104	-1-1-1				
002	0203-1	0509	-1-1-1	0102	0205-1	0206-1	Variables are Z and PCN
007	0310-1	1013	-1-1-1	03-1	0210-1		Variable is $\lambda(W)$ so that bypass ratio varies
009	060302	0302	1002				W3 = W3-W10
002	0304-1	1418	-1-1-1	0405	0214-1	0215-1	Variables are Z and PCN
007	0407-1	1922					HP cool bleed (SV7)
009	060402	0402	0702				W4 = W4-W7
003	0405-1	2325					
004	0506-1	2630	03-1-1	06-1	0228-1		Variable is TF
008	060708						Mixing of cool bleed.
004	0809-1	3135	02-1-1	07-1	0233-1		Variable is TF
010	091011	3638					Mixing of bypass and core
011	1112-1	3941					
005	121301	42-1					
006	01-1-1	-1-1	050104				

The required Brick Data is

BD(No)	Value	BD(No)	Value	BD(No)	Value
1	0.	15	100.	29	2.
2	0.	16	5.75	30	2.
3	0.	17	0.85	31	0.
4	0.98	18	+1.	32	0.903
5	0.833	19	0.097	33	114.
6	100.	20	0.	34	2.3
7	4.25	21	1.	35	1.
8	0.829	22	0.	36	1.
9	-1.	23	0.056	37	1.
10	0.333	24	0.985	38	0.24
11	0.	25	9.5	39	-1.
12	1.	26	0.	40	0.05
13	0.	27	0.9	41	-1.
14	0.814	28	50.	42	-1.

The inlet mass flow is the only SV quantity to be input. This takes the form

01 02 500. (SV(1,2) = mass flow SV1 = 500 lbm/sec)

Therefore, in this simulation:

Number of codewords = 15
Number of Brick Data = 42
Number of SV Data = 1

The "mini" codewords required for the ODP case are:

```
 3                   (Number of changes to be made)
0201-1 36000.        (Altitude = 36000 ft.)
0203-1  0.85         (Flight Mach No. = 0.85)
010506  1810.        (T5 = 1810°K)
```

The deck of cards to simulate this engine would be as follows:

```
-1 07 +1
15 42  1
02 02
COMPRESSOR MAPS
TURBINE MAPS
  CODEWORDS
  BRICK DATA
  SV DATA
  MINI CODEWORDS
```

The outputs for this engine at its DP and ODP case are given in the next 4 pages.

Example 3, Design Point Outputs (Continued on next page)

```
***** DESIGN POINT ENGINE CALCULATIONS *****

***** AMBIENT AND INLET PARAMETERS *****
ALT =       0.0            ISA DEV =     0.000        MACH NO =  0.00          ETAR = 1.0000

***** COMPRESSOR  1 PARAMETERS *****
PRSF =  0.32500E 01       ETASF =  0.99880E 00      WASF =  0.27947E 01
Z = 0.83333               PR =    4.250             ETA = 0.82900
PCN = 100.0000            CN = 1.00000              COMWK =  0.21327E 05

***** COMPRESSOR  2 PARAMETERS *****
PRSF =  0.75508E 00       ETASF =  0.10514E 01      WASF =  0.19408E 01
Z = 0.81433               PR =    5.750             ETA = 0.85000
PCN = 100.0000            CN = 1.00000              COMWK =  0.28080E 05

***** COMBUSTION CHAMBER PARAMETERS *****
ETASF =  0.98500E 00
ETA = 0.98500       DLP = 1.3685       WFB =   9.5000

***** TURBINE  1 PARAMETERS *****
CNSF =  0.11375E 01       ETASF =  0.10005E 01      TFSF =  0.95991E 00     DHSF =  0.83906E-01
TF =  50.000              ETA = 0.90000             CN =  2.000
AUXWK =  0.00000E 00

***** TURBINE  2 PARAMETERS *****
CNSF =  0.11728E 01       ETASF =  0.10062E 01      TFSF =  0.89920E 00     DHSF =  0.23555E 00
TF = 114.000              ETA = 0.90300             CN =  2.300
AUXWK =  0.00000E 00

***** MIXING MACH NUMBERS *****
STATION  9,M = 0.240     STATION 10,M = 0.067    STATION 11,M = 0.156

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  4.7145   EXIT VELOCITY = 1886.07   GROSS THRUST =  41119.57
NOZZLE COEF =  0.96154E 00
```

Example 3, Design Point Outputs (Continued)

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	500.00	1.000	1.000	288.0	288.0	0.0	-1.0000
2	0.0000	500.00	-1.000	1.000	-1.0	288.0	-1.0	-1.0000
3	0.0000	333.34	-1.000	4.250	-1.0	464.6	-1.0	-1.0000
4	0.0000	301.00	-1.000	24.438	-1.0	797.7	-1.0	-1.0000
5	0.0316	310.50	-1.000	23.069	-1.0	1797.0	-1.0	-1.0000
6	0.0316	310.50	-1.000	9.383	-1.0	1505.6	-1.0	-1.0000
7	0.0000	32.33	-1.000	24.438	-1.0	797.7	-1.0	-1.0000
8	0.0285	342.83	-1.000	9.383	-1.0	1444.6	-1.0	-1.0000
9	0.0285	342.83	4.237	4.399	1226.3	1237.0	534.0	8.4413
10	0.0000	166.67	4.237	4.250	464.2	464.6	94.4	8.7830
11	0.0190	509.50	4.270	4.339	1000.1	1004.0	314.7	17.2243
12	0.0190	509.50	-1.000	4.122	-1.0	1004.0	-1.0	-1.0000
13	0.0190	509.50	2.243	4.122	861.9	1004.0	1886.1	4.7145

```
   GROSS THRUST = 41119.57
  MOMENTUM DRAG =     0.00
     NET THRUST = 41119.57
    FUEL BURNED =    9.500
            SFC = 0.831721
      SP THRUST =   82.239
```

```
***** OFF DESIGN ENGINE CALCULATIONS. CONVERGED AFTER  3 LOOPS *****

***** AMBIENT AND INLET PARAMETERS *****
ALT =  36000.0             ISA DEV =     0.000         MACH NO =  0.85          ETAR = 1.0000

***** COMPRESSOR  1 PARAMETERS *****
Z = 0.90766                PR =    5.254               ETA = 0.75507
PCN = 101.4546             CN = 1.09306                COMWK =  0.10519E 05

***** COMPRESSOR  2 PARAMETERS *****
Z = 0.83600                PR =    6.064               ETA = 0.83345
PCN = 100.2578             CN = 1.02260                COMWK =  0.13385E 05

***** COMBUSTION CHAMBER PARAMETERS *****
ETA = 0.98500      DLP = 0.6298      WFB =    4.5538

***** TURBINE  1 PARAMETERS *****
TF =  50.028               ETA = 0.89874               CN =  1.998
AUXWK =  0.00000E 00

***** TURBINE  2 PARAMETERS *****
TF = 115.314               ETA = 0.90453               CN =  2.328
AUXWK =  0.00000E 00

***** MIXING MACH NUMBERS *****
STATION  9,M = 0.253   STATION 10,M = 0.057    STATION 11,M = 0.156

***** CONVERGENT NOZZLE  1 PARAMETERS *****
AREA =  4.7145   EXIT VELOCITY = 1906.83   GROSS THRUST =  19982.78
NOZZLE COEF =  0.91818E 00
```

Example 3, Off Design Outputs (Continued on next page)

Example 3, Off Design Outputs (Continued)

STATION	FUEL/AIR	MASS FLOW	PSTATIC	PTOTAL	TSTATIC	TTOTAL	VEL	AREA
1	0.0000	220.01	0.224	0.360	216.7	248.1	823.5	-1.0000
2	0.0000	220.01	-1.000	0.360	-1.0	248.1	-1.0	-1.0000
3	0.0000	155.91	-1.000	1.890	-1.0	446.6	-1.0	-1.0000
4	0.0000	140.78	-1.000	11.461	-1.0	787.1	-1.0	-1.0000
5	0.0323	145.34	-1.000	10.831	-1.0	1810.0	-1.0	-1.0000
6	0.0323	145.34	-1.000	4.352	-1.0	1513.9	-1.0	-1.0000
7	0.0000	15.12	-1.000	11.461	-1.0	787.1	-1.0	-1.0000
8	0.0292	160.46	-1.000	4.352	-1.0	1451.5	-1.0	-1.0000
9	0.0292	160.46	1.886	1.962	1221.7	1233.0	560.2	8.4413
10	0.0000	64.15	1.885	1.890	446.3	446.6	78.5	8.7830
11	0.0207	224.61	1.906	1.936	1023.2	1027.2	318.0	17.2243
12	0.0207	224.61	-1.000	1.840	-1.0	1027.2	-1.0	-1.0000
13	0.0207	224.61	1.002	1.840	882.9	1027.2	1906.8	4.7145

GROSS THRUST = 19982.78
MOMENTUM DRAG = 5630.99
NET THRUST = 14351.79
FUEL BURNED = 4.554
SFC = 1.142267
SP THRUST = 65.232

APPENDIX D

Comparison with Experimental Verification of National Research Council of Canada Off Design Program

Reference: Chappell, M. S. Grabbe, W. Gas Turbine Cycle Calculations, Experimental Verification of Off Design Point Predictions for a Two-Spool Turbojet with Various Air Bleeds. National Research Council of Canada, Report LR-555, November, 1971.

1. Introduction

As a part of a continuing program of engine studies, the National Research Council of Canada (NRC) developed an ODP engine program. An attempt to obtain experimental verification of the program was made in 1971, with the results being given in the above reference.

It was considered desireable to obtain experimental verification for the program described in this report. In order to fully test the program, it was felt that the test engine must have at least two shafts in order to make it a difficult enough trial. As no two spool engine was available for test at Cranfield, the work done by NRC provided both a source of experimental data and also gave a good comparison with the results of another ODP program.

2. NRC Program and Tests

2.1 The NRC ODP program of the above reference was also described in section 2.3.4 of the main text (this reference is the same as reference 16 in the main text). Basically, it was a program designed with some degree of flexibility since it was capable of dealing with one or two shaft, fan or pure jet engines, with or without reheat. NRC attempted to obtain experimental verification for its program by comparing its results with the experimental values of a two spool turbojet undergoing tests.

2.2 Engine

The engine tested was a J75 - P/3 which is a two spool turbojet with provision for afterburning and a two position nozzle. The layout of the engine is shown below in figure 1.

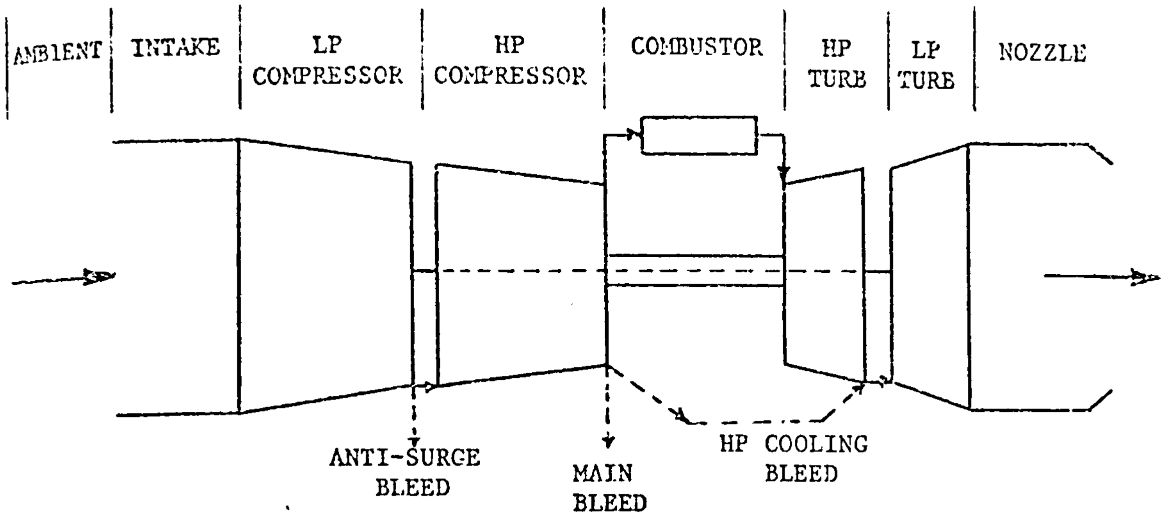


Figure 1 Engine Layout

The anti-surge bleed flow was 6.1% of the airflow at LP speeds < 78%. Bleed ports were installed at the outlet of the HP compressor in order to create the main bleed.

2.3 NRC Program Approximations and Assumptions

The NRC ODP program required the following approximations:

1. component efficiencies remain constant at the DP values (constant polytropic efficiency for the compressors)

2. inlet mass flow is proportional to LP speed

3. cooling bleed air and auxilary power extraction are ignored.

4. nozzle coefficients, combustion pressure loss, and turbine efficiencies (DP values) were all selected to make the computer program agree with the experimental results at the high end of the LP speed range.

2.4 Tests

The object of the experimental work was to measure the major engine parameters while:

1. reducing LP speed (no main bleed) from 100% to 40% in steps of 10%.

2. reducing LP speed with average main bleeds of 3.6% and 5.9%

3. reducing LP speed with no main bleed, but increasing exit nozzle area 7.5%

4. combinations of 2 and 3.

It was then possible to compare the experimental values of the engine parameters with the computer predicted values.

2.5 Results

A summary of the results is given in figure 3 to 16 which compare the NRC computer predictions, the experimental values, and the Cranfield computer predictions. Note that a term which NRC used to measure the influence of main bleed on an engine parameter is the "influence coefficient" which is defined as:

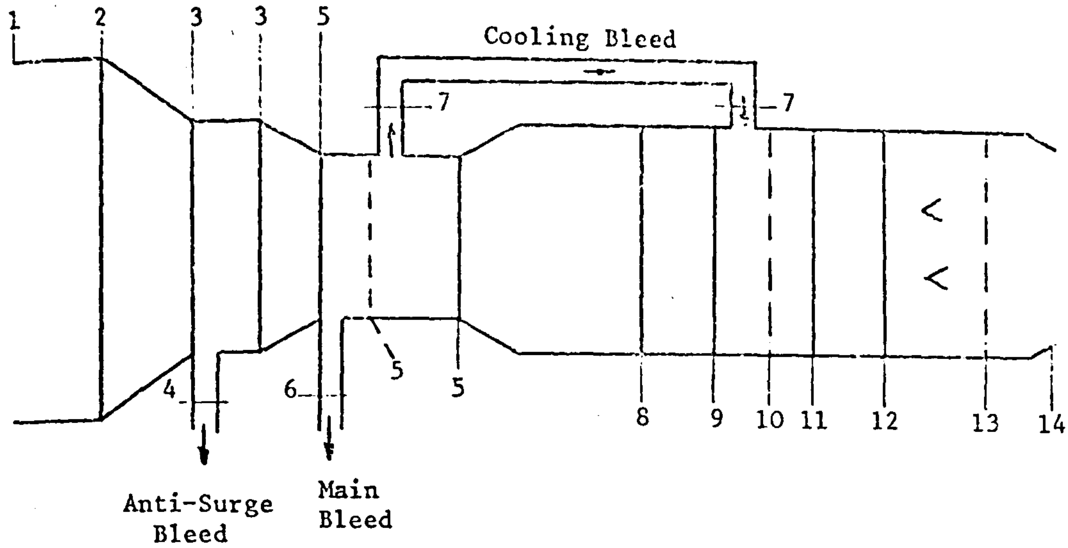
$$\text{influence coefficient} = \left[\frac{A_{MB\%} - A_{NO\ BLEED}}{A_{NO\ BLEED}}\right] \div MB\%$$

where A = any engine parameter
MB% = % main bleed

A similar influence coefficient to account for a change in exit nozzle area was also defined.

3. Cranfield ODP Program

3.1 An attempt was made to provide a more realistic simulation than the NRC program. Naturally, the use of component maps goes a long way to this end. In addition, HP cooling air bleed and jet pipe pressure loss were also incorporated into the engine simulation. The schematic of the engine and the computer "program" follows in figure 2. It will be seen that the program has been written such that six errors are produced for ODP cases. The LP speed is reset for each ODP situation with the TET varying to account for the change.



Engine Schematic; Numbers represent SV's

"Program" Codewords

BR	SV's	BD	EV's REQ.	VAR's	VAR 1	VAR 2	REMARKS
001	0102-1	0104					
002	0203-1	0509	-1-1-1	01-1	0205-1		
007	0304-1	1013					Anti Surge Bleed.
009	060302	0302	0402				W3 = W3-W4
002	0305-1	1418	-1-1-1	0203	0214-1	0215-1	
007	0506-1	1922					Main Bleed
009	060502	0502	0602				W5 = W5-W6
007	0507-1	2326					HP Cool Bleed
009	060502	0502	0702				W5 = W5-W7
003	0508-1	2729	-1-1-1	04-1	010806		
004	0809-1	3034	03-1-1	05-1	0232-1		
008	090710						
007	1011-1	3538					Pressure Loss
004	1112-1	3943	02-1-1	06-1	0241-1		
011	1213-1	4446					
005	131401	47					
006	01-1-1	-1-1	050104				

Figure 2 Engine Schematic and "Program" Codewords

3.2 As with the NRC program, the DP component operating points were chosen to give the best fit to the experimental data. The parameters chosen for the DP of the engine areas follows:

PARAMETER	ACTUAL ENGINE	NRC PROGRAM	CRANFIELD PROGRAM
LP COMPRESSOR			
η_{ISEN}	0.85 (avg)	0.85	0.825
PR	3.80	3.80	3.80
HP COMPRESSOR			
η_{ISEN}	0.90 (avg)	0.91	0.93
PR	3.09	3.09	3.09
ANTI SURGE BLEED FLOW	6.10% (avg) LP SPEED < 78%	6.10% LP SPEED < 78%	6.10% LP SPEED < 78%
COMBUSTOR			
η	(0.98)	0.98	0.985
TET	1187°K	1173°K	1187°K
$\Delta P/P$	(0.05)	0.05	0.05
HP TURBINE			
η	(0.85)	0.85	0.88
PRESS LOSS	0.	0.	0.
LP TURBINE			
η	(0.85)	0.85	0.88
PRESS LOSS	(0.06)	0.06	0.01
NOZZLE COEFFICIENT	(0.96)	0.96	CALCULATED IN PROGRAM
COOLING AIR	UNAVAILABLE	0.	5%
JET PIPE $\Delta P/P$	UNAVAILABLE	0.	0.06
INTAKE η	UNAVAILABLE	100%	98%

<u>Note:</u> Experimental values in parenthesis were either inferred from measurements of other parameters or were assumed.

TABLE 1 DESIGN POINT ENGINE PARAMETERS

3.3 Results

A summary of results is given in figures 3 to 16 which compare the NRC program predictions, experimental values, and Cranfield program predictions.

NOTE: In figures 11 and 12, for main bleed = 5.9%, LP speeds were restricted to the range 60% - 90%. This was due to temperature restrictions being encountered on the test engine. For the same reason (as shown in figures 15 and 16), LP speeds do not exceed 95% for the case of simultaneous increase in exit nozzle area and a main bleed of 95%

Also note that the discontinuity in most curves at LP speeds = 78% is caused by the anti-surge bleed valve opening at this speed.

3.3.1 Figures 3,4,6,7,8,9,10

In most cases, the Cranfield ODP program gave a better agreement with the experimental values than did the NRC ODP program. Also, where the NRC predictions show a linear relationship, the Cranfield predictions, while they may not agree exactly with the experimental values, at least show a properly shaped curve (with few exceptions).

3.3.2 Figures 5 and 6

As may be seen from these figures, experimental HP isentropic efficiency varied widely over the operating range. In fact, reference A states that the curve shown represents "smoothed" data. Nevertheless, the general trend of decreasing efficiency with engine speed is clearly shown. However, LP efficiency is another matter. The measured efficiency is seen to increase with decreasing engine speed contrary to the computer predictions. At low engine speeds, this discrepancy becomes quite large. For example at LP speed = 40%, the difference in measured versus predicted efficiencies is 17%. This helps to explain the difference in predicted versus measured values of turbine inlet temperature, and gross and specific thrust at low engine speeds. On the other hand, at 100% engine speed, the LP measured efficiency is quite low. This may account for the higher predicted values of gross and specific thrust, and lower specific fuel consumption at 100% speed.

This discrepancy in LP compressor predicted values is also seen in figure 6 for speeds of 80% - 100%. The LP pressure ratio falls more sharply than the experimental values. This could perhaps explain the difference in predicted versus measured values of gross and specific thrust in this speed range.

The logical explanation for these LP compressor discrepancies is that the component map used in the computer simulation does not accurately represent the actual characteristics. It would seem that the lines of constant efficiency should be shifted to the left of the design point so that the higher efficiencies are found at the lower speeds. This leads to a possible conclusion that the design point of the engine was not the sea level static condition assumed in both computer simulations.

3.3.3 Figures 11,12,13,14

These graphs show the separate effects of main bleed and exit nozzle area increase on turbine inlet temperature and specific thrust. It will be seen that the Cranfield predicted values for the most part lie between the NRC predictions and the experimental values, thus giving better agreement. However, while the shapes of the curves agree fairly well, there is usually a discrepancy between the values. This difference leads one to suspect that the simulation is not accurate. The main bleed was withdrawn from specially constructed ports in the test engine. Possibly withdrawing quantities of air from these ports upsets the internal air flow such that some new phenomona are not being accounted for in the simulation. Note that the two simulations agree more closely with each other than with the experimental data.

3.3.4 Figures 15,16

These graphs show the change in specific thrust and turbine entry temperature for combined area increase of 7.5% and a main bleed of 9%. Again, the Cranfield predictions lie between the experimental values and the NRC predictions. This time however, a much closer approximation was obtained.

4. Discussion

4.1 General

From the part throttle tests, where LP speeds were reduced gradually from 100% to 40%, it may be seen that the Cranfield ODP program predicted engine performance within an acceptable accuracy. Furthermore, if more specific information was available such as amount of cooling air bleed, nozzle jet pipe pressure losses, etc., and more particularly, more representative component maps for the two compressors, it is felt that an even more accurate simulation would have been produced.

4.2 Effects of Main Bleed Nozzle Area Increase

Unfortunately, the effects of main bleed extraction and nozzle area increase on the engine parameters were not as accurately determined as the part throttle results. The predicted values were closer to the experimentally measured values than were those of the NRC program, but the improvement was not too great. The question is, how accurate should the predictions be to be acceptable. NRC placed great emphasis on the previously mentioned "influence coefficients" and were disappointed when they were out by as large as a factor of two in some cases.

