AD-A047 954 COMPUTER SCIENCES CORP HUNTINGDON VALLEY PA F/G 1/3
F-18 ESCAPE SYSTEM COMPUTER SUPPORT.(U)
OCT 77 N62269-75-C-0001

UNCLASSIFIED NL

1 OF 2
AD
A047954



AD A047954

F-18 ESCAPE SYSTEM COMPUTER SUPPORT

FINAL REPORT

CDRL ITEM #A004

Task Order No. 22

Contract N62269-75-C-0001

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

[redacted]

[redacted]

[redacted]

September 1977

UNCLASSIFIED

COPY

CSC

COMPUTER SCIENCES CORPORATION

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

CSC

(15)

(6) F-18 ESCAPE SYSTEM COMPUTER SUPPORT.

(9) FINAL REPORT.

CDRL ITEM #A004

Task Order No. 22

(15) Contract N62269-75-C-0001 new

Prepared for

NAVAL AIR DEVELOPMENT CENTER

Warminster, Pennsylvania

DDC
RECEIVED
DEC 21 1977
E

(11) October 1977

(12) 182p.

COMPUTER SCIENCES CORPORATION New

101 Masons Mill Business Park
Masons Mill Road
Huntingdon Valley, Pennsylvania, 19006

410506

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

410 506

LB

TABLE OF CONETENTS

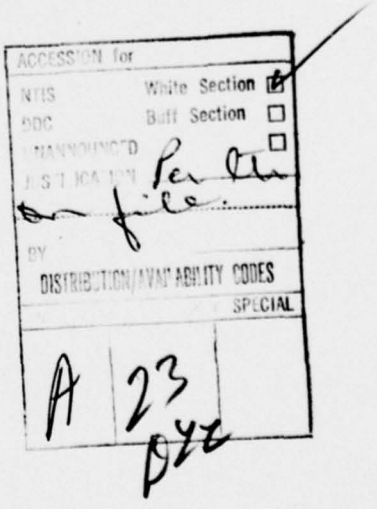
		PAGE
1.0	INTRODUCTION	1
2.0	MATHEMATICAL MODELLING CHANGES AND ADDITIONS	4
2.1	Dynamic Response Index (DRI)	4
2.2	Dynamic Center of Gravity	5
2.3	Martin-Baker Rail and Tube Bending Equations	6
2.4	Martin-Baker Timing and Sequencing	6
3.0	PROGRAMMING CHANGES	8
3.1	Input/Output Modifications	8
3.1.1	Segmentation of Input Routine	8
3.1.2	Additions to Decimal Data	9
3.1.3	Input/Output User Print Options	9
3.1.4	Additional Output Subroutines, Reports and Plot File	9
3.2	Position Event Time Determination	10
3.3	Simplification of Subroutine RKSDEQ Coding	11
3.4	Interfacing of Martin-Baker Timing and Sequencing in Subroutine Status	11
3.5	Documentation of New Subroutines	11
4.0	USER INPUT DATA INSTRUCTIONS	15
4.1	Definition of Input Variables	15
4.1.1	Definition of User's Input/Output Options Card Format	15
4.1.2	Definition of the New Decimal Data Sections	17
4.1.3	Definition of the New Integer Data Variables	21
4.2	Program Load and Execute, Output and Program Load Map Procedures	22
4.2.1	CALL, PRLD	22
4.2.2	CALL, TAPOUT	22
4.2.3	CALL, MAP	24
5.0	HOW TO INSERT NEW DATA AND NEW REPORTS	24
5.1	Additions to the Default Decimal Data	24
5.2	Additions to the Default Integer Data	25
5.3	Implementation of New Reports	26

TABLE OF CONTENTS
(Continued)

		PAGE
6.0	ICARUS FLOW DIAGRAMS	27
6.1	Macro Structure of ICARUS	27
6.2	Macro Subroutine Flow Diagram	28
6.3	Detailed Flow Diagrams of the New Subroutines Inserted into Model	36
REFERENCES		165

APPENDICES

APPENDIX A - Dynamic Response Index (DRI) Equations
APPENDIX B - Dynamic Center of Gravity Equations
APPENDIX C - Martin-Baker Catapult Equations



F-18 ESCAPE SYSTEM COMPUTER SUPPORT

FINAL REPORT

1.0 INTRODUCTION

The following report documents the work completed by Computer Sciences Corporation (CSC) under Task Order Number 22, Contract N62269-75-C-0001 for the Crew Systems Department at the Naval Air Development Center (NADC), Warminster, Pennsylvania. This report documents the modifications and additions that were made to an existing emergency egress, pilot ejection from an aircraft, simulation model. The original model, nicknamed ICARUS by the originators, was developed and implemented in FORTRAN on a CDC 6600 computer by the Aeroballistics Division of the Naval Weapons Laboratory (NWL) located in Dahlgren, Virginia.

Documentation of the original version of ICARUS is covered under NWL Technical Report Number TR-3098, dated February, 1974 and titled: 'Air Crew Automated Escape System Simulation Model' and NWL Technical Note TN-K-7-7/74 dated February, 1974 and titled: 'Program Maintenance Manual for ICARUS A Computer Program to Simulate Escape Systems.'

The original model was developed to simulate the ejection seat system of the AV-8A (Harrier) aircraft. The ejection system was designed and manufactured by the STENCEL Corporation. The model was constructed and programmed in a modular form, thus enabling the current or future users of the model to modify or insert new modules into the model in order to adapt the model to hardware changes to the original ejection system or different ejection systems.

The basic structure of the model can be considered as follows:

- Model Control Section - This section controls the overall model data initialization, the individual case study data initialization, requests for aircraft and the seat ejection normal trajectory computations, trajectory data output, adjustments to the trajectory time step (deltat) and case or problem termination.

- Positional or Time Event Activation/Deactivation Section - This section, called by the above section after each step of the trajectory computation, keeps track of each of the position and time event's status during the trajectory. When an event occurrence has been detected by this section, the section initiates a request to have a message outputted indicating that the particular event has taken place, the time of occurrence to be computed and outputted along with the appropriate trajectory data computed at that time. This section is also responsible for the dynamic scheduling of the various equations of motion to be evaluated at each phase of the trajectory computation.

The first request from the NADC Crew Systems Department was that CSC change the output format of the model (ICARUS) from a paragraph to a tabular format. A series of new reports were designed displaying the trajectory data in tabular form (up to nine components per page) with appropriate page and title headings. The input routine was modified to read-in, immediately after the case label card, an output display selection card. The input routine was further modified to interpret the data read-in from this card and to set program transfer switches within the program to cause the printing or inhabiting of the printing, of the data or reports selected by the user via this card. This card enabled the user of the model to select either the original, new or both report formats to be generated and displayed. If the new report format mode of output data display is selected by the user, the ability to select specific reports contained within the mode was also provided via this card. Simultaneously, the original input routine was segmented into smaller routines and the user of the model was given the option (also via the above card) of selecting which input data segment was to be displayed and whether the recovery chute data and plot data is to be outputted.

CSC was then asked to modify ICARUS to compute and display the pilot's dynamic response index (DRI), a measure of the pilot's spinal compression. A spring-damper system is used as a model of the pilot's spine during the ejection

phase. The DRI is computed as a function of the vertical acceleration imposed on the man during this phase. The computation was implemented in the model and a new report was designed to display this data at each time step of the ejection phase. This accomplished, CSC was then asked by Crew Systems to modify the model to take into account the displacement of the pilot's center of gravity during the ejection phase (from catapult ignition to seat/man separation). The displacement of the pilot's center of gravity was modelled by Crew Systems using spring damper equations (one for each direction of motion). These equations were inserted into the model, evaluated and the appropriate vectors were updated to reflect the displacement of the pilot's center of gravity from its original (static) position. A new output report displaying the displacement, acceleration and force components in each direction of motion, was designed and implemented in the model.

The preceeding modifications to the model are considered common to all ejection systems. The following modifications have been implemented in the model in order to adapt and use the model to simulate a different ejection system namely, the Martin-Baker.

The Martin-Baker catapult was modelled by Crew Systems using spring-damper and tube extension equations. The rail interaction with the seat slippers was modelled using spring equations to simulate and determine the friction and normal forces and moments generated along each axis as the slippers move up the rails during the catapult phase. The forces and moments acting on the seat/man combination due to the bending of the tube while extending were modelled using tube bending equations. The above equations were interfaced and implemented in the model. Additional reports displaying the slipper and tube bending position, force and moment data were designed and implemented in ICARUS.

The model was then modified to simulate the Martin-Baker seat back rocket, drogue and recovery chute sequencing logic.

Three new decimal data input sections were added to the model allowing the user of the model to input, via the model's normal input data procedures, the parameters to the dynamic center of gravity, the Martin-Baker catapult and phasing equations.

The following sections (not necessarily in chronological order of implementation) describe in detail the mathematical modelling, timing and sequencing logic, the input/output changes and additions to the model, a description of the new routines that were implemented and interfaced with the rest of the model, macro and detailed flow diagrams of the model, a definition of the new input variables and how to enter them, guidelines as to how to go about implementing the new input variables, and finally how to run the model.

2.0 MATHEMATICAL MODELLING CHANGES AND ADDITIONS

2.1 Dynamic Response Index (DRI)

A measure of the compressibility of the pilot's spine during the ejection phase of the trajectory has been implemented and interfaced in ICARUS. The pilot's spine is modelled as a mechanical system, a spring-damper. The index of compressibility (DRI) of the man's spine is determined at each time step of the phase as a function of the vertical acceleration (in G's) imposed on the seat/man combination and the resulting displacement value obtained from the spring equation during this phase. Subroutine DERIV has been modified to call a new subroutine titled 'DRICALC' during the ejection phase. Subroutine DRICALC evaluates the spring-damper equation using the velocity and position data obtained after each numerical integration iteration. The dynamic response index is then computed using the displacement value obtained from the integration.

Reference Appendix A for a mathematical presentation of the equations used to determine the DRI.

2.2 Dynamic Center of Gravity

The orignial model was modified to take into account the displacement of the pilot's center of gravity during the ejection phase of the trajectory, from catapult ignition to seat/man separation. The displacement of the man's center of gravity is simulated during this phase by three independent second-order differential equations of motion (spring-damper equations) one for each direction of motion. The equations are numerically integrated at each time step as a function of the resultant axial acceleration imposed on the seat/man combination - yielding the corresponding velocity and displacement components of the man's center of gravity from its static position. The seat/man center of gravity is then recomputed and the appropriate vectors from/to this new position are then updated in the seat/man coordinate system.

The user of the model has been given the option of forcing the model to compute the displacement of the man's center of gravity or ignoring this displacement as was originally done in the model. This option is controlled by the user of the model setting the fifth word of the integer input data array to an integer one if the displacement of the man's center of gravity from its static position is to be computed, a zero or blank entry will force the model to by-pass this computation.

Subroutine DERIV has been modified to call a new subroutine titled 'DYNCG' during the ejection phase. Subroutine DYNCG calls another new subroutine titled 'KUTTA' which performs the numerical integration of the dynamic center of gravity equations, obtaining velocity and displacement components which are then used to evaluate the equations. After the evaluation is completed, DYNCG calls another new subroutine titled 'TRANSFM' which updates the vectors from/to the new seat/man center of gravity location. See Section 4.3 for instructions on how to envoke or inhibit the dynamic center of gravity computation. See Appendix B for a mathematical presentation of the dynamic center of gravity equations.

2.3 Martin-Baker Rail and Tube Bending Equations

The ICARUS seat ejection model has been modified to simulate either the Stencil or the Martin-Baker seat ejection system. The user of the model can select which seat is to be simulated by setting the value of the input decimal variable, 'STNCEL' to a zero (0.0) if the Stencel system is to be simulated or to a decimal one (1.0) if the Martin-Baker system is to be simulated. The variable STNCEL is contained in the decimal portion of the Default data file and its value may be altered by the user of the model via the normal Card Input file procedures.

2.3.1 Rail and Tube Bending Equations

The Martin-Baker catapult system has been modelled using spring-damper and tube bending equations to simulate the forces and moments acting on the seat/man during the catapult phase of the trajectory.

Subroutine SEATMAN has been modified to interpret the value of input variable STNCEL to ascertain what seat is being simulated by the model. If STNCEL equals 1.0 and the model is in the catapult phase of the trajectory - subroutine SLIPFMB is called. This new subroutine contains the Martin-Baker catapult equations of motion. These equations (the rail and tube bending) are evaluated and the forces and moments acting on the seat/man combination due to the rail interaction with the slippers and tube bending are determined as a function of the velocity and displacements resulting from the slippers and tube bending motions.

If on the other hand, STNCEL is input as a zero or blank and the model is in the catapult phase of the trajectory, subroutine SLIDFM is called and the forces and moments attributable to the Stencel catapult will be computed.

2.4 Martin-Baker Timing and Sequencing

Subroutines EXECUT and STATUS have been modified to interpret the input decimal value of variable STNCEL to ascertain which seat ejection system

is being simulated by the model. As previously mentioned in Section 2.3.1 if the value entered into variable STNCEL is 0.0, the Stencel seat ejection system will be simulated by the model, if the location contains the value of 1.0, the Martin-Baker system will be simulated.

In the Martin-Baker mode of simulation, subroutine STATUS has been modified to check, at each time step of the trajectory, if the catapult tubes 1 and 2 have extended beyond the value specified by input variable SBRON contained in the decimal data segment of the Default data file. If the tubes have extended past this value, the time ($\pm$ some delta) when the tube length equalled the value specified by the SBRON variable is acertained. The values of the trajectory variables at this time are computed and displayed and the seat back rocket phase is initiated.

Subroutine INIT has been modified to compute the time of drogue projection (TDP) as a function of the decimal input variables time of catapult ignition (TCI) and the time required to elapse (TMLAP1) before the drogue chute can be projected, TDP = TCI + TMLAP1.

Subroutines STATUS and EXECUTE have been modified to update or hold constant at each time step of the trajectory an internal clock. This clock has a value of zero initially and updating of the clock commences at TCI. The clock is updated under the following conditions:

if the seat/man altitude $<$ ALT1

or

ALT1 $\leq$ seat/man altitude $\leq$ ALT2 and $ACC_Z <$ TIM2GZ

then the clock is updated - the integration time step is added to the clock location (TIMER2=TIMER2+DELTAT).

ALT1, ALT2, TIM2GZ, and TMLAP2 are input via either the Default data file or the Card Input file.

When the internal simulation clock attains the value of TMLAP2, the recovery chute pack is opened and the recovery chute riser line commences coming out.

3.0 PROGRAMMING CHANGES

3.1 Input/Output Modifications

3.1.1 Segmentation of Input Routine

The original ICARUS subroutine INPUT which reads-in the aerodynamic, tabular, decimal and integer data has been segmented into the following ten subroutines: ACT, CONVT, DEFAULT, ENDCASE, INPUT, PTDFLT, TABCONL, TABCRDS, UPDEC and UPINTG. The control routine, subroutine INPUT, computes and at case termination outputs the case execution time, reads-in the case title and user input/output option cards and displays them on the case output listing. The routine then calls subroutine DEFAULT which reads-in the decimal and integer data from the Default file. DEFAULT calls subroutine CONVT which converts the decimal data input as feet or radians to inches or degrees. DEFAULT reads-in the next record contained on the Card Input file and then transfers program control back to subroutine INPUT. Depending on the values of QFLAG read from the Card Input file, subroutines PTDFLT, UPDEC, UPINTG and ENDCASE are called. Subroutines UPDEC and UPINTG reads-in from the Card Input file updates to the decimal and integer data.

Subroutine PTDFLT transfers to the user's output file the updated decimal and integer default data. PTDFLT then calls subroutine TABCONL which controls the reading-in and output of the aerodynamic and tabular data via subroutines ACT and TABCRDS. Subroutine ACT, reads and outputs the aerodynamic data contained on file AERO. Subroutine TABCRDS reads and outputs the tabular data contained on the Card Input file. Before transferring program control back to subroutine INPUT, PTDFLT calls subroutine CONVT which converts the decimal data back to feet or degrees.

Finally, subroutine ENDCASE is called by INPUT and a check is made within this subroutine to ensure that all input data required to execute the case has been read.

3.1.2 Additions to the Decimal Data

Three additional input sections have been implemented in ICARUS, Sections 12, 13 and 14. ICARUS' input and output routines have been modified to read the data contained in these sections, from the Default and Card Input files and to display this information along with the other decimal data. Section 12 contains the input parameters to the dynamic center of gravity equations, Section 13 the parameters of the Martin-Baker catapult equations and Section 14 the parameters to the Martin-Baker timing and sequencing logic. For a detailed definition of the variables contained in the above new sections see Section 4.1.2 of this report.

3.1.3 Input/Output User Print Options

An additional input card has been added to the Card Input file. This card must immediately follow the case label card. The card has been added to the model input card stream to allow the user of the model to select which input data and output reports are to be transferred to the user's output file for subsequent printing. See Section 4.1.1 of this report for a detailed explanation of how this card is used to control the input/output data display.

3.1.4 Additional Output Subroutines, Reports and Plot File

3.1.4.1 Tabular Displays

At the request of Crew Systems, ICARUS' output format was changed from a paragraph format to a tabular format display. A series of new reports were designed displaying the data in tabular form (up to nine components per page) with appropriate page and column headings. To accomplish this, three new subroutines (FETCH, NREP, and REP1) have been implemented and interfaced with ICARUS' input and output routines. Subroutine PROUT has been modified to call subroutine REP1 when integer variable IPROP, read-in from the user's input/output options card, has a value other than zero, indicating that the user has selected at least one of the new output reports to be generated and displayed. Subroutine REP1 interrogates phase indicator variables and based upon the pre-

sent phase of the trajectory a transfer within the routine is made to the appropriate Fortran subroutine call statement to subroutine FETCH. The parameters of the call statement indicate the new reports that can be considered for output in this phase.

Subroutine FETCH ascertains which of the new reports that are logically possible to output at this time have actually been selected by the user to be displayed. The values of the trajectory data displayed in the user selected reports are then passed, one report at a time, to subroutine NREP which formats and transfers the line of data to the associated report tableau for subsequent output.

3.1.4.2 Plot File

An additional new subroutine, CUTTAPE, has been interfaced and implemented in ICARUS. This routine is called by subroutine FETCH. The purpose of subroutine CUTTAPE is to format and output, to a plot file at each step of the trajectory, seat/man positional, rate and angular data. The plot file generated by CUTTAPE is then read by a separate Fortran/PLOT10 program via the Tektronix terminal. The plots of the data generated by this program are copied using the Tektronix hard copying attachment.

3.2 Position Event Time Determination

Subroutine HIT controls the determination of the exact time (within a given tolerance) in the trajectory that a specified position value has been obtained, e.g., the instant in time when the catapult tube has extended its full length implying seat/man separation from the Martin-Baker catapult. After the exact time of the event has been determined and output by a call to subroutine PROUT along with the associated values of the variables at that time, HIT has been modified to set NTRY equal to 4 and to call RKSDEQ. Entry into subroutine RKSDEQ with NTRY equal to 4 forces the routine to by-pass the updating of the work array which contains the values of integration variables at time t_i, t_{i-1}, and t_{i-2}. Thus the values of the variables at time t_i upon entry into the routine are not retained. The routine then performs the integration over the interval from the event time back to

the time when the event was detected as occurring.

Program control is returned to HIT which then calls PROUT forcing the output of the new values of the variables at the time of the event detection. HIT has been modified to update the Lagrange array YSAL with the new values of the variables obtained from the above integration. In addition, HIT has been modified (Martin-Baker seat only) to compute the seat back rocket burnout time immediately after it has determined the time of seat back rocket ignition.

3.3 Simplification of RKSDEQ Coding

The overall coding of subroutine RKSDEQ (the main numerical integrating routine) has been simplified and an additional transfer point was added to the routine. When the routine is called with NTRY set to 4, the updating of the work array before integration is not accomplished and the values of the current time variables are not retained.

3.4 Interfacing of the Martin-Baker Time and Sequencing in Subroutine STATUS

Subroutine STATUS has been modified to take into account the anomalies that exist between the Stencel and the Martin-Baker event determination. See Section 2.4 of this report for a detailed explanation of the modifications made to STATUS.

3.5 Documentation of New Subroutines

The following new subroutines have been implemented in ICARUS:

ACT	ENDCASE	RKSDEQ (modified and recorded)
CENGRAV	FETCH	RUNGE
CONVT	IDEBUG	PTDFLT
CUTTAPE	INPUT	SLIPFMB
DANGLE	KUTTA	STORE
DEFAULT	NREP	TABCONL
DRICALC	RECALL	TABCRDS
DYNCG	REP1	TRANSM

UPDEC

UPINTG

Section 4.3 of this report contains the detailed flow charts of these new subroutines.

3.5.1 Subroutine ACT, called by TABCONL, reads and outputs the aerodynamic data contained on file AERO. The output of the aerodynamic coefficients may be inhibited by the user of the model via the input/output options card.

3.5.2 Subroutine CENGRAV, called by DYNCG, evaluates the equations of motion (spring equations) used to determine the displacement of the pilot's center of gravity from its static position during the seat/man portion of the ejection trajectory.

3.5.3 Subroutine CONVT is called by DEFAULT and PTDFLT. Based upon the values contained in the locations of the formal parameters passed, CONVT will convert the decimal data represented in feet and radians to inches and degrees or vice versa.

3.5.4 Subroutine CUTTAPE generates a file containing data to be plotted. At present the following data is output:

(X_S, Y_S, Z_S) - position of the seat/man in (EFCS)

(M_S, V_S, W_S) - velocity of the seat/man in (SMCS)

(ψ, θ, φ) - yaw, pitch and roll angles of seat/man

($\dot{\psi}$, $\dot{\theta}$, $\dot{\varphi}$) - yaw, pitch and roll angular rates of seat/man

(Xaccel, Yaccel, Zaccel) - acceleration of seat/man in (SMCS)

3.5.5 Subroutine DANGLE, called by SEATMAN, computes the derivatives of the yaw, pitch and roll angles (ψ, θ, φ).

3.5.6 Subroutine DEFAULT, called by INPUT, reads-in the decimal and integer data from the Default file.

3.5.7 Subroutine DRICALC called by subroutine DERIV evaluates the equation of motion (spring damper) used to determine the index of compressibility of the man's spine during the seat/man phase of the ejection trajectory.

3.5.8 Subroutine DYNCG called by subroutine DERIV computes the new location of the man's and the seat/man's center of gravity as a function of the displacement computed in CENGRAV during the seat/man portion of the trajectory.

3.5.9 Subroutine ENDCASE called by INPUT checks to see that all input data required to execute the case has been read-in.

3.5.10 Subroutine FETCH called by subroutine REP1 ascertaines which of the new reports that can be generated and output during this phase have actually been selected by the user to be output. The user selected reports are then further processed, one at a time, to determine which of the variables contained in the report are to be output during this phase, and pointers are set to the core locations where the variables have been stored during this phase. This information is then passed to subroutine NREP which transfers the line of data to the associated report output file.

3.5.11 Subroutine IDEBUG produces reports that are not normally required to be displayed. These reports are used as aides in debugging the model or to help in the analysis of a given simulation test case.

See Section 4.2.3 of this report for instructions as to how to envoke these reports.

3.5.12 Subroutine INPUT called by EXECUT computes and outputs the case execution time at the termination of each case, reads the case title, user's Input/Output Options card, and outputs these two cards as part of the users data listing. INPUT then calls subroutine ACT, DEFAULT, ENDCASE, PTDFLT, TABCONL, UPDEC, and UPINTG. These routines read and output the aerodynamic, tabular, default and card input data files, and insure that all data required to execute the problem case have been read.

3.5.13 Subroutine KUTTA called by DYNCG is the fourth order Gill's variation of the Runge-Kutta numerical integration algorithm. This routine is used to integrate the dynamic center of gravity equations.

3.5.14 Subroutine NREP called by subroutines FETCH and INPUT formats and transfers the line of data passed to it by either subroutine FETCH or INPUT, to the output file designated to display the data.

3.5.15 Subroutine RECALL called by subroutine RKSDEQ replaces the t_i values of seat/man center of gravity position, rate data, seat/man vectors, Martin-Baker catapult, dynamic response index, and the derivates for the dynamic center of gravity and dynamic response equations with their corresponding values at t_{i-1}. These values are then used to determine the exact time an event has taken place.

3.5.16 Subroutine REP1 is called by subroutine PROUT when the new report format has been selected by the user to display output data. Subroutine REP1 interrogates phase indicator variables and based upon the phase of the trajectory subroutine FETCH is called with a formal parameter string indicating which of the new reports may be output during this phase.

3.5.17 Subroutine RKSDEQ called by subroutines CKTME, CKVARB, EXECUT and STATUS has been modified to by-pass the updating of the work array when entered with NTRY set to 4. The overall coding of the routine has been simplified.

3.5.18 Subroutine RUNGE called by subroutine DRICALC numerically integrates the dynamic reflex index equation. The numerical technique used is a Runge-Kutta forth order method.

3.5.19 Subroutine PTDFLT, called by subroutine INPUT, prints the default data. PTDFLT then calls subroutine TABCONL which controls the reading-in and outputting of the aerodynamic and tabular data via subroutines ACT and TABCRDS.

3.5.20 Subroutine SLIPFMB, called by subroutine SEATMAN during the Martin-Baker catapult phase, computes the forces and moments of the seat/man

due to the rail interaction with the seat slippers and catapult tube bending.

3.5.21 Subroutine STORE called by subroutine INIT and RKSDEQ saves the values at time t_i of the: seat/man center of gravity position and rate data, Martin-Baker catapult data, DRI data, and the deriviatives of the dynamic center of gravity and reflex index, after each integration.

3.5.22 Subroutine TABCONL called by subroutine PTDFLT controls the reading of the aerodynamic and tabular data via subroutines ACT and TABCRDS.

3.5.23 Subroutine TABCARDS called by subroutine TABCONL reads the tabular data from the Card Input file and transfers the data to the output file.

3.5.24 Subroutine TRANSFM called by subroutine DYNCG updates the vectors from/to the seat/man center of gravity during the ejection phase of the trajectory.

3.5.25 Subroutine UPDEC called by subroutine INPUT reads from the Card Input file updates to the decimal data.

3.5.26 Subroutine UPINTG called by subroutine INPUT reads from the Card Input file updates to the integer data.

Section VI-3 of this report contains detailed flow diagrams of the above new routines.

4.0 USER INPUT DATA INSTRUCTIONS

4.1 Definition of Input Variables

4.1.1 Definition of User's Input/Output Options Card Format (6I2, 3X, 35I1).

Subroutine INPUT has been modified to read-in an additional input card. This card is mandatory and must immediately follow the case title card. The card contains the following user coded output display switch settings:

Card Col.	Col. Name	
2	IPROP	A zero or blank entry will force the program to display output variables in original ICARUS format. A '1' keypunched in this column will force the program to display output variables in the new report format, Reports 2-23 A '2' keypunched in this column will force the program to display output variables in both the original and new formats.
4	IPLOT	A zero or blank entry in column 4 of the card will inhibit the generation of the file. A '1' keypunched in column 4 of the card will force the program to generate a file containing data to be plotted.
6	IRECH	A zero or blank entry in column 6 of the card will inhibit the outputting of the recovery chute angle and reference area. A '1' keypunched in column 6 of the card will force the program to display the recovery chute angle of attack and reference area after each iteration.
8	IAERO	A zero or blank entry in column 8 of the card will inhibit the outputting of the seat/man aerodynamic data. A '1' keypunched in column 8 of the card will force the program to output the aerodynamic data.
10	IDFLT	A zero or blank entry in column 10 of the card will inhibit the outputting of the default data. A '1' keypunched in column 10 of the card will force the program to output the default data.
12	ITABP	A zero or blank entry in column 12 of the card will inhibit the outputting of the tabular data.

A '1' keypunched in column 12 of the card will force the program to output the tabular data.

Card Cols.	Col. Name	
16,17,18,...35	IRSW (1,2...23)	A '0' or blank entry in these columns will force the program to inhibit the printing of corresponding new reports (Reports: 1, 2, 3 23).
		A '1' keypunched in these columns will force the program to print the corresponding new reports (Reports: 1,2,3...23).