However, as an example, consider the change in turbine inlet temperature caused by a main bleed extraction of 3.6% as shown in figure 11. For the worst case at an LP speed of 40%, the measured temperature change was from 640°K to 673°K, a change of +5%. The Cranfield computer predicted change was from 700°K to 722°K, a change of +3%. In other words, the two changes varied by 2% in relation to the absolute temperatures. At higher LP speeds, this percentage variation is less. This would seem a more valid way of comparing the values rather than comparing the percentages and saying that since the actual change was 5% and the predicted change was 3%, therefore the predicted change was in error by 40%. This is what occurs if influence coefficients are compared. In other words, comparing influence coefficients may be an overly severe test of a program.

4.3 Summary

As mentioned in paragraph 3.3.2, because of discrepancies between predicted and measured values of LP compressor isentropic efficiency (particularly at low engine speeds), and LP pressure ratio at higher engine speeds, it would seem that the LP compressor map does not accurately reflect the actual component characteristics. Also, possibly a wrong choice has been made in assuming that sea level static conditions represent the engine design point. Although these two facts may seem small, they are not insignificant and it is considered that they represent the largest source of error in the simulation. A large discrepancy in efficiencies, for example, will mean discrepancies in parameters all the way through the engine. This fact alone, it is felt, is responsible for the over-estimation of turbine inlet temperatures at low engine speeds. It is not surprising then, that the response of some engine parameters to main bleed or nozzle area increases will not be predicted with the desired accuracy.

In short, it is felt that proper component operating lines have not been accurately enough established (largely due to insufficient information) for a simulation to be made with high accuracy. However, with the information available, it is considered that the simulation presented here has demonstrated that the program is capable of a proper analysis of an engine. The fact that the shape of most curves is correct is in itself proof. Also, it is considered that the changes in certain parameters in response to main bleed and/or nozzle area increase have been predicted with an acceptable accuracy considering the information available to simulate the engine.

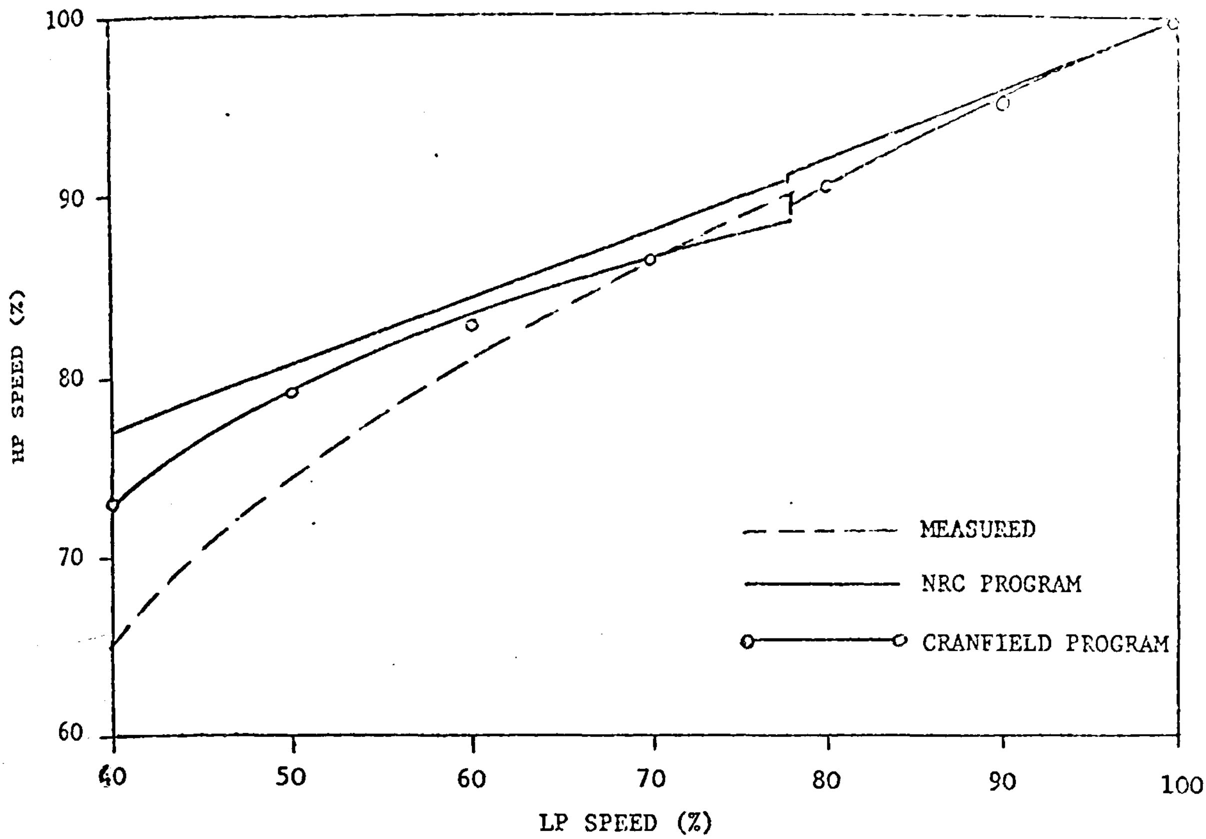


FIGURE 3 HP SPOOL SPEED VS. LP SPOOL SPEED

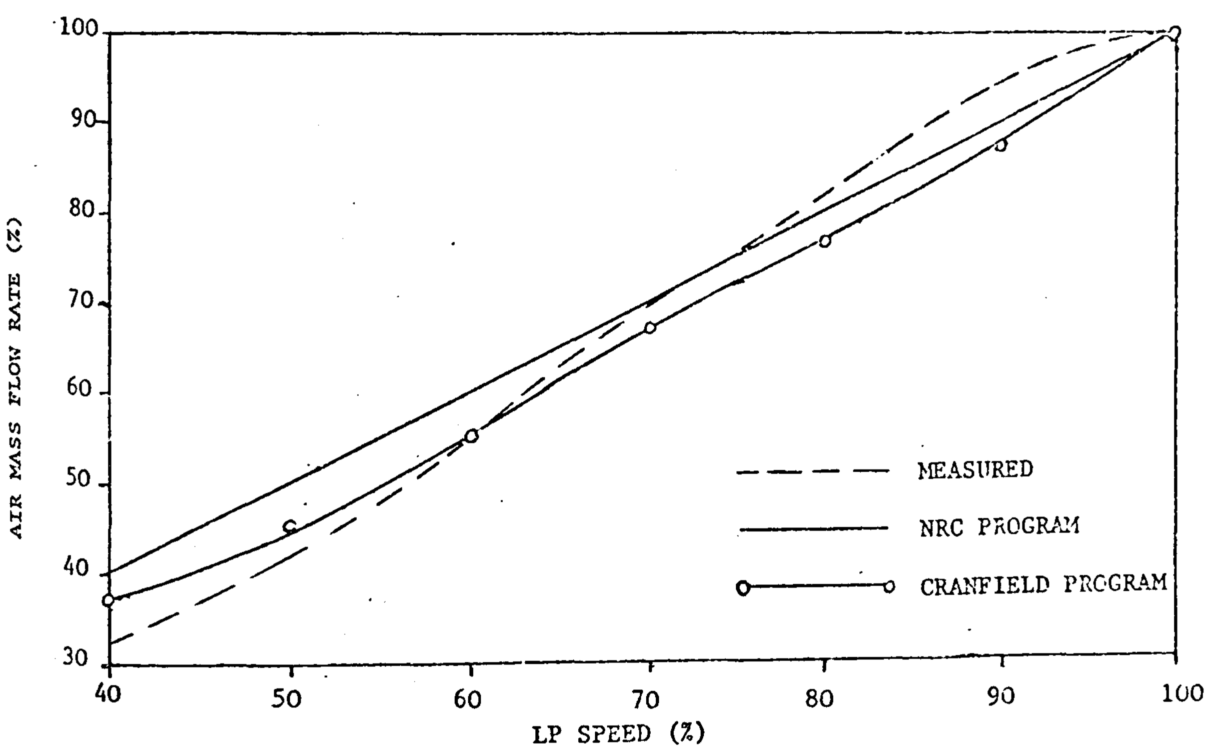


FIGURE 4 AIRMASS FLOW RATE AS A PERCENT OF DESIGN
VS. LP SPOOL SPEED

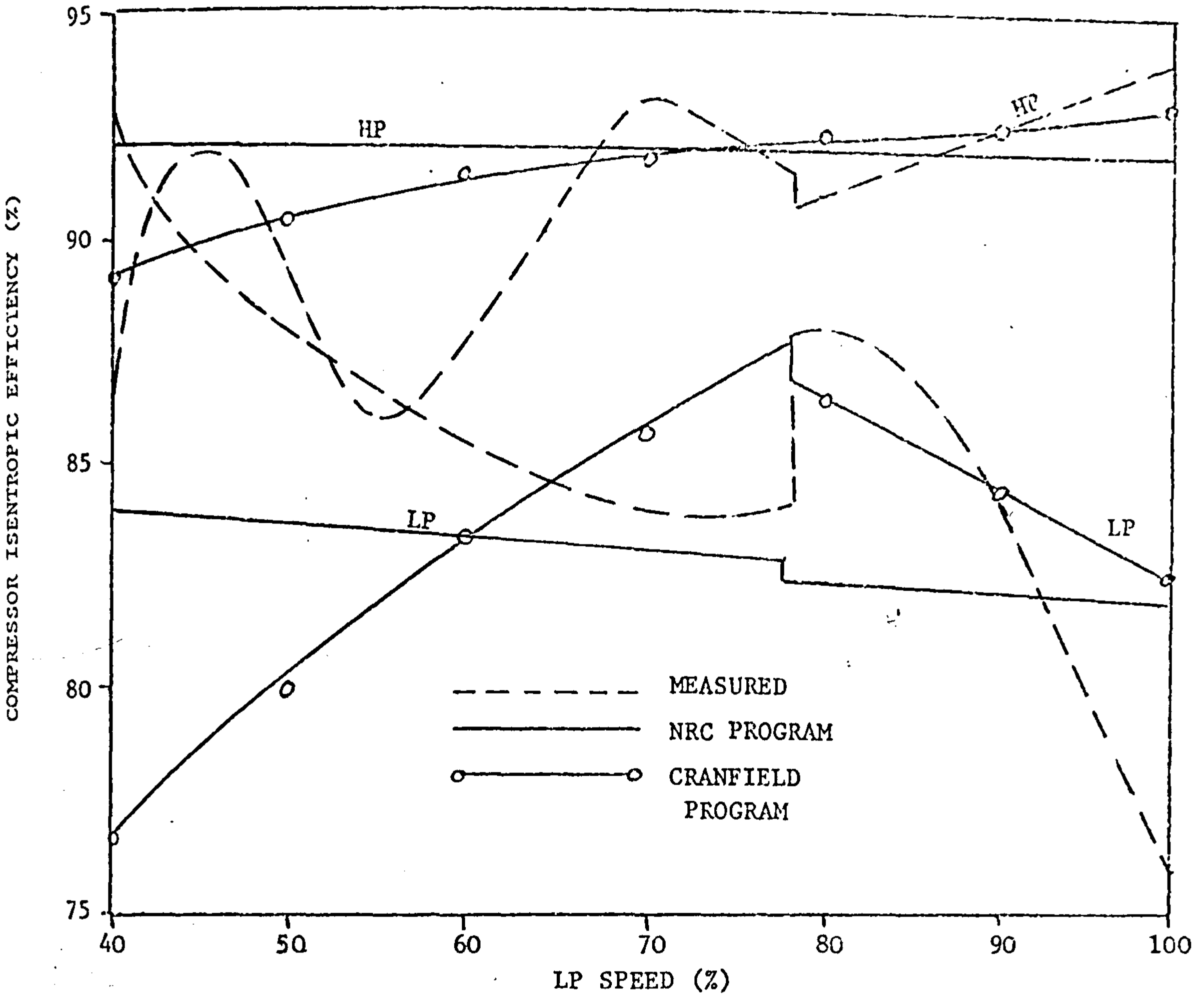


FIGURE 5 COMPRESSOR ISENTROPIC EFFICIENCY VS. LP SPOOL SPEED

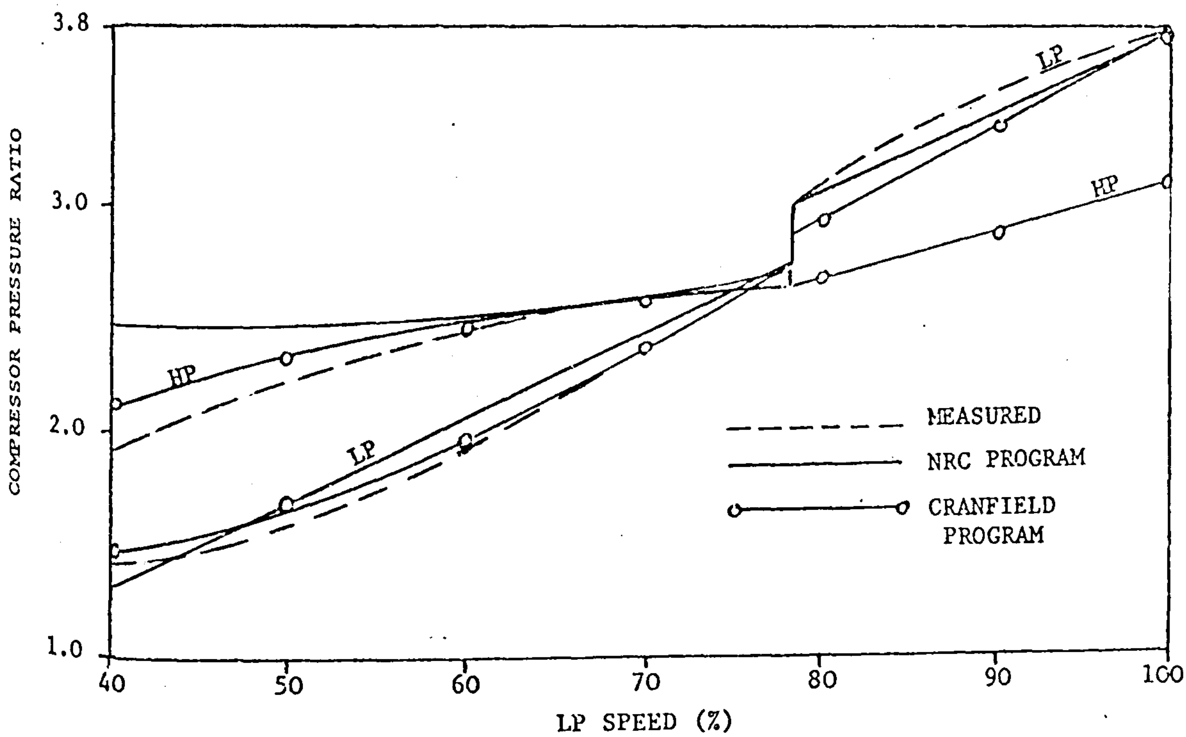


FIGURE 6 COMPRESSOR PRESSURE RATIO VS. LP SPOOL SPEED

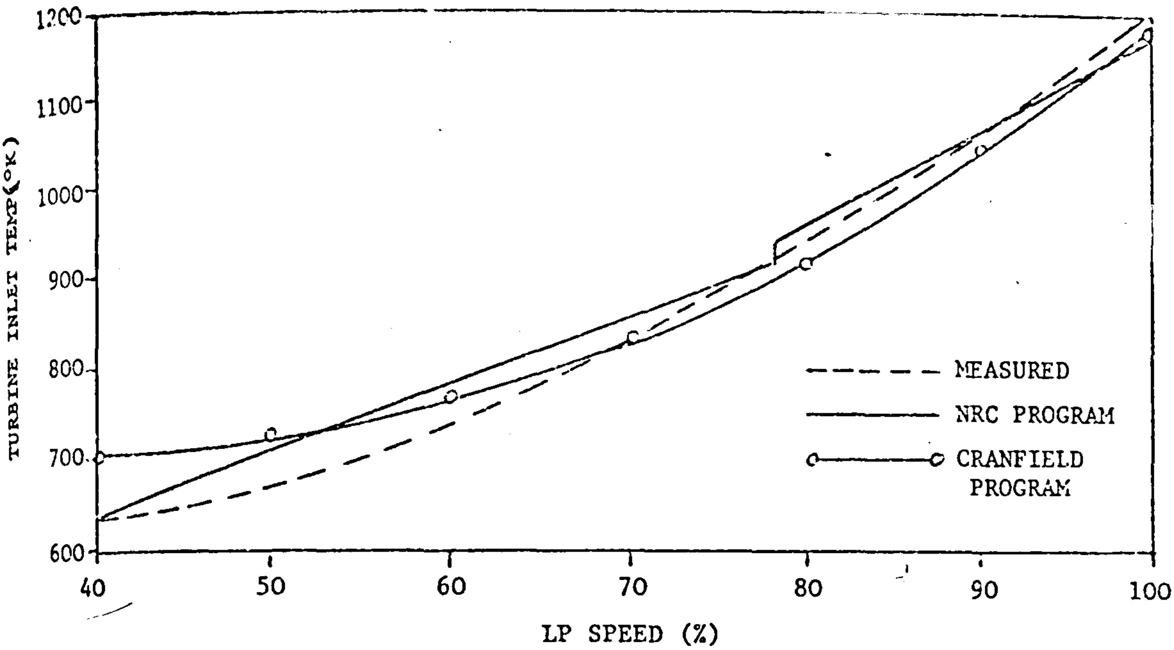


FIGURE 7 TURBINE INLET TEMPERATURE VS. LP SPOOL SPEED

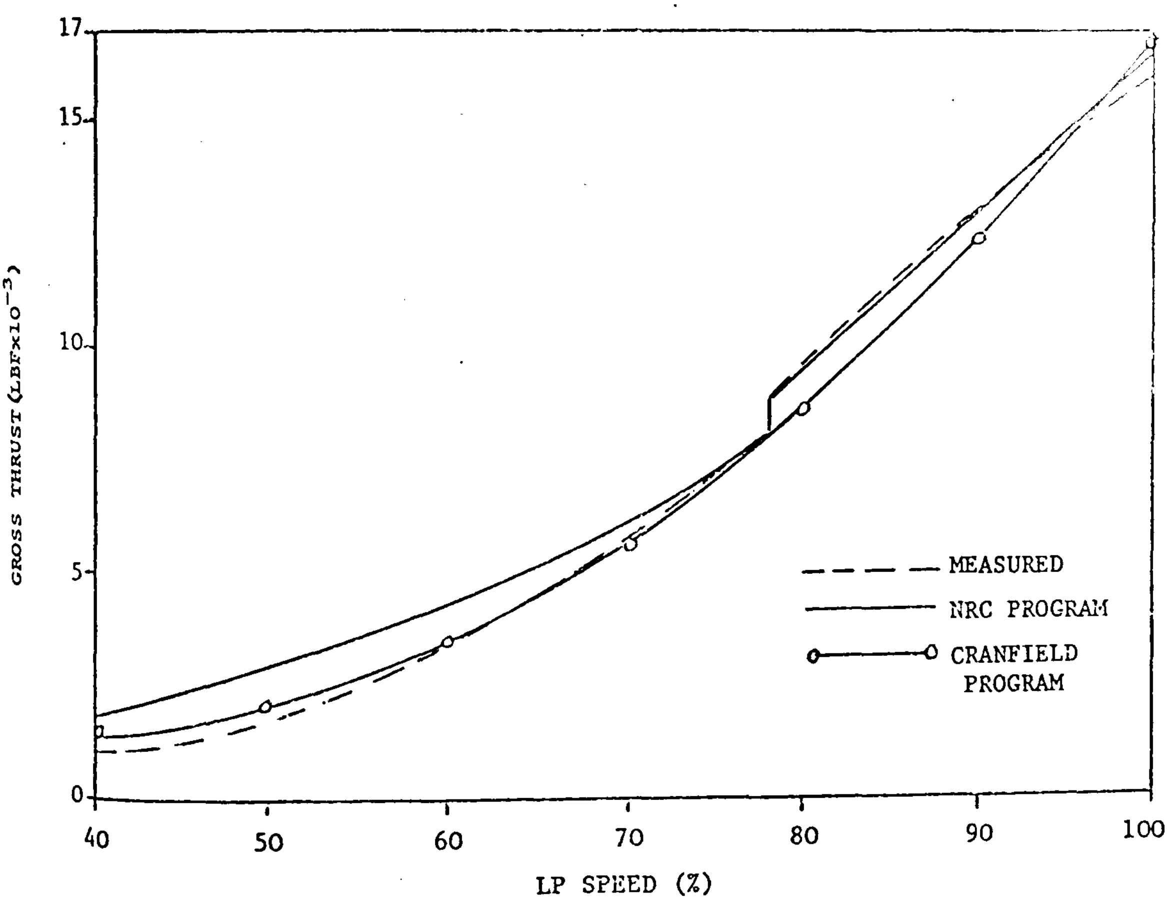


FIGURE 8 GROSS THRUST VS. LP SPOOL SPEED

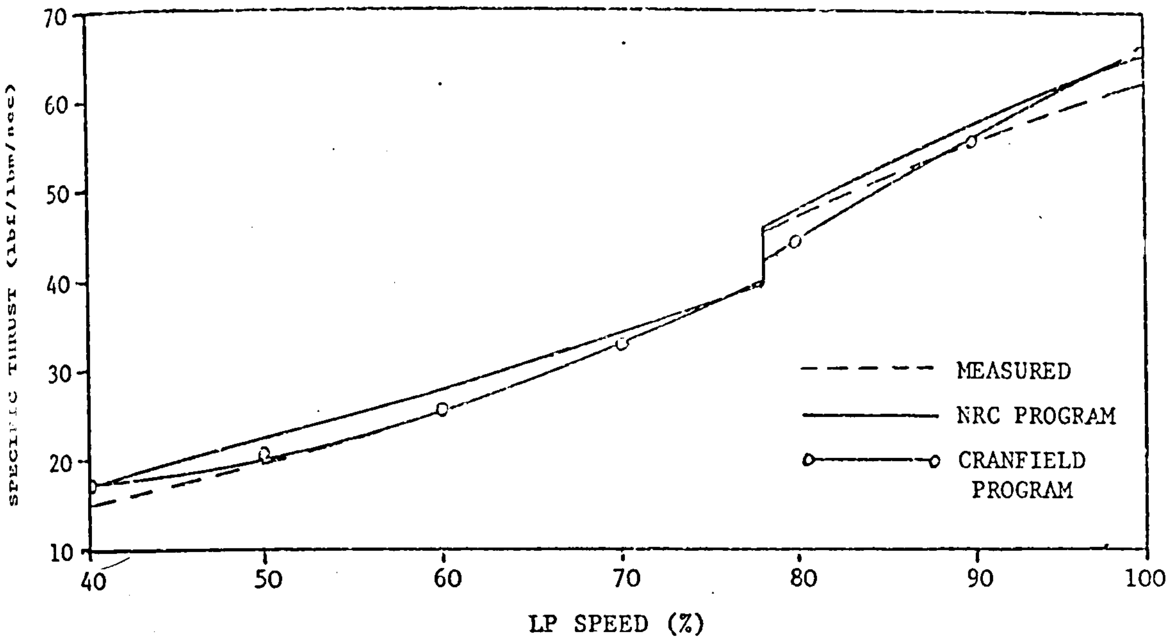


FIGURE 9 SPECIFIC THRUST VS. LP SPOOL SPEED

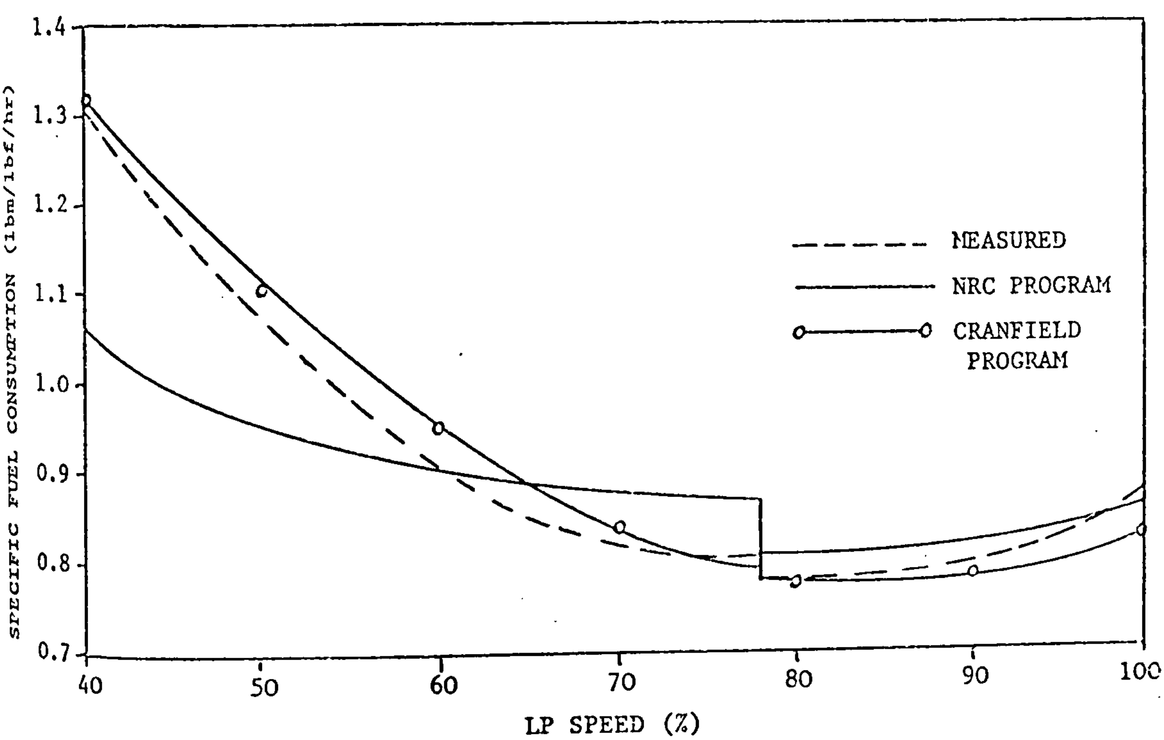
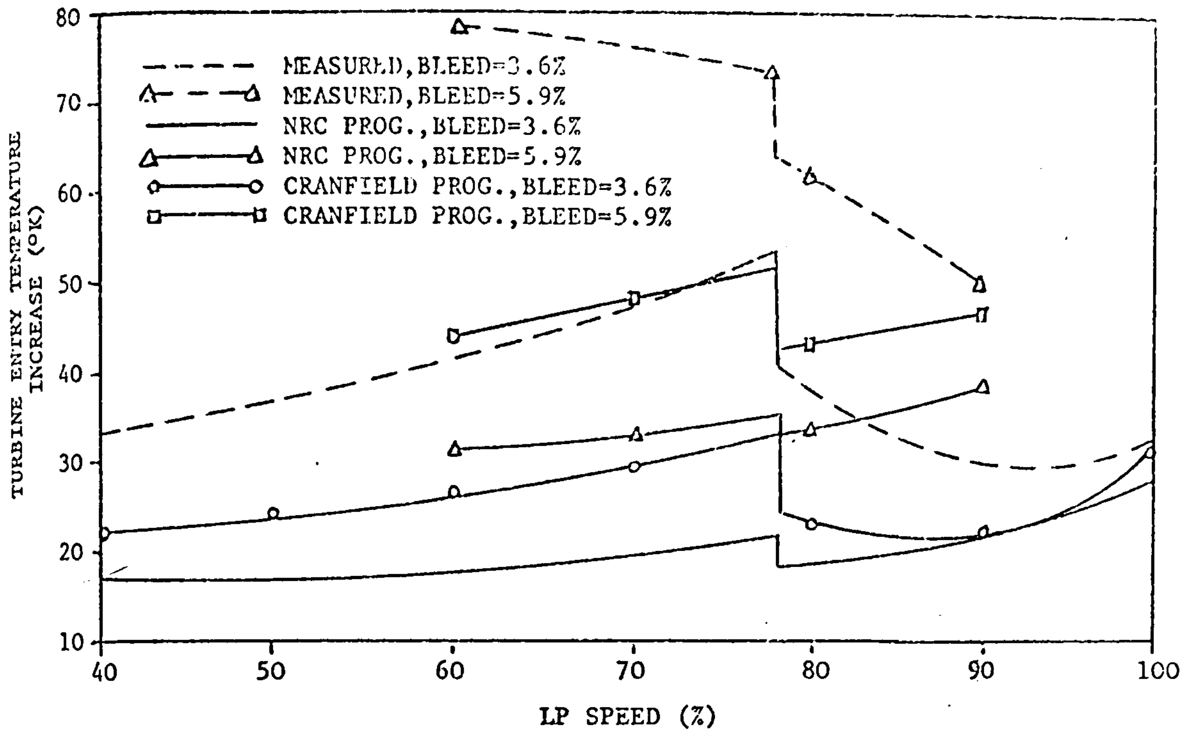


FIGURE 10 SPECIFIC FUEL CONSUMPTION VS. LP SPOOL SPEED



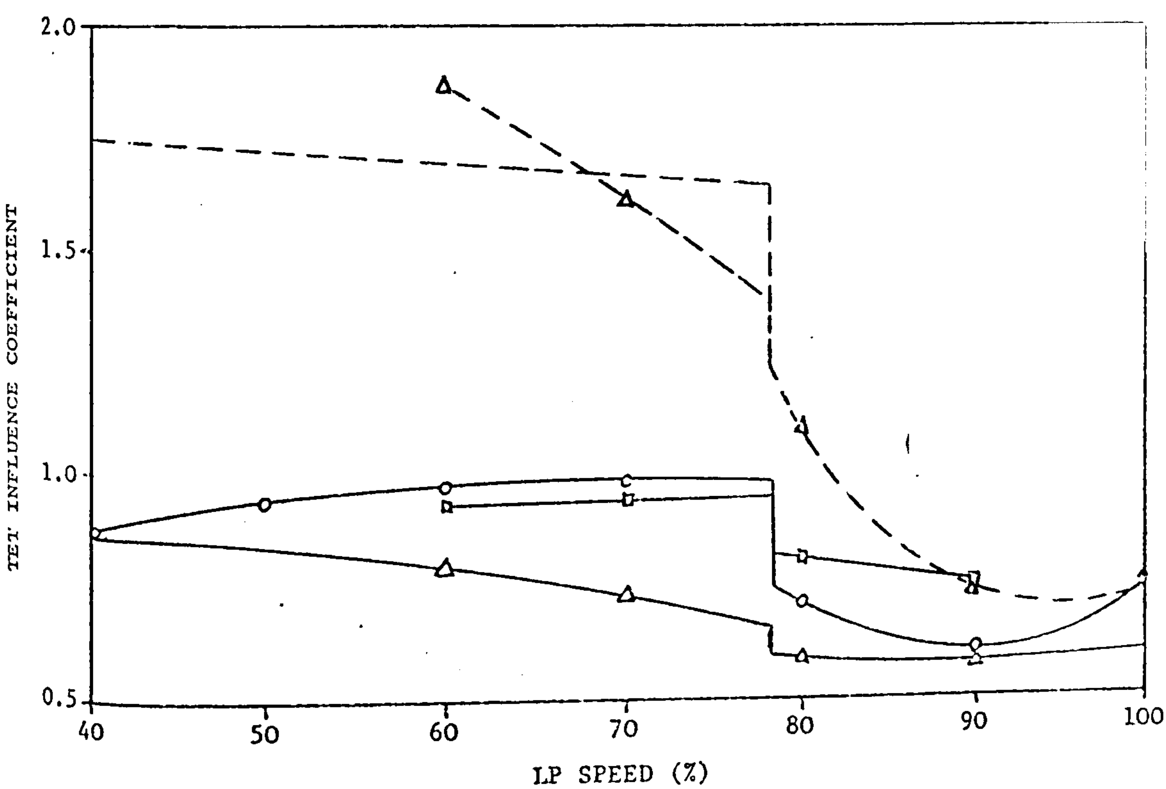


FIGURE 11 TURBINE INLET TEMPERATURE CHANGE VS. LP SPOOL SPEED, MAIN BLEED = 3.6%, 5.9%.