Presently, reports 2 to 23 have been defined and implemented. Subroutines ICARUS, INPUT, PROUT and RECOV have been modified and three new subroutines (FETCH, NREP, REP1) have been added to the model to interpret the above inputted switch settings and to generate the user requested output reports.

4.1.2 Definition of the New Decimal Data Sections

The following three additional input sections have been added to the decimal data segment of the Default file:

4.1.2.1 Section 12, Dynamic Center of Gravity

Section 12 contains the following 12 input parameters to the dynamic center of gravity equations:

CX	XSLACK	SXP
SXN	CY	SY
CZ	ZSLACK	SZP
ZBOT	SZN1	SZN2

Where:	CX, CY, and CZ	are the damping constants in the X, Y and Z directions respectively, expressed in units of per-second.
	SXP, SXN	spring modulus constants in the X direction, expressed in units of per-second ^{2}.

XSLACK	dead zone in X direction ($0 < X \leq$ XSLACK) where the frictional acceleration term (FX) is set to zero.
SY	spring modulus constant in the Y direction, expressed in units of per second ^{2}.
SZP, SZN1, and SZN2	spring modulus constants in the Z direction, expressed in units of per second ^{2}.
ZBOT	bottoming distance in the Z direction, expressed in feet.
ZSLACK	upward slack in the Z direction ($0 \leq Z \leq$ ZSLACK) where the frictional acceleration term (FZ) is set to zero, expressed in feet.

Reference Appendix B for a mathematical presentation of the above.

4.1.2.2 Section 13, Martin-Baker Catapult

Section 13 contains the following 26 input parameters to the rail and tube bending equations plus the program control switch STNCEL indicating which ejection seat system is to be simulated by the model (STNCEL set to 0.0 implies the Stencel system, set to 1.0 implies the Martin-Baker system):

XKSLIP	ZKSLIP	XCSLIP
ZCSLIP	UMSLIP	XYKTOR
XYCTOR	YKTUB	PKTUB
RKTUB	YCTUB	PCTUB
RCTUB	UMTUB	TUBLT1
TUBLT2	TUBLT3	ELMOD
DIAMI1	DIAMO1	DIAMI2
DIAMO2	XCTCP	YCTCP
ZCTCP	STNCEL	RAILNT

Where: XKSLIP, ZKSLIP are the spring modulus constants in the X and Z directions of the rail/slipper equations in units of /sec ^{2}.

XCSLIP, ZCSLIP	are the damping constants in the X and Z directions of the rail/slipper equations in units of /sec.
UMSLIP	coefficient of friction used in rail/slipper equations (Y direction).
XYKTOR	rail/slipper torsional spring modulus constant in the pitch plane (X-Y) in units of (ft-lbs/rad).
XYCTOR	rail/slipper torsional spring-damper constant in the pitch plane (X-Y) in units of (ft-lbs - sec/rad).
YKTUB, PKTUB, RKTUB	catapult torsional spring modulus constants in the yaw, pitch and roll direction in units of (ft-lbs/rad).
YCTUB, PCTUB, RCTUB	catapult torsional spring damper constants in the yaw, pitch and roll directions in units of (ft-lbs-sec/rad).
UMTUB	catapult coefficient of friction in Y direction.
TUBLT1, TUBLT2, TUBLT3	length of tubes 1, 2 and 3 respectively in feet.
DIAMI1, DIAMI2	inside diameters of tubes 1 and 2 respectively in feet.
DIAMO1, DIAM02	outside diameters of tubes 1 and 2 respectively in feet.
XCTCP, YCTCP, ZCTCP	coordinates of the seat contact point with catapult tube in feet (CCS).
RAILNT	length of slipper rails in feet.

See Appendix C for a mathematical presentation of the use of the above variables.

4.1.2.3 Section 14, Martin-Baker Phasing

Section 14 contains the following 7 input parameters to the Martin-Baker drogue chute, seat back rocket ignition, recovery chute and seat/man separation simulation:

TMLAP1	TMLAP2	ALT2
ALT1	TIM2GZ	SMSEPF
SBRON		

Where:

TMLAP1	is the time (from catapult ignition) of drogue projection for the Martin-Baker system.
TMLAP2	is the time that the recovery chute clock must advance to, from a zero initial value, before the Martin-Baker recovery chute pack will open.
ALT1	altitude below which the recovery chute clock is updated automatically until TMLAP2 is reached (in feet).
ALT2	if seat/man are above this altitude, the recovery chute clock is not updated.
TIM2GZ	vertical acceleration (in G's) which will stop the clock from being updated if the man's altitude is below ALT2 and above ALT1 and his vertical acceleration is greater than TIM2GZ.
SMSEPF	resultant force that the recovery chute must exert (greater than or equal to) on seat/man to cause the man to separate from the seat, in units of lbs.
SBRON	distance seat must travel up the catapult tube before the seat back rocket will ignite, in feet.

Changes to the decimal data, defined above for Sections 12, 13 and 14, can be accomplished at run-time via the normal Card Input procedures, i.e., entering in the decimal data section of the file: the variable name(s) (columns 2-7)

28-33 and 54-59), and corresponding values (columns 8-27, 34-53, and 60-79).

Reference Section 3 of NWL Technical Note TN-K 7/74 dated February 1974.

4.1.3 Definitions of the New Integer Data Variables

The second segment of the Default data file contains the integer input data to the model. The first word of this segment contains the count (30) of the number of integer data values that immediately follow. The next two records contain the integer values to be read-in under a 16I5 format. The first four words of the 30 word array were used by the original implementors of the model. The remaining 26 locations were set aside for future use.

In the revised version of ICARUS 5 of the 26 extra words, words 5 through 9 of the 30 word integer array INT, have been used as follows:

Word	Card Columns	
5	21-25	Set by user of the model to control the computation of the dynamic center of gravity. 0 - entered in this word (via the Default file only) will force the model to bypass the dynamic center of gravity computation. 1 - entered in this word will force the model to compute the dynamic center of gravity.
6 7 8	26-30 31-35 36-40	Set by the user of the model to request additional output reports not normally displayed - model debugging or case analysis aides. Entering, via the Default file only, the following values into these words will force the model to generate the following reports: 1 - Prints all forces and moments on seat/man. 2 - Prints values contained in locations NTRY, NUMEQS, DELTAT, DELTARP, WORK (1-5).

		3 - Prints forces and moments due to catapult. 4 - Prints Martin-Baker catapult displacements and rates. 5 - Prints the Quaternions. 6 - Prints CTLWDS common block.
9	41-45	Set by the user of the model to control the generation of the Tektronix plot file. 0 - entered in this word will inhibit the generation of this file. 1 - entered will force the model to generate the file.

4.2 Program Load, Execute, Output and Load Map Procedures

4.2.1 CALL, PRLD - this procedure loads the ICARUS program and associated data files and executes the program.

KRONOS PROCEDURE PRLD:

```
00110 ASSIGN.MS.OUTPUT.
00111 REWIND.OUTPUT.TAPE11.TAPE12.TAPE13.TAPE14.TAPE15.TAPE16.TAPE17.
00112 REWIND.TAPE18.TAPE19.TAPE20.TAPE21.TAPE22.TAPE23.TAPE24.TAPE25.
00113 REWIND.TAPE26.TAPE27.TAPE28.TAPE29.TAPE30.TAPE31.TAPE32.TAPE33.
00114 REWIND.TAPE34.TAPE35.TAPE36.TAPE37.TAPE38.TAPE39.TAPE40.TAPE41.
00115 REWIND.TAPE42.TAPE43.TAPE44.TAPE45.
00120 GET.TAPE1=DFDATA.
00130 GET.CARDIN.
00140 GET.TAPE2=AERO.
00150 GET.NEW.
00155 GET.COMCLIB/UN=SYSTEM.
00160 GET.MATLIB/UN=SYSTEM.
00163 REL.127777.
00170 LDSET.LIB=MATLIB.
00171 LDSET.LIB=COMCLIB.
00180 LOAD.NEW.
00190 EXECUTE.ICARUS.CARDIN.
00200 EXIT.
```

4.2.2 CALL, TAPOUT - this procedure copies the new report files onto the output file.

KRONOS PROCEDURE TAPOUT:

```
REWIND,TAPE11,TAPE12,TAPE13,TAPE14,TAPE15,TAPE16,TAPE17,TAPE18,TAPE19.
REWIND,TAPE21,TAPE22,TAPE23,TAPE24,TAPE25,TAPE26,TAPE27,TAPE28,TAPE29.
REWIND, TAPE20,TAPE30,TAPE31,TAPE32,TAPE33.
REWIND,TAPE34,TAPE35,TAPE36,TAPE37,TAPE38,TAPE39,TAPE40,TAPE41.
REWIND,TAPE42,TAPE43,TAPE44,TAPE45.
COPYBF,TAPE11.
COPYBF,TAPE12.
COPYBF,TAPE13.
COPYBF,TAPE14.
COPYBF,TAPE15.
COPYBF,TAPE16.
COPYBF,TAPE17.
COPYBF,TAPE33.
COPYBF,TAPE30.
COPYBF,TAPE31.
COPYBF,TAPE32.
COPYBF,TAPE18.
COPYBF,TAPE19.
COPYBF,TAPE20.
COPYBF,TAPE21.
COPYBF,TAPE22.
COPYBF,TAPE23.
COPYBF,TAPE24.
COPYBF,TAPE25.
COPYBF,TAPE26.
COPYBF,TAPE27.
COPYBF,TAPE28.
COPYBF,TAPE29.
COPYBF,TAPE34.
COPYBF,TAPE35.
COPYBF,TAPE36.
COPYBF,TAPE37.
COPYBF,TAPE38.
COPYBF,TAPE39.
COPYBF,TAPE40.
COPYBF,TAPE41.
COPYBF,TAPE42.
COPYBF,TAPE43.
COPYBF,TAPE44.
COPYBF,TAPE45.
```

4.2.3 CALL, MAP - this procedure loads the ICARUS program and associated data files and generates a full load and subroutine cross reference map of the program.

KRONOS PROCEDURE MAP:

```
GET,TAPE1=DFDATA.
GET,COMCLIB/UN=SYSTEM.
GET,CARDIN.
GET,TAPE2=AERO.
GET,NEW.
GET,MATLIB/UN=SYSTEM.
RFL,127777.
ASSIGN,MS,OUTPUT.
LDSET,LIB=MATLIB,MAP=SBEX.
LDSET,LIB=COMCLIB.
LOAD,NEW,MATLIB.
NOGO.
DISPOSE,OUTPUT=PR/E1=CD8002.
```

5.0 HOW TO INSERT NEW DATA AND NEW REPORTS

5.1 Additions to the Default Decimal Data (Segment One of the File)

Whenever additional decimal input parameters are required by the model, i.e., extensions to the model such as the dynamic center of gravity equations, a new data section can be appended to the last section (currently Section 14) by:

- Changing the first record (one word in length, I5 format) of the Default data file (DFDATA) to reflect the number of additional decimal words to be input. Currently this word contains the integer value of 253. If a new section were to be added con-

taining six additional variables, the word count would be increased to 259. The values of the six new variables to be read-in would be inserted immediately after the 253rd word (words 254, 255,... 259) in a 3E20.14 format.

- Subroutines INPUT, INIT, PTDFLT, UPDEC, DEFAULT and the appropriate labelled common blocks, dimension and integer array must be updated to reflect the additional variables.

The 253 (current) decimal data values are displayed on the user's output file in the 14 data sections as follows:

Section	Begin Word	End Word	Length	
1	1	12	12	Angular data
2	13	85	73	Positional data
3	86	97	12	A/C data
4	98	103	6	(Seat back rocket angles)
5	104	108	5	Slider block
6	109	111	3	Dart
7	112	145	34	S/M inertia, weights, area
8	146	163	18	Chute data
9	164	174	11	Chute data
10	175	183	9	Event time date
11	184	207	24	Event time steps
12	208	219	12	Dynamic C.G.
13	220	246	27	M.B. catapult
14	247	253	7	M.B. phasing

5.2 Additions to the Default Integer Data

(Segment 2 of the File)

The first record after the decimal data, 1 word in length (I5-format) indicates the number of integer words that immediately follow. Currently this count is set at 30. The next two records contain the integer data values in a 16

I5 format. The first 4 words of the 30 word integer array have been formally defined by the original implementors of the model and have been equivalenced to the following integer data names: ISMA, ISTOP, ISMS and ISEAT respectively. see pp. - 63, 71, 72 of the Program Maintenance Manual (Reference 2) for a detailed definition of their use in the model.

The first 4 integer data inputs may be updated via the Card Input file. The remaining 26 words have not been formally defined for updating using the Card Input file procedure. The 26 words can currently only be modified by changing their values in the Default file. Words 5 to 9 are currently being used for plotting, program case analysis and debug purposes.

See Section 4.1.3 of this report for further details regarding their use in the model.

5.3 Implementation of New Reports

The first card immediately after the case title card is the user INPUT/OUTPUT Display option card. This card contains 35 switches (one for each report) of which 23 are currently in use. These switches are set by the user to control the printing or non-printing of each of the 35 reports. If a new report is to be added, switches 24 to 35 are available to be assigned to the report. Report page and column headings can be entered by inserting the appropriate page and column headings in the data statement array allocated in subroutine NREP. Subroutines REP1 and FETCH must be updated to call the new reports with formal parameters pointing to the data locations to be printed and the format to be used in displaying the locations.

6.0 ICARUS FLOW DIAGRAMS

6.1 Macro Structure of ICARUS

Control Section

Overall model data initialization.

Case study data initialization.

A/C and seat/man requests for trajectory computations and output.

Adjustments to the trajectory time steps.

Termination of case studies.



Event Control

Monitors the status of position and time events.

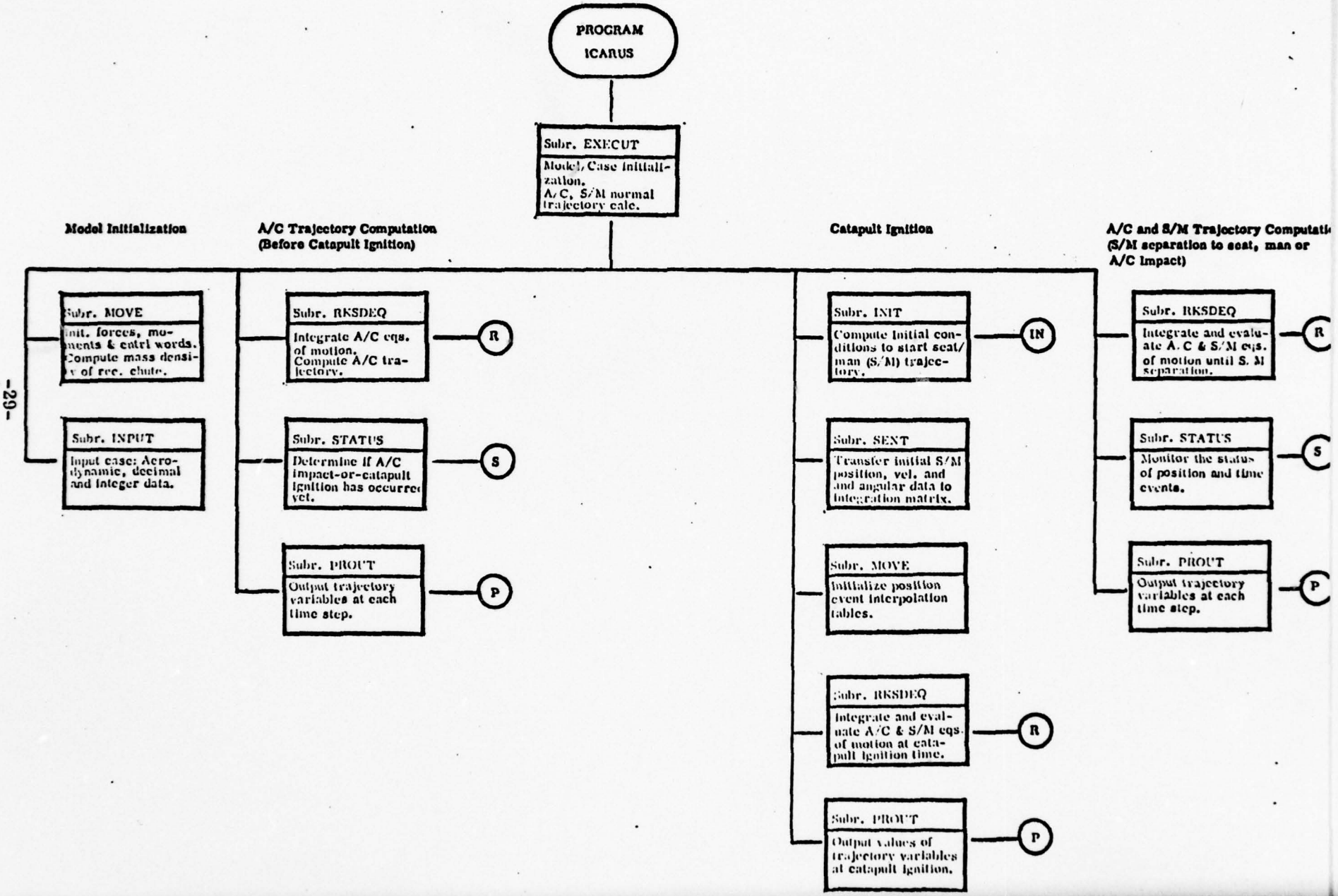
Initiates output request for message indicating a particular event has occured.

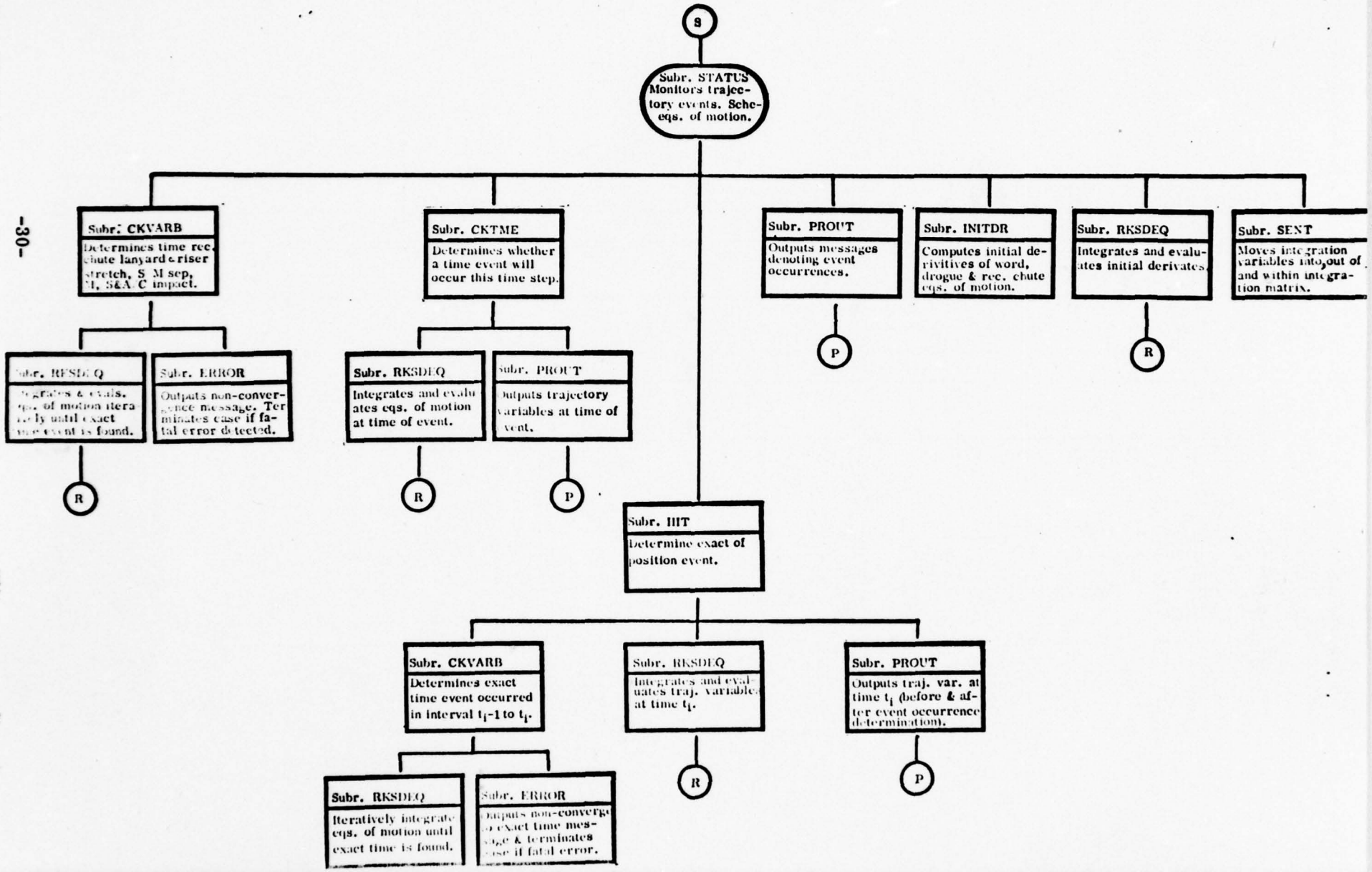
Initiates requests for the determination of the exact time that an event has occurred and to output this time and the values of the trajectory variables at that time.

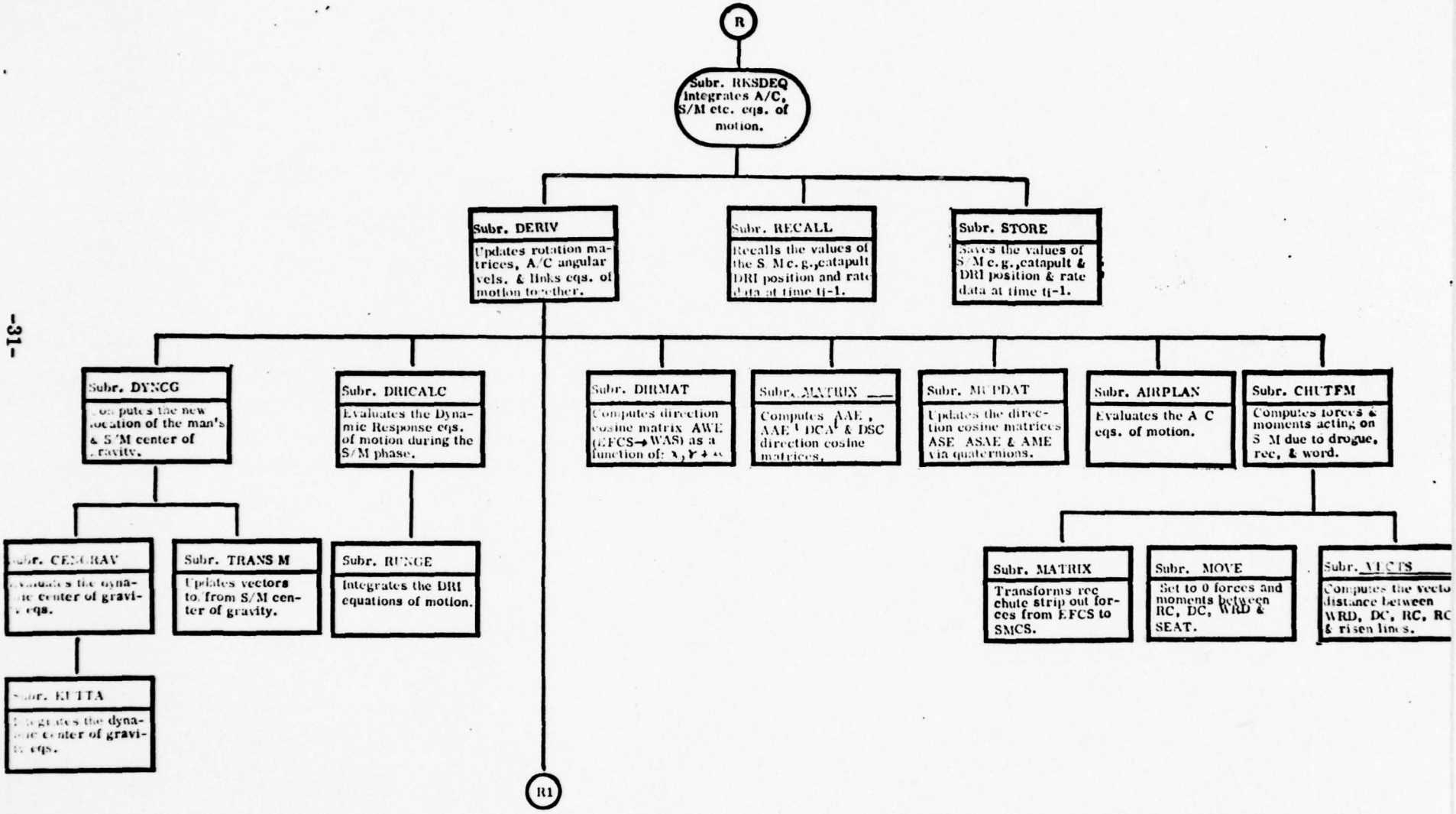
Dynamic scheduling and initialization of the equations of motion to be evaluated at each phase of the trajectory.

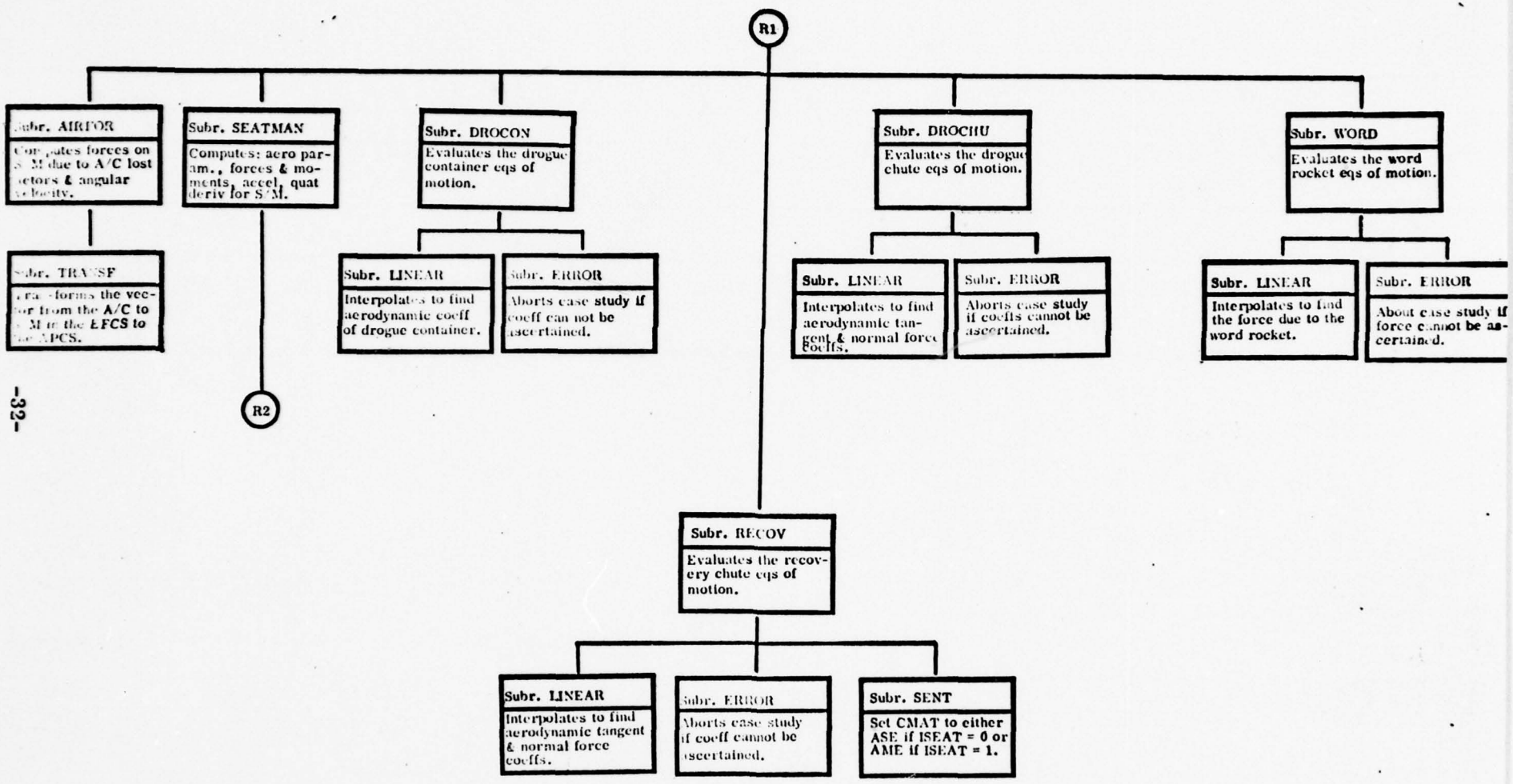
Controls overall time step (Δt) to be used during each phase.

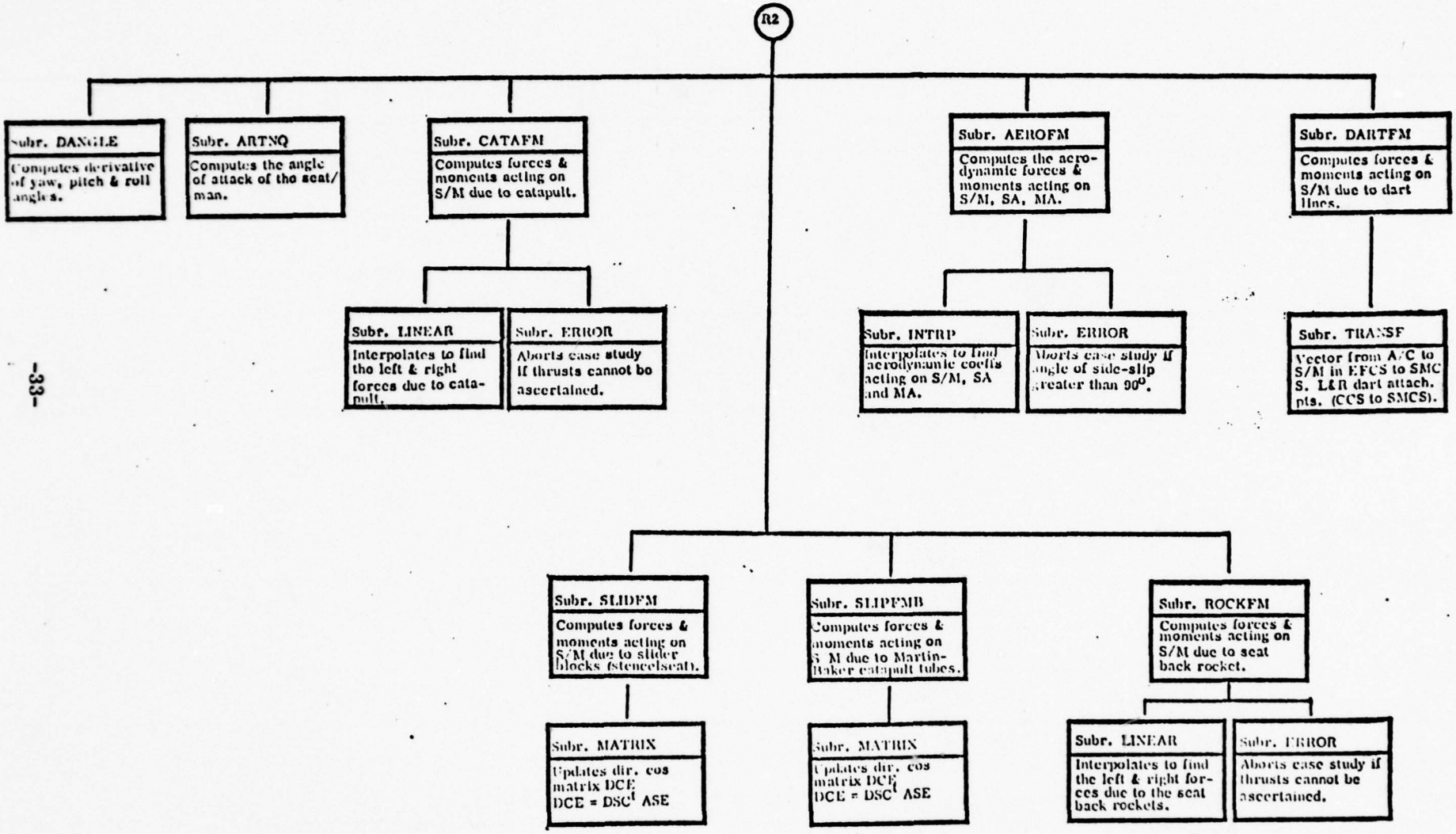
6.2 Macro Subroutine Flow Diagram

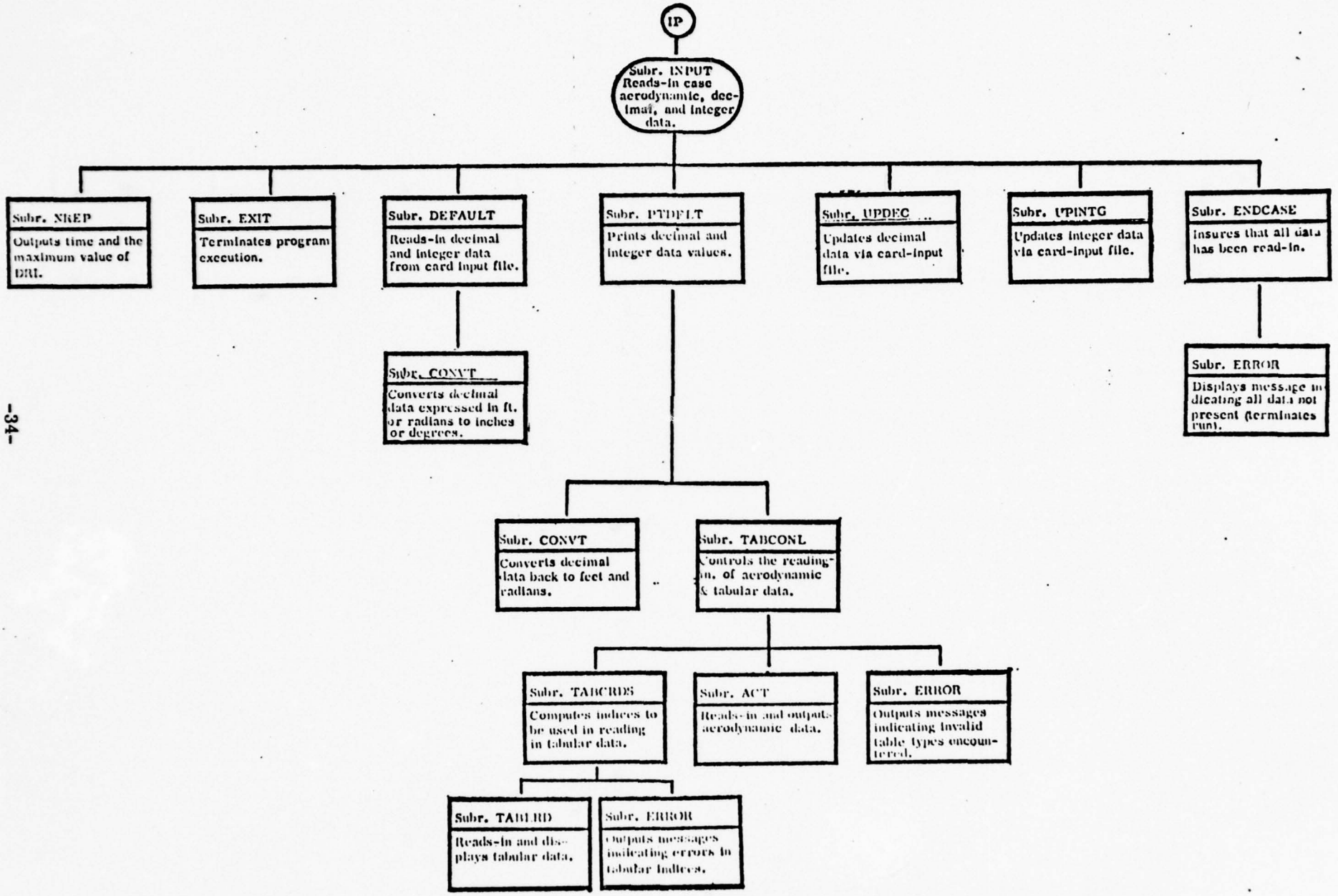

PROGRAM ICARUS
Subr. EXECUT
Model/Case initialization.
A/C, S/M normal trajectory calc.
Model Initialization
A/C Trajectory Computation (Before Catapult Ignition)
Catapult Ignition
A/C and S/M Trajectory Computation (S/M separation to seat, man or A/C Impact)
Subr. MOVE
Init. forces, moments & cntrl words. Compute mass density of rec. chute.
Subr. INPUT
Input case: Aerodynamic, decimal and integer data.
Subr. RKSDEQ
Integrate A/C eqs. of motion. Compute A/C trajectory.
R
Subr. STATUS
Determine if A/C impact-or-catapult ignition has occurred yet.
S
Subr. PROUT
Output trajectory variables at each time step.
P
Subr. INIT
Compute initial conditions to start seat/man (S/M) trajectory.
IN
Subr. SENT
Transfer initial S/M position, vel. and and angular data to integration matrix.
Subr. MOVE
Initialize position event interpolation tables.
Subr. RKSDEQ
Integrate and evaluate A/C & S/M eqs. of motion at catapult ignition time.
R
Subr. PROUT
Output values of trajectory variables at catapult ignition.
P
Subr. RKSDEQ
Integrate and evaluate A/C & S/M eqs. of motion until S/M separation.
R
Subr. STATUS
Monitor the status of position and time events.
S
Subr. PROUT
Output trajectory variables at each time step.
P

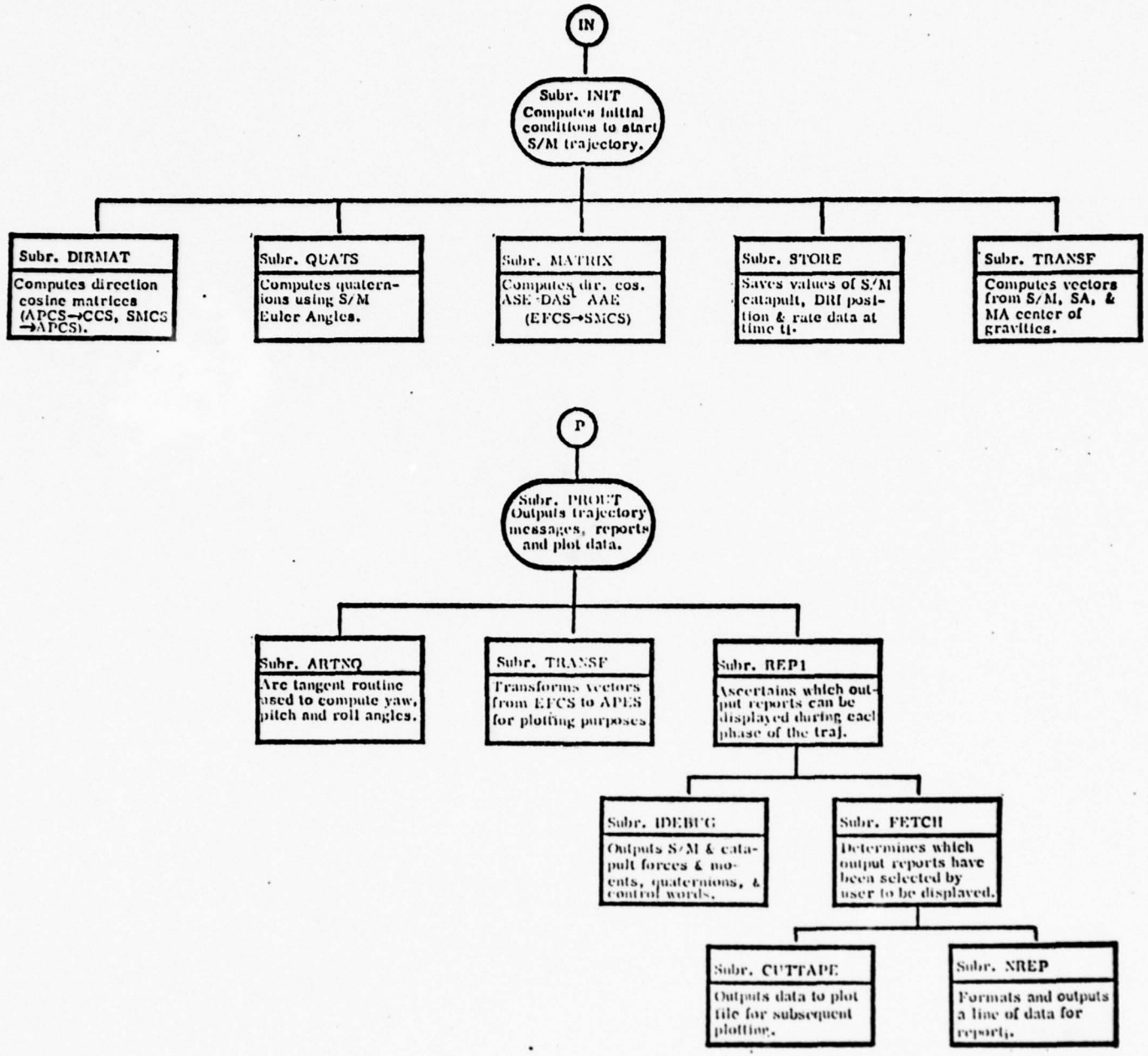

S
Subr. STATUS Monitors trajectory events. Schc- eqs. of motion.
Subr. CKVARB
Determines time rec. chute lanyard & riser stretch, S M sep, M, S&A C impact.
Subr. CKTME
Determines whether a time event will occur this time step.
Subr. PROUT
Outputs messages denoting event occurrences.
Subr. INITDR
Computes initial derivitives of word, drogue & rec. chute eqs. of motion.
Subr. RKSDEQ
Integrates and evaluates initial derivates.
Subr. SENT
Moves integration variables into, out of and within integration matrix.
Subr. RKSDEQ
Subr. ERROR
Outputs non-convergence message. Terminates case if fatal error detected.
Subr. RKSDEQ
Integrates and evaluates eqs. of motion at time of event.
Subr. PROUT
Outputs trajectory variables at time of event.
R
R
P
P
R
Subr. IIIT
Determine exact of position event.
Subr. CKVARB
Determines exact time event occurred in interval t_{i-1} to t_i.
Subr. RKSDEQ
Integrates and evaluates traj. variables at time t_i.
Subr. PROUT
Outputs traj. var. at time t_i (before & after event occurrence determination).
Subr. RKSDEQ
Iteratively integrate eqs. of motion until exact time is found.
Subr. ERROR
Outputs non-converge exact time message & terminates case if fatal error.
R
P


R
Subr. RKSDEQ
Integrates A/C, S/M etc. eqs. of motion.
Subr. DERIV
Updates rotation matrices, A/C angular vels. & links eqs. of motion together.
Subr. RECALL
Recalls the values of the S/M c.g., catapult DRI position and rate data at time tj-1.
Subr. STORE
Saves the values of S/M c.g., catapult & DRI position & rate data at time tj-1.
Subr. DYNCG
Computes the new location of the man's & S/M center of gravity.
Subr. DRICALC
Evaluates the Dynamic Response eqs. of motion during the S/M phase.
Subr. DIRMAT
Computes direction cosine matrix AWE (EFCS→WAS) as a function of:
Subr. MATRIX
Computes AAE, AAE^T DCA^T & DSC direction cosine matrices.
Subr. MUPDAT
Updates the direction cosine matrices ASE ASAE & AME via quaternions.
Subr. AIRPLAN
Evaluates the A/C eqs. of motion.
Subr. CHUTFM
Computes forces & moments acting on S/M due to drogue, rec, & word.
Subr. CENGRAV
Evaluates the dynamic center of gravity eqs.
Subr. TRANS M
Updates vectors to/from S/M center of gravity.
Subr. RUNGE
Integrates the DRI equations of motion.
Subr. MATRIX
Transforms rec chute strip out forces from EFCS to SMCS.
Subr. MOVE
Set to 0 forces and moments between RC, DC, WRD & SEAT.
Subr. VECTS
Computes the vecto distance between WRD, DC, RC, RC & risen lines.
Subr. KUTTA
Integrates the dynamic center of gravity eqs.
R1


R1
Subr. AIRFOR
Computes forces on S/M due to A/C lost sectors & angular velocity.
Subr. TRANSF
Transforms the vector from the A/C to S/M in the EFCS to the APCS.
Subr. SEATMAN
Computes: aero param., forces & moments, accel, quat deriv for S/M.
R2
Subr. DROCON
Evaluates the drogue container eqs of motion.
Subr. LINEAR
Interpolates to find aerodynamic coeff of drogue container.
Subr. ERROR
Aborts case study if coeff can not be ascertained.
Subr. DROCHU
Evaluates the drogue chute eqs of motion.
Subr. LINEAR
Interpolates to find aerodynamic tangent & normal force coeffs.
Subr. ERROR
Aborts case study if coeffs cannot be ascertained.
Subr. WORD
Evaluates the word rocket eqs of motion.
Subr. LINEAR
Interpolates to find the force due to the word rocket.
Subr. ERROR
About case study if force cannot be ascertained.
Subr. RECOV
Evaluates the recovery chute eqs of motion.
Subr. LINEAR
Interpolates to find aerodynamic tangent & normal force coeffs.
Subr. ERROR
Aborts case study if coeff cannot be ascertained.
Subr. SENT
Set CMAT to either ASE if ISEAT = 0 or AME if ISEAT = 1.

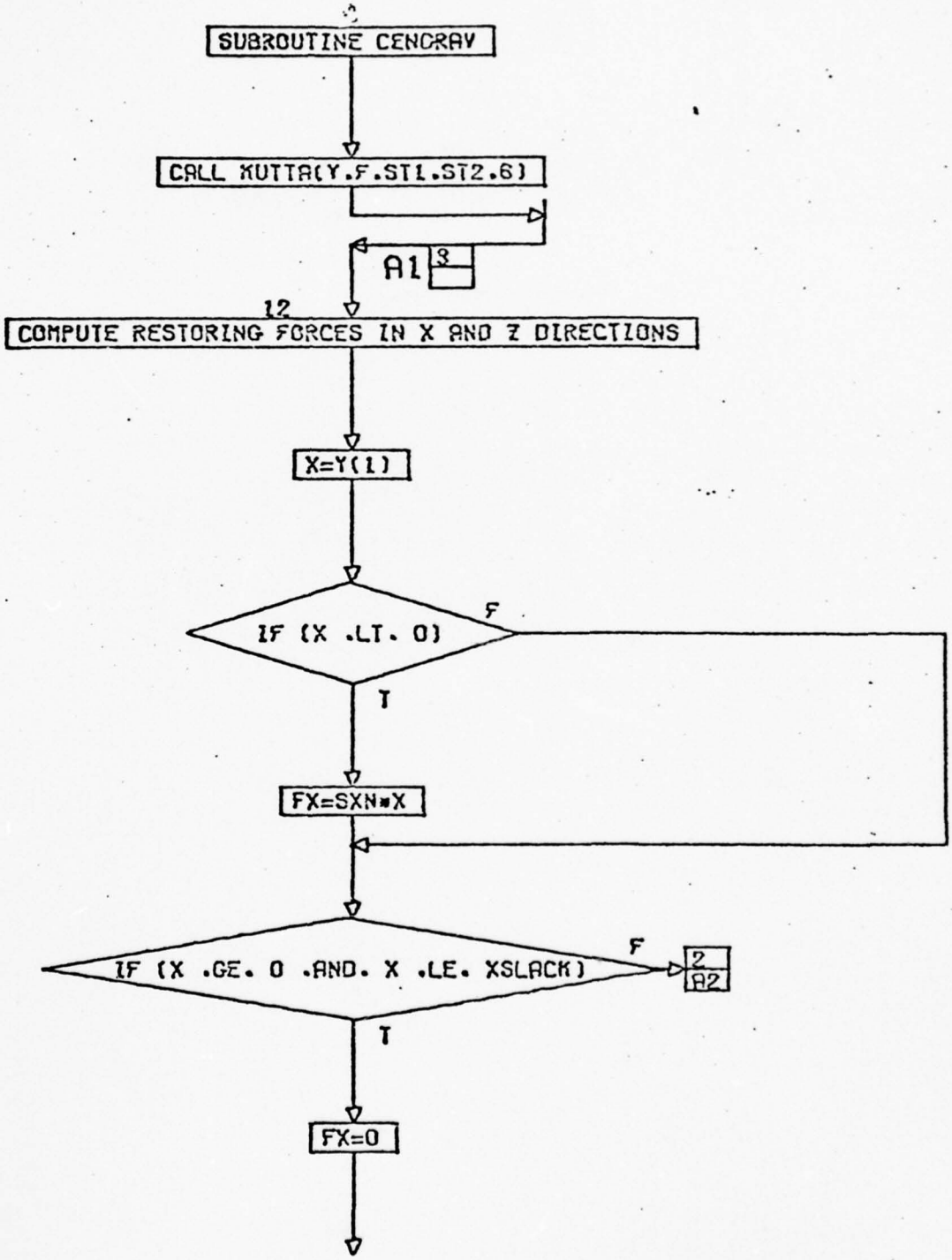

R2
Subr. DANGLE
Computes derivative of yaw, pitch & roll angles.
Subr. ARTNQ
Computes the angle of attack of the seat/man.
Subr. CATAFM
Computes forces & moments acting on S/M due to catapult.
Subr. LINEAR
Interpolates to find the left & right forces due to catapult.
Subr. ERROR
Aborts case study if thrusts cannot be ascertained.
Subr. AEROFM
Computes the aerodynamic forces & moments acting on S/M, SA, MA.
Subr. INTRP
Interpolates to find aerodynamic coefts acting on S/M, SA and MA.
Subr. ERROR
Aborts case study if angle of side-slip greater than 90°.
Subr. DARTFM
Computes forces & moments acting on S/M due to dart fins.
Subr. TRANSF
Vector from A/C to S/M in EFCS to SMCS. L&R dart attach. pts. (CCS to SMCS).
Subr. SLIDFM
Computes forces & moments acting on S/M due to slider blocks (stencelseat).
Subr. MATRIX
Updates dir. cos matrix DCE
DCE = DSC^t ASE
Subr. SLIPFMB
Computes forces & moments acting on S/M due to Martin-Baker catapult tubes.
Subr. MATRIX
Updates dir. cos matrix DCE
DCE = DSC^t ASE
Subr. ROCKFM
Computes forces & moments acting on S/M due to seat back rocket.
Subr. LINEAR
Interpolates to find the left & right forces due to the seat back rockets.
Subr. ERROR
Aborts case study if thrusts cannot be ascertained.


IP
Subr. INPUT
Reads-in case aerodynamic, decimal, and integer data.
Subr. NREP
Outputs time and the maximum value of DRI.
Subr. EXIT
Terminates program execution.
Subr. DEFAULT
Reads-in decimal and integer data from card input file.
Subr. PTDFLT
Prints decimal and integer data values.
Subr. UPDEC
Updates decimal data via card-input file.
Subr. UPINTG
Updates integer data via card-input file.
Subr. ENDCASE
Insures that all data has been read-in.
Subr. CONVT
Converts decimal data expressed in ft. or radians to inches or degrees.
Subr. ERROR
Displays message indicating all data not present (terminates run).
Subr. CONVT
Converts decimal data back to feet and radians.
Subr. TABCONL
Controls the reading-in. of aerodynamic & tabular data.
Subr. TABCRDS
Computes indices to be used in reading in tabular data.
Subr. ACT
Reads-in and outputs aerodynamic data.
Subr. ERROR
Outputs messages indicating invalid table types encountered.
Subr. TABLRD
Reads-in and displays tabular data.
Subr. ERROR
Outputs messages indicating errors in tabular indices.


IN
Subr. INIT
Computes initial conditions to start S/M trajectory.
Subr. DIRMAT
Computes direction cosine matrices (APCS→CCS, SMCS →APCS).
Subr. QUATS
Computes quaternions using S/M Euler Angles.
Subr. MATRIX
Computes dir. cos. ASE→DAS→AAE (EFCS→SMCS)
Subr. STORE
Saves values of S/M catapult, DRI position & rate data at time t1.
Subr. TRANSF
Computes vectors from S/M, SA, & MA center of gravities.
P
Subr. PROUT
Outputs trajectory messages, reports and plot data.
Subr. ARTNQ
Arc tangent routine used to compute yaw, pitch and roll angles.
Subr. TRANSF
Transforms vectors from EFCS to APES for plotting purposes
Subr. REP1
Ascertains which output reports can be displayed during each phase of the traj.
Subr. IDEBUG
Outputs S/M & catapult forces & moments, quaternions, & control words.
Subr. FETCH
Determines which output reports have been selected by user to be displayed.
Subr. CUTTAPE
Outputs data to plot file for subsequent plotting.
Subr. NREP
Formats and outputs a line of data for reports.