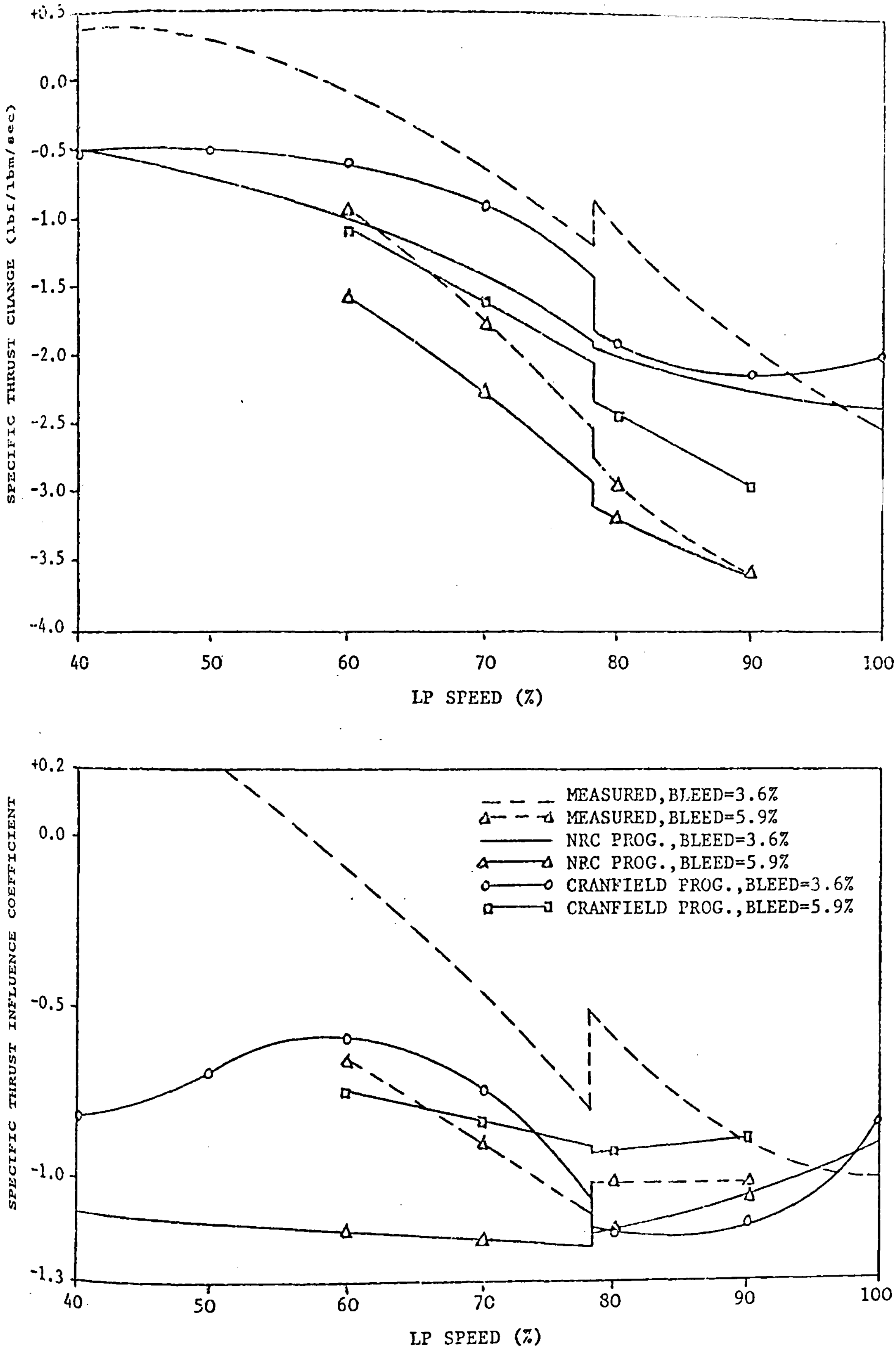


FIGURE 12 SPECIFIC THRUST CHANGE VS. LP SPOOL SPEED, MAIN BLEED = 3.6%, 5.9%

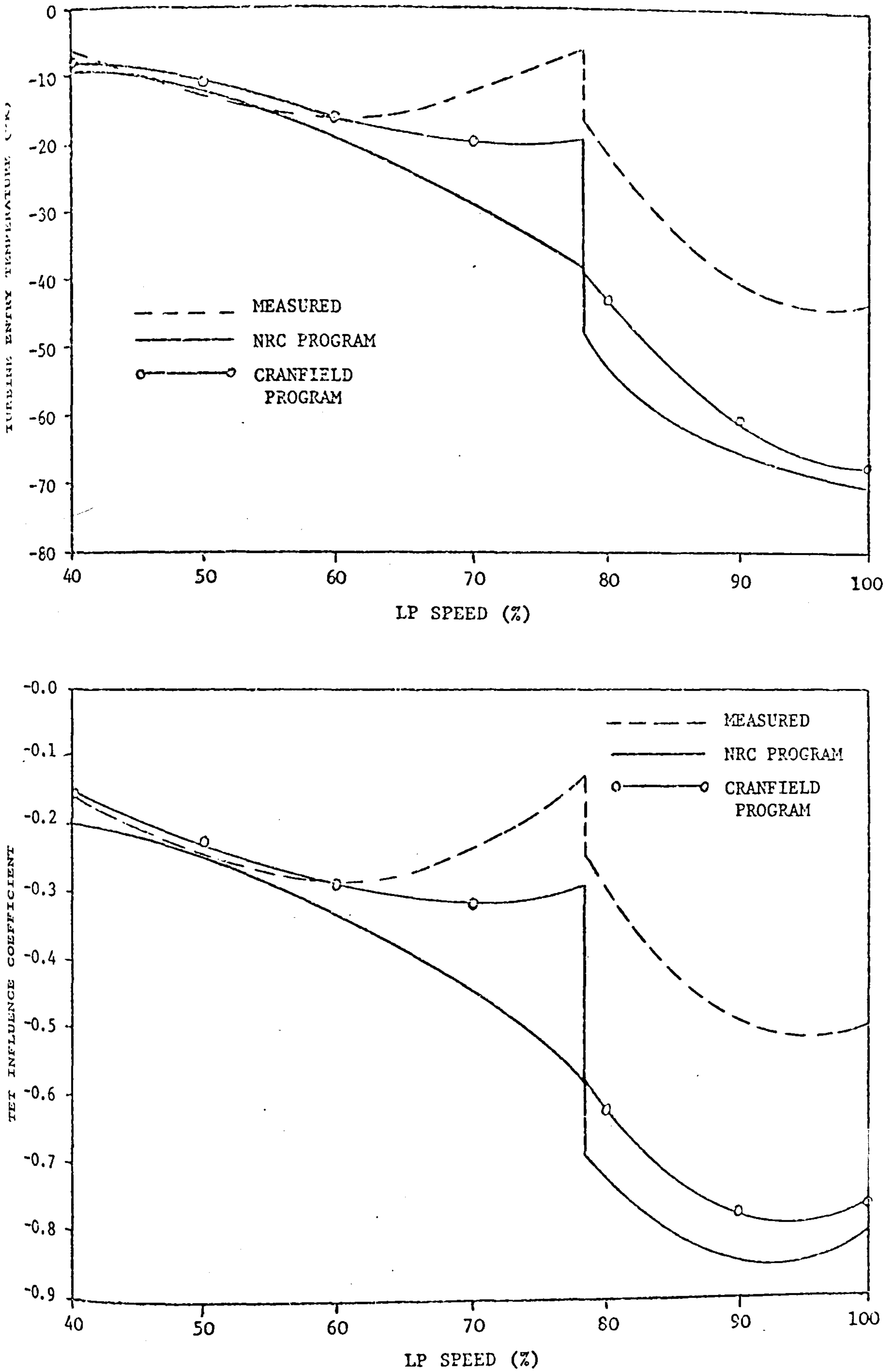


FIGURE 13 TURBINE INLET TEMPERATURE CHANGE VS. LP SPOOL SPEED, EXIT AREA CHANGE = +7.5%

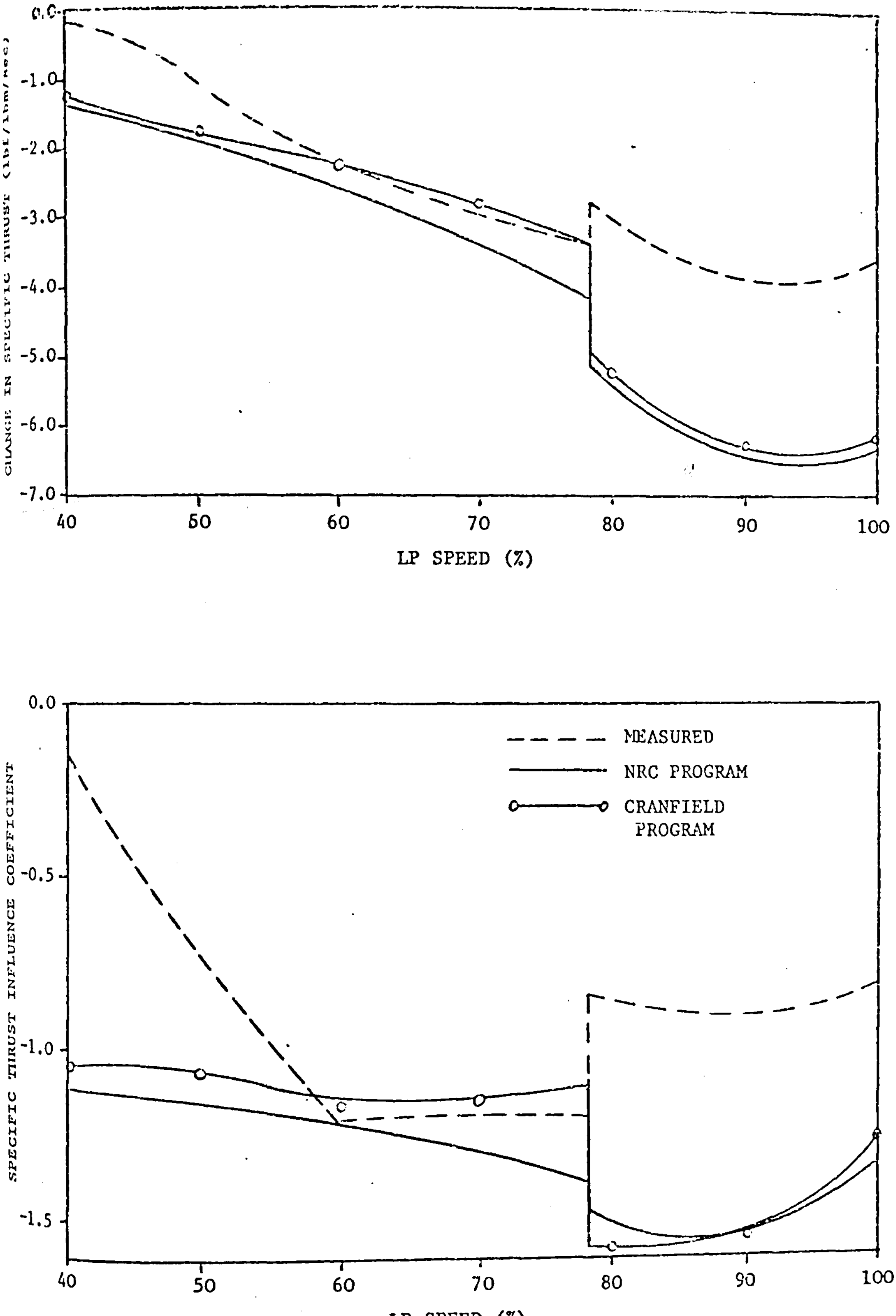


FIGURE 14 SPECIFIC THRUST CHANGE VS. LP SPOOL SPEED,
EXIT AREA CHANGE = +7.5%

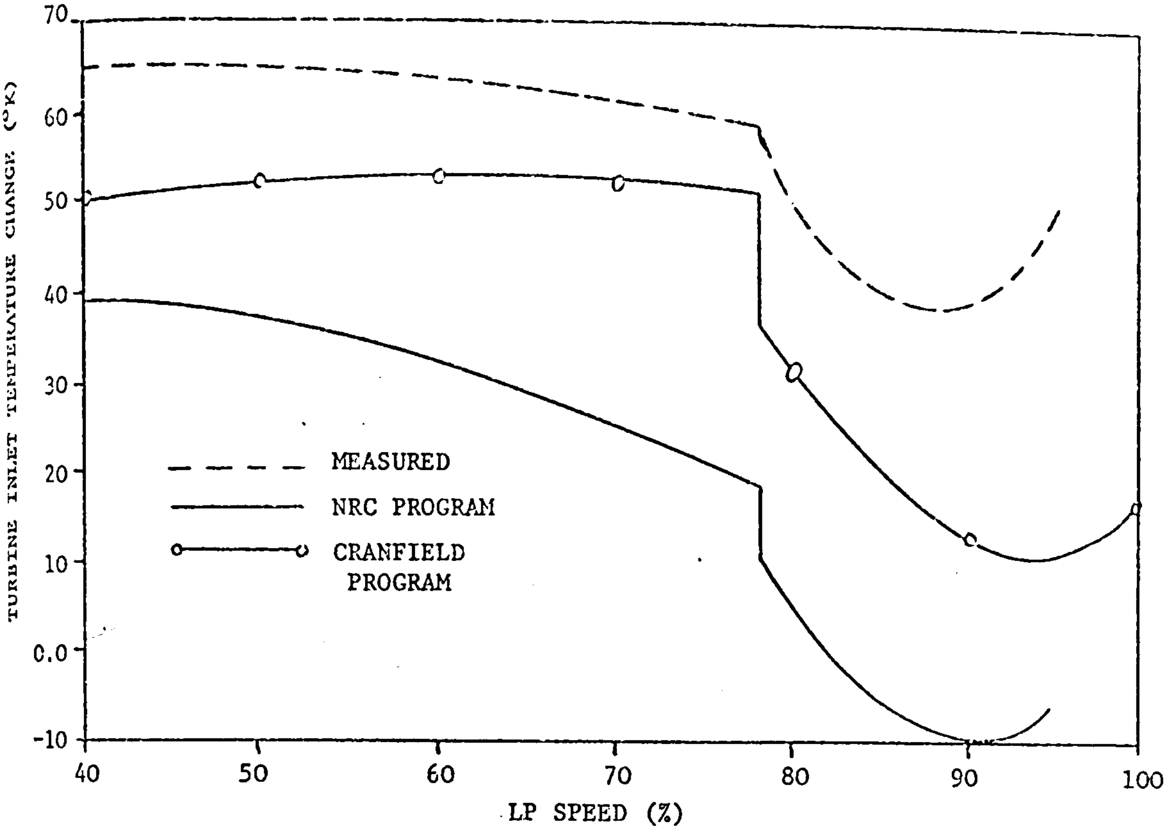


FIGURE 15 TURBINE INLET TEMPERATURE CHANGE VS. LP SPOOL SPEED, AREA CHANGE = +7.5%, MAIN BLEED = 9%

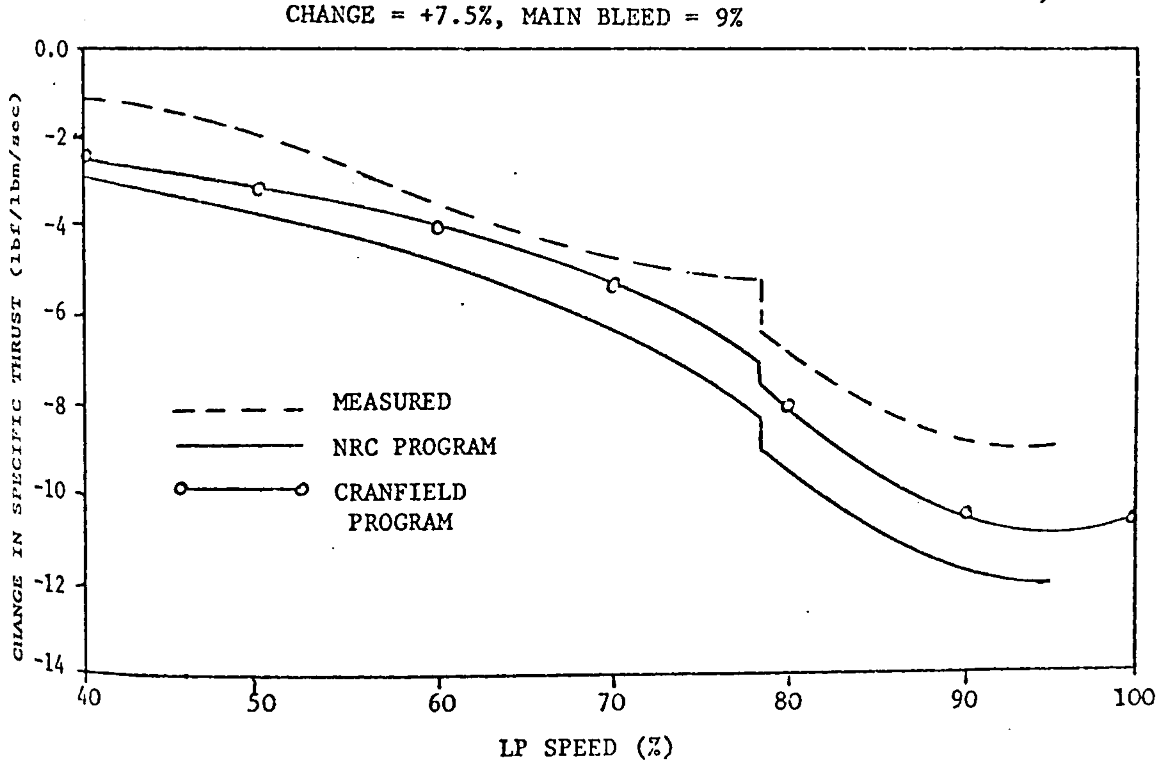


FIGURE 16 SPECIFIC THRUST CHANGE VS. LP SPOOL SPEED, AREA CHANGE = +7.5%, MAIN BLEED = 9%

Appendix E

Series/Parallel Engine Cycle Simulations

1. This appendix gives information on how two types of series/parallel variable cycle engines have been successfully simulated using the Cranfield computer program. The first engine is a two spool engine with series/parallel LP and HP fans. It corresponds to Engine 3 mentioned in section 5.2 of the main text. The second engine is three spool, with series/parallel LP fan and aft fan. This corresponds to Engine 4 of section 5.2.

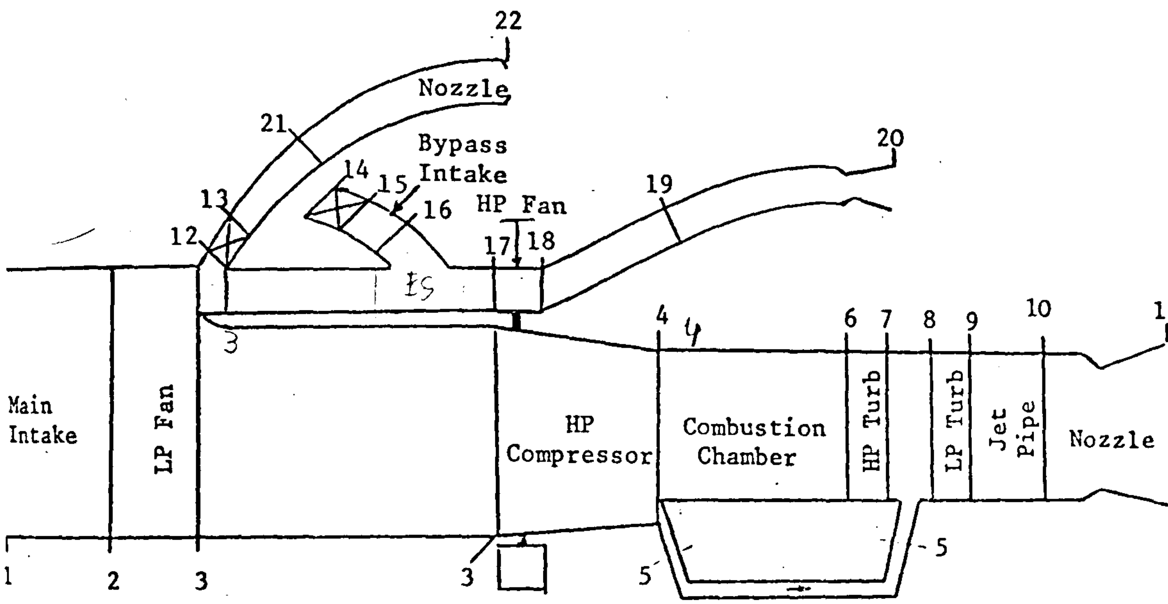
In addition, these simulations provide examples of two "programming" techniques mentioned in section 3.8.4 of the main text.

The first technique, mentioned in section 3.8.4, paragraph 5, is that of a boosted compressor and is found in the first engine. The HP fan is actually connected to the front stages of the HP compressor. It is simulated by considering it to be two separate compressors running on the same shaft but with separate air inlets and outlets. A separate component map is required for each part of this compressor. Two uses of Brick 9 set the shaft speeds (PCN) equal and also sum the values of work required (COMWK) before this value is input into the HP turbine brick.

The second technique, mentioned in section 3.8.4, paragraph 4 concerns an unusual engine inlet mass flow and is found in both engine examples at the air inlets to the HP fan and aft fan respectively. In these cases, the brick at the outlet of B1 is not B2 but is B8. In order for the mass flow of B1 to reflect the mass flow demands of the downstream compressors at ODP conditions, the value of the compressor mass flow is transferred back to the B1 outlet by means of Brick 9.

2. Series/Parallel LP and HP Fans

The schematic diagram for this simulation is shown below. The components between 12→13 and 14→15 represent on-off valves which are simulated by Brick 7 with λ(W) set either to 1.0 (valve open) or 0.0 (valve closed). Section 4.3.2 of the main text describes this type of engine and its simulation in greater detail.



The codewords necessary to simulate the engine are given on the following page. Note that eight variables are necessary to simulate this engine. The eight errors are generated in the following components for parallel operation.

Error	Component
1	HP compressor
2,3	HP turbine
4,5	LP turbine
6,7	Convergent/divergent nozzles
8	Convergent nozzle

In series operation, since only the two convergent/divergent nozzle are in use, an extra error must be generated. This is done by having the HP fan produce an error (set BD(22) = +1.0)

Thus, the engine will operate in series or parallel modes according to the following BD values.

	Parallel	Series	Remarks
BD(14)	+1.0	0.0	Controls on-off valve
BD(22)	-1.0	+1.0	Controls error production of HP fan

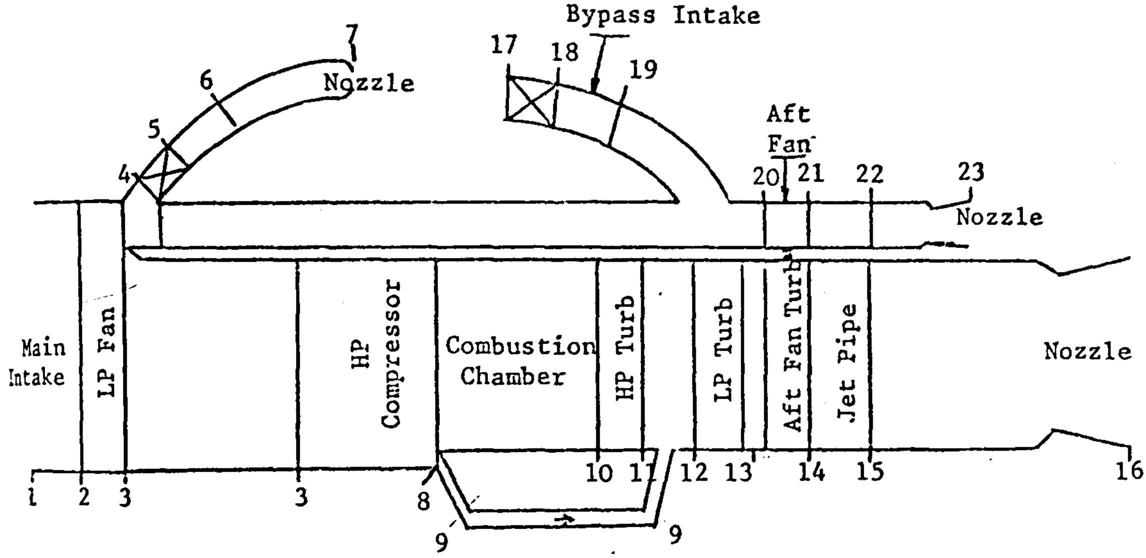
In order to perform the transition from one mode to the other, the "mini" codewords must reflect the above BD values.

The codewords required are:

BR	SV's	BD	EV's REQ.	VAR's	VAR 1	VAR 2	REMARKS
001	0102-1	0104					Main inlet
002	0203-1	0509	-1-1-1	0102	0205-1	0206-1	Variables are Ƶ and PCN
007	0312-1	1013	-1-1-1	03-1	0210-1		Variable is λ(W) so that bypass ratio varies
009	060302	0302	1202				W3=W3-W12
007	1213-1	1417					First on-off valve
009	061202	1202	1302				W12=W12-W13
007	1415-1	1417					Second on-off valve
001	1516-1	0104					Bypass inlet
008	161217						"Y" junction
002	1718-1	1822	-1-1-1	0405	0218-1	0219-1	Variables are Ƶ and PCN
009	050024	0019	0013				BD(24)=BD(19)+BD(13) (note: BD(13)=0.0)
002	0304-1	2327	-1-1-1	06-1	0223-1		Variable is Ƶ. The speed of this compressor, BD(24)=BD(19), is already a variable.
007	0405-1	2931					HP cool bleed
009	060402	0402	0502				W4=W4-W5
003	0406-1	3234					
009	05-105	-105	-104				Sums work of compressors 2+3
004	0607-1	3539	05-1-1	07-1	0237-1		Variable is TF
008	050708						
004	0809-1	4044	02-1-1	08-1	0242-1		Variable is TF
011	0910-1	4547					
012	101101	4849					
011	1819-1	5052					
012	192001	5354					
011	1321-1	5557					
005	212201	58					
009	05-107	-107	-108				} Sums XC of the
009	05-107	-107	-109				three nozzles
009	05-101	-101	-103				
009	051602	1802	0013				Sets: W16=W18+BD(13)
006	0105-1	-1-1	070106				

3. Series/Parallel Aft Fan

This simulation is similar to the previous engine, the main difference being that the engine has three shafts with the third turbine driving an aft fan. The on-off valves required for cycle transition operate between stations 4→5 and 17→18.



In this simulation, ten variables are necessary. The ten errors are generated by the following components for parallel operation.

Error	Component
1.2	HP turbine
3,4	LP turbine
5,6	Aft fan turbine
7	Aft fan (compressor)
8,9	Convergent/divergent nozzles
10	Convergent nozzle

As with the previous simulation, series or parallel operation is controlled by the following BD values

	Parallel	Series
BD(14)	+1.0	0.0
BD(22)	−1.0	+1.0

The codewords required are:

BR	SV's	BD	EV's REQ.	VAR's	VAR 1	VAR 2	REMARKS
001	0102-1	0104					Main inlet
002	0203-1	0509	-1-1-1	0102	0205-1	0206-1	Variables are Z and PCN
007	0304-1	1013	-1-1-1	03-1	0210-1		Varies bypass ratio
009	060302	0302	0402				W3=W3-W4
007	0405-1	1417					First on-off valve
009	060402	0402	0502				W4=W4-W5
002	0308-1	1822	-1-1-1	0405	0218-1	0219-1	Variables are Z and PCN
007	0809-1	2326					HP cool bleed
009	060802	0802	0902				W8=W8-W9
003	0810-1	2729					
004	1011-1	3034	03-1-1	06-1	0232-1		Variable is TF
008	110812						
004	1213-1	3539	02-1-1	07-1	0237-1		Variable is TF
007	1718-1	1417					Second on-off valve
001	1819-1	0104					Aft fan inlet
009	05-101	-101	-105				Sums XR of two inlets
008	041920						
002	2021-1	4044	-1-1-1	0809	0240-1	0241-1	Variables are Z and PCN
004	1314-1	4549	06-1-1	10-1	0247-1		Variable is TF
011	1415-1	5052					
012	151601	5354					
011	2122-1	5557					
012	222301	5859					
011	0506-1	6062					
005	060701	63					
009	05-107	-107	-108				Sums XG of three
009	05-107	-107	-109				nozzles
009	051902	2102	0013				Sets: W19=W21+BD(13)
006	0118-1	-1-1	070104				

4. The component parameters at cruise conditions of the four engines compared in section 5.2 of the main text (including the two variable engines described in this appendix) are as follows.

	Engine 1 (Turbojet)	Engine 2 (Turbofan)	Engine 3 (LP+HP Fans)	Engine 4 (LP+Aft Fans)
Intake Pressure Recovery *	92.5%	92.5%	92.5%	92.5%
LP Fan: PR η	- -	1.99 83.0%	1.70 88.6%	1.72 88.6%
HP or Aft Fan: PR η	- -	- -	1.25 81.2%	1.21 79.9%
HP Compressor: PR η	9.68 82.9%	5.27 80.3%	5.92 81.2%	6.02 81.5%
Combustion: η $\Delta P/P$	98.0% 5.8%	98.0% 5.5%	98.0% 5.7%	98.0% 5.7%
HP Turbine η	90.6%	90.0%	90.0%	90.0%
LP Turbine η	-	89.0%	89.2%	89.2%
Aft Fan Turbine η	-	-	-	89.6%
Nozzle Velocity Coefficient: * Core Bypass	 0.945 -	 0.970 0.970	 0.970 0.970	 0.970 0.970
Bypass Ratio	0.0	0.915	0.915	0.895
HP Cooling Bleed	8.0%	8.0%	8.0%	8.0%