6.3 Detailed Flow Diagrams of the New Subroutines Inserted into ICARUS

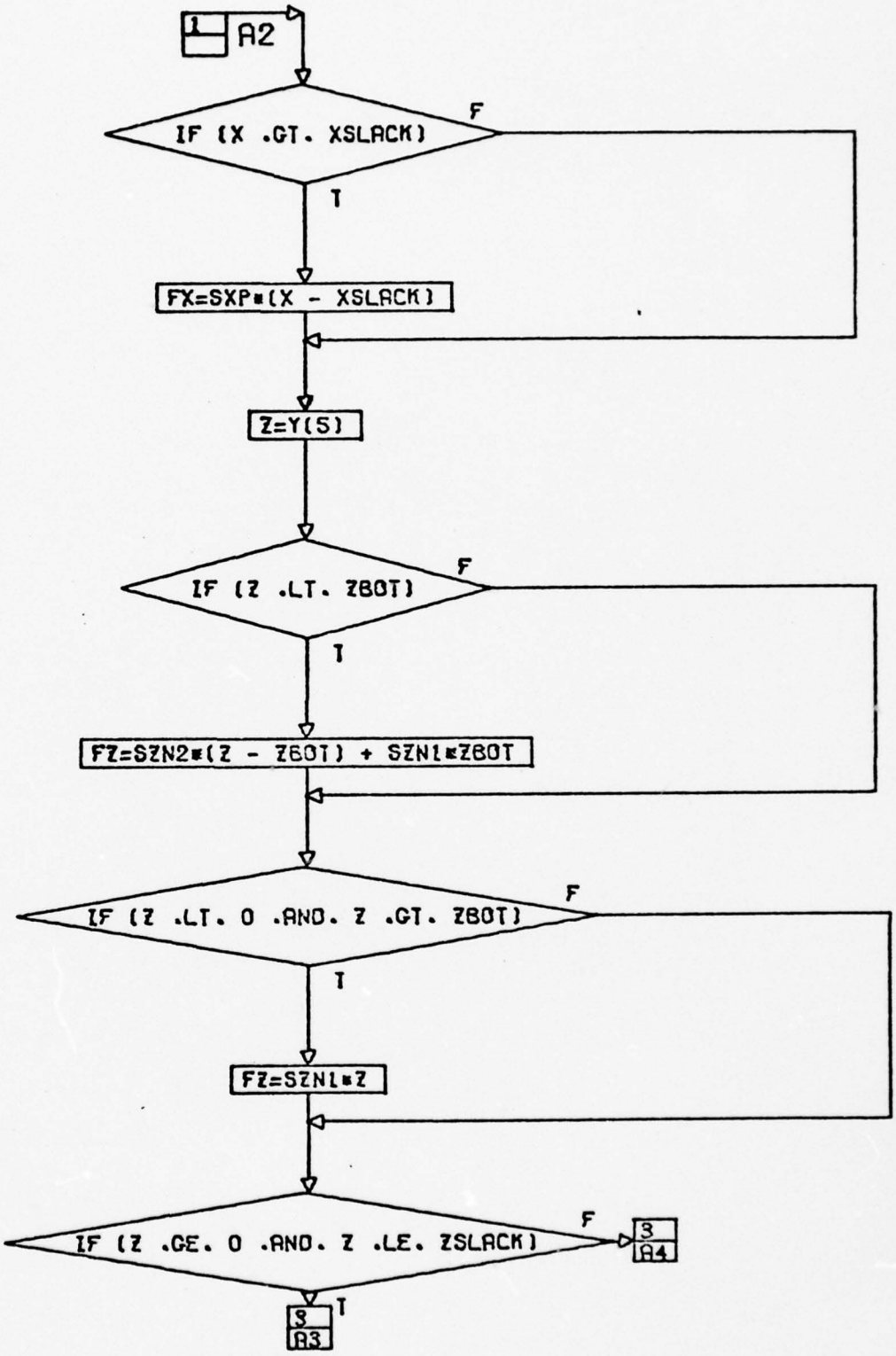
See Figure 1



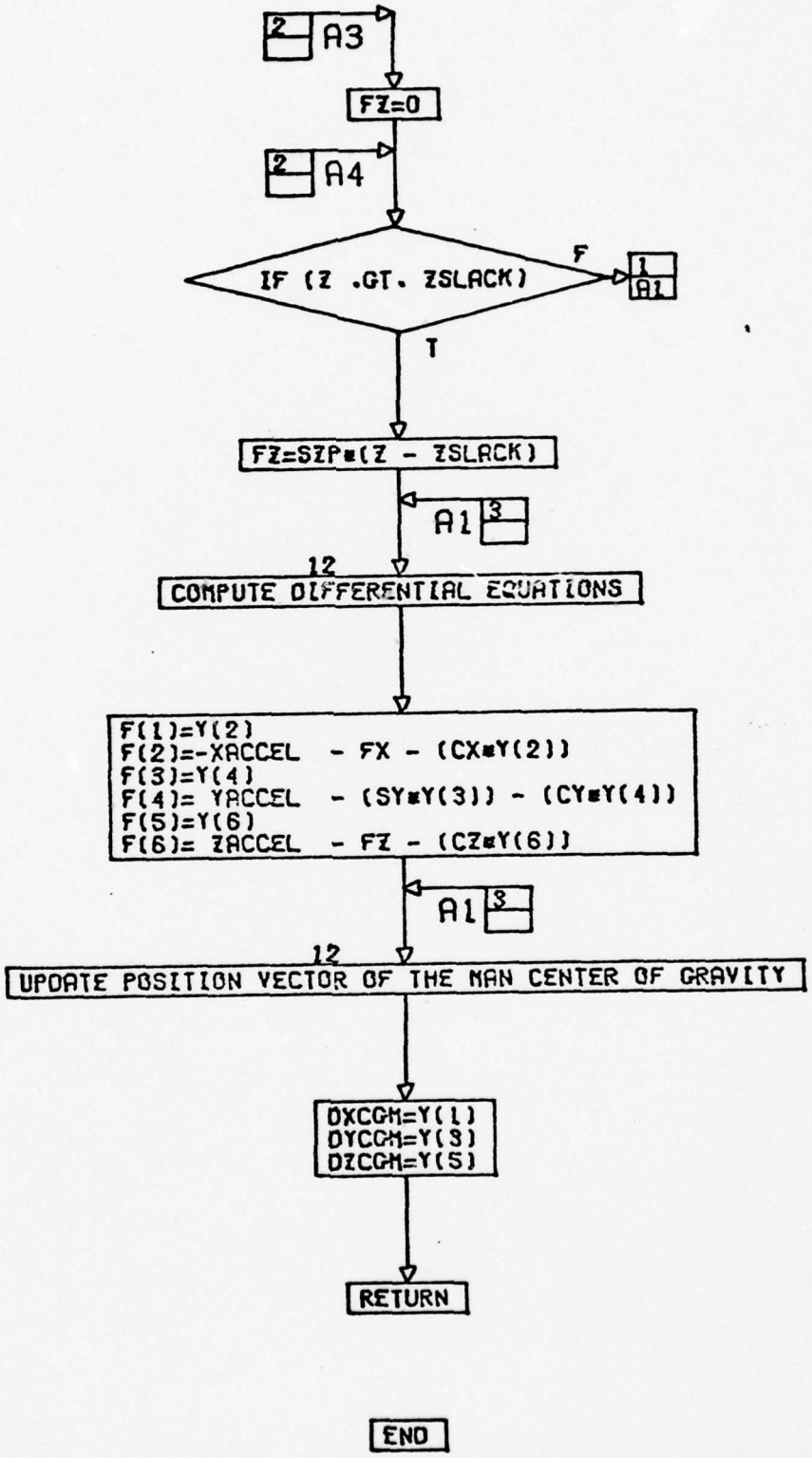
CONT. ON PG 2

FIGURE 1

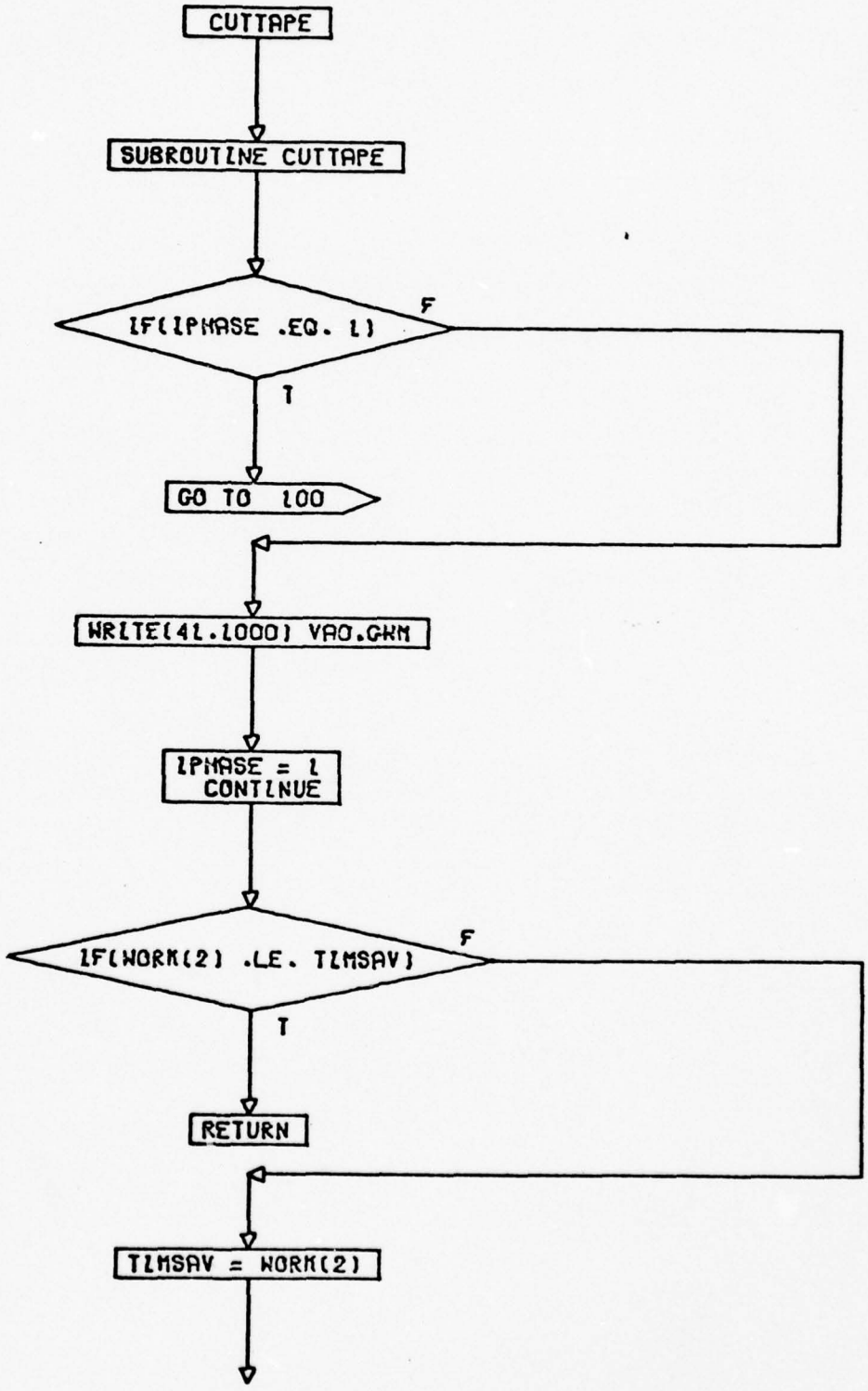
PG 1 OF 3


1
A2
IF (X .GT. XSLACK)
F
T
FX=SXP*(X - XSLACK)
Z=Y(S)
IF (Z .LT. ZBOT)
F
T
FZ=SZN2*(Z - ZBOT) + SZN1*ZBOT
IF (Z .LT. 0 .AND. Z .GT. ZBOT)
F
T
FZ=SZN1*Z
IF (Z .GE. 0 .AND. Z .LE. ZSLACK)
F
3
A4
3
A3
T

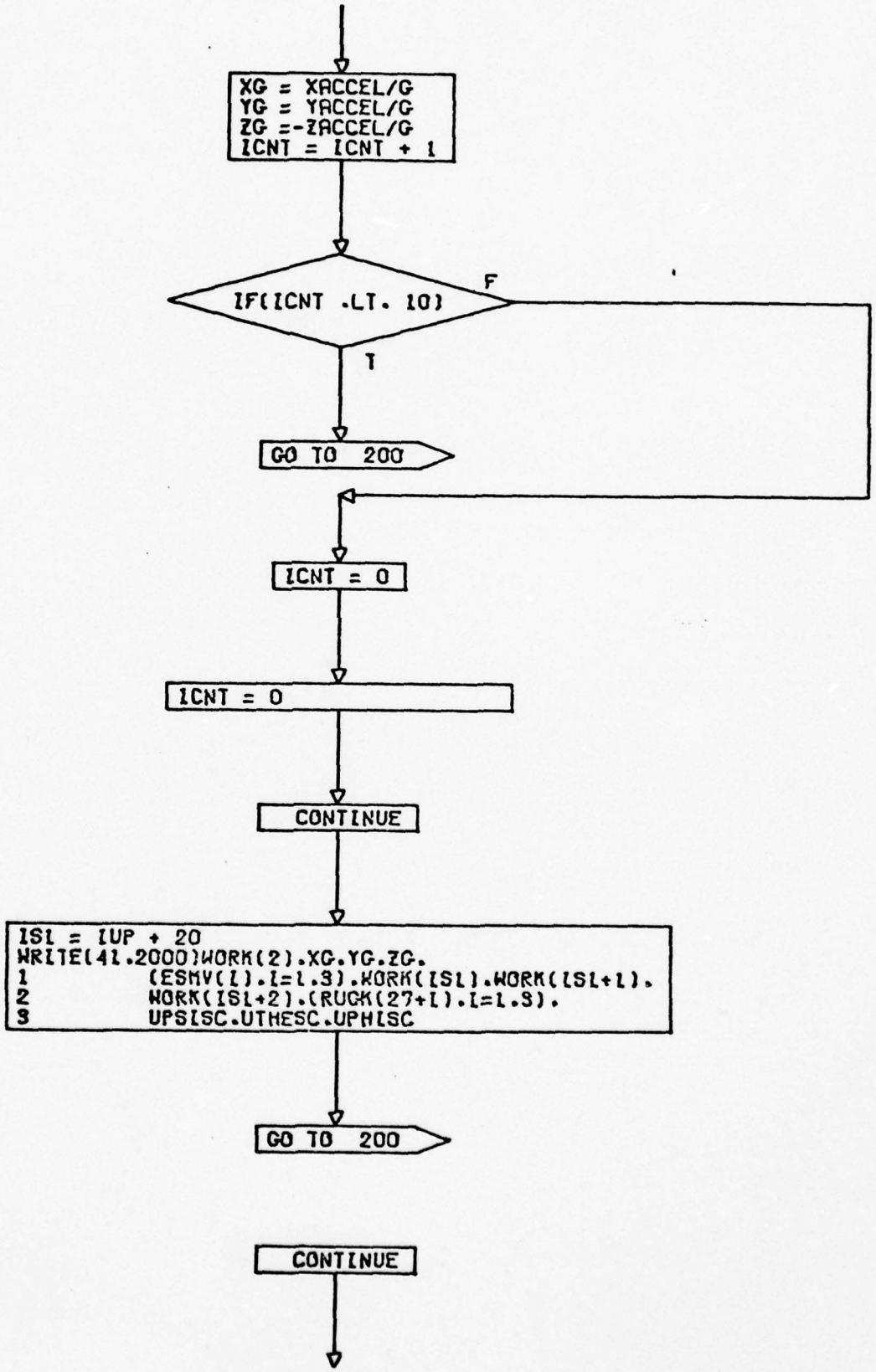
CONT. ON PG 3

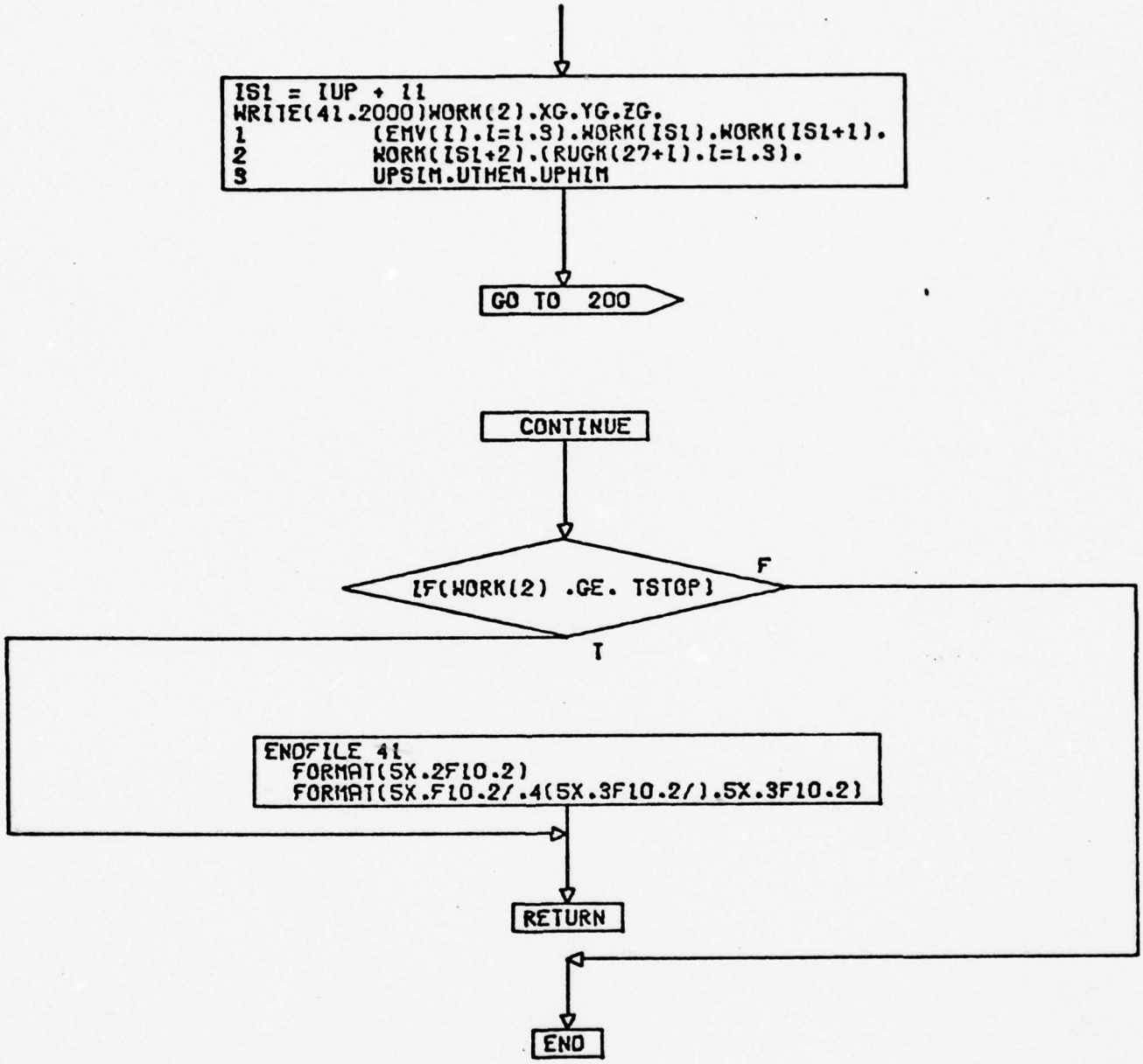

2 A3
FZ=0
2 A4
IF (Z .GT. ZSLACK)
F
1 A1
T
FZ=SZP*(Z - ZSLACK)
A1 3
12
COMPUTE DIFFERENTIAL EQUATIONS
F(1)=Y(2)
F(2)=-XACCEL - FX - (CX*Y(2))
F(3)=Y(4)
F(4)= YACCEL - (SY*Y(3)) - (CY*Y(4))
F(5)=Y(6)
F(6)= ZACCEL - FZ - (CZ*Y(6))
A1 3
12
UPDATE POSITION VECTOR OF THE MAN CENTER OF GRAVITY
DXCGM=Y(1)
DYCGM=Y(3)
DZCGM=Y(5)
RETURN
END

PG 3 FINAL

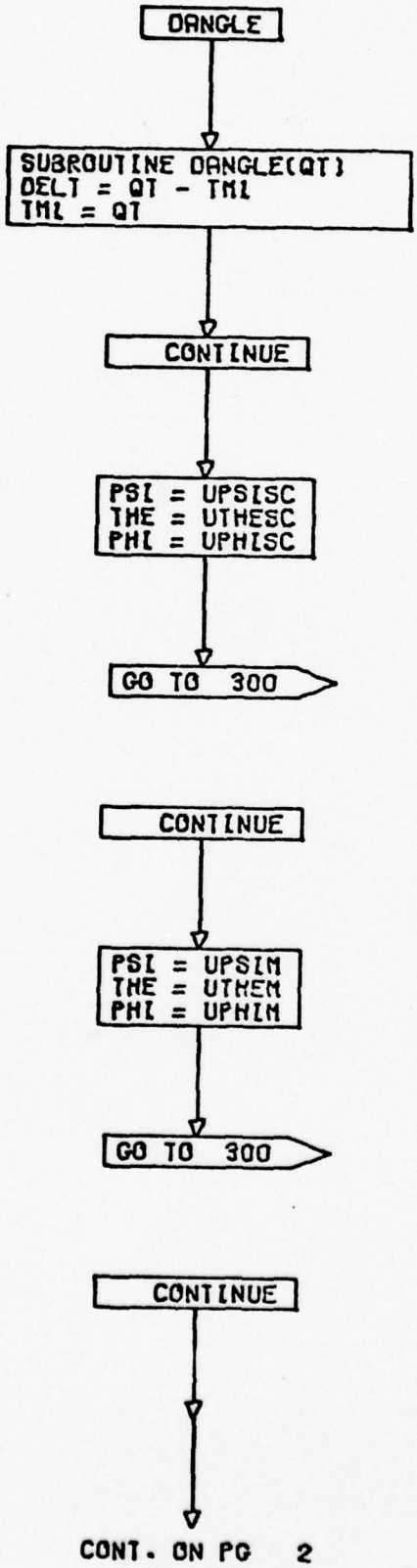

CUTTAPE
SUBROUTINE CUTTAPE
IF(IPHASE .EQ. 1)
F
T
GO TO 100
WRITE(41,1000) VAO,GHM
IPHASE = 1
CONTINUE
IF(WORK(2) .LE. TIMSAV)
F
T
RETURN
TIMSAV = WORK(2)

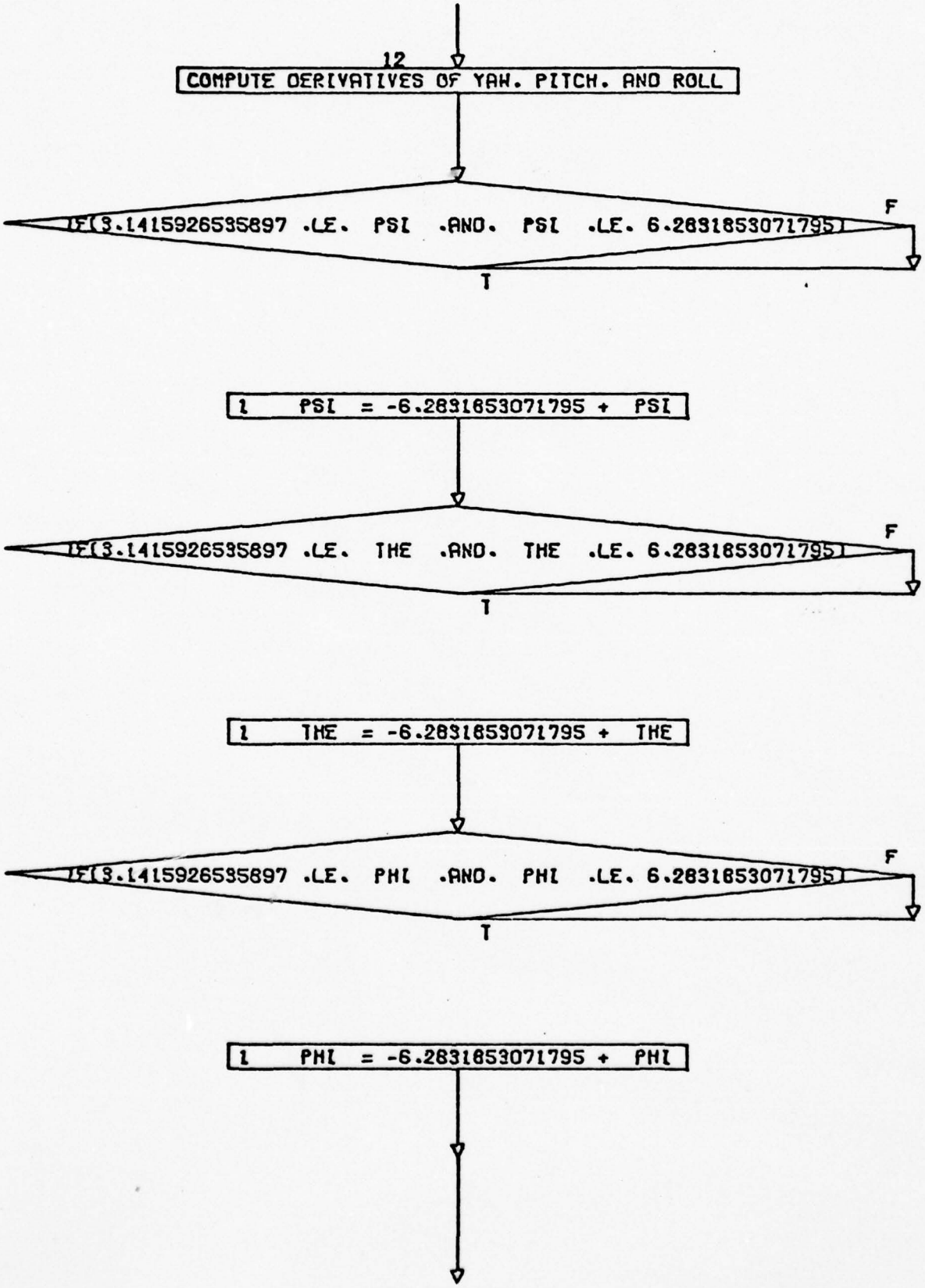
CONT. ON PG 2

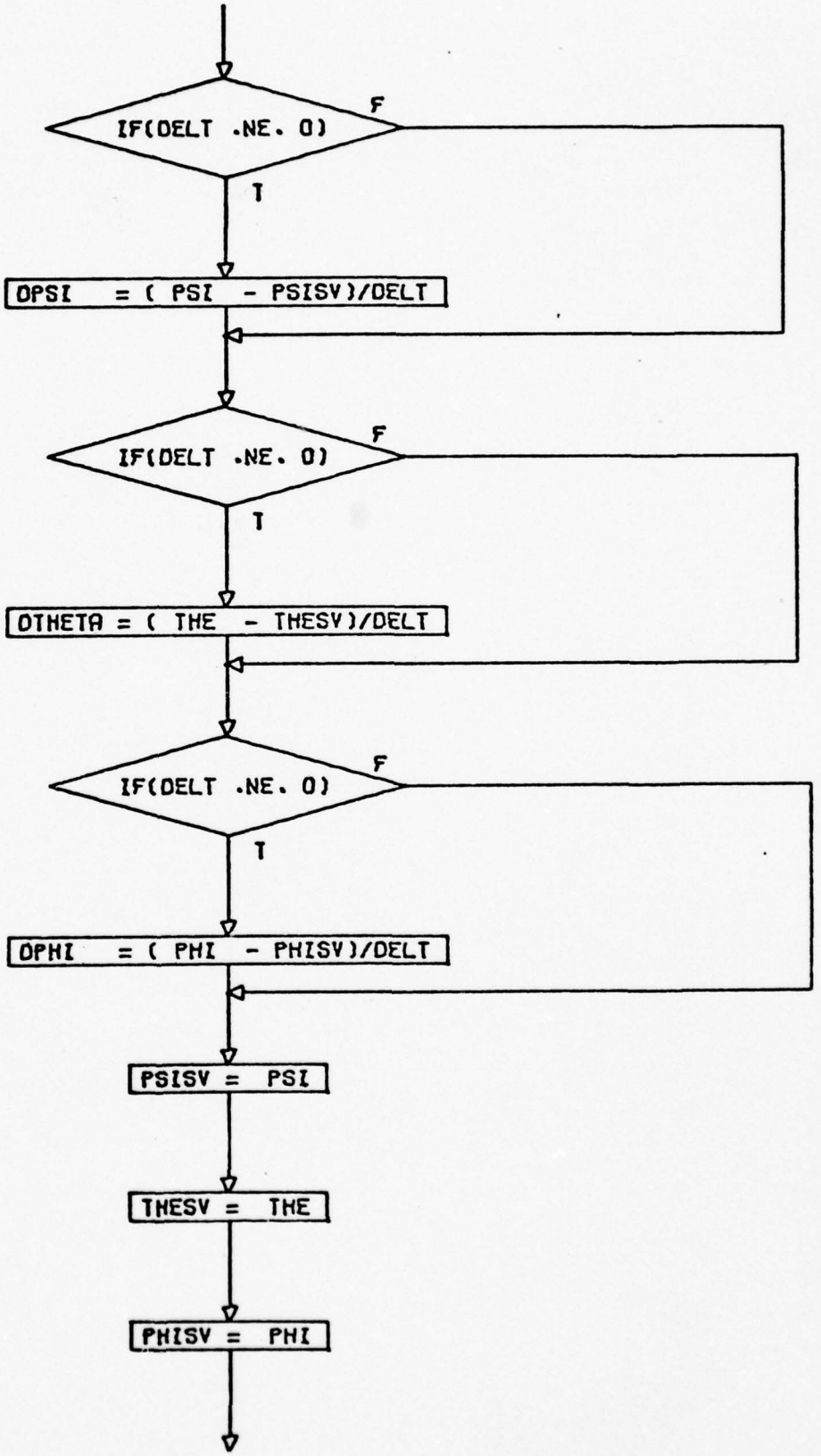

XG = XACCEL/G
YG = YACCEL/G
ZG =-ZACCEL/G
ICNT = ICNT + 1
IF(ICNT .LT. 10)
F
T
GO TO 200
ICNT = 0
ICNT = 0
CONTINUE
ISL = IUP + 20
WRITE(41,2000)WORK(2),XG,YG,ZG,
1 (ESMV(I),I=1,3),WORK(ISL),WORK(ISL+1),
2 WORK(ISL+2),(RUCK(27+I),I=1,3),
3 UPSISC,UTHESC,UPHISC
GO TO 200
CONTINUE
CONT. ON PG 3