* indicates quantities internally generated by the computer program

Appendix F

Computer Program Listing

```
      MASTER MASTER SEGMENT
      INTEGER SA,SB,SC,CW(45,17),RR,DES,SBMAX,REMAP,FINISHED
      DIMENSION MEXT(5),MCOMP(5),MTURB(5),MCONVRG(5),MDUCT(5),
     1MCONDI(5),MBURN(5)
      DIMENSION EV(20),BD(90),SV(25,8),D(5)
      DIMENSION A(10),DEL(10),NPC(4),V(5),LP(5),LOOPDO(5)
      DIMENSION VAR(10),EROR(10,10),BERR(10),DVAR(10),VALUE(5),
     1E(10,10),DELVAR(10),NCW(5,3)
      DIMENSION UNITSWORDS(8,2)
      COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
     1UNITSPOWER,UNITSMASS
      COMMON/COMP1/CN1(10),PR1(10,5),WAC1(10,5),ETA1(10,5)
      COMMON/COMP2/CN2(10),PR2(10,5),WAC2(10,5),ETA2(10,5)
      COMMON/COMP3/CN3(10),PR3(10,5),WAC3(10,5),ETA3(10,5)
      COMMON/TURB1/TF1(10),CNT1(10,6),DH1(10,6),ETAT1(10,6)
      COMMON/TURB2/TF2(10),CNT2(10,6),DH2(10,6),ETAT2(10,6)
      COMMON/TURB3/TF3(10),CNT3(10,6),DH3(10,6),ETAT3(10,6)
      COMMON/INITIAL/IKOUNT,NOW,SBMAX,KOUNT,NOMIS,MISMAT,NLOOP,
     1NUMAP,FINISHED,REMAP
      DATA SV/200*-1./,TOLNCE/0.005/,DES/1/,LP/5*0/,MAXLOOPS/0/
      DATA UNITSWORDS/6HFEET  ,8H FEET   ,7HLBM/SEC,6H  FEET,
     17HLBF    ,7HLBM/LBF,7HLBF/LBM,7HCHU/SEC,6HMETERS,8H METERS ,
     27HKG/SEC ,6HMETERS,7HNEWTONS,7H   KG/N,7H   N/KG,7HWATTS  /
      LOOPING=-1
      WRITE(2,114)
      READ(1,100)IDES,KN,KUNITS
      WRITE(2,115)IDES,KN
      READ(1,100)NI,ND,ISV
      WRITE(2,116)NI,ND,ISV
      READ(1,126)NCMAP,NTMAP
      I=0
      READ(1,127)(CN1(M),M=1,10),((PR1(M,N),WAC1(M,N),ETA1(M,N),
     1N=1,5),M=1,10)
      I=I+1
      IF(I.EQ.NCMAP)GOTO40
      READ(1,127)(CN2(M),M=1,10),((PR2(M,N),WAC2(M,N),ETA2(M,N),
     1N=1,5),M=1,10)
      I=I+1
      IF(I.EQ.NCMAP)GOTO40
      READ(1,127)(CN3(M),M=1,10),((PR3(M,N),WAC3(M,N),ETA3(M,N),
     1N=1,5),M=1,10)
40    I=0
      READ(1,128)(TF1(M),M=1,10),((CNT1(M,N),DH1(M,N),ETAT1(M,N),
     1N=1,6),M=1,10)
      I=I+1
      IF(I.EQ.NTMAP)GOTO41
      READ(1,128)(TF2(M),M=1,10),((CNT2(M,N),DH2(M,N),ETAT2(M,N),
     1N=1,6),M=1,10)
      I=I+1
      IF(I.EQ.NTMAP)GOTO41
      READ(1,128)(TF3(M),M=1,10),((CNT3(M,N),DH3(M,N),ETAT3(M,N),
     1N=1,6),M=1,10)
41    CALL RESET
      GOTO(80,81)KUNITS
80    UNITSAREA=1.
      UNITSFORCE=1.
      UNITSLENGTH=1.
      UNITSMASS=1.
      UNITSPOWER=1.
      GOTO89
```

```
81    UNITSAREA=.092903
      UNITSFORCE=4.4482
       UNITSLENGTH=.3048
      UNITSMASS=.45359237
      UNITSPOWER=.52754/3600.
89    READ(1,107)((CW(I,N),N=1,17),I=1,NI)
      WRITE(2,118)((CW(I,N),N=1,17),I=1,NI)
99    DO19I=1,ND
      READ(1,102)N,BD(N)
19    WRITE(2,119)N,BD(N)
      DO1I=1,ISV
      READ(1,103)NSV,NSVI,SV(NSV,NSVI)
      WRITE(2,120)NSV,NSVI,SV(NSV,NSVI)
1     IF(NSVI.EQ.2)SV(NSV,2)=SV(NSV,2)/UNITSMASS
      CALL TIME(TIMENOW)
      WRITE(2,101)TIMENOW
      WRITE(2,174)
      WRITE(2,123)(UNITSWORDS(I,KUNITS),I=1,8)
      WRITE(2,105)
15    K=1
20    L=0
      I=1
      NCOMP=0
      NTURB=0
      NCONVRG=0
      NDUCT=0
      NBURN=0
      NCONDI=0
      NEXT=0
14    NER=0
      R1=-1.
      R2=-1.
      R3=-1.
      R4=-1.
      BR=CW(I,1)
      SA=CW(I,2)
      SB=CW(I,3)
      IF(SB.LT.SBMAX)GOTO51
      SBMAX=SB
51    SC=CW(I,4)
      K1=CW(I,5)
      K2=CW(I,6)
      IF(BR.EQ.9)GOTO17
      IF(K1.LE.0)GOTO17
58    IF(K2.LE.0)GOTO18
      NN=1
      DO2N=K1,K2
      IF(BR.EQ.2.AND.NN.EQ.2)NPCN(NCOMP+1)=N
66    D(NN)=BD(N)
2     NN=NN+1
18    D(1)=BD(K1)
      IF(NOW.GT.0)GOTO59
17    NE1=CW(I,7)
      NE2=CW(I,8)
      NE3=CW(I,9)
      IF(NE1.LE.0.OR.BR.EQ.9)GOTO5
      E1=EV(NE1)
      IF(NE2.LE.0)GOTO5
      E2=EV(NE2)
      IF(NE3.LE.0)GOTO5
```

```
      F3=EV(NE3)
5     CONTINUE
      IF(IDES.GT.0.OR.FINISHED.EQ.1)GOTO39
      M=12
21    IF(CW(I,M).LE.0)GOTO39
      IF(DES.LE.0.AND.REMAP.LE.0)GOTO32
      MX=CW(I,M)
      M1=CW(I,M+1)
      GOTO(22,23)MX
22    M2=CW(I,M+2)
      VAR(K)=SV(M1,M2)
      GOTO27
23    VAR(K)=BD(M1)
27    K=K+1
      IF(M.EQ.15)GOTO39
      M=M+3
      GOTO21
32    IF(KOUNT.EQ.1)GOTO39
52    IF(IKOUNT.NE.1)GOTO57
      DO53I=1,NI
      IF(CW(I,10).EQ.K)GOTO55
      IF(CW(I,11).EQ.K)GOTO56
53    CONTINUE
      GOTO54
57     IF(CW(I,10).EQ.K)GOTO55
       IF(CW(I,11).EQ.K)GOTO56
      GOTO39
56    M=15
55    MX=CW(I,M)
      M1=CW(I,M+1)
      GOTO(37,38)MX
37.   M2=CW(I,M+2)
      SV(M1,M2)=VAR(K)
      GOTO36
38    BD(M1)=VAR(K)
      IF(BR.EQ.2.AND.M1.EQ.K1)K5=K
      IF(IKOUNT.EQ.1)GOTO54
      NOW=1
      GOTO58
59    NOW=-1
36    IF(IKOUNT.EQ.1)GOTO54
      IF(KOUNT.NE.3)GOTO39
      K=K+1
      GOTO32
39    CONTINUE
      GOTO(301,302,303,304,305,306,307,308,309,310,311,312,313,
     1314)BR
301   D(1)=D(1)/UNITSLENGTH
      CALL B1(D(1),D(2),D(3),SV(SA,7),SV(SA,5),SV(SA,3),SV(SA,6),
     1SV(SA,4),SV(SA,1),SV(SA,2),SV(SB,6),SV(SB,4),SV(SB,1),
     2SV(SB,2),D(4),R1,DES,FINISHED)
      GOTO13
302    NCOMP=NCOMP+1
      IF(D(1).LE.1.)GOTO3020
      BD(K1)=.999
      VAR(K5)=BD(K1)
      WRITE(2,3021)K5,BD(K1)
      GOTO15
3021  FORMAT(9H NEW VAR(,I3,4H) = ,F7.5)
```

```
3020  CALLB2(D(1),D(2),SV(SA,6),SV(SA,4),SV(SA,1),SV(SA,2),D(3),
     1D(4),DES,SV(SB,6),SV(SB,4),SV(SB,2),SV(SB,1),R1,NCOMP,NER,
     2ER1,NCODE,FINISHED,D(5),E1)
      GOTO13
303   CALL B3( DES,SV(SA,6),SV(SA,4),D(1),SV(SA,2),SV(SA,1),D(2),
     1SV(SB,6),SV(SB,4),SV(SB,2),R1,SV(SB,1),NCODE,FINISHED,D(3))
      GOTO13
304   NTURB=NTURB+1
      NCOMTURB=IFIX(D(5))
      N=NPCN(NCOMTURB)
      E2=BD(N)
      D(1)=D(1)/UNITSPOWER
      CALL B4(SV(SA,6),SV(SA,4),SV(SA,1),SV(SA,2),E1,D(1),D(2),
     1D(3),D(4),SV(SB,6),SV(SB,4), E2,DES,NCODE,SV(SB,1),SV(SB,2)
     2,NER,ER1,ER2,NTURB,REMAP,NUMAP,FINISHED)
      IF(REMAP.LE.0.OR.NCODE.EQ.7)GOTO13
      BD(N)=E2
      DO67N=1,4
      IF(CW(I-N,1).EQ.3)GOTO68
67    CONTINUE
68    SV(SA-N+1,6)=SV(SA,6)
      K21=K2-1
      N=1
      DO72NN=K1,K21
      BD(NN)=D(N)
72    N=N+1
      GOTO69
305   NCONVRG=NCONVRG+1
      CALL B5(SV(SA,1),SV(SA,6),SV(SA,4),SV(SC,3), DES,SV(SA,2),
     1SV(SB,7),SV(SB,3),SV(SB,8),ICON,R1,NER,ER1,SV(SB,4),SV(SB,6)
     2,SV(SB,5),SV(SB,1),SV(SB,2),NCONVRG,NCODE,FINISHED,D(1))
      IF(NCODE.EQ.5)GOTO332
      GOTO13
306   IF(DES.LE.0.AND.FINISHED.LE.0)GOTO200
      AIRIN=SV(SA,2)
      IF(SB.GT.0)AIRIN=AIRIN+SV(SB,2)
      IF(SC.GT.0)AIRIN=AIRIN+SV(SC,2)
      XG=E1*UNITSFORCE
      XR=E2*UNITSFORCE
      XN=XG-XR
      WFB=E3*UNITSMASS.
      SFC=WFB/XN*3600.
      SXN=XN/AIRIN/UNITSMASS
      GOTO200
307   D(2)=D(2)/UNITSMASS
      SV(SB,2)=D(1)*SV(SA,2)+D(2)
      SV(SB,4)=D(3)*SV(SA,4)+D(4)
      SV(SB,6)=SV(SA,6)
      SV(SB,1)=SV(SA,1)
      GOTO270
308   CALLB8(SV(SA,1),SV(SB,1),SV(SC,1),SV(SA,2),SV(SB,2),SV(SC,2)
     1,SV(SA,4),SV(SC,4),SV(SA,6),SV(SB,6),SV(SC,6),SV(SB,4))
      GOTO270
309   IF(SB.EQ.-1)OP1=EV(SC)
      IF(SB.EQ.0)OP1=BD(SC)
      IF(SB.GT.0)OP1=SV(SB,SC)
      IF(K1.LT.0)OP2=EV(K2)
      IF(K1.EQ.0)OP2=BD(K2)
      IF(K1.GT.0)OP2=SV(K1,K2)
      IF(NE2.LE.0)GOTO4
```

```
3     IF(NE1.EQ.-1)OP3=EV(NE2)
      IF(NE1.EQ.0)OP3=BD(NE2)
      IF(NE1.GT.0)OP3=SV(NE1,NE2)
4     CALL B9(SA,OP1,OP2,OP3,R1)
      IF(R1.GE.0)GOTO13
      IF(SB.EQ.-1)EV(SC)=OP1
      IF(SB.EQ.0)BD(SC)=OP1
      IF(SB.GT.0)SV(SB,SC)=OP1
      GOTO270
310   N=IFIX(D(1))
      N=NPCN(N)
      PCN=BD(N)
      Z=BD(N-1)
      CALL B10(SV(SA,1),SV(SB,1),SV(SC,1),SV(SA,2),SV(SB,2),
     1SV(SC,2),SV(SA,3),SV(SB,3),SV(SC,3),SV(SA,4),SV(SB,4),
     2SV(SC,4),SV(SA,5),SV(SB,5),SV(SC,5),SV(SA,6),SV(SB,6),
     3SV(SC,6),SV(SA,7),SV(SB,7),SV(SC,7),SV(SA,8),SV(SC,8),
     4SV(SC,8),DES,NCODE,FINISHED,PCN,Z,D(2),D(3),REMAP,NER,ER1,
     5AM1,AM2,AM3)
      IF(DES.GT.0.OR.FINISHED.GT.0)WRITE(2,129)SA,AM1,SB,AM2,S3,
     1AM3
      IF(REMAP.LE.0.OR.NCODE.EQ.7)GOTO13
      BD(N)=PCN
      BD(N-1)=Z
      GOTO69
311   NDUCT=NDUCT+1
      IF(D(1).GT.0.)NBURN=NBURN+1
      CALL B11(SV(SA,1),SV(SB,1),SV(SA,2),SV(SB,2),SV(SA,4),
     1SV(SB,4),SV(SA,6),SV(SB,6),DES,FINISHED,NDUCT,NBURN,D(1),
     2D(2),D(3),R1)
      GOTO13
312   NCONDI=NCONDI+1
      CALL B12(SV(SA,1),SV(SB,1),SV(SA,2),SV(SB,2),SV(SA,3),
     1SV(SB,3),SV(SA,4),SV(SB,4),SV(SA,5),SV(SB,5),SV(SA,6),
     2SV(SB,6),SV(SB,7),SV(SB,8),DES,NCODE,FINISHED,NER,ER1,
     3SV(SC,3),R1,D(1),NCONDI,D(2))
      GOTO13
313   IF (IDES.LE.0)GOTO270
      IF(SA.LT.MAXLOOPS)GOTO3131
      MAXLOOPS=SA
3131  LOOPING=1
      LP(SA)=LP(SA)+1
      MCONVRG(SA)=NCONVRG
      MTURB(SA)=NTURB
      MCOMP(SA)=NCOMP
      MCONDI(SA)=NCONDI
      MBURN(SA)=NBURN
      MDUCT(SA)=NDUCT
      MEXT(SA)=NEXT
      N=IFIX(D(1))
      NN=IFIX(D(2))
      IF(N.LE.0)GOTO75
      V(SA)=SV(N,NN)
      GOTO76
75    V(SA)=BD(NN)
76    CALL B13(V(SA),D(3),D(4),D(5),LP(SA),I,LOOPDO(SA))
      IF(N.LE.0)GOTO77
      SV(N,NN)=V(SA)
      GOTO270
77    BD(NN)=V(SA)
```

```
      GOTO270
314   NTURB=NTURB+1
      D(2)=D(2)/UNITSPOWER
      CALL B4(SV(SA,6),SV(SA,4),SV(SA,1),SV(SA,2),0.,D(2),D(5),
     1D(3),D(4),SV(SB,6),SV(SB,4),D(1),DES,NCODE,SV(SB,1),SV(SB,2)
     2,NER,ER1,ER2,NTURB,0,0,FINISHED)
      IF(DES.LE.0.AND.FINISHED.LE.0)GOTO13
      D(2)=D(2)*UNITSPOWER
      E2=E1*3600./D(2)*UNITSMASS
      WRITE(2,1314)D(1),D(2),E2
1314  FORMAT(36H ADDITIONAL FREE TURBINE PARAMETERS;,/9H SPEED =
     2,F5.1,1H%,10X,8HPOWER = ,E12.5,5X,6HSFC = ,E12.5/)
      GOTO13
13    IF(NCODE.EQ.7)GOTO331
      IF(R1.GE.0.)GOTO8
      GOTO12
8     NEXT=NEXT+1
      EV(NEXT)=R1
      IF(R2.GE.0.)GOTO9
      GOTO12
9     NEXT=NEXT+1
      EV(NEXT)=R2
      IF(R3.GE.0.)GOTO10
      GOTO12
10    NEXT=NEXT+1
       EV(NEXT)=R3
      IF(R4.GE.0.)GOTO11
      GOTO12
11    NEXT=NEXT+1
      EV(NEXT)=R4
12    NUMAP=0
200   CONTINUE
      IF(IDES.GT.0.OR.DES.GT.0.OR.NER.EQ.0.OR.FINISHED.EQ.1)GOTO270
      IF(KOUNT.EQ.1.OR.KOUNT.EQ.3)GOTO400
      L=L+1
      EROR(L,K)=ER1
28    IF(NER.EQ.1)GOTO270
      L=L+1
      EROR(L,K)=ER2
      GOTO270
400   L=L+1
      BERR(L)=-ER1
      IF(NER.EQ.1)GOTO270
      L=L+1
      BERR(L)=-ER2
270   I=I+1
      IF(I.LE.NI)GOTO14
      IF(KOUNT.EQ.0.OR.FINISHED.EQ.1)GOTO330
      IF(KOUNT.EQ.2)GOTO71
      DO73LL=1,KN
73    WRITE(2,173)LL,BERR(LL)
224   DO24LL=1,KN
      IF(ABS(BERR(LL)).GT.TOLNCE)GOTO70
24    CONTINUE
      FINISHED=1
      WRITE(2,106)NLOOP
      GOTO15
70    CONTINUE
C *** NOW STARTS OFF DESIGN
71    DES=-1
```

```
      IF(KOUNT.NE.0)GOTO30
      WRITE(2,121)
      READ(1,108)NC
      WRITE(2,122)NC
      DO74I=1,NC
      READ(1,109)(NCW(I,N),N=1,3),VALUE(I)
74    WRITE(2,110)(NCW(I,N),N=1,3),VALUE(I)
      WRITE(2,174)
69    DO45I=1,NC
      IF(NCW(I,1).EQ.2)GOTO46
      NSV=NCW(I,2)
      NSVI=NCW(I,3)
      SV(NSV,NSVI)=VALUE(I)
      IF(NSVI.EQ.2)SV(NSV,2)=VALUE(I)/UNITSMASS
      IF(NSVI.EQ.8)SV(NSV,8)=VALUE(I)/UNITSAREA
      GOTO45
46    NBD=NCW(I,2)
      BD(NBD)=VALUE(I)
45    CONTINUE
      KOUNT=1
      GOTO15
30    IF(KOUNT.EQ.3.AND.MISMAT.GT.0)GOTO61
      IF(KOUNT .EQ.2)GOTO42
      IF(NLOOP.LT. 20)GOTO65
      WRITE(2,125)NLOOP
      FINISHED=1
      GOTO15
65    KOUNT=2
      DO50K1=1,KN
      A(K1)=1.
      IF(VAR(K1).LE.1.)A(K1)=100.
      IF(VAR(K1).GE.1000.)A(K1)=0.1
      DEL(K1)=0.001*A(K1)*VAR(K1)
50    DVAR(K1)=DEL(K1)/A(K1)
      VAR(1)=VAR(1)-DVAR(1)
      GOTO15
42    VAR(K)=VAR(K)+DVAR(K)
      IF(K.EQ.KN)GOTO43
      M=12
      IKOUNT=1
      GOTO52
54    K=K+1
      IKOUNT=-1
      VAR(K)=VAR(K)-DVAR(K)
      GOTO20
43    CONTINUE
      DO44K1=1,KN
      DO44K2=1,KN
44    E(K2,K1)=(-BERR(K2)-EROR(K2,K1))/DEL(K1)
62    CALL MATRIX(E,DELVAR,BERR,KN)
      NLOOP=NLOOP+1
      VMTAVE=0.
      KBIG=0
      VARBIG=0.
      DO60K=1,KN
      ABSVAR=ABS(DELVAR(K))
      VMTAVE=VMTAVE+ABSVAR
      IF(ABSVAR.LE.0.1*VAR(K)*A(K))GOTO60
      IF(ABSVAR.LE.VARBIG)GOTO60
      KBIG=K
```

```
      VARBIG=ABSVAR
60    CONTINUE
      VMTAVE=VMTAVE/FLOAT(KN)
      VRATIO=1.0
      IF(KBIG.GT.0)VRATIO=0.1*VAR(KBIG)*A(KBIG)/VARBIG
      IF(MISMAT.GT.0)GOTO63
      IF(NOMIS.EQ.0)MISMAT=1
      GOTO64
63    MISMAT=MISMAT+1
      IF(VMTAVE.LT.0.85*VMTAVX)GOTO64
      IF(MISMAT.LT.4)NOMIS=1
      MISMAT=0
      GOTO65
64    WRITE(2,164)NLOOP
      DO47K=1,KN
      DELVAR(K)=VRATIO*DELVAR(K)/A(K)
47    VAR(K)=VAR(K)+DELVAR(K)
      IKOUNT=-1
      KOUNT=3
      GOTO15
61    VMTAVX=VMTAVE
      GOTO62
330   WRITE(2,112)
      DO16M=1,SBMAX
      IF(SV(M,2).GT.0.1)GOTO1615
      DO1615N=1,7
      SV(M,N)=0.
1615  CONTINUE
      SV(M,2)=SV(M,2)*UNITSMASS
      SV(M,7)=SV(M,7)*UNITSLENGTH
      SV(M,8)=SV(M,8)*UNITSAREA
16    WRITE(2,104)M,(SV(M,N),N=1,8)
      WRITE(2,113)XG,XR,XN,WFB,SFC,SXN
      CALL TIME(TIMENOW)
      WRITE(2,101)TIMENOW
      WRITE(2,174)
      DO1616M=1,SBMAX
      SV(M,2)=SV(M,2)/UNITSMASS
      SV(M,7)=SV(M,7)/UNITSLENGTH
1616  SV(M,8)=SV(M,8)/UNITSAREA
      IF(IDES.GT.0)GOTO332
      CALL RESET
      GOTO71
332   IF(LOOPING.LE.0)GOTO331
      DO333M=1,MAXLOOPS
      N=MAXLOOPS-M+1
      IF(LOOPDO(N).GT.0)GOTO334
333   LP(N)=0
      GOTO331
334   I=LOOPDO(N)
      NCONDI=MCONDI(N)
      NBURN=MBURN(N)
      NDUCT=MDUCT(N)
      NCONVRG=MCONVRG(N)
      NTURB=MTURB(N)
      NCOMP=MCOMP(N)
      NEXT=MEXT(N)
      GOTO14
331   CONTINUE
100    FORMAT(3I0)
```

```
101    FORMAT(90X,10H TIME NOW ,A8,/1H1)
102    FORMAT(I0,F0.0)
103    FORMAT(2I0,F0.0)
104    FORMAT(4X,I2,5X,F7.4,5X,F6.2,3X,F6.3,3X,F6.3,2X,F6.1,3X,F6.1,2X,
      1F6.1,2X,F7.4)
105    FORMAT(1H1,//5X,39H ***** DESIGN POINT ENGINE CALCULATIONS,
      66H *****///)
106    FORMAT(1H1,//5X,39H ***** OFF DESIGN ENGINE CALCULATIONS. ,
      116HCONVERGED AFTER ,I3,12H LOOPS *****///)
107    FORMAT(I3,16I2)
108    FORMAT(I0)
109    FORMAT(3I2,F0.0)
110    FORMAT(1X,3I2,E17.5)
112    FORMAT(//47H  STATION  FUEL/AIR  MASS FLOW  PSTATIC  PTOTAL,
      132H  TSTATIC  TTOTAL   VEL     AREA)
113    FORMAT(//17H  GROSS THRUST = ,F8.2,/17H MOMENTUM DRAG = ,
      1F8.2,/4X,13HNET THRUST = ,F8.2,/4X,14HFUEL BURNED = ,F8.3/
      217H          SFC =  ,F8.6, /,17H     SP THRUST = ,F8.3)
114    FORMAT(/26H1INPUT ''PROGRAM'' FOLLOWS)
115    FORMAT(1X,2I2)
116    FORMAT(1X,3I3)
118    FORMAT(1X,I3,16I2)
119    FORMAT(1X,I3,E15.6)
120    FORMAT(1X,2I3,E15.6)
121    FORMAT(/36H CHANGED CONDITIONS CODEWORDS FOLLOW)
122    FORMAT(1X,I2)
123    FORMAT(//////17X,38H THE UNITS FOR THIS RUN ARE AS FOLLOWS,
      1//48H TEMPERATURE = DEGREE K   PRESSURE = ATMOSPHERES ,
      212H   LENGTH = ,A6,12H   AREA = SQ,A8,//13H MASS FLOW = ,A7
      3,6X,11HVELOCITY = ,A6,16H/SEC     FORCE = ,A7,9H   SFC = ,A7
      4,3H HR,//12H SP THRUST = ,A7,4H/SEC,16X,8HPOWER = ,A7)
125    FORMAT(48H NO CONVERGENCE. ENGINE HAS NOT CONVERGED AFTER ,
      1I3,49HLOOPS. LAST SERIES OF ENGINE CALCULATIONS FOLLOWS/)
126    FORMAT(2I0)
127    FORMAT(10F0.0,150F0.0)
128    FORMAT(10F0.0,180F0.0)
129    FORMAT(32H ***** MIXING MACH NUMBERS ***** ,/9H STATION ,I2,
      15H,M= ,F5.3,3X,9H STATION ,I2,5H,M = ,F5.3,4X,8HSTATION ,I2,
      25H,M = ,F5.3/)
164    FORMAT(/6H LOOP ,I3)
173    FORMAT(6H BERR(,I1,4H) = ,E14.5)
174    FORMAT(48H ***********************************************)
       STOP
       END


END OF SEGMENT, LENGTH  5105, NAME  MASTERSEGMENT

       SUBROUTINE B1(Z,DEV,AM,V,TAS,PAS,TA,PA,FA,WA,TB,PB,FB,WB,
      1ETAR,XR,IDES,LASTONE)
       COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
      1UNITSPOWER,UNITSMASS
       IF(WA.GT..1)GOTO6
       WB=WA
       XR=0.00001
       RETURN
6      IF(Z.GT.36089.)GOTO1
       TAS=288.-0.0019813*Z
       PAS=(TAS/288.)**5.2546
       TAS=TAS+DEV
```

```
       GOTO2
1     TAS=216.5+DEV
      PAS=0.2233*EXP((36089.-Z)/20796.)
2     SP=TRM(0.,TAS,1)
      V=AM*SQRT(SP/(SP-0.068567)*0.068567*TAS*45071.595)
      TA=TAS*1.14
      HA=TRM(0.,TAS,2)+V*V/90143.19
      CALL AIRTAB(1,0.,TA,HA)
      PA=PAS*EXP(TRM(0.,TA,3)-TRM(0.,TAS,3))
      IF(ETAR.GT.0.)GOTO3
      IF(AM.GT.1.)GOTO4
      ETAR=1.
      GOTO3
4     IF(AM.GT.5.)GOTO5
      ETAR=1.-0.075*((AM-1.)**1.35)
      GOTO3
5     ETAR=800./((AM**4)+935.)
3     PB=ETAR*PA
      TB=TA
      FA=0.
      FB=FA
      IF(WB.GE.0.)WA=WB
      XR=WA*V/32.174049
      WB=WA
      IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
      ZUNITS=Z*UNITSLENGTH
      WRITE(2,100)ZUNITS,DEV,AM,ETAR
100   FORMAT(41H ***** AMBIENT AND INLET PARAMETERS *****,/7H ALT =
     1,F8.1,9X,11H ISA DEV = ,F8.3,7X,10HMACH NO = ,F5.2,9X,
     27HETAR = ,F6.4/)
      RETURN
      END


END OF SEGMENT, LENGTH   340, NAME  B1


      SUBROUTINE B2(ZC,PCN,TA,PA,FA,WA,PR,ETA,IDES,TB,PB,WB,FB,
     1COMUK,NCOMP,NER,ER1,NCODE,LASTONE,EROR)
      DIMENSION PRCF(3),ETACF(3),WACF(3),WLH(2),TRATD(3)
      COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
     1UNITSPOWER,UNITSMASS
      COMMON/COMP1/CN1(10),PR1(10,5 ),WAC1(10,5 ),ETA1(10,5)
      COMMON/COMP2/CN2(10),PR2(10,5 ),WAC2(10,5 ),ETA2(10,5)
      COMMON/COMP3/CN3(10),PR3(10,5 ),WAC3(10,5 ),ETA3(10,5)
      DATA WLH/6H (LO) ,6H (HI) /
      WB=WA
      IF(WA.LT..1)RETURN
      WA1=WA
      ER1=0.
      TRAT=SQRT(TA/288.149)
      IF(IDES.LT.0)GOTO1
       TRATD(NCOMP)=TRAT
      PRD=PR
      WAD=WA
      ETAD=ETA
1     CNC=PCN*TRATD(NCOMP)/(100.*TRAT)
      IF(ZC.LT.0.)ZC=0.
      IF(ZC.GT.1.)ZC=1.
```

```
      CNS=CNC
      GOTO(3,4,5)  NCOMP
3     CALL SEARCH(ZC ,CNC,PR ,WAC ,ETA ,CN1(1),10 ,PR1(1,1),
     1WAC1(1,1),ETA1(1,1),5,10,5,NCODE)
      GOTO6
4     CALL SEARCH(ZC ,CNC,PR ,WAC ,ETA ,CN2(1),10 ,PR2(1,1),
     2WAC2(1,1),ETA2(1,1),5,10,5,NCODE)
      GOTO6
5     CALL SEARCH(ZC ,CNC,PR ,WAC ,ETA ,CN3(1),10 ,PR3(1,1),
     2WAC3(1,1),ETA3(1,1),5,10,5,NCODE)
6     WA=WAC*PA/TRAT/UNITSMASS
      IF(NCODE.EQ.1.OR.NCODE.EQ.2)WRITE(2,101)NCOMP,CNS,WLH(NCODE)
      IF(IDES.LT.0)GOTO2
      PRCF(NCOMP)=(PRD-1.)/(PR-1.)
      ETACF(NCOMP)=ETAD/ETA
      WACF(NCOMP)=WAD/WA
2     PR=PRCF(NCOMP)*(PR-1.)+1.
      ETA=ETACF(NCOMP)*ETA
      WA=WACF(NCOMP)*WA
      FB=FA
       PHIBS=TRM(FA,TA,3)+ALOG(PR)
      TBS=TA+100.
      CALL AIRTAB(2,FB,TBS,PHIBS)
      ENTHA=TRM(FA,TA,2)
      ENTHB=ENTHA+(TRM(FB,TBS,2)-ENTHA)/ETA
      TB=TBS+30.
      CALL AIRTAB(1,FB,TB,ENTHB)
      PB=PA*PR
      COMWK=(ENTHB-ENTHA)*WA
      WB=WA
      IF(EROR.LE.0.)GOTO7
      NER=1
      ER1=(WA1-WA)/WA
7     IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
      WRITE(2,12)NCOMP
      COMWKUNITS=COMWK*UNITSPOWER
      IF(IDES.GT.0)WRITE(2,10)PRCF(NCOMP),ETACF(NCOMP),WACF(NCOMP)
      WRITE(2,11)ZC,PR,ETA,PCN,CNC,COMWKUNITS
      RETURN
10    FORMAT(8H PRSF = ,E12.5,5X,8HETASF = ,E12.5,5X,7HWASF = ,E12.5)
11    FORMAT(5H Z = ,F7.5,13X,5HPR = ,F7.3,13X,6HETA = ,F7.5,/
     17H PCN = ,F8.4,10X,5HCN = ,F7.5, 13X,8HCOMWK = ,E12.5/)
12    FORMAT(18H ***** COMPRESSOR ,I2,17H PARAMETERS ***** )
101   FORMAT(12H COMPRESSOR ,I2/,17H * * * CN OFF MAP,F10.4,2X,A6
     1,6H * * *)
      END

END OF SEGMENT, LENGTH   570, NAME  B2

      SUBROUTINE B3(IDES,TA,PA,DELP,WA,FA,ETA,TB,PB,WB,WFB,FB,
     1NCODE,LASTONE,WFB1)
      COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
     1UNITSPOWER,UNITSMASS
      COMMON/CUMB/PSI(10),DELT(10,8),ETAX(10,8)
      COMMON/FUEL/CV,FS,WTMOL
      DIMENSION Q(9)
      IF(WA.GT..1)GOTO8
      WB=WA
      WFB=0.
```

```
8      Q(2)=0.
       Q(3)=0.
       MODE=0
       WFB=WFB1/UNITSMASS
       IF(WFB.GT.0.)MODE=1
       IF(IDES.LT.0)GOTO1
       ETAD=ETA
       DLP=DELP*PA
       CA= DLP*PA/(WA*WA*TA)
1       DLP=CA*WA*WA*TA/PA
       IF(TB.LE.0.)TB=1000.
3      DT=TB-TA
       PB=PA-DLP
       CALL SEARCH(-1.,PA,DT,ETA,DUMMY,PSI(1),10 ,DELT(1,1),
      1ETAX(1,1),DELT(1,1),8,10,8,NCODE)
       IF(NCODE.NE.7)GOTO4
       WRITE(2,101)
       RETURN
4      IF(IDES.LT.0)GOTO2
       ETACF=ETAD/ETA
2      ETA=ETACF*ETA
       DELH=TRM(FA,TB,2)-TRM(FA,TA,2)
       ECV=CV+TRM(0.,TB,2)/FS-TRM(FS,TB,2)*(1.+1./FS)
       FB=DELH/ECV/ETA
       WAA=(1.-FA)*WA
       WFBX=FB*WAA
       IF(MODE.NE.1)GOTO7
       ERRW=(WFB-WFBX)/WFB
       DIR=SQRT(WFB/WFBX)
       CALL AFQUIR(Q(1),TB,ERRW,0.,20.,0.0001,DIR,TBT,IGO)
       GOTO(5,7,6)IGO
5      TB=TBT
       GOTO3
6      NCODE=7
       WRITE(2,102)
       RETURN
7      WFB=WFBX
       WB=WA+WFB
       IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
       WFBUNITS=WFB*UNITSMASS
       WRITE(2,10)
       IF(IDES.GT.0)WRITE(2,11)ETACF
       WRITE(2,12)ETA,DLP,WFBUNITS
10     FORMAT(42H ***** COMBUSTION CHAMBER PARAMETERS *****)
11     FORMAT(9H ETASF = ,E12.5)
12     FORMAT(7H ETA = ,F7.5,5X,6HDLP = ,F6.4,5X,6HWFB = ,F8.4/)
101    FORMAT(44H *** ERROR ***POINT NOT FOUND ON COMBUSTION  ,
      111HCHAMBER MAP)
102    FORMAT(47H *** ERROR *** UNABLE TO CONVERGE ON COMBUSTION ,
      124HCHAMBER EXIT TEMPERATURE)
       RETURN
       END


END OF SEGMENT, LENGTH   451, NAME  B3


       SUBROUTINE B4(TA,PA,FA,WA,COMWK,AUXWK,ETA,TFU,CN,TB,PB,PCNC,
      1IDES,IGO,FB,WB,NER,ER1,ER2,NTURB,MAPGO,NUMAP,LASTONE)
       DIMENSION CNCF(3),TFCF(3),ETACF(3),DHCF(3)
```

```
      COMMON/TURB1/TFF1(10),CN1(10,6),DH1(10,6),ETA1(10,6)
      COMMON/TURB2/TFF2(10),CN2(10,6),DH2(10,6),ETA2(10,6)
      COMMON/TURB3/TFF3(10),CN3(10,6),DH3(10,6),ETA3(10,6)
      COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
     1UNITSPOWER,UNITSMASS
      WB=WA
      IF(WA.LT..1)RETURN
      NER=2
      SQRTA=SQRT(TA)
      TF=WA*SQRTA/PA
      TFF=TFU/UNITSMASS
      IF(IDES.LT.0)GOTO2
      CND=CN
      ETAD=ETA
      CNCF(NTURB)=CND*SQRTA/PCNC
      TFFD=TFF
      TFCF(NTURB)=TFFD/TF
2     CN=CNCF(NTURB)*PCNC/SQRTA
      TFSU=TFF*UNITSMASS
      CNS=CN
      FB=FA
      ENTHA=TRM(FA,TA,2)
      ENTHB=ENTHA-(COMWK+AUXWK)/WA
      DELHAB=(ENTHA-ENTHB)/TA
      GOTO(11,12,13)NTURB
11    CALL SEARCH(-1.,TFU,CN,DELHT,ETA,TFF1(1),10,CN1(1,1),DH1(1,1),
     1ETA1(1,1),6,10,6,IGO)
      GOTO10
12     CALL SEARCH(-1.,TFU,CN,DELHT,ETA,TFF2(1),10,CN2(1,1),DH2(1,1),
     1ETA2(1,1),6,10,6,IGO)
      GOTO10
13     CALL SEARCH(-1.,TFU,CN,DELHT,ETA,TFF3(1),10,CN3(1,1),DH3(1,1),
     1ETA3(1,1),6,10,6,IGO)
10    IF(IGO.GT.0)WRITE(2,24)NTURB
24    FORMAT(9H TURBINE ,I2,1H;)
      IF(IGO.EQ.1.OR.IGO.EQ.11.OR.IGO.EQ.21)WRITE(2,20)TFSU
      IF(IGO.EQ.2.OR.IGO.EQ.12.OR.IGO.EQ.22)WRITE(2,21)TFSU
      IF(IGO.EQ.10.OR.IGO.EQ.11.OR.IGO.EQ.12)WRITE(2,22)CNS
      IF(IGO.EQ.20.OR.IGO.EQ.21.OR.IGO.EQ.22)WRITE(2,23)CNS
      IF(IGO.NE.7)GOTO3
      WRITE(2,101)NTURB
      RETURN
3     DELHT=DELHT/UNITSPOWER*UNITSMASS
      TFF=TFU/UNITSMASS
      MAPGO=0
      IF(ABS(TFSU-TFU).LE.0.001*TFSU)GOTO4
      MAPGO=1
      IF(ABS(CNS-CN).GT.0.001*CNS)MAPGO=3
      GOTO5
4     IF(ABS(CNS-CN).GT.0.001*CNS)MAPGO=2
5     IF(MAPGO.EQ.0)GOTO6
      CALL MAPBAC(MAPGO,TFSU,TFU,CNS,CN,PCNC,TA,NUMAP)
      RETURN
6     DELH=DELHAB*TA
      IF(IDES.LT.0)GOTO1
      ETACF(NTURB)=ETAD/ETA
      DHCF(NTURB)=DELHAB/DELHT
1     ETA=ETACF(NTURB)*ETA
      TF=TFCF(NTURB)*TF
      DELHT=DHCF(NTURB)*DELHT
```

```
      ER1=(TF-TFF)/TF
      ER2=(DELHAB-DELHT)/DELHAB
      ENTHBS=ENTHA-DELH/ETA
      TBS=TA-400.
      CALL AIRTAB(1,FB,TBS,ENTHBS)
      PB=PA*EXP(TRM(FB,TBS,3)-TRM(FA,TA,3))
      TB=TBS-20.
      CALL AIRTAB(1,FB,TB,ENTHB)
      IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
      AUXWKUN=AUXWK*UNITSPOWER
      WRITE(2,110)NTURB
      IF(IDES.GT.0)WRITE(2,111)CNCF(NTURB),ETACF(NTURB),TFCF(NTURB)
     1,DHCF(NTURB)
      WRITE(2,112)TFU,ETA,CN,AUXWKUN
20    FORMAT(7H TFF LO,F17.5)
21    FORMAT(7H TFF HI,E17.5)
22    FORMAT(7H CN  LO,E17.5)
23    FORMAT(7H CN  HI,E17.5)
101   FORMAT(48H *** ERROR *** POINT NOT FOUND ON MAP OF TURBINE,
     1I3)
110   FORMAT(15H ***** TURBINE ,I2,17H PARAMETERS *****)
111   FORMAT(8H CNSF = ,E12.5,5X,8HETASF = ,E12.5,5X,7HTFSF = ,
     1E12.5,5X,7HDHSF = ,E12.5)
112   FORMAT(6H TF = ,F7.3,12X,6HETA = ,F7.5,12X,5HCN = ,F6.3,/
     19H AUXWK = ,E12.5/)
      RETURN
      END


END OF SEGMENT, LENGTH   794, NAME  B4

      SUBROUTINE B5(FA,TA,PA,PATM,IDES,WA,VB,PSB,A,ICON,XG,NER,
     1ER1,PB,TB,TSB,FB,WB,NNOZ,NCODE,LASTONE,FLOAT)
      COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
     1UNITSPOWER,UNITSMASS
      WB=WA
      FB=FA
      XG=0.00001
      IF(WA.LT..1)RETURN
      CJ=1400.868
      PPSF=2116.217
      NER=1
      RA=GASCONST(FA)
      ENTHA=TRM(FA,TA,2)
      J=0
      TSS=0.833*TA
1     J=J+1
      SPHTSS=TRM(FA,TSS,1)
      GAM=SPHTSS/(SPHTSS-RA)
      CSS=SQRT(GAM*RA*TSS*45071.595)
      HSCAL=ENTHA-CSS*CSS/(2.*45071.595)
      DELH=HSCAL-TRM(FA,TSS,2)
      IF(ABS(DELH).LE.0.00025*HSCAL)GOTO2
      TSS=TSS+DELH/SPHTSS
      IF(J-15)1,1,3
3     NCODE=7
      WRITE(2,100)NNOZ
      RETURN
2     IF(IDES.LE.0.AND.FLOAT.LE.0.)GOTO4
      NER=0
```

```
C *** ISENTROPIC EXPANSION (DESIGN)
5     J=0
      TSA=TA*(PATM/PA)**0.286
      SA=RA*(TRM(FA,TA,3)-ALOG(PA))
6     J=J+1
      SSA=RA*(TRM(FA,TSA,3)-ALOG(PATM))
      SAA=SA-SSA
      IF(ABS(SAA).LE.0.0001*SA)GOTO7
      TSA=TSA*EXP(4.*SAA)
22    IF(J-30)6,6,3
7     DH=ENTHA-TRM(FA,TSA,2)
      IF(DH.GE.0.)GOTO71
      NCODE=5
      WRITE(2,102)NNOZ
      RETURN
102   FORMAT(53H *** ERROR *** IMAGINARY EXIT VELOCITY CONVERG NOZZLE
     1,I3)
71    VAS=SQRT(2.*45071.595*DH)
      IF(VAS.GE.CSS)GOTO8
C *** SUBSONIC DESIGN, CALCULATE AB
      VB=VAS
      TSB=TSA
      PSB=PATM
      RHO=PPSF*PSB/(RA*TSB*CJ)
      A=WA/(RHO*VB)
      SPHTSB=TRM(FA,TSB,1)
      AMB=VB/SQRT(SPHTSB/(SPHTSB-RA)*RA*TSB*45071.595)
9     TB=TA
      PB=PA
      XG=1.
      IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
      AUNITS=A*UNITSAREA
      VBUNITS=VB*UNITSLENGTH
      CALL NOZCO(PB/PATM,1.,COEF)
      XG=WA*VB*COEF/32.174049+(PSB-PATM)*PPSF*A
      XGUNITS=XG*UNITSFORCE
      WRITE(2,101)NNOZ,AUNITS,VBUNITS,XGUNITS,COEF
      RETURN
C *** SONIC DESIGN,  CALCULATE AB
8     VB=CSS
      TSB=TSS
      PSB=PA*(TSB/TA)**(GAM/(GAM-1.))
      RHO=PPSF*PSB/(RA*TSB*CJ)
      A=WA/(RHO*VB)
      AMB=1.0
      GOTO9
C *** OFF DESIGN, CALCULATE CRITICAL CONDITIONS
4     VB=CSS
      TSB=TSS
       PSB=PATM
      RHO=PPSF*PSB/(RA*TSB*CJ)
      ABCRIT=WA/(RHO*VB)
      AMB=1.0
      PREQ=PSB*(TA/TSB)**(GAM/(GAM-1.))
      IF(A.GT.ABCRIT)GOTO11
C *** OFF DESIGN, CRITICAL AND SUPERCRITICAL CONDITIONS (SONIC)
10    PSB=PSB*ABCRIT/A
      PREQ=PREQ*ABCRIT/A
      ER1=(PREQ-PA)/PREQ
      GOTO9
```

```
C *** OFF DESIGN  SUBSONIC
11    PSB=PATM
      J=0
      TSB=0.833*TSB
12    J=J+1
      RHO=PPSF*PSB/(RA*TSB*CJ)
      VB=WA/(RHO*A)
      HSCAL=ENTHA-VB*VB/(2.*45071.595)
      DELH=HSCAL-TRM(FA,TSB,2)
      SPHTSB=TRM(FA,TSB,1)
      IF(ABS(DELH).LE.0.0005*HSCAL)GOTO13
      TSB=TSB+DELH/SPHTSB
      IF(J-15)12,12,3
13    GAM=SPHTSB/(SPHTSB-RA)
      AMB=VB/SQRT(GAM*RA*TSB*45071.595)
      PREQ=PSB*(TA/TSB)**(GAM/(GAM-1.))
      ER1=(PREQ-PA)/PREQ
      GOTO9
100   FORMAT(48H *** ERROR *** UNABLE TO CONVERGE ON TEMPERATURE
     1,19H IN CONVRG NOZZLE ,I2)
101   FORMAT(25H ***** CONVERGENT NOZZLE ,I2,17H PARAMETERS *****,/
     118H AREA = ,F7.4,19H   EXIT VELOCITY = ,F7.2,
     221H   GROSS THRUST = ,F9.2,/15H NOZZLE COEF = ,E12.5/)
      END

END OF SEGMENT, LENGTH   801, NAME  B5

      SUBROUTINE B8(F1,F2,F3,W1,W2,W3,P1,P3,T1,T2,T3,P2)
C     1 = MAINSTREAM, 2 = BLEED STREAM, 3 = MIXED STREAM
      W3=W1+W2
      IF(W1.GE.W2)P3=P1
      IF(W2.GT.W1)P3=P2
      C=W1/(1.+F1)+W2/(1.+F2)
      F3=(W3-C)/C
      H3=(TRM(F1,T1,2)*W1+TRM(F2,T2,2)*W2)/W3
      T3=(T1*W1+T2*W2)/W3
      CALL AIRTAB(1,F3,T3,H3)
      RETURN
      END