IS1 = IUP + 11
WRITE(41,2000)WORK(2),XG,YG,ZG,
1 (EMV(I),I=1,3),WORK(IS1),WORK(IS1+1),
2 WORK(IS1+2),(RUGK(27+I),I=1,3),
3 UPSIM,UTHEM,UPHIM
GO TO 200
CONTINUE
IF(WORK(2) .GE. TSTOP)
F
T
ENDFILE 41
FORMAT(5X,2F10.2)
FORMAT(5X,F10.2/,4(5X,3F10.2/),5X,3F10.2)
RETURN
END

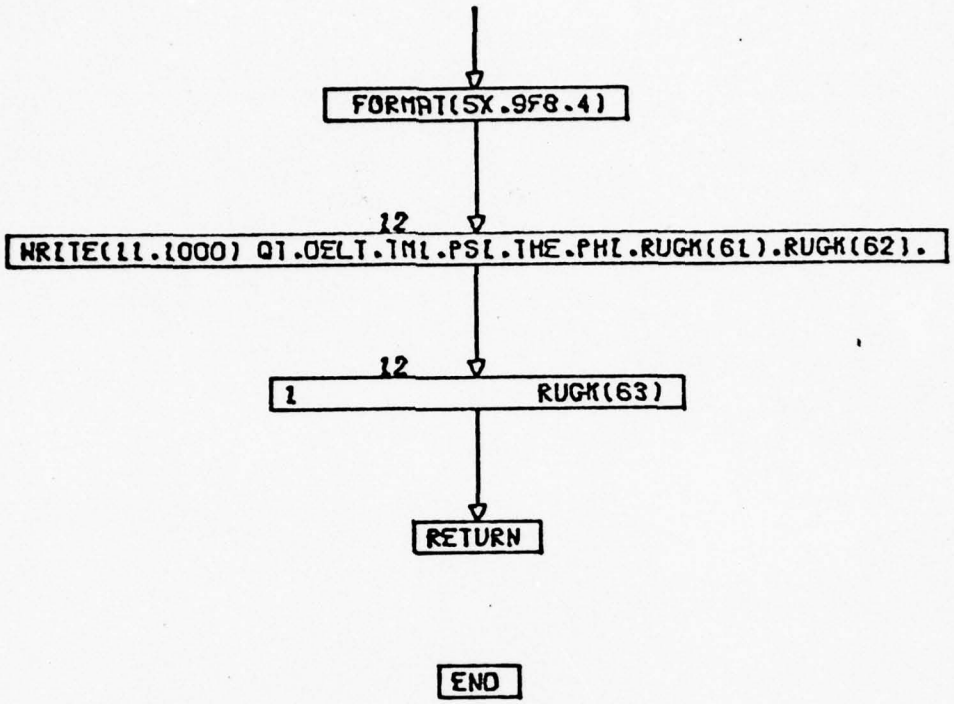
PG 3 FINAL


DANGLE
SUBROUTINE DANGLE(QT)
DELT = QT - TM1
TM1 = QT
CONTINUE
PSI = UPSISC
THE = UTHESC
PHI = UPHISC
GO TO 300
CONTINUE
PSI = UPSIM
THE = UTHEM
PHI = UPHIM
GO TO 300
CONTINUE
CONT. ON PG 2

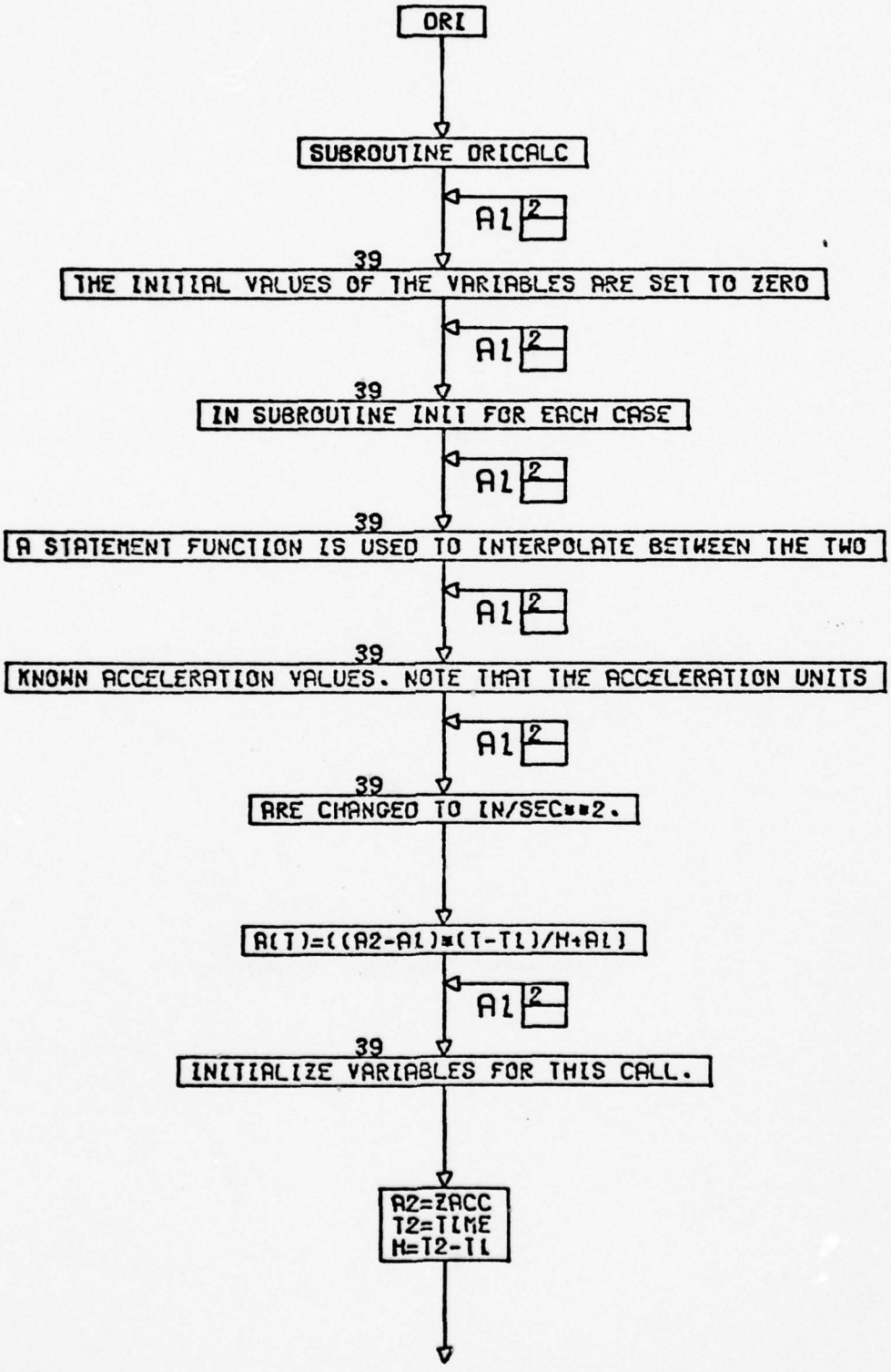

12
COMPUTE DERIVATIVES OF YAW. PITCH. AND ROLL
IF(3.1415926535897 .LE. PSI .AND. PSI .LE. 6.2831853071795)
F
T
1 PSI = -6.2831853071795 + PSI
IF(3.1415926535897 .LE. THE .AND. THE .LE. 6.2831853071795)
F
T
1 THE = -6.2831853071795 + THE
IF(3.1415926535897 .LE. PHI .AND. PHI .LE. 6.2831853071795)
F
T
1 PHI = -6.2831853071795 + PHI
CONT. ON PG 3


IF(DELT .NE. 0)
F
T
DPSI = (PSI - PSISV)/DELT
IF(DELT .NE. 0)
F
T
DTHETA = (THE - THESV)/DELT
IF(DELT .NE. 0)
F
T
DPHI = (PHI - PHISV)/DELT
PSISV = PSI
THESV = THE
PHISV = PHI

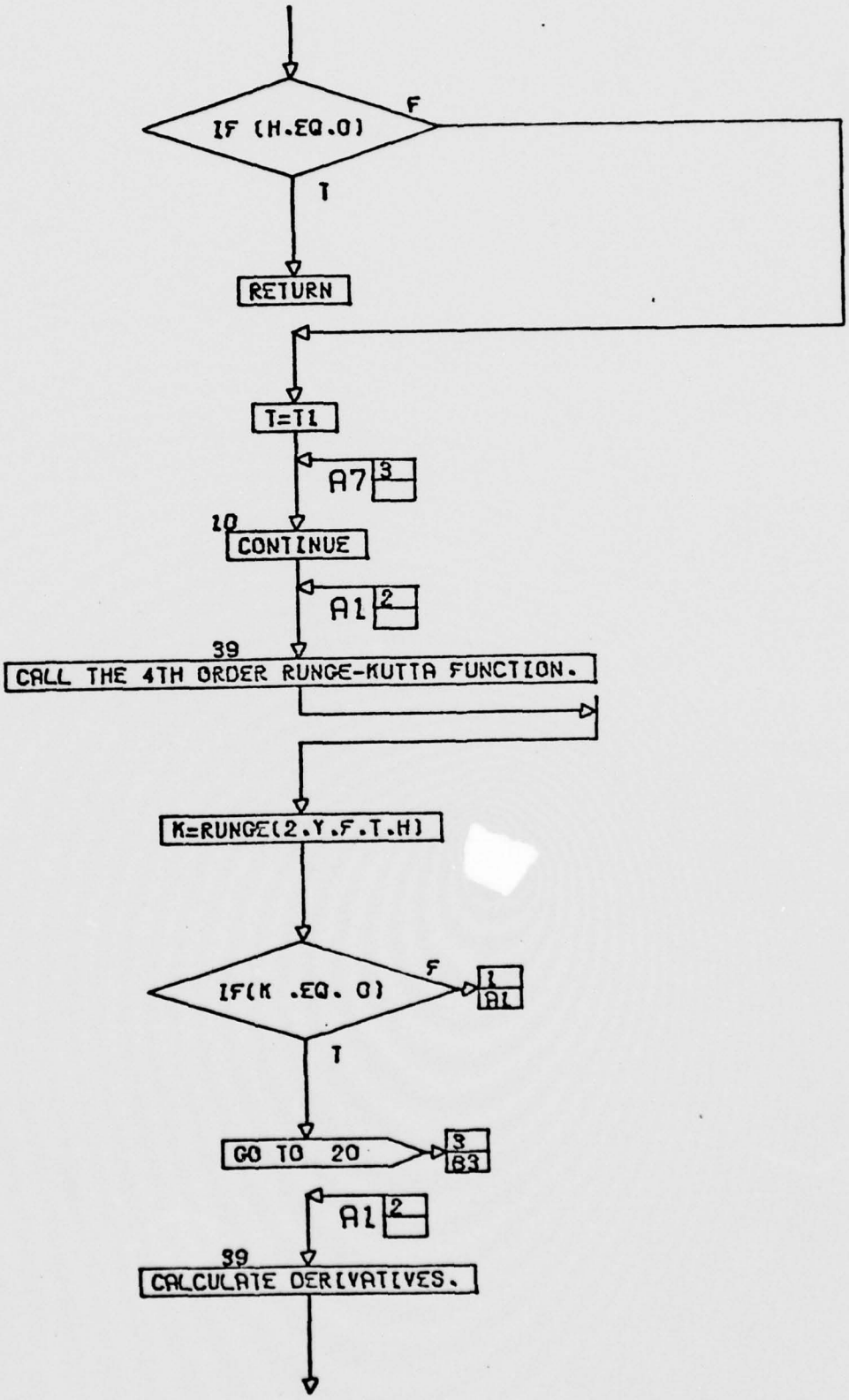
CONT. ON PG 4


FORMAT(5X,9F8.4)
12
WRITE(11,1000) QT,DELT,TMI,PSI,THE,PHI,RUGK(61),RUGK(62),
12
1 RUGK(63)
RETURN

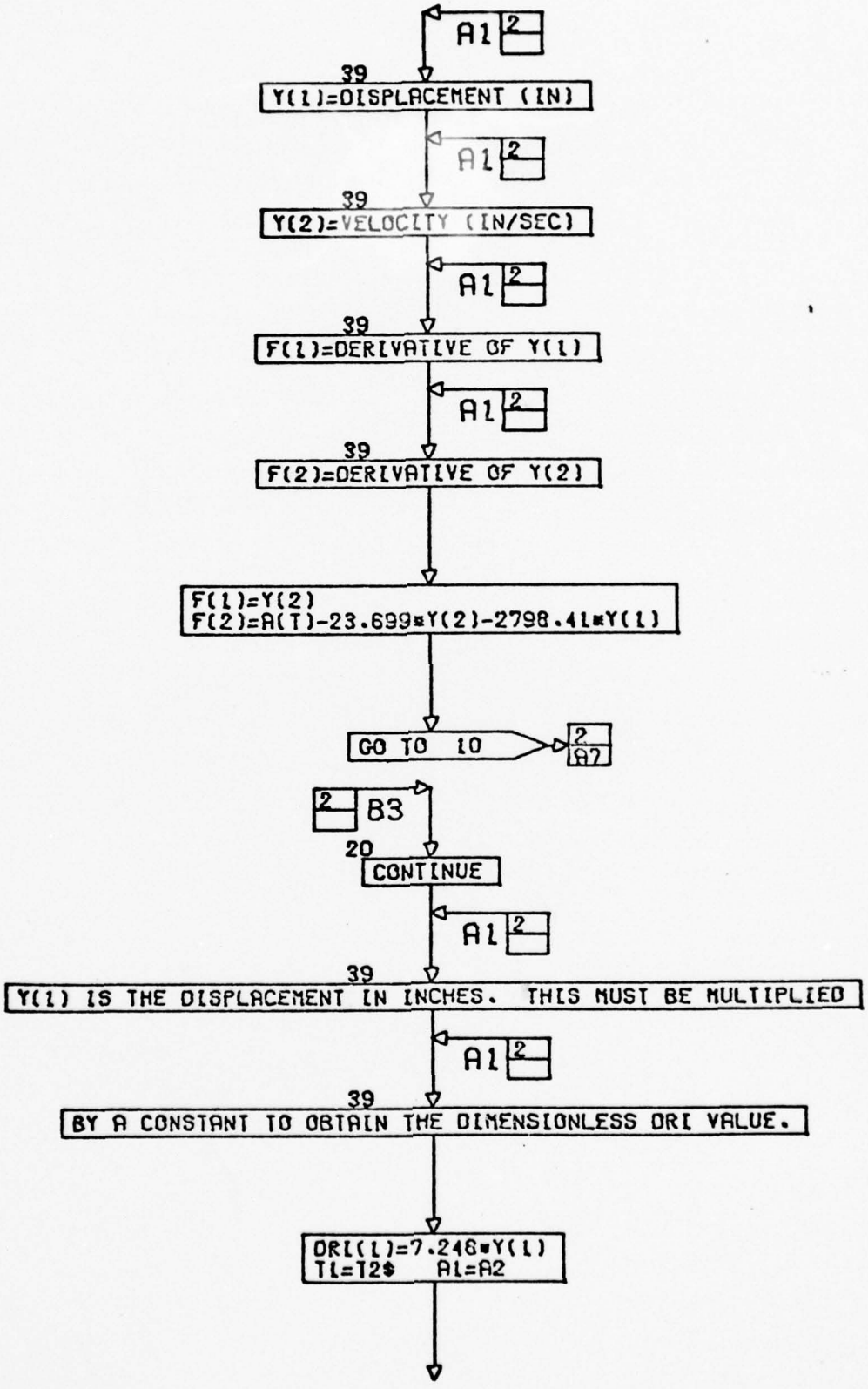
END


ORI
SUBROUTINE ORICALC
A1 2
39
THE INITIAL VALUES OF THE VARIABLES ARE SET TO ZERO
A1 2
39
IN SUBROUTINE INIT FOR EACH CASE
A1 2
39
A STATEMENT FUNCTION IS USED TO INTERPOLATE BETWEEN THE TWO
A1 2
39
KNOWN ACCELERATION VALUES. NOTE THAT THE ACCELERATION UNITS
A1 2
39
ARE CHANGED TO IN/SEC**2.
A(T)=((A2-A1)*(T-T1)/H+A1)
A1 2
39
INITIALIZE VARIABLES FOR THIS CALL.
A2=ZACC
T2=TIME
H=T2-T1

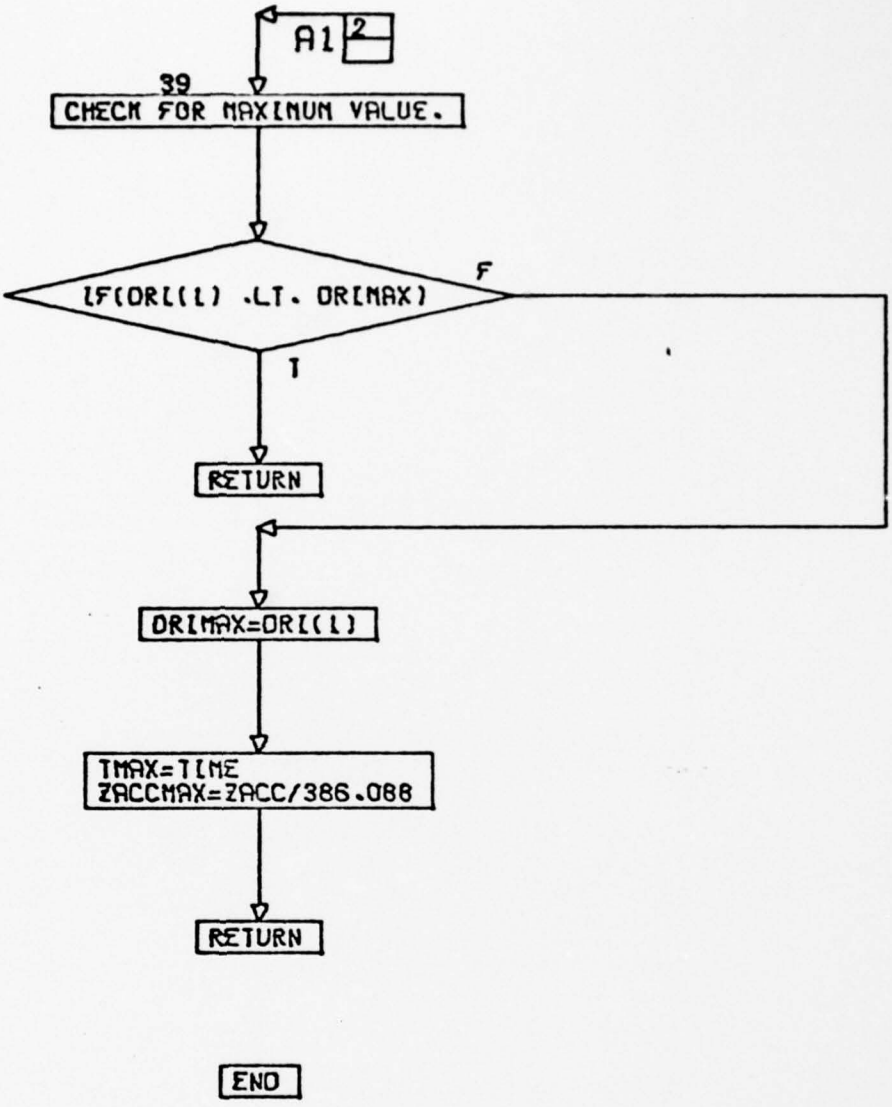
CONT. ON PG 2


IF (H.EQ.0)
F
T
RETURN
T=T1
A7 3
10 CONTINUE
A1 2
39 CALL THE 4TH ORDER RUNGE-KUTTA FUNCTION.
K=RUNGE(2.Y.F.T.H)
IF(K .EQ. 0)
F
1 A1
T
GO TO 20
3 B3
A1 2
39 CALCULATE DERIVATIVES.

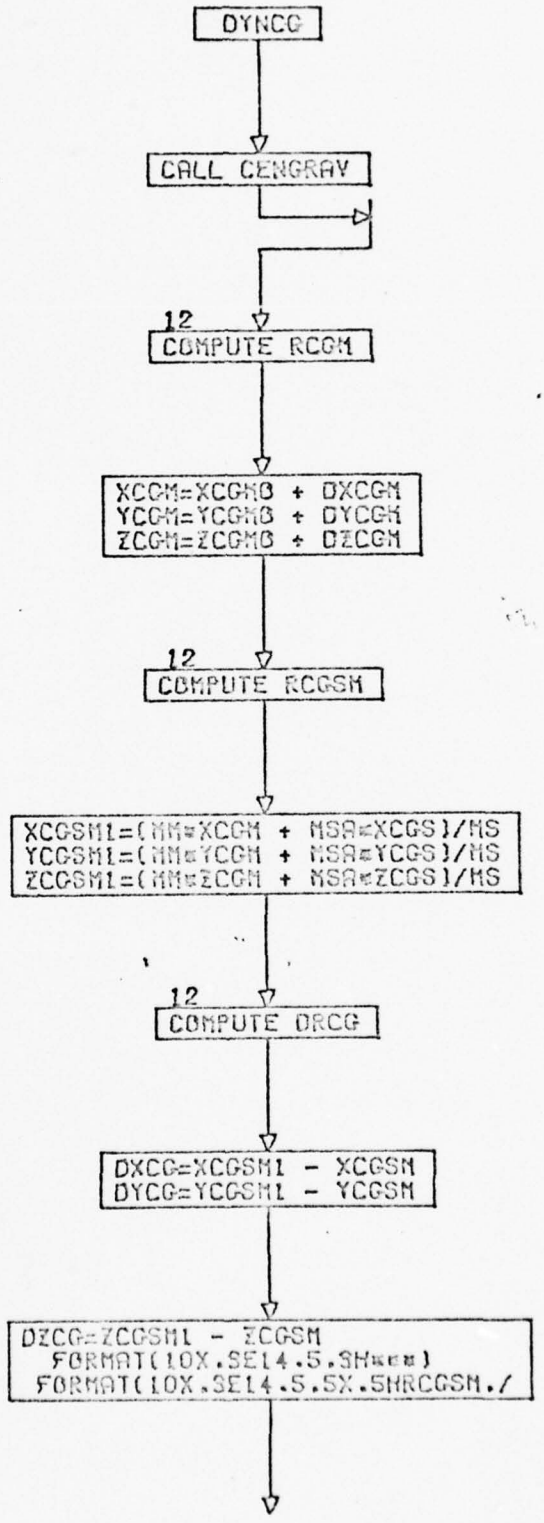
CONT. ON PG 3


A1 2
39
Y(1)=DISPLACEMENT (IN)
A1 2
39
Y(2)=VELOCITY (IN/SEC)
A1 2
39
F(1)=DERIVATIVE OF Y(1)
A1 2
39
F(2)=DERIVATIVE OF Y(2)
F(1)=Y(2)
F(2)=A(T)-23.699*Y(2)-2798.41*Y(1)
GO TO 10
2
A7
2 B3
20
CONTINUE
A1 2
39
Y(1) IS THE DISPLACEMENT IN INCHES. THIS MUST BE MULTIPLIED
A1 2
39
BY A CONSTANT TO OBTAIN THE DIMENSIONLESS ORI VALUE.
ORI(1)=7.246*Y(1)
T1=T2$ A1=A2

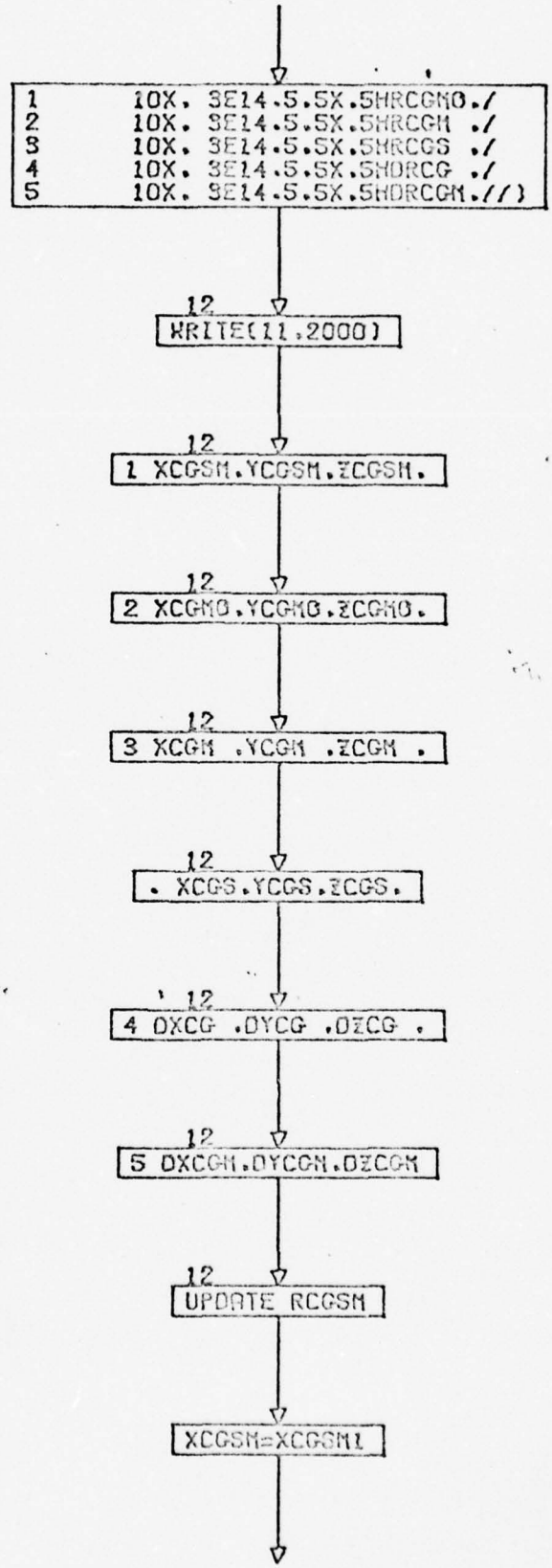
CONT. ON PG 4


A1 2
39
CHECK FOR MAXIMUM VALUE.
IF(ORI(I) .LT. ORIMAX)
F
T
RETURN
ORIMAX=ORI(I)
TMAX=TIME
ZACCMAX=ZACC/386.088
RETURN
END

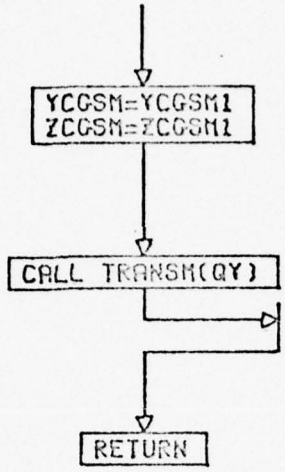
PG 4 FINAL


DYNCG
CALL CENGRAV
12
COMPUTE RCGM
XCGM=XCGMB + DXCGM
YCGM=YCGMB + DYCGM
ZCGM=ZCGMB + DZCGM
12
COMPUTE RCGSM
XCGSM1=(MM*XCGM + MSA*XCGS)/MS
YCGSM1=(MM*YCGM + MSA*YCGS)/MS
ZCGSM1=(MM*ZCGM + MSA*ZCGS)/MS
12
COMPUTE DRCG
DXCG=XCGSM1 - XCGSM
DYCG=YCGSM1 - YCGSM
DZCG=ZCGSM1 - ZCGSM
FORMAT(10X,3E14.5,3H***)
FORMAT(10X,3E14.5,5X,5HRCGSM,/

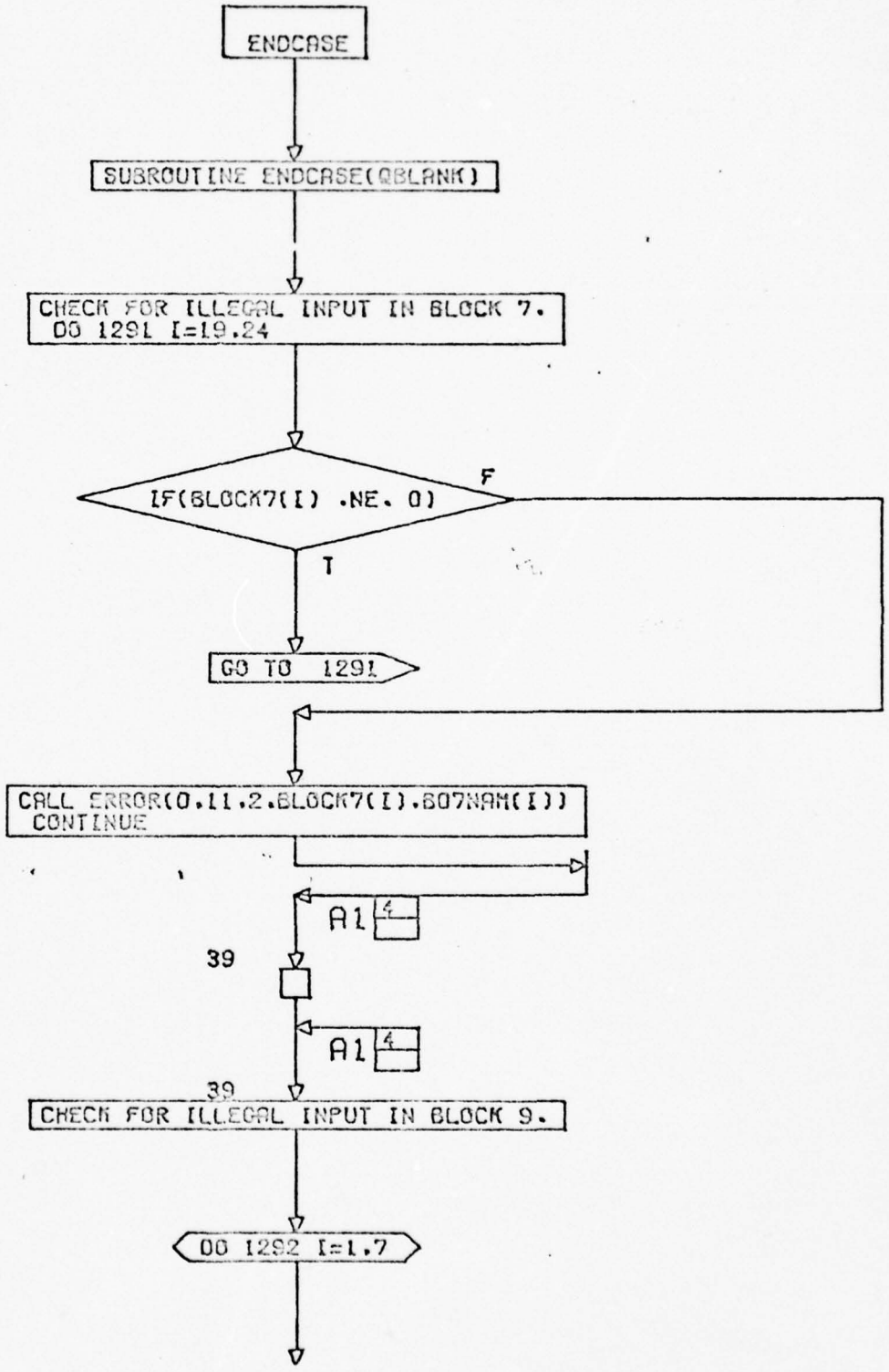
CONT. ON PG 2


1 10X. 3E14.5.5X.5HRCGMO./
2 10X. 3E14.5.5X.5HRCGM ./
3 10X. 3E14.5.5X.5HRCGS ./
4 10X. 3E14.5.5X.5HDRCG ./
5 10X. 3E14.5.5X.5HDRCGM.//)
12
WRITE(11,2000)
12
1 XCGSM.YCGSM.ZCGSM.
12
2 XCGMO.YCGMO.ZCGMO.
12
3 XCGM .YCGM .ZCGM .
12
. XCGS.YCGS.ZCGS.
12
4 DXCG .DYCG .DZCG .
12
5 DXCGM.DYCGM.DZCGM
12
UPDATE RCGSM
XCGSM=XCGSM1

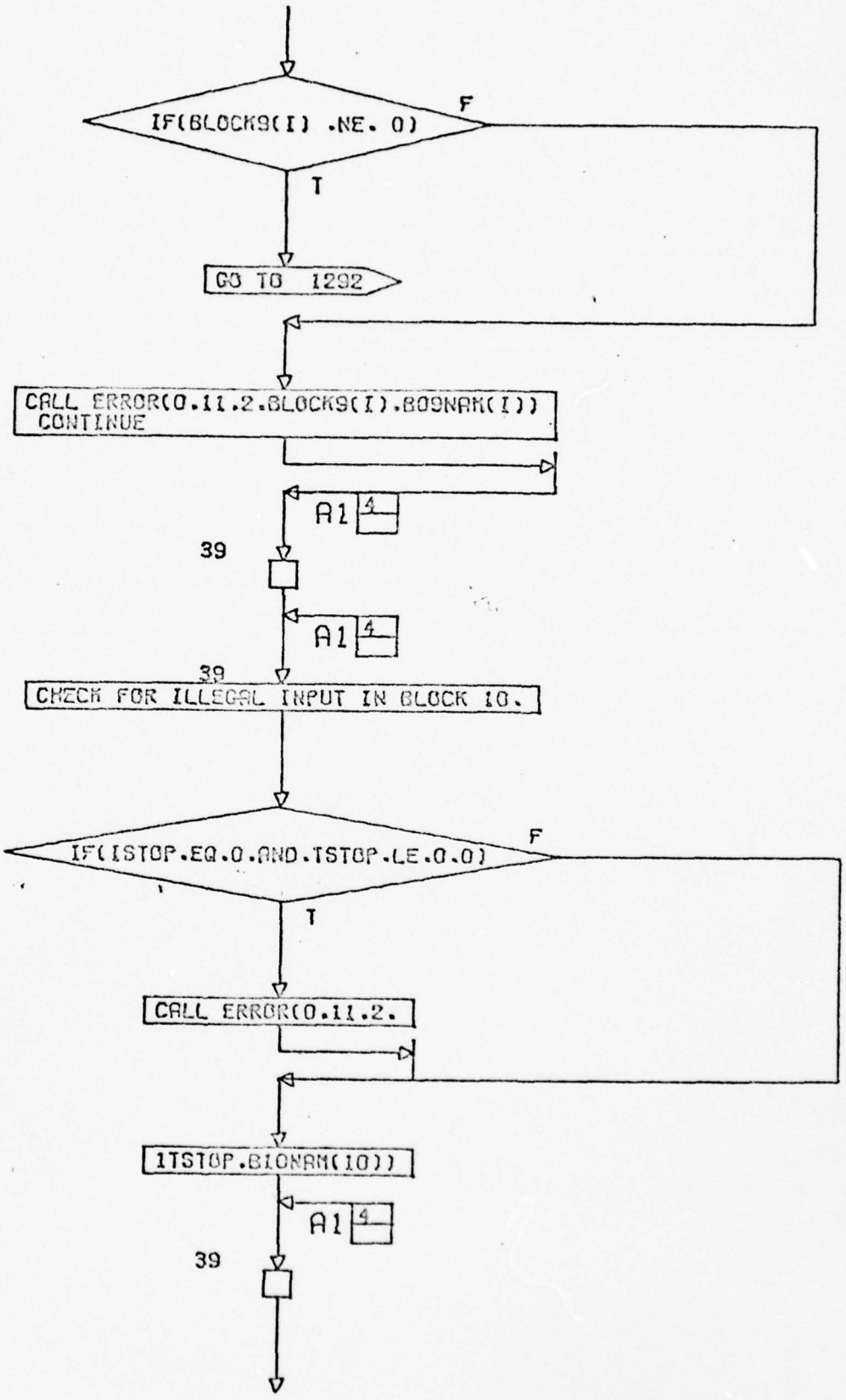
CONT. ON PG 3



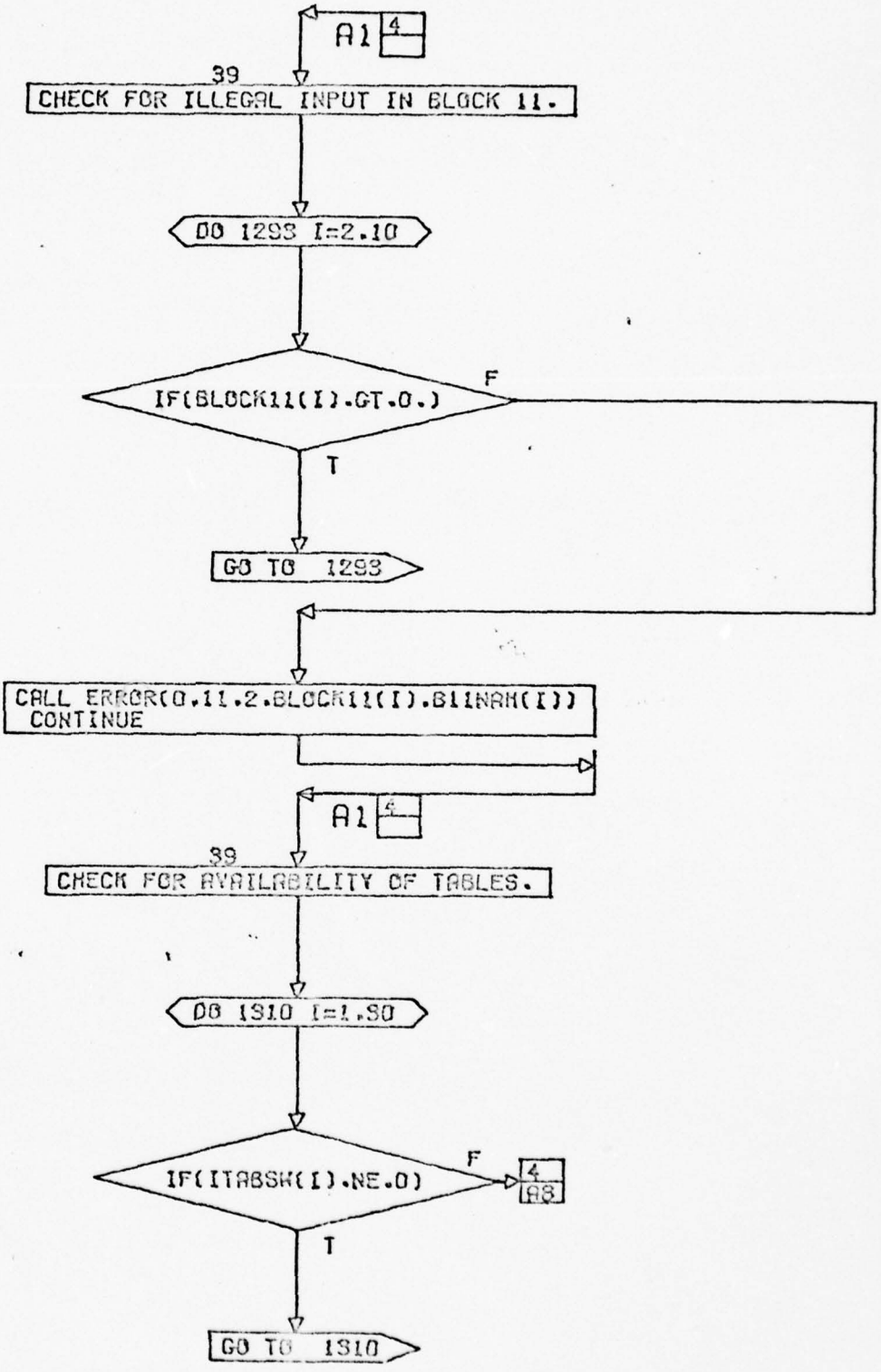
END


ENDCASE
SUBROUTINE ENDCASE(QBLANK)
CHECK FOR ILLEGAL INPUT IN BLOCK 7.
DO 1291 I=19,24
IF(BLOCK7(I) .NE. 0)
F
T
GO TO 1291
CALL ERROR(0,11,2,BLOCK7(I),BO7NAM(I))
CONTINUE
A1
4
39
A1
4
39
CHECK FOR ILLEGAL INPUT IN BLOCK 9.
DO 1292 I=1,7

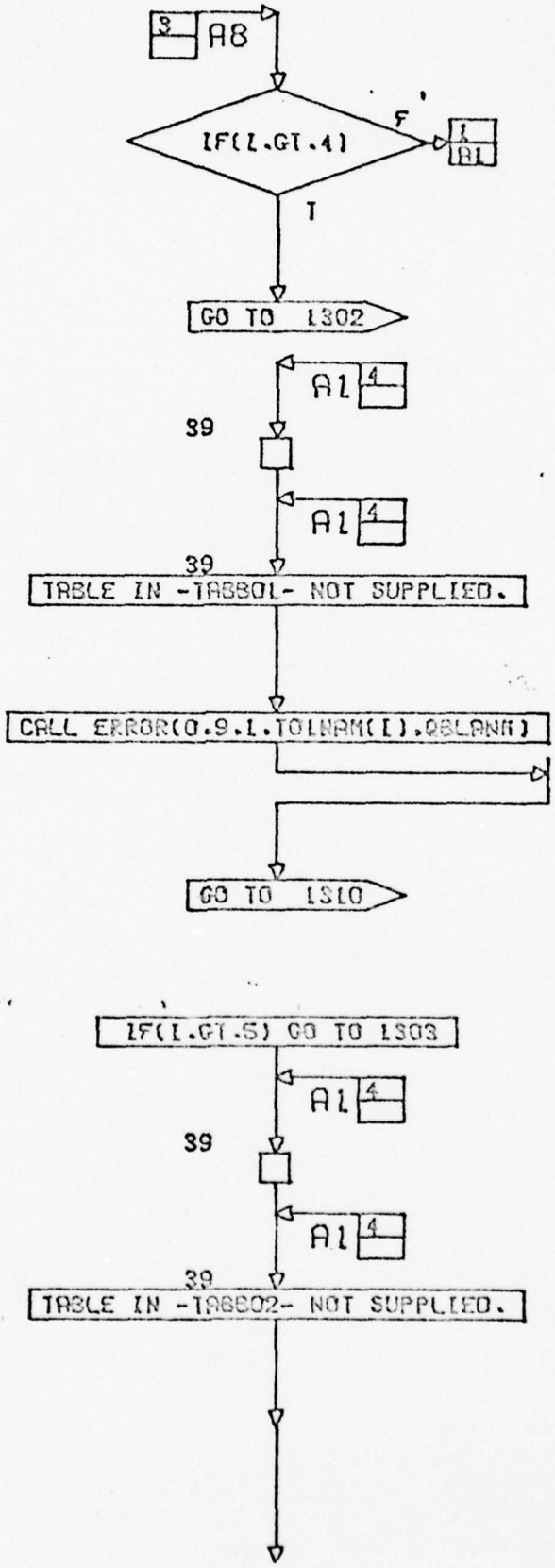
CONT. ON PG 2


IF(BLOCK9(I) .NE. 0)
F
T
GO TO 1292
CALL ERROR(0.11.2.BLOCK9(I).BOGNAM(I))
CONTINUE
A1 4
39
A1 4
39
CHECK FOR ILLEGAL INPUT IN BLOCK 10.
IF(ISTOP.EQ.0.AND.TSTOP.LE.0.0)
F
T
CALL ERROR(0.11.2.
ITSTOP.BIONAM(10))
A1 4
39

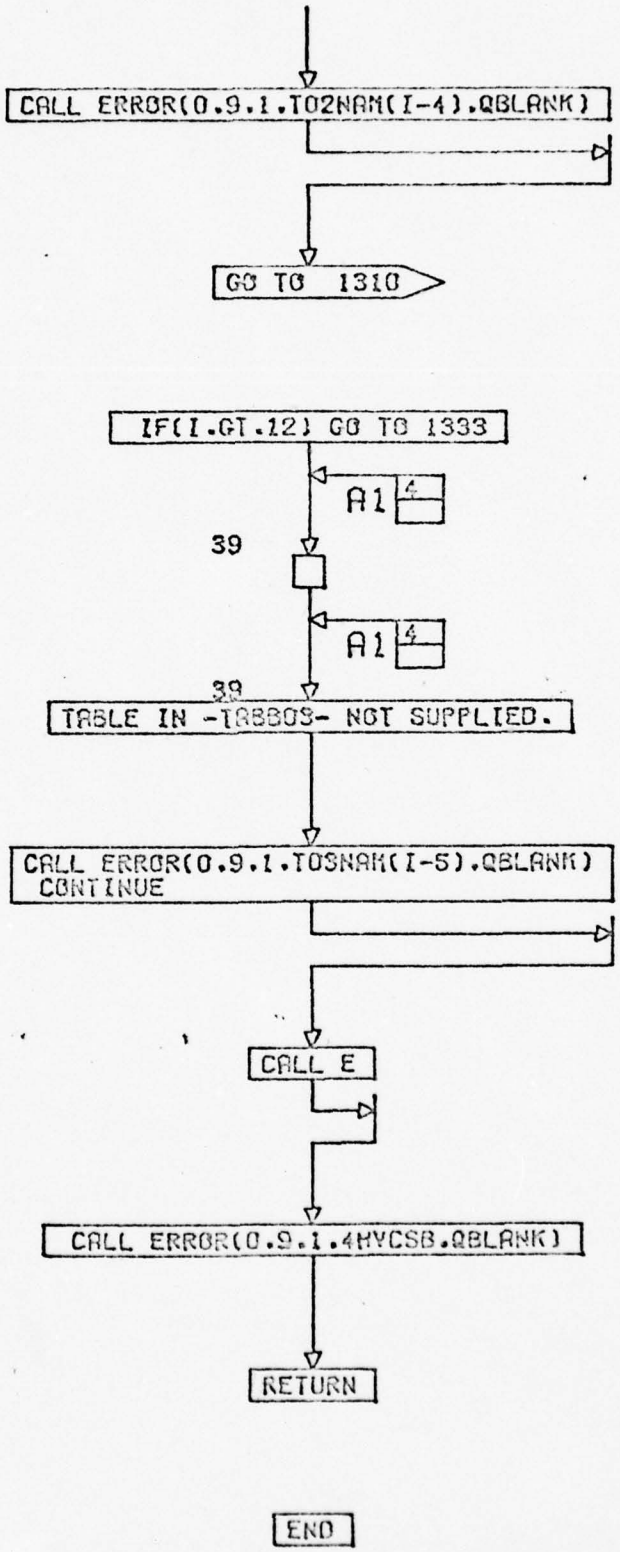
CONT. ON PG 3


A1 4
39
CHECK FOR ILLEGAL INPUT IN BLOCK 11.
DO 1293 I=2,10
IF(BLOCK11(I).GT.0.)
F
T
GO TO 1293
CALL ERROR(0,11,2,BLOCK11(I),BLINAM(I))
CONTINUE
A1 4
39
CHECK FOR AVAILABILITY OF TABLES.
DO 1310 I=1,30
IF(ITABSW(I).NE.0)
F
4
186
T
GO TO 1310

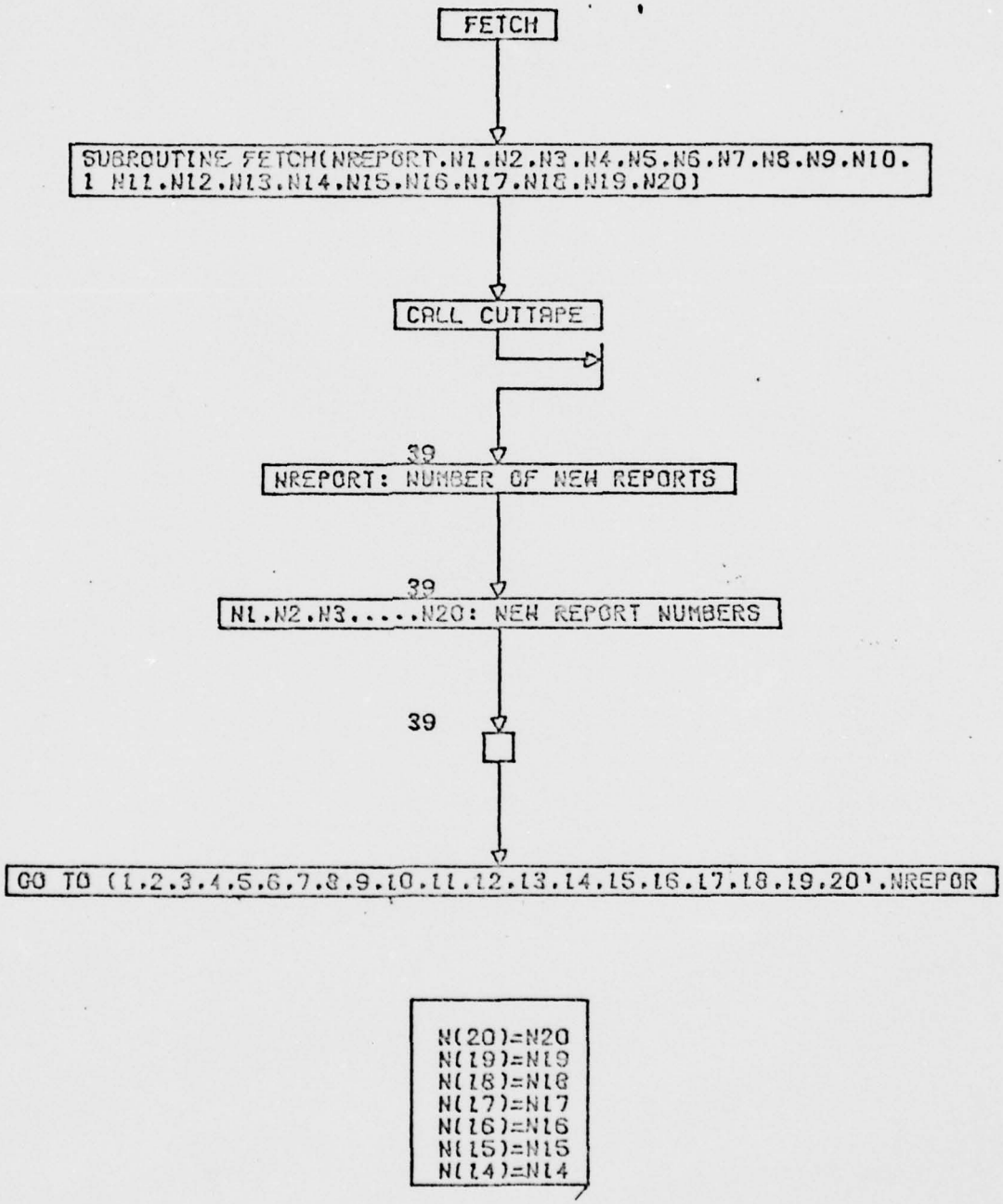
CONT. ON PG 4


A8
IF(I.GT.4)
F
T
GO TO 1302
A1
39
A1
39
TABLE IN -TABBOL- NOT SUPPLIED.
CALL ERROR(0.9.1.TOLNAM(I).QBLANK)
GO TO 1510
IF(I.GT.5) GO TO 1503
A1
39
A1
39
TABLE IN -TABBO2- NOT SUPPLIED.

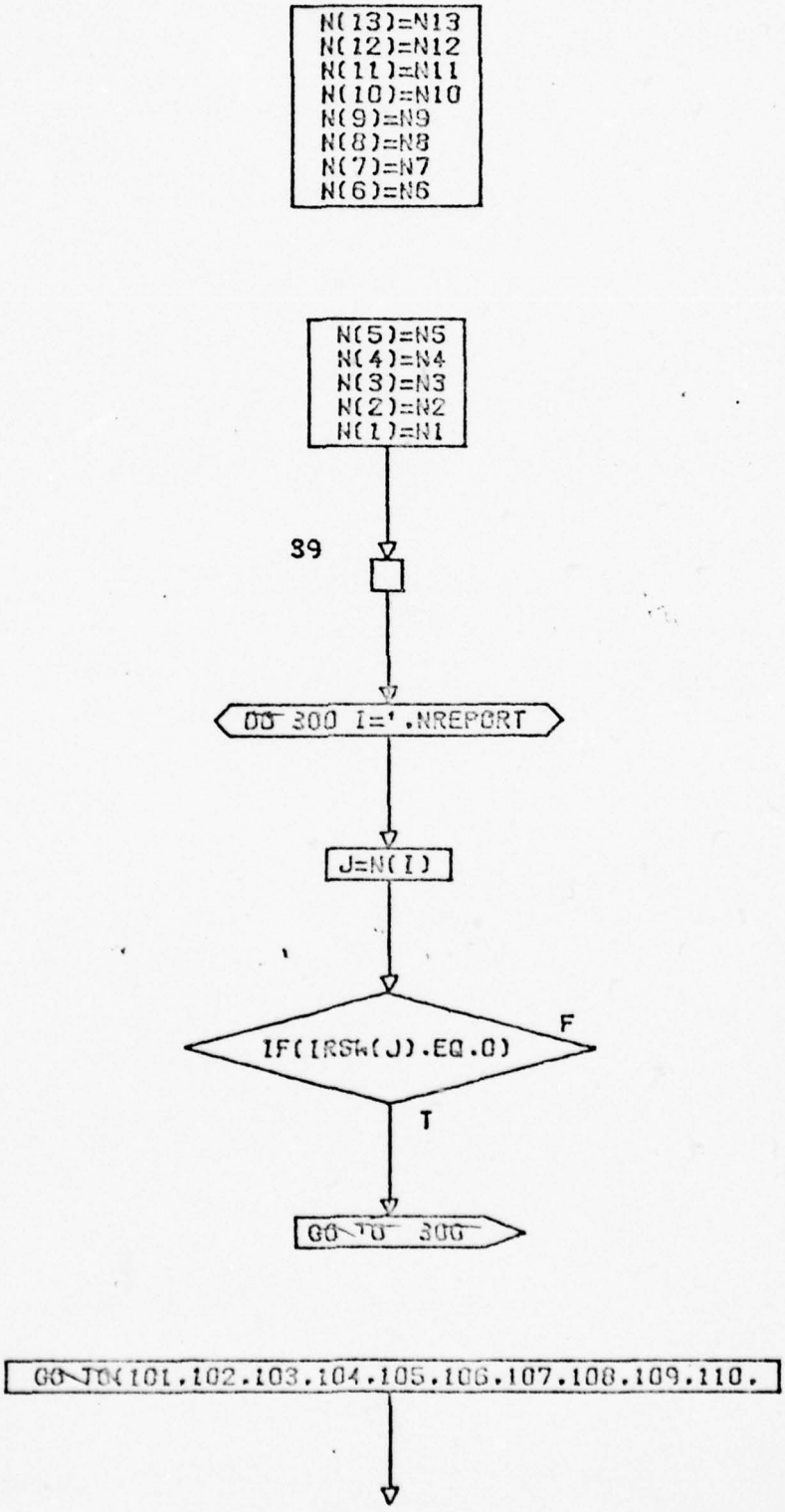
CONT. ON PG 5


CALL ERROR(0.9.1.TO2NAM(I-4).QBLANK)
GO TO 1310
IF(I.GT.12) GO TO 1333
A1 4
39
A1 4
39
TABLE IN -TABBOS- NOT SUPPLIED.
CALL ERROR(0.9.1.TO3NAM(I-5).QBLANK)
CONTINUE
CALL E
CALL ERROR(0.9.1.4HVCSB.QBLANK)
RETURN
END