END OF SEGMENT, LENGTH   125, NAME  B8

      SUBROUTINE B9(SA,OP1,OP2,OP3,R1)
      INTEGER SA
      R1=-1.
      GOTO(1,2,3,4,5,6,7,8)SA
1     R1=OP1+OP2
      RETURN
2     R1=OP1-OP2
      RETURN
3     R1=OP1*OP2
      RETURN
4     R1=OP1/OP2
      RETURN
5     OP1=OP2+OP3
      RETURN
```

```
6       OP1=OP2-OP3
        RETURN
7       OP1=OP2*OP3
        RETURN
8       OP1=OP2/OP3
        RETURN
        END

END OF SEGMENT, LENGTH   116, NAME  89

        SUBROUTINE B10 (F1,F2,F3,W1,W2,W3,PS1,PS2,PS3,P1,P2,P3,TS1,
       1TS2,TS3,T1,T2,T3,V1,V2,V3,A1,A2,A3,IDES,NCODE,LASTONE,PCN,Z,
       2SWITCH,VALUE,NOMAP,NER,ER1,AM1,AM2,AM3)
        DIMENSION QQ(9)
        IF(SWITCH.LE.0.)PS1=VALUE
        IF(SWITCH.GT.0.)AM1=VALUE
35      CJ=1400.868
        G=32.174049
        GCJ=45071.595
        NOMAP=0
        PPSF=2116.217
        R1=GASCONST(F1)
        R2=GASCONST(F2)
        H1=TRM(F1,T1,2)
        H2=TRM(F2,T2,2)
        PHI1=TRM(F1,T1,3)
        PHI2=TRM(F2,T2,3)
        IF(IDES.LE.0)GOTO1
C       CALCULATE A1 AND A2 WITH PS1=PS2
        IF(SWITCH.GT.0.)GOTO3
C       PS1 IS GIVEN
        TS1=T1*(PS1/P1)**0.286
        DO2I=1,15
        PHIS=PHI1-ALOG(P1/PS1)
        DELPHI=PHIS-TRM(F1,TS1,3)
        IF(ABS(DELPHI).LE.0.0001*PHIS)GOTO5
2       TS1=TS1*EXP(4.*DELPHI/R1)
12      NCODE=7
        WRITE(2,101)
        RETURN
C       PS1 IS NOT GIVEN BUT M1 (AM1) IS
3       TS1=0.875*T1
        DO4I=1,15
        SP1=TRM(F1,TS1,1)
        GAM1=SP1/(SP1-R1)
        CS1=SQRT(GAM1*R1*TS1*GCJ)
        V1=AM1*CS1
        HSCAL=H1-V1*V1/(2.*GCJ)
        HS1=TRM(F1,TS1,2)
        DELHS=HSCAL-HS1
        IF(ABS(DELHS).LE.0.0005*HSCAL)GOTO6
4       TS1=TS1+DELHS/SP1
        GOTO12
6       PS1=P1/EXP(PHI1-TRM(F1,TS1,3))
        IF(PS1.LE.P2)GOTO5
        PFNEW=PS1/P2*1.02
        PCNEW=1./PFNEW
        WRITE(2,102)PFNEW,PCNEW
        NCODE=7
```

```
      RETURN
102   FORMAT(51H MIXING IMPOSSIBLE AS PSTATIC (MAIN STREAM) GREATER
     144H THAN PTOTAL (BYPASS STREAM)./,18H TRY NEW FAN PR = ,
     213HOLD FAN PR X ,F6.3,/19H AND NEW COMP PR = ,
     314HOLD COMP PR X ,F6.3)
C     CALCULATE A1,A2
5     HS1=TRM(F1,TS1,2)
      IF(H1.GT.HS1)GOTO7
      GOTO12
7     V1=SQRT(2.*GCJ*(H1-HS1))
      RHO=PPSF*PS1/(CJ*R1*TS1)
      A1=W1/(RHO*V1)
      SP1=TRM(F1,TS1,1)
      GAM1=SP1/(SP1-R1)
      CS1=SQRT(GAM1*R1*TS1*GCJ)
      AM1=V1/CS1
8     PS2=PS1
      TS2=T2*(PS2/P2)**0.286
      S2=R2*(PHI2-ALOG(P2))
      DO9I=1,15
      SS2=R2*(TRH(F2,TS2,3)-ALOG(PS2))
      DELS=S2-SS2
      IF(ABS(DELS).LE.0.0001*S2)GOTO10
9     TS2=TS2*EXP(4.*DELS)
      GOTO12
10    HS2=TRM(F2,TS2,2)
      IF(H2.GT.HS2)GOTO11
      GOTO12
11     V2=SQRT(2.*GCJ*(H2-HS2))
      RHO=PPSF*PS2/(CJ*R2*TS2)
       A2=W2/(RHO*V2)
      SP2=TRM(F2,TS2,1)
      GAM2=SP2/(SP2-R2)
      CS2=SQRT(GAM2*R2*TS2*GCJ)
      AM2=V2/CS2
      GOTO27
1     WQA=W1/A1
C     OFF DESIGN, CALCULATE PS1 AND PS2 USING DESIGN A1 AND A2
      C1=P1*SQRT(G/(T1*CJ))*PPSF
      MCON=0
      QQ(2)=0.
      QQ(3)=0.
      AM1=.5
      TS1=.875*T1
13    DO14I=1,15
      SP1=TRM(F1,TS1,1)
      GAM1=SP1/(SP1-R1)
      CS1=SQRT(GAM1*R1*TS1*GCJ)
      V1=AM1*CS1
      HSCAL=H1-V1*V1/(2.*GCJ)
      DELHS=HSCAL-TRM(F1,TS1,2)
      IF(ABS(DELHS).LE.0.0005*HSCAL)GOTO15
14    TS1=TS1+DELHS/SP1
      GOTO12
15    WQAT=C1*SQRT(GAM1/R1)*AM1/(1.+(GAM1-1.)*AM1*AM1/2.)**((GAM1
     1+1.)/(2.*(GAM1-1.)))
      AMX=AM1
      IGOGO=0
16    DIR=WQA/WQAT
      FW=(WQA-WQAT)/WQA
```

```
      CALL AEQUIR(QQ(1),AMX,EW,0.,30.,0.0005,DIR,AMXT,ICON)
      GOTO(17,22,12)ICON
17    IF(AMXT.LE.1.)GOTO20
      AMXT=.7
      MCON=MCON+1
      IF(MCON.LE.1)GOTO20
      PCN=PCN*1.01
      Z=.99*Z
18    NOMAP=1
      RETURN
20    IF(IGOGO.EQ.1)GOTO21
      AM1=AMXT
      GOTO13
21    AM2=AMXT
      GOTO23
22    IF(IGOGO.EQ.1)GOTO26
      PS1=P1/EXP(PHI1-TRM(F1,TS1,3))
      WQA=W2/A2
      C1=P2*SQRT(G/(T2*CJ))*PPSF
      MCON=0
      QQ(2)=0.
      QQ(3)=0.
      AM2=.25
      TS2=.875*T2
23    DO24I=1,15
      SP2=TPM(F2,TS2,1)
      GAM2=SP2/(SP2-R2)
      CS2=SQRT(GAM2*R2*TS2*GCJ)
      V2=AM2*CS2
      HSCAL=H2-V2*V2/(2.*GCJ)
      DELHS=HSCAL-TRM(F2,TS2,2)
      IF(ABS(DELHS).LE.0.0005*HSCAL)GOTO25
24    TS2=TS2+DELHS/SP2
      GOTO12
25    WQAT=C1*SQRT(GAM2/R2)*AM2/(1.+(GAM2-1.)*AM2*AM2/2.)**((GAM2
     1+1.)/(2.*(GAM2-1.)))
      AMX=AM2
      IGOGO=1
      GOTO16
26    PS2=P2/EXP(PHI2-TRM(F2,TS2,3))
27    W3=W1+W2
      IF(IDES.LE.0)NER=1
      EP1=(PS2-PS1)/PS2
      C1=W1/(1.+F1)+W2/(1.+F2)
      F3=(W3-C1)/C1
      H3=(W1*H1+W2*H2)/W3
      T3=(W1*T1+W2*T2)/W3
      CALL AIRTAB(1,F3,T3,H3)
      C1=PS1*A1*(1.+GAM1*AM1*AM1)+PS2*A2*(1.+GAM2*AM2*AM2)
      TS3=.833*T3
      R3=GASCONST(F3)
      DO32I=1,15
      SP3=TRM(F3,TS3,1)
      GAM3=SP3/(SP3-R3)
      CS3=SQRT(GAM3*R3*TS3*GCJ)
      C2=W3*SQRT(CJ*R3*T3/(GAM3*G))
      C3=C2/PPSF/C1
      C4=(GAM3-1.)/2.- (C3*GAM3)**2
      C5=1.-2.*GAM3*C3*C3
      C6=C5*C5+4.*C4*C3*C3
```

```
      IF(C6)12,29,30
29    AM3G=-C5/(2.*C4)
      GOTO31
30    AM3G=(SQRT(C6)-C5)/(2.*C4)
31    IF(AM3G.LE.0.)GOTO12
      AM3=SQRT(AM3G)
      V3=AM3*CS3
      HSCAL=H3-V3*V3/(2.*GCJ)
      DELHS=HSCAL-TRM(F3,TS3,2)
      IF(ABS(DELHS).LE.0.0005*HSCAL)GOTO34
32    TS3=TS3+DELHS/SP3
      GOTO12
34    A3=A1+A2
      C7=SQRT(1.+(GAM3-1.)*AM3G/2.)
      PS3=C2/(PPSF*A3*AM3*C7)
      P3=PS3*EXP(TRM(F3,T3,3)-TRM(F3,TS3,3))
101   FORMAT(44H ***** ERROR ***** ERROR IN MIXING UNABLE TO,
     123H CONVERGE WITHIN LIMITS)
      S1=R1*(TRM(F1,T1,3)-ALOG(P1))
      S2=R2*(TRM(F2,T2,3)-ALOG(P2))
      S3=R3*(TRM(F3,T3,3)-ALOG(P3))
      S3AVE=(W1*S1+W2*S2)/W3
      IF(S3.GE.S3AVE)GOTO36
      S3=S3AVE
      P3=EXP(S3/R3-TRM(F3,T3,3))
36    CONTINUE
      RETURN
      END


END OF SEGMENT, LENGTH  1600, NAME  B10


      SUBROUTINE B11(FA,FB,WA,WB,PA,PB,TA,TB,IDES,LASTONE,NDUCT,
     1NBURN,BURN,DELP,ETA,WFB)
      COMMOM/FUEL/CV,FS,WTMOL
      COMMON/UNITSFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
     1UNITSPOWER,UNITSMASS
      DIMENSION CA(3)
      IBURN=IFIX(BURN)
      IF(WA.GT.01)GOTO20
      WB=WA
      IF(IBURN.GE.1)WFB=0.00001
      RETURN
20    IF(IDES.LE.0)GOTO1
      DLP=DELP*PA
      CA(NDUCT)=DLP*PA/(WA*WA*TA)
1     DLP=CA(NDUCT)*WA*WA*TA/PA
      PB=PA-DLP
      IF(IBURN.GE.1)GOTO2
      TB=TA
      FB=FA
      WB=WA
      RETURN
2     IF(IBURN.EQ.2)GOTO3
      TB=TA
      FB=FA
      WB=WA
      WFB=0.00001
      GOTO4
```

```
3       DELH=TRM(FA,TB,2)-TRM(FA,TA,2)
        ECV=CV+TRM(0.,TB,2)/FS-TRM(FS,TB,2)*(1.+1./FS)
        FB=DELH/ECV/ETA
        WAA=(1.-FA)*WA
        WFB=FB*WA
        FB=(WFB+WA-WAA)/WAA
        WB=WA+WFB
4       IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
        WFBUNITS=WFB*UNITSMASS
        WRITE(2,100)NBURN,ETA,DLP,WFBUNITS
100     FORMAT(25H ***** DUCT/AFTER BURNING,I2,17H PARAMETERS *****/,
       17H ETA = ,F7.4,11X,6HDLP = ,F6.4,13X,6HWFB = ,F8.4/)
        RETURN
        END


END OF SEGMENT, LENGTH   311, NAME  B11

        SUBROUTINE B12(FA,FB,WA,WB,PSA,PSB,PA,PB,TSA,TSB,TA,TB,VB,AB
       1,IDES,NCODE,LASTONE,NER,ERR,PATM,XG,FLOAT,NOZ,ATHROAT)
        COMMON/UNITFACTORS/UNITSAREA,UNITSFORCE,UNITSLENGTH,
       1UNITSPOWER,UNITSMASS
        DIMENSION Q(9),AT(3),WORDS(4)
        DATA WORDS/5H OVER,5HUNDER,3H IN,3HOUT/
        Q(2)=0.
        Q(3)=0.
        L1=0
        L2=0
        FB=FA
        WB=WA
        XG=0.00001
        IF(WA.LE.1.)RETURN
        IF(ATHROAT.GT.0.)AT(NOZ)=ATHROAT/UNITSAREA
2       PHIA=TRM(FA,TA,3)
        CJ=1400.868
        G=32.174049
        GCJ=45071.595
        PPSF=2116.217
        R=GASCONST(FA)
        HA=TRM(FA,TA,2)
        TSS=0.833*TA
        J=0
1       J=J+1
        SPHTSS=TRM(FA,TSS,1)
        GAM=SPHTSS/(SPHTSS-R)
        CSS=SQRT(GAM*R*TSS*GCJ)
        HSCAL=HA-CSS*CSS/(2.*GCJ)
        DELH=HSCAL-TRM(FA,TSS,2)
        IF(ABS(DELH).LE.0.0005*HSCAL)GOTO4
        TSS=TSS+DELH/SPHTSS
        IF(J-15)1,1,3
3       NCODE=7
        WRITE(2,102)NOZ
102     FORMAT(24H ERROR IN CON/DI NOZZLE ,I2)
        RETURN
4       IF(IDES.LE.0.AND.FLOAT.LE.0.)GOTO11
        NER=0
C       SONIC DESIGN, CALCULATE AT
5       VT=CSS
        TST=TSS
```

```
      PST=PA*(TST/TA)**(GAM/(GAM-1.))
      RHO=PPSF*PST/(CJ*R*TST)
      AT(NOZ)=WA/(RHO*VT)
      AMT=1.0
C     IDEAL EXPANSION, CALCULATE AB
      PSB=PATM
      TSB=TA*(PSB/PA)**.286
      SA=R*(TRM(FA,TA,3)-ALOG(PA))
      DO7I=1,15
      SSB=R*(TRM(FA,TSB,3)-ALOG(PSB))
      DELS=SA-SSB
      IF(ABS(DELS).LE.0.0001*SA)GOTO8
7     TSB=TSB*EXP(4.*DELS)
      GOTO3
8     VB=SQRT(2.*GCJ*(HA-TRM(FA,TSB,2)))
      SPHTSB=TRM(FA,TSB,1)
      GAMSB=SPHTSB/(SPHTSB-R)
      CSB=SQRT(GAMSB*R*TSB*GCJ)
      AMB=VB/CSB
      AB=(AT(NOZ)/AMB)*(2.*(1.+(GAMSB-1.)*AMB*AMB/2.)/(GAMSB+1.))
     1**((GAMSB+1.)/(2.*(GAMSB-1.)))
      PAR=PA
9     TB=TA
      PB=PA
10    TT=TA
      PT=PA
      XG=1.
      IF(IDES.GT.0.OR.FLOAT.GT.0.)GOTO43
      NER=1
      ERR=(PAR-PA)/PAR
43    IF(IDES.LE.0.AND.LASTONE.LE.0)RETURN
      CALL NOZCO(PB/PATM,AB/AT(NOZ),COEF)
      XG=WA*VB*COEF/G+(PSB-PATM)*PPSF*AB
      XGU=XG*UNITSFORCE
      VTU=VT*UNITSLENGTH
      VBU=VB*UNITSLENGTH
      ABU=AB*UNITSAREA
      ATU=AT(NOZ)*UNITSAREA
      WRITE(2,100)NOZ,ATU,VTU,AMT,ABU,VBU,AMB,COEF,XGU
      IF(LASTONE.GT.0.AND.L1.GT.0)WRITE(2,101)WORDS(L1),WORDS(L2)
      RETURN
C     ASSUME SONIC THROAT AND ISENTROPIC EXPANSION TO AB
11    VT=CSS
      AMT=1.0
      TST=TSS
      RHO=WA/(AT(NOZ)*VT)
      PST=RHO*CJ*R*TST/PPSF
      PAR=PST*(TA/TST)**(GAM/(GAM-1.))
      IF(PST-PATM)12,27,27
12    TSB=0.95*TA
      NAM=0
13    SPHTSB=TRM(FA,TSB,1)
      GAMSB=SPHTSB/(SPHTSB-R)
      CSB=SQRT(GAMSB*R*TSB*GCJ)
      AMB=SQRT(2.*((TA/TSB)-1.)/(GAMSB-1.))
      ABCAL=(AT(NOZ)/AMB)*(2.*(1.+(GAMSB-1.)*AMB*AMB/2.)/(GAMSB+1.
     2))**((GAMSB+1.)/(2.*(GAMSB-1.)))
      EA=(AB-ABCAL)/AB
      DIR=SQRT(AB/ABCAL)
      CALL AEQUIR(Q(1),TSB,EA,0.,100.,0.0001,DIR,TSBT,JCON)
```

```
      GOTO(14,18,3)JCON
14    TSB=TSBT
      IF(TSB-TA)15,13,16
15    TSC=2.*TA/(GAMSB+1.)
      IF(TSB.GT.TSC)GOTO17
16    TSB=0.98*TA
      GOTO13
17    IF(Q(2).LT.30..OR.AMB.LT.0.95.OR.MAM.EQ.1)GOTO13
      TSB=2.*TA/(2.+0.98*(GAMSB-1.))

      MAM=1
      GOTO13
18    PSB=PAR*(TSB/TA)**(GAMSB/(GAMSB-1.))
      IF(PSB-PATM)20,19,27
C     CRITICAL FLOW, ISENTROPIC EXPANSION TO PATM
19    VB=AMB*CSB
      ICON=1
      GOTO9
C     SUBSONIC FLOW
20    PSB=PATM
      Q(2)=0.
      Q(3)=0.
      J=0
      TSB=0.833*TA
21    J=J+1
      SPHTSB=TRM(FA,TSB,1)
      GAMSB=SPHTSB/(SPHTSB-R)
      RHO=PPSF*PSB/(CJ*R*TSB)
      VB=WA/(RHO*AB)
      HSCAL=HA-VB*VB/(2.*GCJ)
      DELHS=HSCAL-TRM(FA,TSB,2)
      IF(ABS(DELHS)-0.0005*HSCAL)23,23,22
22    TSB=TSB+DELHS/SPHTSB
      IF(J-15)21,21,3
23    AMB=VB/SQRT(GAMSB*R*TSB*GCJ)
      PAR=PSB*(TA/TSB)**(GAMSB/(GAMSB-1.))
      TST=TSB
24    SPHTST=TRM(FA,TST,1)
      GAMST=SPHTSS/(SPHTSS-R)
      CST=SQRT(GAMST*R*TST*GCJ)
      PST=PAR*(TST/TA)**(GAMST/(GAMST-1.))
      RHO=PST*PPSF/(CJ*R*TST)
      VT=WA/(RHO*AT(NOZ))
      HSCAL=HA-VT*VT/(2.*GCJ)
      HST=TRM(FA,TST,2)
      EH=(HSCAL-HST)/HSCAL
      DIR=1.+(HSCAL-HST)/(SPHTSS*TST)
      CALL AEQUIR(Q(1),TST,EH,0.,20.,0.0005,DIR,TSTT,JCON)
      GOTO(25,26,3)JCON
25    TST=TSTT
      GOTO24
26    AMT=VT/CST
      GOTO9
C    SUPERCRITICAL FLOW, ISENTROPIC EXPANSION TO PATM
27    PSB=PATM
      J=0
      TSB=TA*(PSB/PAR)**.286
      SA=R*(TRM(FA,TA,3)-ALOG(PAR))
28    J=J+1
      SSB=R*(TRM(FB,TSB,3)-ALOG(PSB))
```

```
      DELS=SA-SSB
      IF(ABS(DELS).LE.0.0001*SA)GOTO30
      TSB=TSB*EXP(4.*DELS)
      IF(J-15)28,28,3
30    VB=SQRT(2.*GCJ*(HA-TRM(FA,TSB,2)))
      SPHTSB=TRM(FA,TSB,1)
      GAMSB=SPHTSB/(SPHTSB-R)
      CSB=SQRT(GAMSB*R*TSB*GCJ)
      AMB=VB/CSB
      ABAD=(AT(NOZ)/AMB)*(2.*(1.+(GAMSB-1.)*AMB*AMB/2.)/(GAMSB+1.
     1))**((GAMSB+1.)/(2.*(GAMSB-1.)))
      N=0
      IF(AB-ABAD)31,9,32
C     SUPERCRITICAL FLOW, ISENTROPIC EXPANSION TO AB
31    N=1
32    TSB=0.833*TA
      J=0
33    J=J+1
      SPHTSB=TRM(FA,TSB,1)
      GAMSB=SPHTSB/(SPHTSB-R)
      CSB=SQRT(GAMSB*R*TSB*GCJ)
      AMB=SQRT(2.*((TA/TSB)-1.)/(GAMSB-1.))
      ABCAL=(AT(NOZ)/AMB)*(2.*(1.+(GAMSB-1.)*AMB*AMB/2.)/(GAMSB+1.
     1))**((GAMSB+1.)/(2.*(GAMSB-1.)))
      DELA=AB-ABCAL
      IF(ABS(DELA)-0.0001*AB)35,35,34
34    TSB=TSB*SQRT(ABCAL/AB)
      IF(J-50)33,33,3
35    IF(N)37,37,36
C     UNDEREXPANDED, SHOCK OUTSIDE NOZZLE
36    PSB=PAR*(TSB/TA)**(GAMSB/(GAMSB-1.))
      VB=AMB*CSB
      L1=2
      L2=4
      GOTO9
C     OVEREXPANDED, FIND SHOCK POSITION
37    PSX=PAR*(TSB/TA)**(GAMSB/(GAMSB-1.))
      PSY=PSX*(2.*GAMSB*AMB*AMB/(GAMSB+1.)-(GAMSB-1.)/(GAMSB+1.))
      IF(PATM-PSY)38,39,39
C     OVEREXPANDED, SHOCK OUTSIDE NOZZLE
38    PSB=PSX
      VB=AMB*CSB
      L1=2
      L2=4
      GOTO9
C     OVEREXPANDED, SHOCK INSIDE NOZZLE
39    PSB=PATM
      L1=1
      L2=3
      J=0
      TSB=0.833*TA
40    J=J+1
      SPHTSB=TRM(FA,TSB,1)
      GAMSB=SPHTSB/(SPHTSB-R)
      RHO=PPSF*PSB/(CJ*R*TSB)
      VB=WA/(RHO*AB)
      HSCAL=HA-VB*VB/(2.*GCJ)
      DELHS=HSCAL-TRM(FA,TSB,2)
      IF(ABS(DELHS)-0.0005*HSCAL)42,42,41
41    TSB=TSB+DELHS/SPHTSB
```

```
      IF(J-15)40,40,3
42    AMB=VB/SQRT(GAMSB*R*TSB*GCJ)
      TB=TA
      PB=PSB*(TB/TSB)**(GAMSB/(GAMSB-1.))
      GOTO10
100   FORMAT(34H ***** CONVERGENT/DIVERGENT NOZZLE,I2,
     117H PARAMETERS ***** ,/15H THROAT AREA = ,F7.4,5X,
     2 18HTHROAT VELOCITY = ,F7.2,5X,17HTHROAT MACH NO = ,F7.4,/
     315H   EXIT AREA = ,F7.4,5X,18H  EXIT VELOCITY = ,F7.2,5X,
     417H  EXIT MACH NO = ,F7.4,/15H NOZZLE COEF = ,F7.4,
     512X,15HGROSS THRUST = ,F8.2/)
101   FORMAT(11H NOZZLE IS ,A5,19HEXPANDED. SHOCK IS ,A3,
     112HSIDE NOZZLE,//)
      END

END OF SEGMENT, LENGTH  1824, NAME  B12

      SUBROUTINE B13(Z,ZINIT,DELZ,ZFIN,LOOP,I,K)
      IF(LOOP.GT.1)GOTO1
      Z=ZINIT
      K=I
      RETURN
1     Z=Z+DELZ
      IF(Z.GE.ZFIN)K=-1
      IF(Z.LT.ZFIN)K=I
      RETURN
      END

END OF SEGMENT, LENGTH    87, NAME  B13

      FUNCTION GASCONST(F)
      COMMON /FUEL/ CV,FS,WTMOL
      GASCONST=1.98586/28.969*(1.+WTMOL/28.969*F)/(1.+F)
      RETURN
      END

END OF SEGMENT, LENGTH    33, NAME  GASCONST

      FUNCTION TRM(F,T,N)
C     CALCULATES THERMODYNAMIC QUANTITIES GIVEN T AND F.
C     SPLIT INTO TWO RANGES; 200 - 800 DEG K, 800 - 2000 DEG K.
      COMMON/AIRCONST/CL0,CL1,CL2,CL3,CL4,CH0,CH1,CH2,CH3,CH4,TCP0,
     1TCP1,TCP2,TCP3,TCP4,TH0,TH1,TH2,TH3,TH4,TFI0,TFI1,TFI2,
     2TFI3,TFI4,CLIH,CHIH,CLIF,CHIF,B
      FF=F/(1.+F)
      IF(T.GE.800.)GOTO1
      IF(N.GT.1)GOTO3
      TRM=((((CL4+FF*TCP4)*T+CL3+FF*TCP3)*T+CL2+FF*TCP2)*T+
     1CL1+FF*TCP1)*T+CL0+FF*TCP0
      RETURN
3     IF(N.GT.2)GOTO2
      TRM=(((((CL4/5.)*T+CL3/4.+FF*TH4)*T+CL2/3.+FF*TH3)*T+CL1/2.
     1+FF*TH2)*T+CL0+FF*TH1)*T+CLIH+FF*TH0
      RETURN
2     TRM=((((B*CL4/4.+FF*TFI4)*T+B*CL3/3.+FF*TFI3)*T+B*CL2
     1/2.+FF*TFI2)*T+B*CL1+FF*TFI1)*T+B*CL0*ALOG(T)+CLIF+FF*TFI0
      RETURN
1     IF(N.GT.1)GOTO4
```

```
      TRM=((((CH4+FF*TCP4)*T+CH3+FF*TCP3)*T+CH2+FF*TCP2)*T+
     1CH1+FF*TCP1)*T+CH0+FF*TCP0
      RETURN
4     IF(N.GT.2)GOTO5
      TRM=(((((CH4/5.)*T+CH3/4.+FF*TH4)*T+CH2/3.+FF*TH3)*T+CH1/2.
     1+FF*TH2)*T+CH0+FF*TH1)*T+CHIH+FF*TH0
      RETURN
5     TRM=(((B*CH4/4.+FF*TFI4)*T+B*CH3/3.+FF*TFI3)*T+B*CH2
     1/2.+FF*TFI2)*T+B*CH1+FF*TFI1)*T+B*CH0*ALOG(T)+CHIF+FF*TFI0
      RETURN
      END

END OF SEGMENT, LENGTH   458, NAME  TRM

      SUBROUTINE AIRTAB(I,F,T,TAB)
C     INVERSE THERMODYNAMIC RELATIONSHIPS. FINDS T GIVEN F AND
C     THERMO QUANTITY
C     SPLIT INTO TWO RANGES; 200 - 800 DEG K, 800 - 2000 DEG K.
      COMMON/AIRCONST/CL0,CL1,CL2,CL3,CL4,CH0,CH1,CH2,CH3,CH4,TCP0,
     1TCP1,TCP2,TCP3,TCP4,TH0,TH1,TH2,TH3,TH4,TFI0,TFI1,TFI2,
     2TFI3,TFI4,      CLIH,CHIH,CLIF,CHIF,B
      FF=F/(1.+F)
      IF(I.GT.1)GOTO1
C     FIND T GIVEN F AND H.
2     IF(T.GE.800.)GOTO4
      TCAL=(TAB-CLIH-FF*TH0)/(((((CL4/5.)*T+CL3/4.+FF*TH4)*T+
     1CL2/3.+FF*TH3)*T+CL1/2.+FF*TH2)*T+CL0+FF*TH1)
      GOTO5
4     TCAL=(TAB-CHIH-FF*TH0)/(((((CH4/5.)*T+CH3/4.+FF*TH4)*T+
     1CH2/3.+FF*TH3)*T+CH1/2.+FF*TH2)*T+CH0+FF*TH1)
5     IF(ABS(T-TCAL).LE..01)GOTO61
      T=TCAL
      GOTO2
61    T=TCAL
      RETURN
C     FIND T GIVEN F AND PHI.
1     IF(T.GE.800.)GOTO6
      TCAL=EXP((TAB-CLIF-FF*TFI0-((((B*CL4/4.+FF*TFI4)*T+B*
     1CL3/3.+FF*TFI3)*T+B*CL2/2.+FF*TFI2) *T+B*CL1+FF*TFI1)*T)/
     1(B*CL0))
      GOTO7
6     TCAL=EXP((TAB-CHIF-FF*TFI0-(((( B*CH4/4.+FF*TFI4)*T+B*
     1CH3/3.+FF*TFI3)*T+B*CH2/2.+FF*TFI2) *T+B*CH1+FF*TFI1)*T)/
     2(B*CH0))
7     IF(ABS(T-TCAL).LE..01)GOTO3
      T=TCAL
      GOTO1
3     T=TCAL
      RETURN
      END

END OF SEGMENT, LENGTH   408, NAME  AIRTAB

      SUBROUTINE MAPBAC(MAPGO,TFS,TFF,CNS,CN,PCN,T,NUM)
      DATA WT,WS/6H TFF  ,6HSPEED /
      IF(NUM.GT.0)GOTO1
      NUMH=0
```

```
1       GOTO(2,3,5)MAPGO
2       TFF=TFF+0.1*(TFF-TFS)
        WRITE(2,100)WT,TFS,TFF
        RETURN
3       CN=CN+0.05*(CN-CNS)
        PCN=PCN*(CN/CNS)
        T=T*(CNS/CN)**2
        WRITE(2,100)WS,CNS,CN
        IF(NUMH.GT.2)GOTO4
        NUM=1
        NUMH=NUMH+1
        RETURN
4       DELCN=CN-CNS
        IF(DELCN.GE.0) RETURN
        TFF=TFF*(1.+DELCN/CN)
        WRITE(2,101)WT,TFS,TFF
        RETURN
5       TFF=TFF+0.1*(TFF-TFS)
        WRITE(2,100)WT,TFS,TFF
        GOTO3
100     FORMAT(1X,12HTURBINE MAP ,A6,4HWAS=,E13.6,10H AND NOW=,E13.6)
101     FORMAT(1X,A6,22HWAS ALSO CHANGED FROM ,E13.6,5H  TO ,E13.6)
        END


END OF SEGMENT, LENGTH   203, NAME  MAPBAC

        SUBROUTINE MATRIX(E,V,A,N)
        DIMENSION E(10,10),V(10),A(10),PIV(11),T(10,11)
        NN=N+1
        NM=N-1
        DO1I=1,N
        T(I,NN)=A(I)
        DO1J=1,N
1       T(I,J)=E(I,J)
        DO7I=1,N
        TEMP=0.
        DO2J=I,N
        IF(TEMP.GT.ABS(T(J,I)))GOTO2
        TEMP=ABS(T(J,I))
        IPIV=J
2       CONTINUE
        IP1=I+1
        DO3J=IP1,NN
3       PIV(J)=T(IPIV,J)/T(IPIV,I)
        IFROM=N
        ITO=N
4       IF(IFROM.EQ.IPIV)GOTO6
        RM=-T(IFROM,I)
        DO5J=IP1,NN
5       T(ITO,J)=T(IFROM,J)+RM*PIV(J)
        ITO=ITO-1
6       IFROM=IFROM-1
        IF(IFROM.GE.I)GOTO4
        DO7J=IP1,NN
7       T(I,J)=PIV(J)
        DO8I=1,NM
        J=NN-I
        K=N-I
        DO8L=J,N
8       T(K,NN)=T(K,NN)-T(K,L)*T(L,NN)
```

```
      DO9I=1,N
9     V(I)=T(I,NN)
       RETURN
      END


END OF SEGMENT, LENGTH   416, NAME  MATRIX

      SUBROUTINE SEARCH(P,A,B,C,D,AX,NA,BX,CX,DX,NO,NAM,NOM,NCODE)
      DIMENSION AX(NAM), BX(NAM,NOM),CX(NAM,NOM),DX(NAM,NOM),Q(9)
      NCODE=0
      C=0.
      D=0.
C *** START BY FINDING A
      DO1I=1,NA
      IH=I
      IF(A.LT.AX(I))GOTO2
C *** IE: SCAN ALONG AX      LINES UNTIL REACH ONE HIGHER, THUS BRACKET
C *** THE 'A' LINE
1     CONTINUE
      IF(A.GT.AX(IH))NCODE=2
      A=AX(IH)
      GOTO3
2     IF(IH.GT.1)GOTO3
      NCODE=1
      IH=2
      A=AX(1)
3     IL=IH-1
C *** NOW FIND 'B'
      PRM=(A-AX(IL))/(AX(IH)-AX(IL))
      PP=P
      IF(P.GE.0.)GOTO6
      BL=BX(IL,1)+PRM*(BX(IH,1)-BX(IL,1))
      BH=BX(IL,NO  )+PRM*(BX(IH,NO  )-BX(IL,NO  ))
      IF(B.GE.BL)GOTO4
      NCODE=NCODE+10
      B=BL
      GOTO5
4     IF(B.LE.BH)GOTO5
      NCODE=NCODE+20
      B=BH
5     PP=0.5
      Q(2)=0.
      Q(3)=0.
6     BH=PP*(BX(IH,NO  )-BX(IH,1))+BX(IH,1)
      BL=PP*(BX(IL,NO  )-BX(IL,1))+BX(IL,1)
C *** IE; SCAN ALONG AX(IH) AND AX(IL) UNTIL FIND BX'S .GE. BL AND BH
      DO7J=2,NO
      JH=J
      IF(BH.LT.BX(IH,J))GOTO8
7     CONTINUE
8      JL=JH-1
      DO9K=2,NO
      KH=K
      IF(BL.LT.BX(IL,K))GOTO10
9     CONTINUE
10    KL=KH-1
C *** WHAT FOLLOWS IS A LINEAR INTERPOLATION (USING SIMILAR TRIANGLES)
C *** BETWEEN BH AND BL TO FIND POINT BT. SIMILARLY FOR CH,CL,AND CT;
C *** DH,DL,AND DT
```

```
      PR=(BX(IH,JL)-BH)/(BX(IH,JH)-BX(IH,JL))
      CH=CX(IH,JL)-PR*(CX(IH,JH)-CX(IH,JL))
      DH=DX(IH,JL)-PR*(DX(IH,JH)-DX(IH,JL))
      PR=(BX(IL,KL)-BL)/(BX(IL,KH)-BX(IL,KL))
      CL=CX(IL,KL)-PR*(CX(IL,KH)-CX(IL,KL))
      DL=DX(IL,KL)-PR*(DX(IL,KH)-DX(IL,KL))
      BT=BL+PRM*(BH-BL)
      CT=CL+PRM*(CH-CL)
      DT=DL+PRM*(DH-DL)
      IF(P.GE.0.)GOTO13
      DIR=SQRT(B/BT)
      ERR=(B-BT)/B
      CALL AFQUIR(Q(1),PP,ERR,0.,25.,0.001,DIR,PT,ICON)
      GOTO(11,13,12),ICON
11    PP=PT
      IF(PP.LT.0.)PP=0.
      IF(PP.GT.1.)PP=1.
      GOTO6
12    NCODE=7
13    B=BT
      C=CT
      D=DT
      RETURN
      END

END OF SEGMENT, LENGTH   698, NAME  SEARCH

      SUBROUTINE AFQUIR(X,AIND,DEPEND,ANS,AJ,TOL,DIR,ANEW,ICON)
      DIMENSION X(9)
      Y=0.
      IF(ANS)1,2,1
1     DEP=DEPEND-ANS
      TOLANS=TOL*ANS
      GOTO3
2     DEP=DEPEND
      TOLANS=TOL
3     IF(ABS(DEP)-TOLANS)5,5,4
4     IF(X(2)-AJ)8,8,7
5     ANEW=AIND
      X(2)=0.
      ICON=2
      RETURN
6     ANEW=Y
      X(2)=X(2)+1.
      ICON=1
      RETURN
7     ANEW=Y
      X(2)=0.
      ICON=3
      RETURN
8     IF(X(3))9,9,12
C *** FIRST GUESS USING DIR
9     X(3)=1.
      X(8)=DEP
      X(9)=AIND
      IF(AIND)10,11,10
10    Y=DIR*AIND
      GOTO6
11    Y=DIR
```

```
      GOTO6
12    IF(X(3)-1.)13,13,16
C *** LINEAR GUESS
13    X(3)=2.
      X(6)=DEP
      X(7)=AIND
      IF(X(8)-X(6))14,9,14
14    IF(X(9)-X(7)) 15,9,15
15    A=(X(9)-X(7))/(X(8)-X(6))
      Y=X(9)-A*X(8)
      IF(ABS(10.*X(9))-ABS(Y))9,9,6
C *** QUADRATIC GUESS
16    X(4)=DEP
      X(5)=AIND
      IF(X(7)-X(5))18,17,18
17    IF(X(6)-X(4))13,9,13
18    IF(X(6)-X(4))19,13,19
19    IF(X(9)-X(5))23,20,23
20    IF(X(8)-X(4))21,22,21
21    X(9)=X(7)
      X(8)=X(6)
      GOTO13
22     X(9)=X(7)
      X(8)=X(6)
      X(3)=1.
      IF(X(9))10,11,10
23    IF(X(8)-X(4))24,21,24
24    F=(X(6)-X(4))/(X(7)-X(5))
      A=(X(8)-X(4)-F*(X(9)-X(5)))/((X(9)-X(7))*(X(9)-X(5)))
      B=F-A*(X(5)+X(7))
      C=X(4)+X(5)*(A*X(7)-F)
      IF(A)27,25,27
25    IF(B)26,7,26
26    Y=-C/B
      GOTO47
27    IF(B)32,28,32
28    IF(C)30,29,30
29    Y=0.
      GOTO47
30    G=-C/A
      IF(G)7,7,31
31    Y=SQRT(G)
      YY=-SQRT(G)
      GOTO37
32    IF(C)34,33,34
33    Y=-B/A
      YY=0.
      GOTO37
34    D=4.*A*C/B**2
      IF(1.-D)13,35,36
35    Y=-B/(2.*A)
      GOTO47
36    E=SQRT(1.-D)
      Y=(-B/(2.*A))*(1.+E)
      YY=(-B/(2.*A))*(1.-E)
37    J=4
      DEPMIN=ABS(X(4))
      DO39I=6,8,2
      IF(DEPMIN-ABS(X(I)))39,39,38
38    J=1
```

```
      DEPMIN=ABS(X(I))
39    CONTINUE
       K=J+1
      IF((X(K)-Y)*(X(K)-YY))42,42,40
40    IF(ABS(X(K)-Y)-ABS(X(K)-YY))47,47,41
41    Y=YY
      GOTO47
42    IF(J-6)43,44,44
43    JJ=J+2
      KK=K+2
      GOTO45
44    JJ=J-2
      KK=K-2
45    SLOPE=(X(KK)-X(K))/(X(JJ)-X(J))
      IF(SLOPE*X(J)*(X(K)-Y))46,46,47
46    Y=YY
47    X(9)=X(7)
      X(8)=X(6)
      X(7)=X(5)
      X(6)=X(4)
      GOTO6
      END

END OF SEGMENT, LENGTH   995, NAME  AFQUIR

      SUBROUTINE NOZCO(PRAT,ARAT,CF)
      DIMENSION PR(30),AR(10),CFR(30,10),XX(3),YY(3),ZZ(2)
      DATA PR/1.0,1.25,1.5,1.75,2.0,2.25,2.5,2.75,3.0,3.25,3.5,3.75,4.0,
     14.25,4.5,4.75,5.,5.5,6.,6.5,7.,8.,9.,11.,13.,15.,17.,19.,21.,23./
      DATA AR/1.,1.05,1.092,1.228,1.318,1.423,1.8,1.9,1.97,2.15/
      DATA CFR/
     1.97,.975,.978,3*.98,.978,.977,.974,2*.97,.966,.963,.96,.957,.954,
     2.95,.965,.938,.932,.927,.92,.91,.9,.886,.876,.867,.86,.852,.844,.9
     363,.966,.97,.974,.978,3*.981,.979,.976,.974,2*.97,.967,.964,.96,.9
     46,.954,.95,.944,.94,.93,.924,.91,.9,.89,.884,.876,.87,.86,.956,.96
     5,.964,.967,.97,.973,.975,.978,4*.98,.976,.974,.972,2*.97,.964,.96,
     6.955,.95,.945,.94,.928,.92,.91,.9,.89,.883,.876,.947,2*.95,2*.955,
     7.96,.963,.967,.97,.974,.976,.978,2*.977,.976,.975,.974,.97,.967,.9
     864,.96,.955,.95,.94,.93,.923,.915,3*.9,3*.95,.947,.944,3*.94,.95,.
     9956,.96,.964,.967,.968,5*.97,.968,.966,.96,2*.944,.935,.927,.92,.9
     A13,2*.90,4*.948,.947,.944,.936,.928,.92,2*.916,.93,.94,.946,.95,.9
     B53,.956,4*.96,.957,.953,.945,.936,.928,.92,.91,.9,.892,.81,.815,.8
     C25,.833,.84,.852,.86,.87,.88,.89,.9,.914,.924,.935,.943,.95,.958,.
     D968,.975,2*.98,.982,.98,.978,.974,.97,.965,.96,.954,.95,.796,.8,.8
     E1,.82,.83,.838,.847,.856,.865,.876,.886,.898,.91,.92,.93,.94,.946,
     F.958,.966,.97,.975,2*.98,.977,.974,.97,.967,.962,.96,.954,.79,2*.8
     G,.81,.82,.83,.837,.846,.855,.864,.874,.884,.894,.91,.915,.925,.933
     H,.95,.96,.967,.97,2*.976,.98,.977,.974,.97,.966,.96,.956,.774,.78,
     I.787,.794,.8,.81,.817,.825,.833,.84,.85,.86,.87,.88,.887,.896,.91,
     J .926,.944,.953,.96,.966,.97,.973,3*.97,.966,.964,.962/
      Y(X1,X2,X,Y1,Y2)=(Y1-Y2)*(X-X1)/(X1-X2)+Y1
      N=0
      DO1J=1,30
1     IF(PRAT.GE.PR(J))N=J-1
      IF(N.EQ.0)N=1
      IF(N.GE.29)N=28
      M=0
      DO2K=1,10
2     IF(ARAT.GE.AR(K))M=K
```

```
      IF(M.EQ.10.OR.M.EQ.0)GOTO5
      DO4L=1,2
      LL=M+L-1
      DO3I=1,3
      NN=N+I-1
      XX(I)=PR(NN)
3     YY(I)=CFR(NN,LL)
      CALL PARABO(XX,YY,PRAT,ANS)
4     ZZ(L)=ANS
      CF=Y(AR(M),AR(M+1),ARAT,ZZ(1),ZZ(2))
      RETURN
5     WRITE(2,7)ARAT,AR(10)
      DO6I=1,3
      NN=N+I-1
      XX(I)=PR(NN)
6     YY(I)=CFR(NN,10)
      CALL PARABO(XX,YY,PRAT,CF)
      RETURN
7     FORMAT(8H ARATIO=,F7.3,4X,26H OUT OF RANGE,USE DATA FOR,F8.3)
      END

END OF SEGMENT, LENGTH   295, NAME  NOZCO

      SUBROUTINE PARABO(X,Y,XD,YANS)
      DIMENSION X(3),Y(3)
      A=((X(1)-X(2))*(Y(1)-Y(3))-(X(1)-X(3))*(Y(1)-Y(2)))/((X(1)-X(2))*(
     1X(1)-X(3))*(X(3)-X(2)))
      B=((X(1)**2-X(2)**2)*(Y(1)-Y(3))-(X(1)**2-X(3)**2)*(Y(1)-Y(2)))/((
     1X(1)-X(2))*(X(1)-X(3))*(X(2)-X(3)))
      D=(Y(1)*X(2)**2-Y(2)*X(1)**2-B*X(2)*X(1)*(X(2)-X(1)))/(X(2)**2-X(1
     1)**2)
      YANS=(A*XD+B)*XD+D
      RETURN
      END

END OF SEGMENT, LENGTH   280, NAME  PARABO

      SUBROUTINE RESET
      COMMON/INITIAL/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10
      N1=-1
      N2=-1
      N3=0
      N4=0
      N5=0
      N6=0
      N7=0
      N8=0
      N9=0
      N10=0
      RETURN
      END

END OF SEGMENT, LENGTH    64, NAME  RESET
```

```
 BLOCK DATA
 COMMON/AIRCONST/CL0,CL1,CL2,CL3,CL4,CH0,CH1,CH2,CH3,CH4,TCP0,
1TCP1,TCP2,TCP3,TCP4,      TH0,TH1,TH2,TH3,TH4,     TFI0,TFI1,TFI2,
2TFI3,TFI4,      CLIH,CHIH,CLIF,CHIF,B
 DATA CL0,CL1,CL2,CL3,CL4/.24336328,-.32921648E-04,.47395140E-07,
1.10126885E-09,-.89883655E-13/
 DATA CH0,CH1,CH2,CH3,CH4/.19075549,.12752498E-03,-.54651988E-07,
1.89378182E-11,0./
 DATA TCP0,TCP1,TCP2,TCP3,TCP4      /.11799097,.44934282E-03,-.148
145732E-06,.48002619E-10,-.11380192E-13/
 DATA TH0,TH1,TH2,TH3,TH4     /-.29457579E+02,.14564900,.17564917E
1-03,-.95734930E-08,-.34413546E-11/
 DATA TFI0,TFI1,TFI2,TFI3,TFI4      /-.40453568E+01,.13713103E-01,
1-.63212887E-05,.23746611E-08,-.39087250E-12/
 DATA CLIH,CHIH,CLIF,CHIF,B/-.40567100,.11317630E+02,.32050096E+01,
1.70344726E+01,.14587635E+02/
 END


 BLOCK DATA
 COMMON/CUMB/  P(10),DELT(10,8),ETA(10,8)
 DATA P/0.3341,0.6684,1.0027,1.3368,1.6711,2.0053,2.3390,
12.6737,6.8046,34.0230/
 DATA DELT/10*111.1,10*166.65,10*222.2,10*277.75,10*333.3,
110*444.4,10*555.5,10*888.8/
 DATA ETA/
1.600, .726, .777, .806, .826, .855,2*.870,
2.758, .825, .858, .875, .888, .906, .914, .915,
3.868, .893, .911, .925, .935, .947,2*.953,
4.925, .936, .946, .955, .963, .974, .978, .979,
8.960, .966, .972, .977, .982, .990, .993, .995,
6.988, .991, .992, .994, .995, .998,2*.999,32*1./
 END

 BLOCK DATA
 COMMON/FUEL/ CV,FS,WTMOL
 DATA  CV,FS,WTMOL/10300.,.06823,28.969/
 END

 FINISH
```