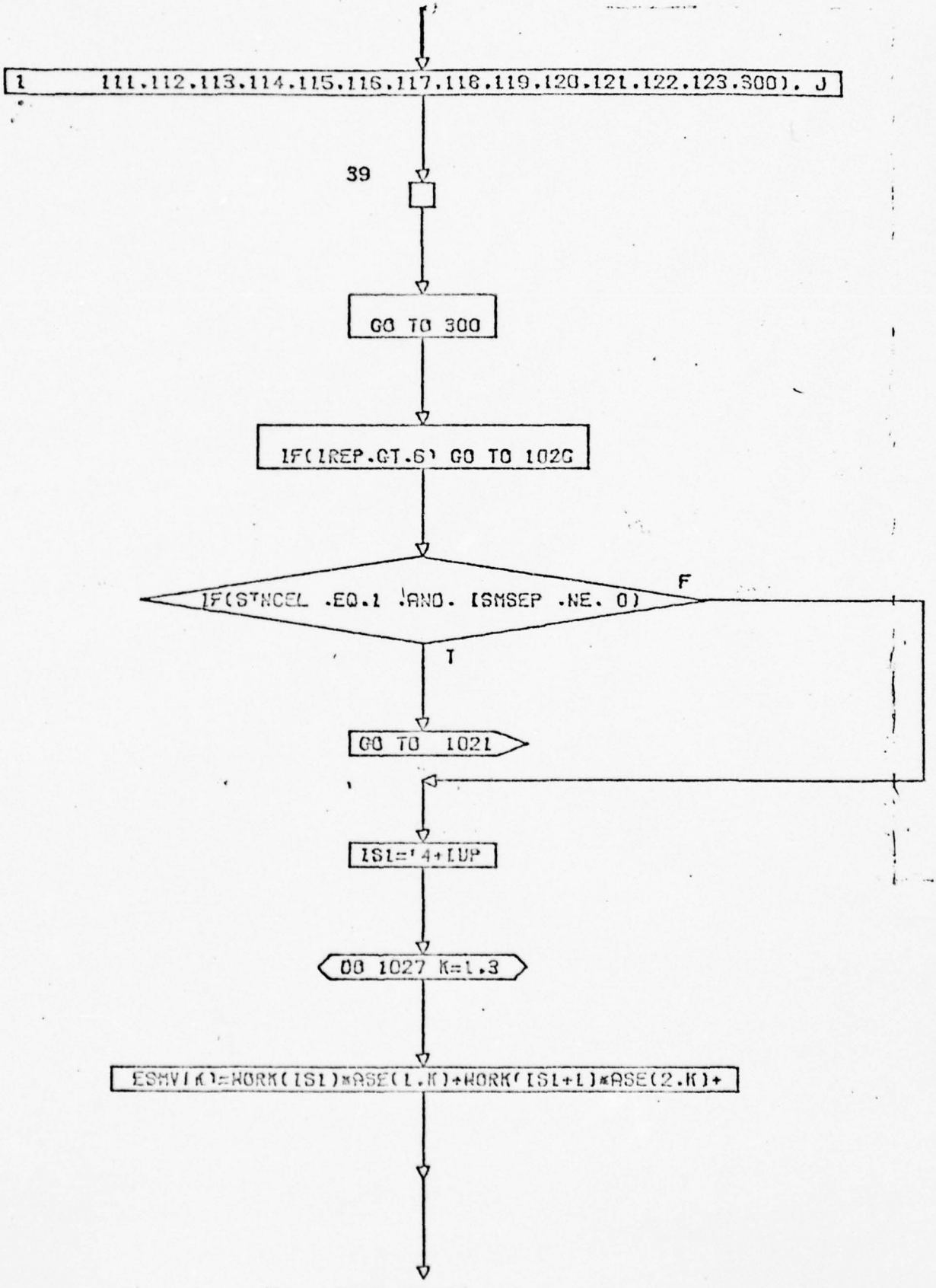
PG 5 FINAL


FETCH
SUBROUTINE FETCH(NREPORT,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
1 N11,N12,N13,N14,N15,N16,N17,N18,N19,N20)
CALL CUTTAPE
39
NREPORT: NUMBER OF NEW REPORTS
39
N1,N2,N3.....N20: NEW REPORT NUMBERS
39
GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),NREPOR
N(20)=N20
N(19)=N19
N(18)=N18
N(17)=N17
N(16)=N16
N(15)=N15
N(14)=N14

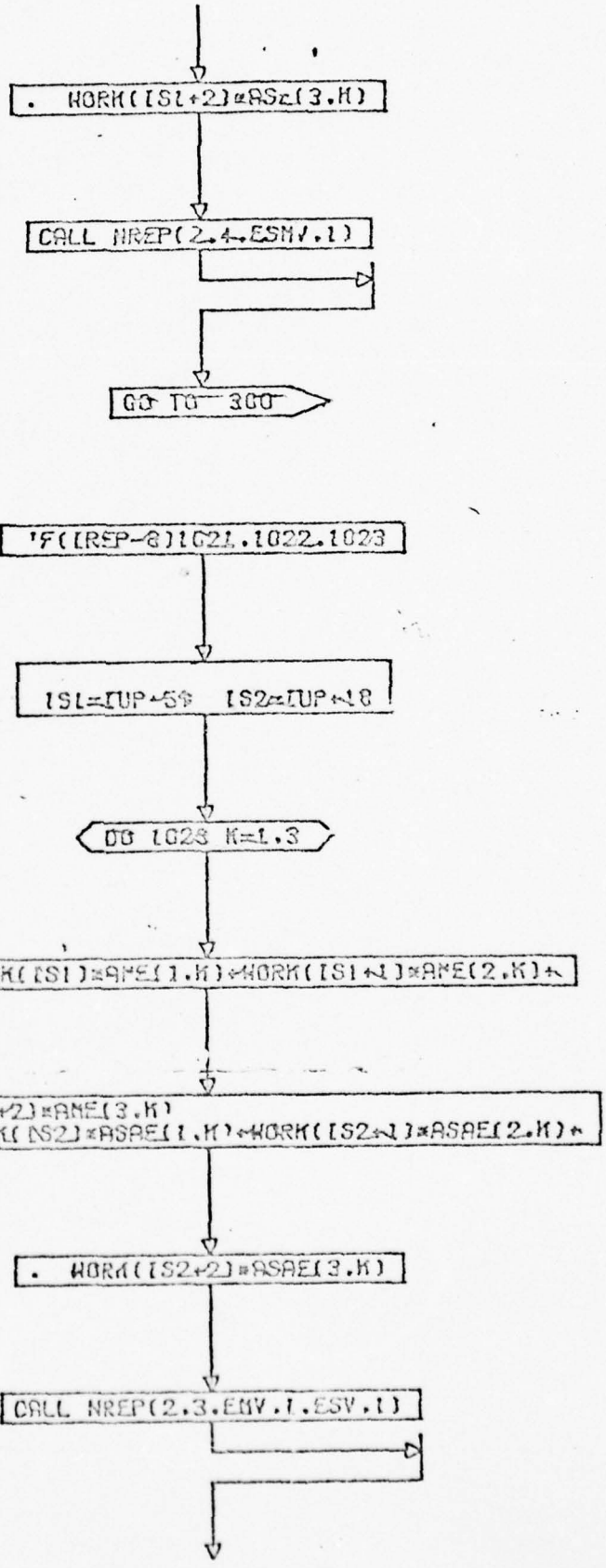
CONT. ON PG 2


N(13)=N13
N(12)=N12
N(11)=N11
N(10)=N10
N(9)=N9
N(8)=N8
N(7)=N7
N(6)=N6
N(5)=N5
N(4)=N4
N(3)=N3
N(2)=N2
N(1)=N1
39
DO 300 I=1,NREPORT
J=N(I)
IF(IRSW(J).EQ.0)
F
T
GO TO 300
GO TO(101,102,103,104,105,106,107,108,109,110,

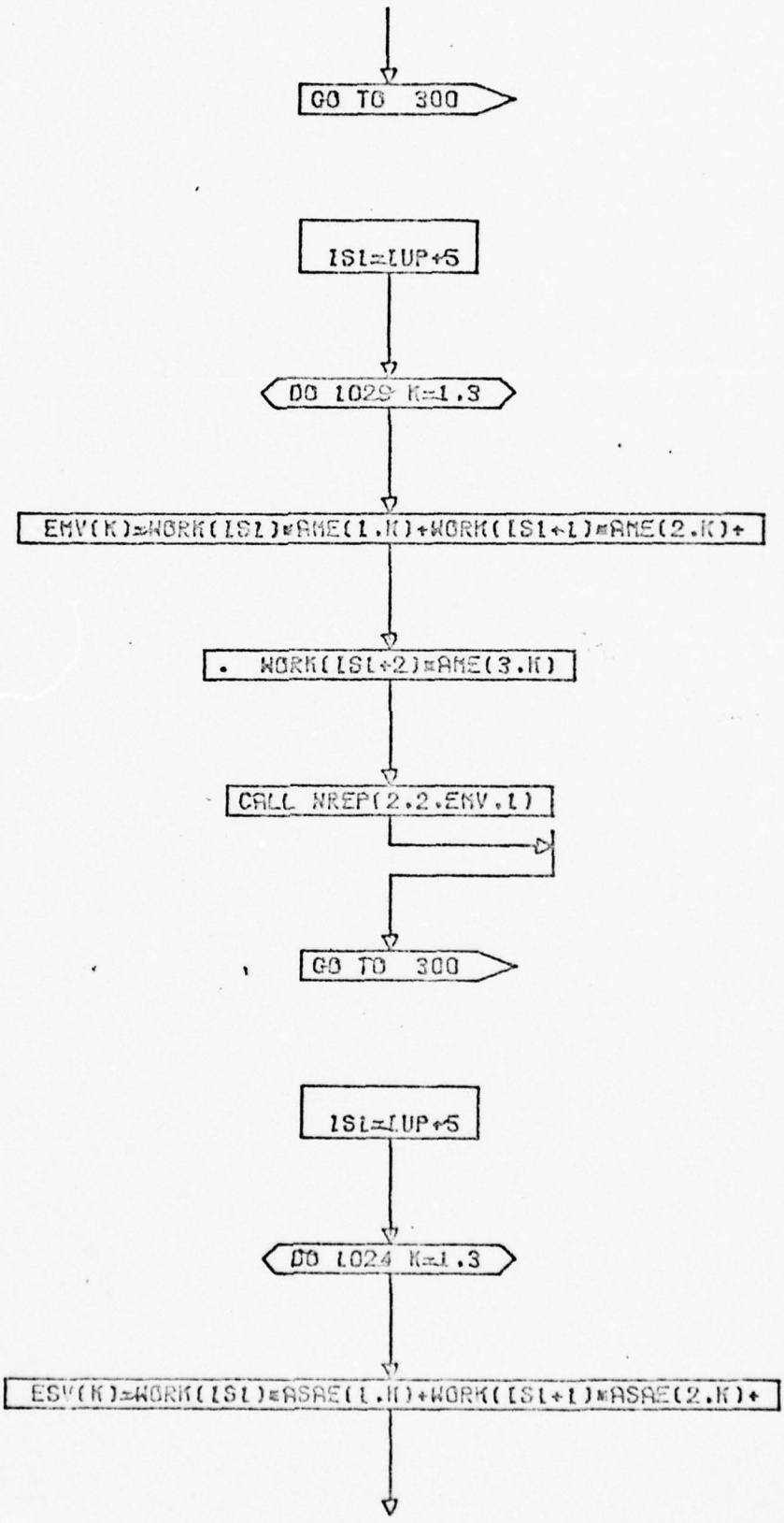
CONT. ON PG 3


1 111.112.113.114.115.116.117.118.119.120.121.122.123.300). J
39
GO TO 300
IF(IREP.GT.6) GO TO 1020
IF(STNCEL .EQ.1 .AND. ISMSEP .NE. 0)
F
T
GO TO 1021
IS1='4+IUP
DO 1027 K=1,3
ESMV(K)=WORK(IS1)*ASE(1,K)+WORK(IS1+1)*ASE(2,K)+

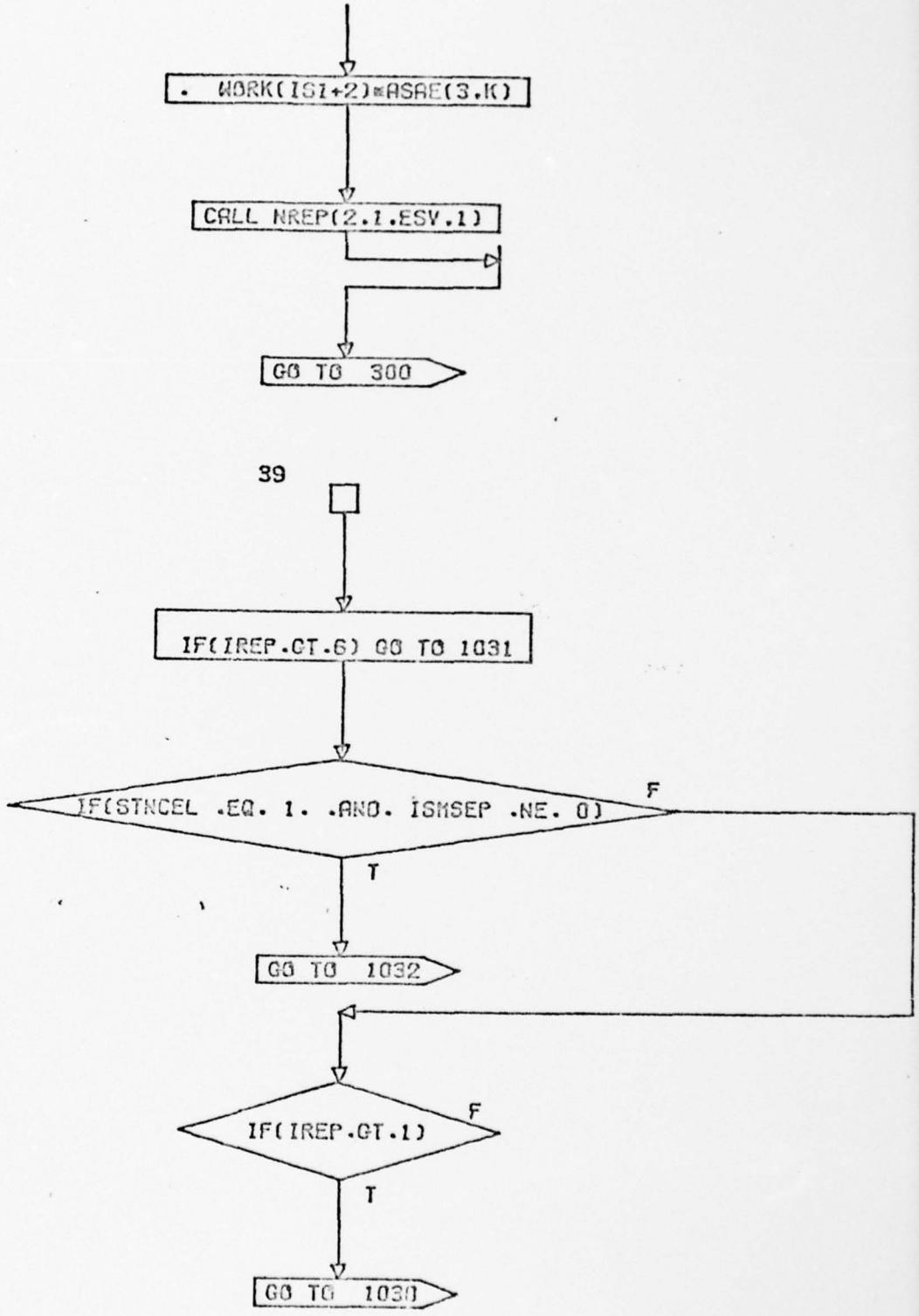
CONT. ON PG 4



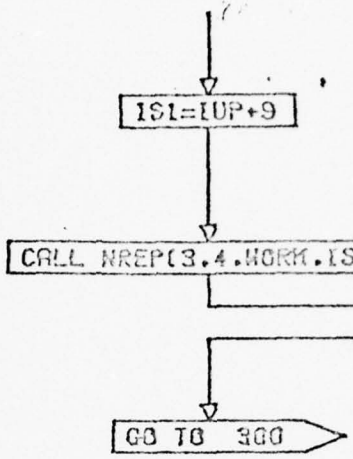
-62- CONT. ON PG 5

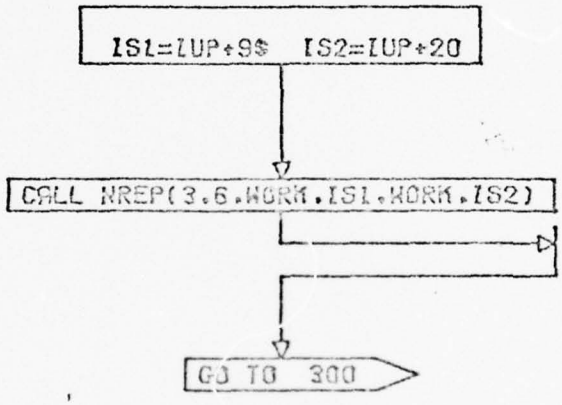

GO TO 300
ISL=LUP+5
DO 1029 K=1,3
ENV(K)=WORK(ISL)*AME(1,K)+WORK(ISL+1)*AME(2,K)+
. WORK(ISL+2)*AME(3,K)
CALL NREP(2,2,ENV,1)
GO TO 300
ISL=LUP+5
DO 1024 K=1,3
ESV(K)=WORK(ISL)*ASAE(1,K)+WORK(ISL+1)*ASAE(2,K)+

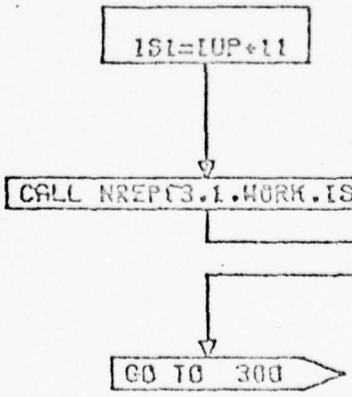
CONT. ON PG 6


WORK(IS1+2)=ASAE(3,K)
CALL NREP(2,1,ESV,1)
GO TO 300
39
IF(IREP.GT.6) GO TO 1031
IF(STNCEL .EQ. 1. .AND. ISMSEP .NE. 0)
F
T
GO TO 1032
IF(IREP.GT.1)
F
T
GO TO 1030

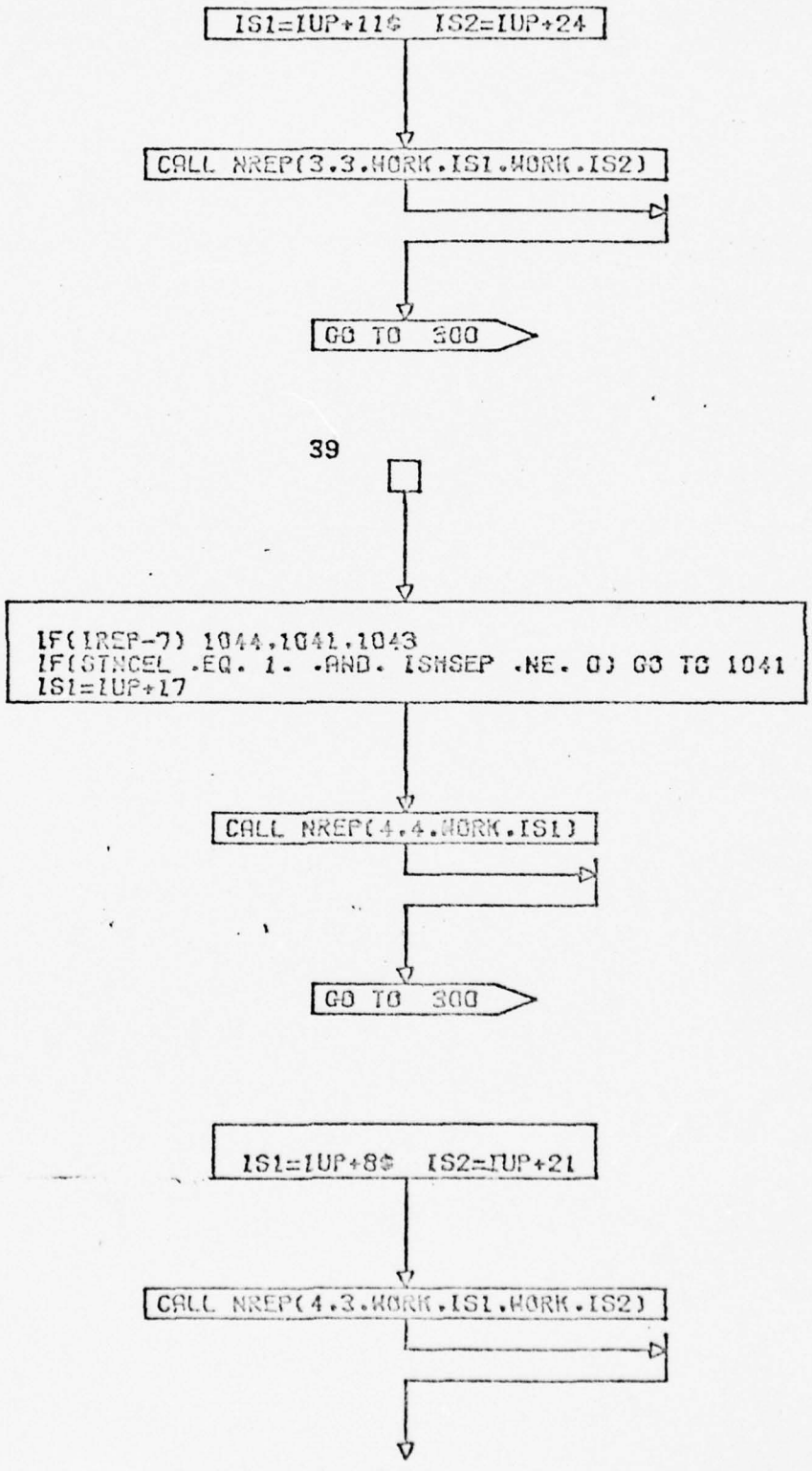
CONT. ON PG 7


IS1=IUP+9
CALL NREP(3,4,WORK,IS1)
GO TO 300

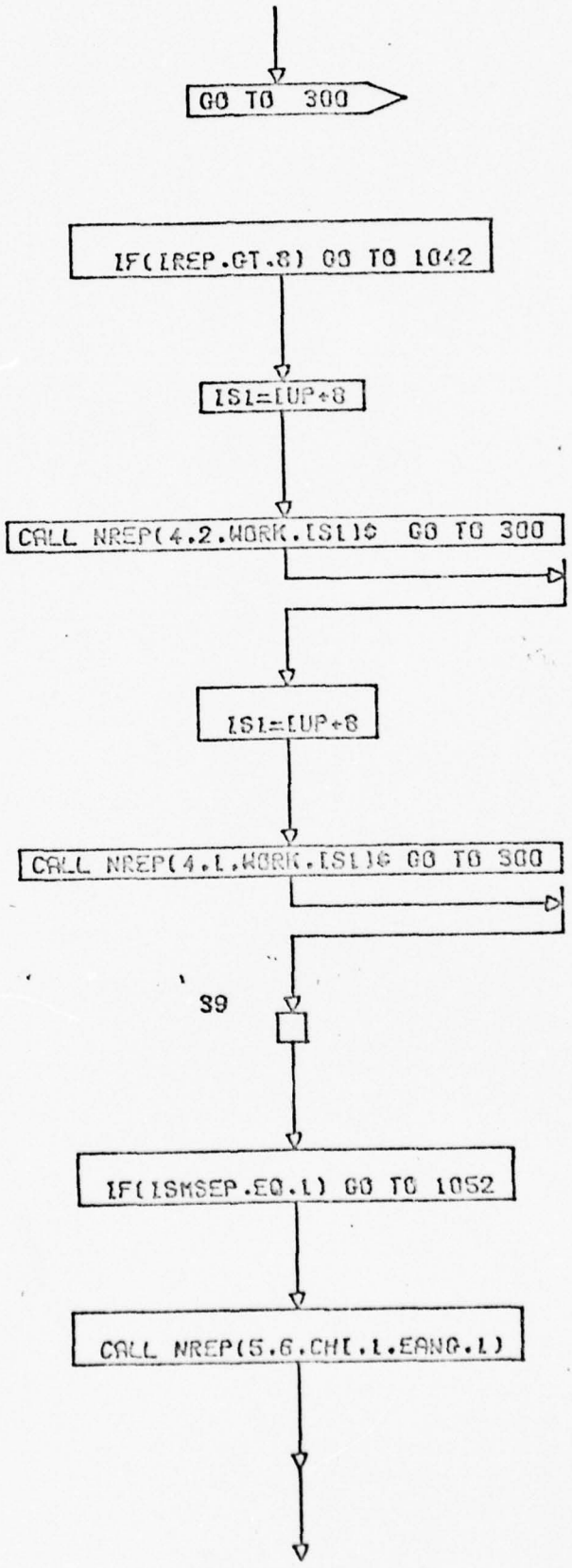

IS1=IUP+9$ IS2=IUP+20
CALL NREP(3,6,WORK,IS1,WORK,IS2)
GO TO 300


IS1=IUP+11
CALL NREP(3,1,WORK,IS1)
GO TO 300

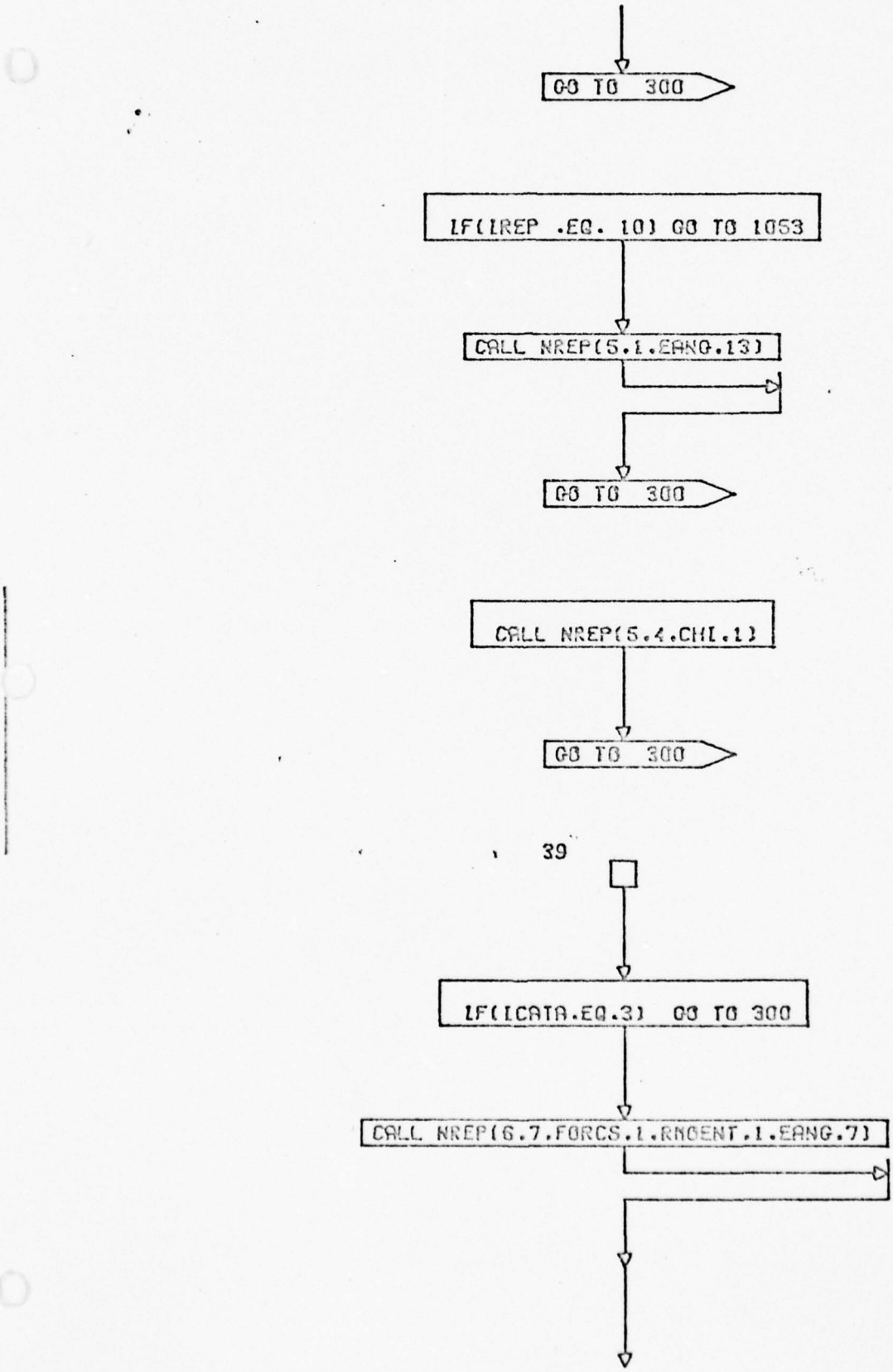
CONT. ON PG 8

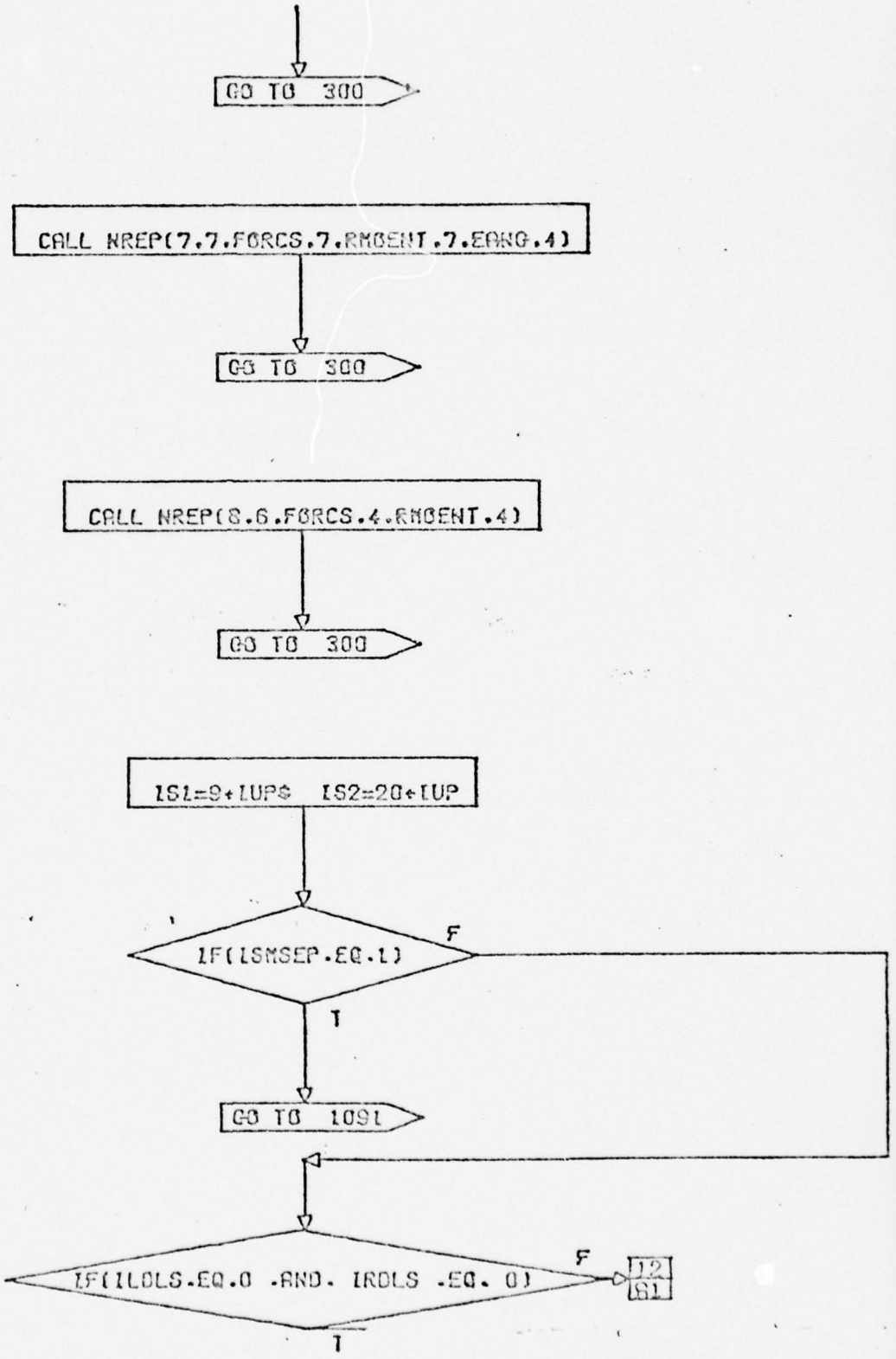

IS1=IUP+11$ IS2=IUP+24
CALL NREP(3,3,WORK,IS1,WORK,IS2)
GO TO 300
39
IF(IREP-7) 1044,1041,1043
IF(STNCEL .EQ. 1. .AND. ISHSEP .NE. 0) GO TO 1041
IS1=IUP+17
CALL NREP(4,4,WORK,IS1)
GO TO 300
IS1=IUP+8$ IS2=IUP+21
CALL NREP(4,3,WORK,IS1,WORK,IS2)

CONT. ON PG 9

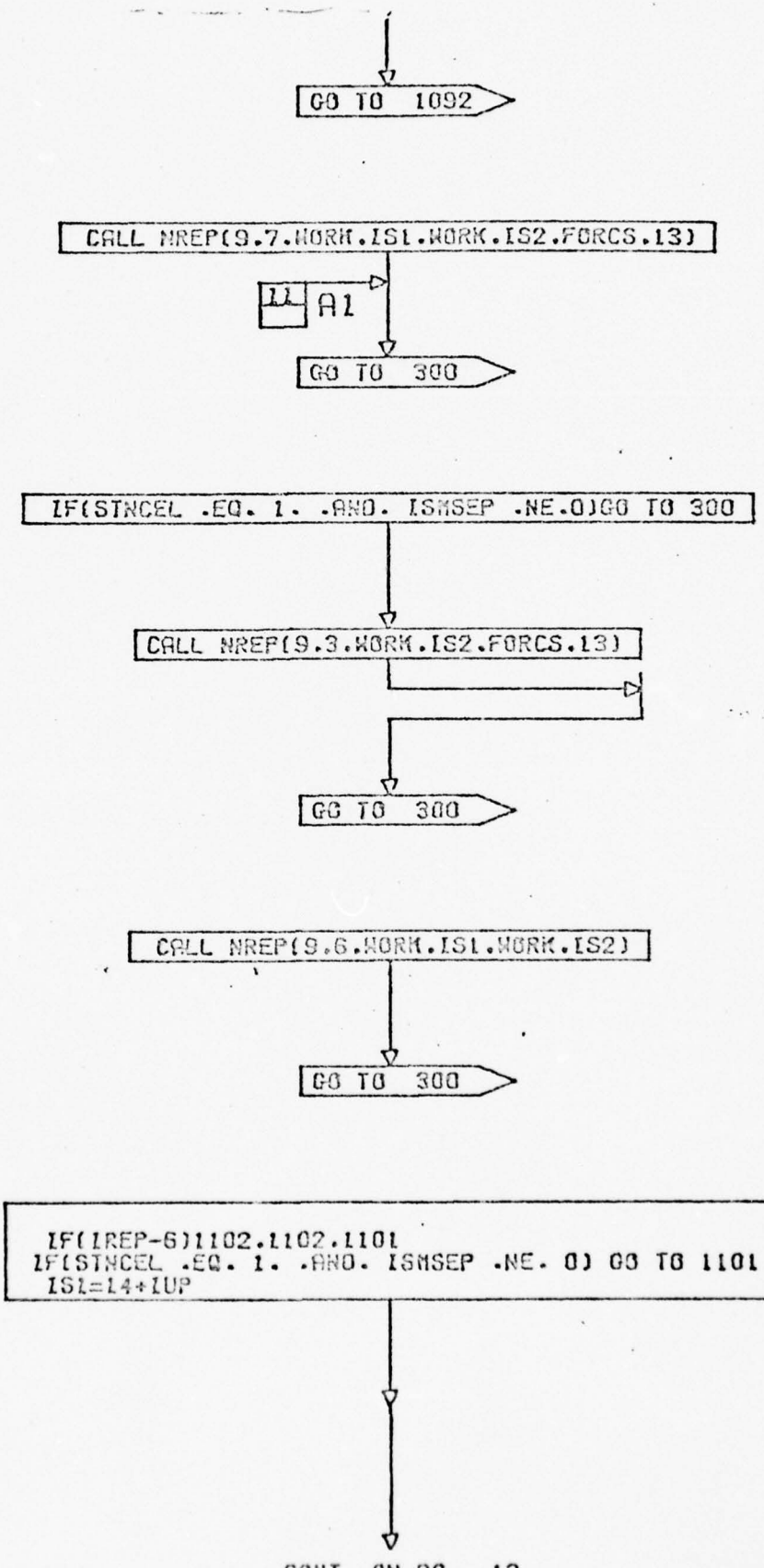


-67- CONT. ON PG 10

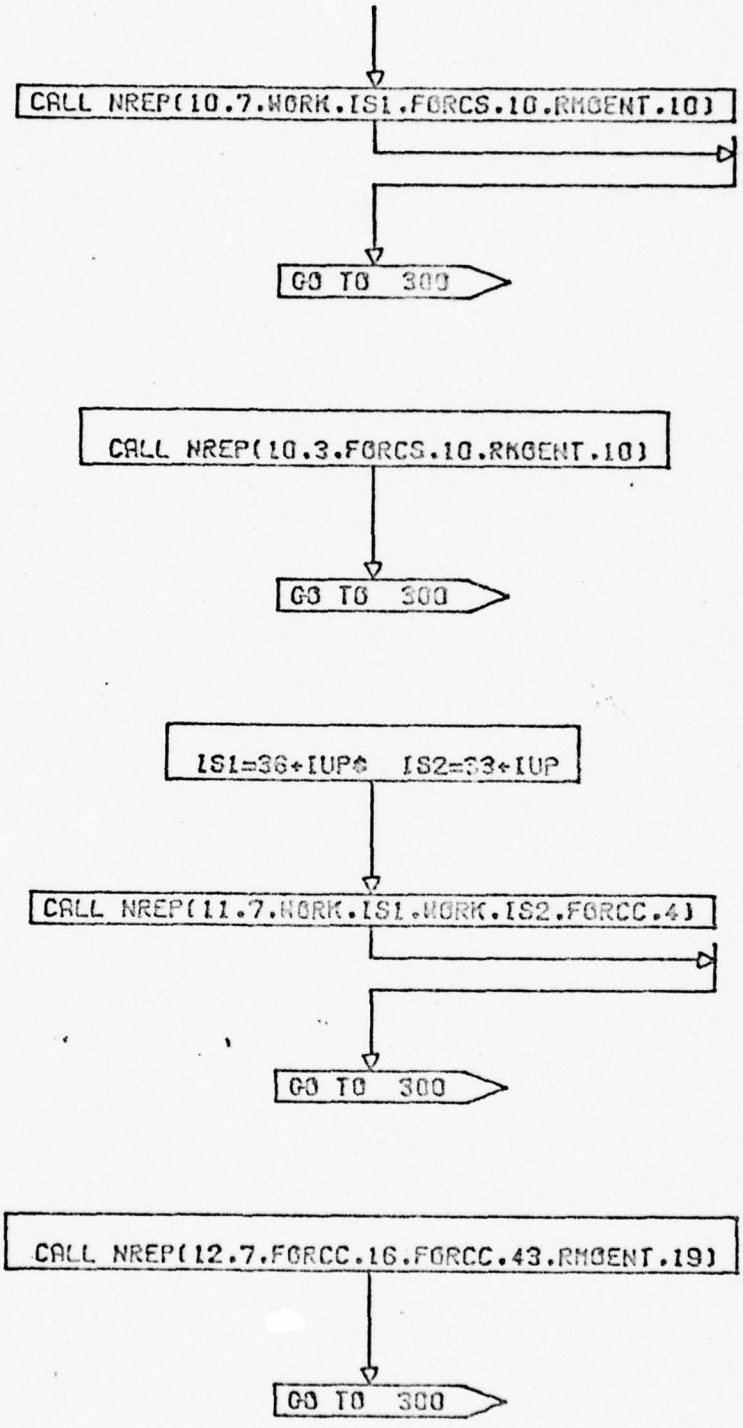

GO TO 300
IF(IREP .EQ. 10) GO TO 1053
CALL NREP(5,1,EANG,13)
GO TO 300
CALL NREP(5,4,CHI,1)
GO TO 300
39
IF(ICATA.EQ.3) GO TO 300
CALL NREP(6,7,FORCS,1,RNOENT,1,EANG,7)
CONT. ON PG 11


GO TO 300
CALL NREP(7,7,FORCS,7,RNGENT,7,ERNG,4)
GO TO 300
CALL NREP(8,6,FORCS,4,RNGENT,4)
GO TO 300
IS1=9+IUP$ IS2=20+IUP
IF(ISMSEP.EQ.1)
F
T
GO TO 1091
IF(ILOLS.EQ.0 .AND. IROLS .EQ. 0)
F
T
J2
S1

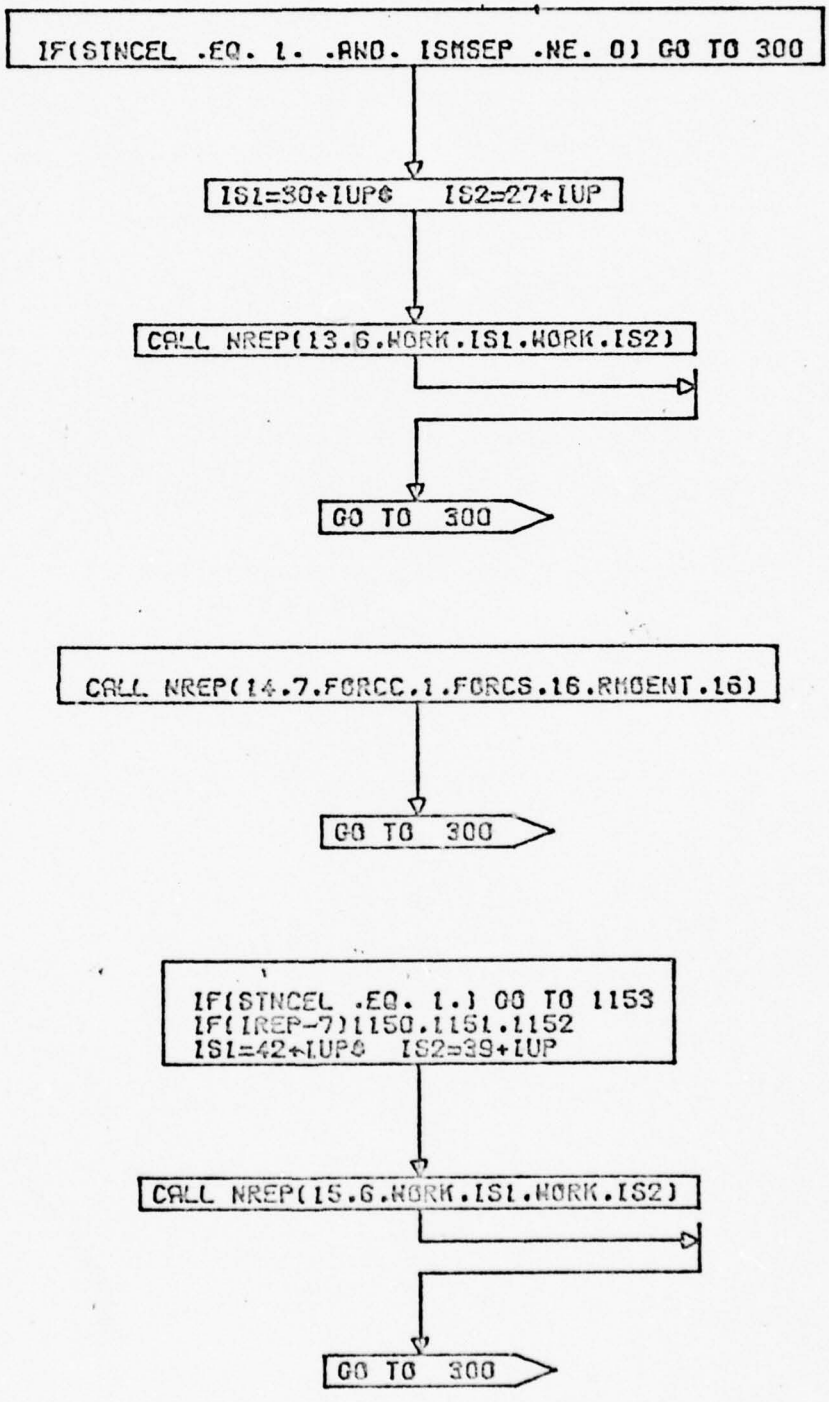
CONT. ON PG 12



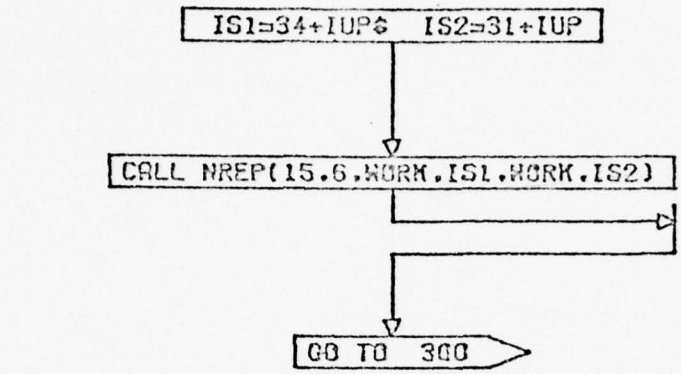
-70- CONT. ON PG 13

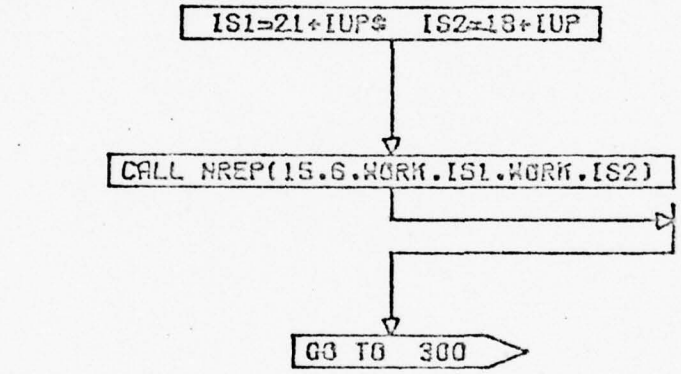

CALL NREP(10,7,WORK,IS1,FORCS,10,RHOENT,10)
GO TO 300
CALL NREP(10,3,FORCS,10,RHOENT,10)
GO TO 300
IS1=36+IUP$ IS2=33+IUP
CALL NREP(11,7,WORK,IS1,WORK,IS2,FORCC,4)
GO TO 300
CALL NREP(12,7,FORCC,16,FORCC,43,RHOENT,19)
GO TO 300

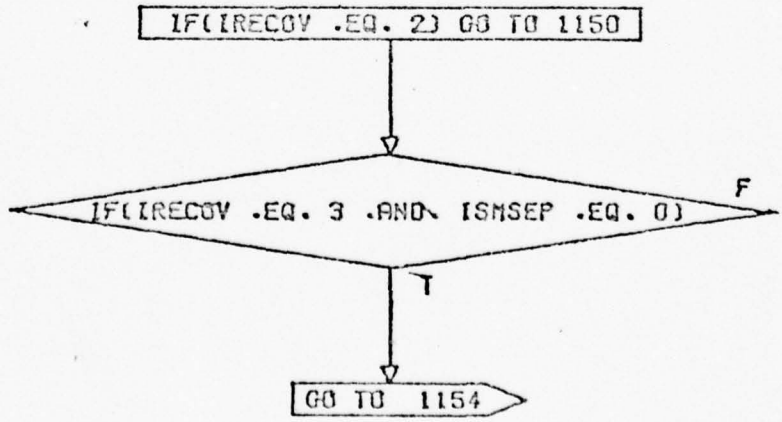
CONT. ON PG 14


IF(STNCEL .EQ. 1. .AND. ISMSEP .NE. 0) GO TO 300
IS1=30+IUP$ IS2=27+IUP
CALL NREP(13.6.WORK.IS1.WORK.IS2)
GO TO 300
CALL NREP(14.7.FORCC.1.FORCS.16.RHOENT.16)
GO TO 300
IF(STNCEL .EQ. 1.) GO TO 1153
IF(IREP-7)1150.1151.1152
IS1=42+IUP$ IS2=39+IUP
CALL NREP(15.6.WORK.IS1.WORK.IS2)
GO TO 300

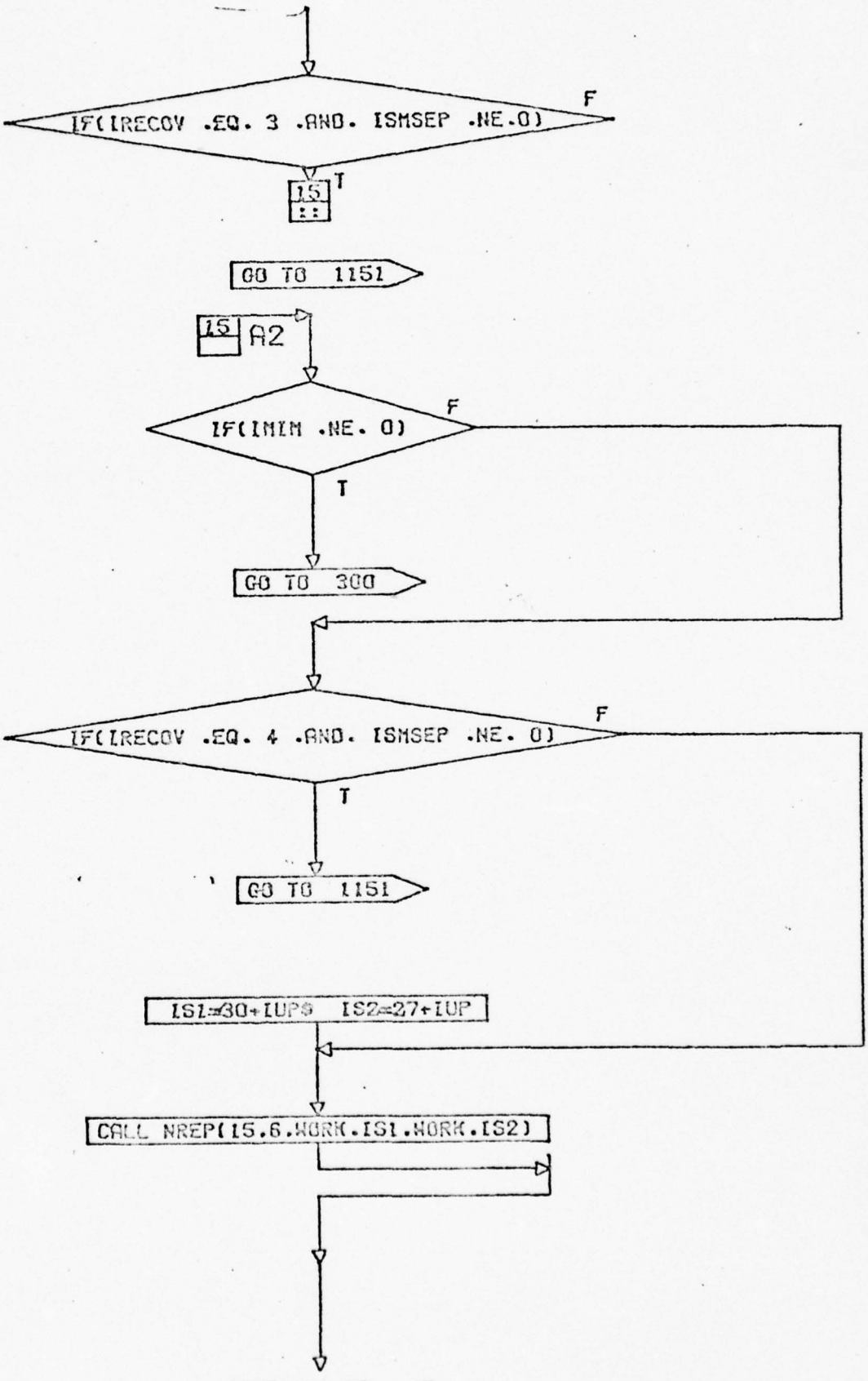
CONT. ON PG 15


IS1=34+IUP$ IS2=31+IUP
CALL NREP(15,6,WORK,IS1,WORK,IS2)
GO TO 300

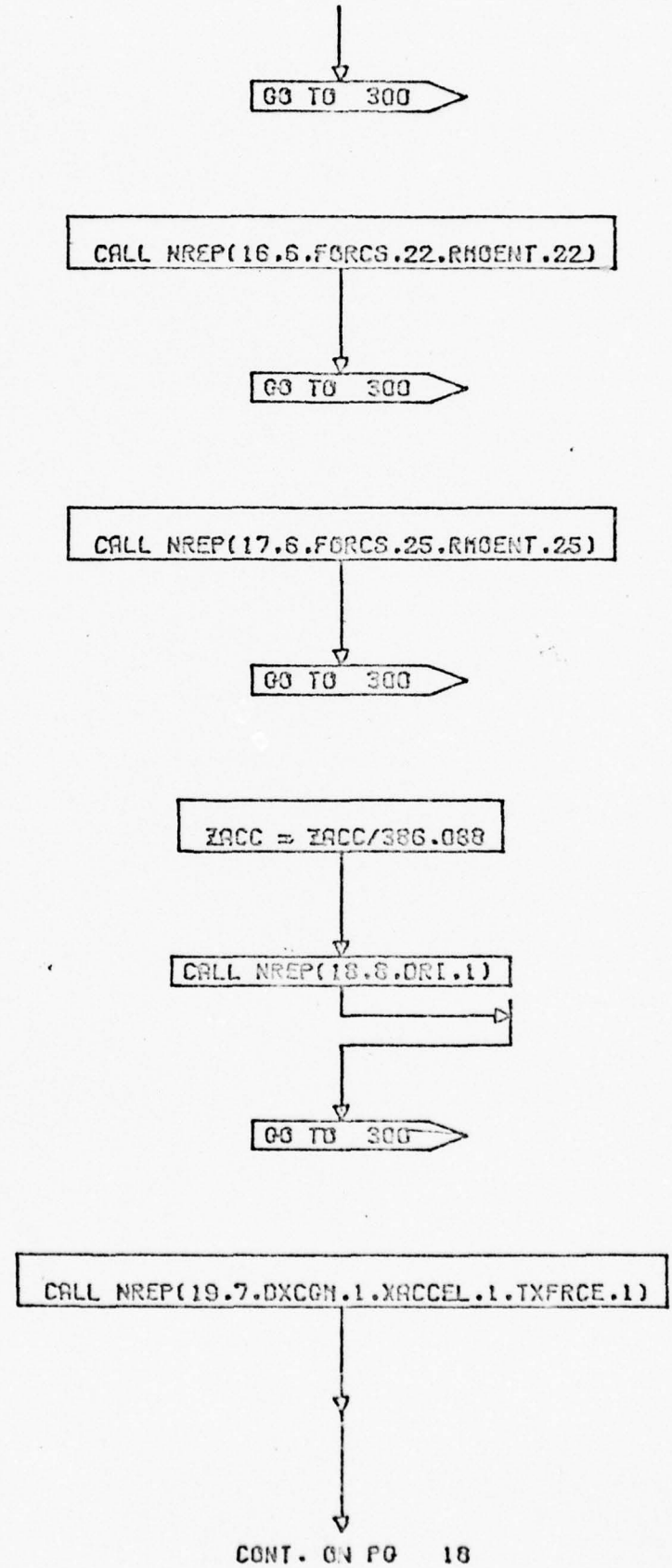

IS1=21+IUP$ IS2=18+IUP
CALL NREP(15,6,WORK,IS1,WORK,IS2)
GO TO 300


IF(IRECOV .EQ. 2) GO TO 1150
IF(IRECOV .EQ. 3 .AND. ISMSEP .EQ. 0)
F
T
GO TO 1154

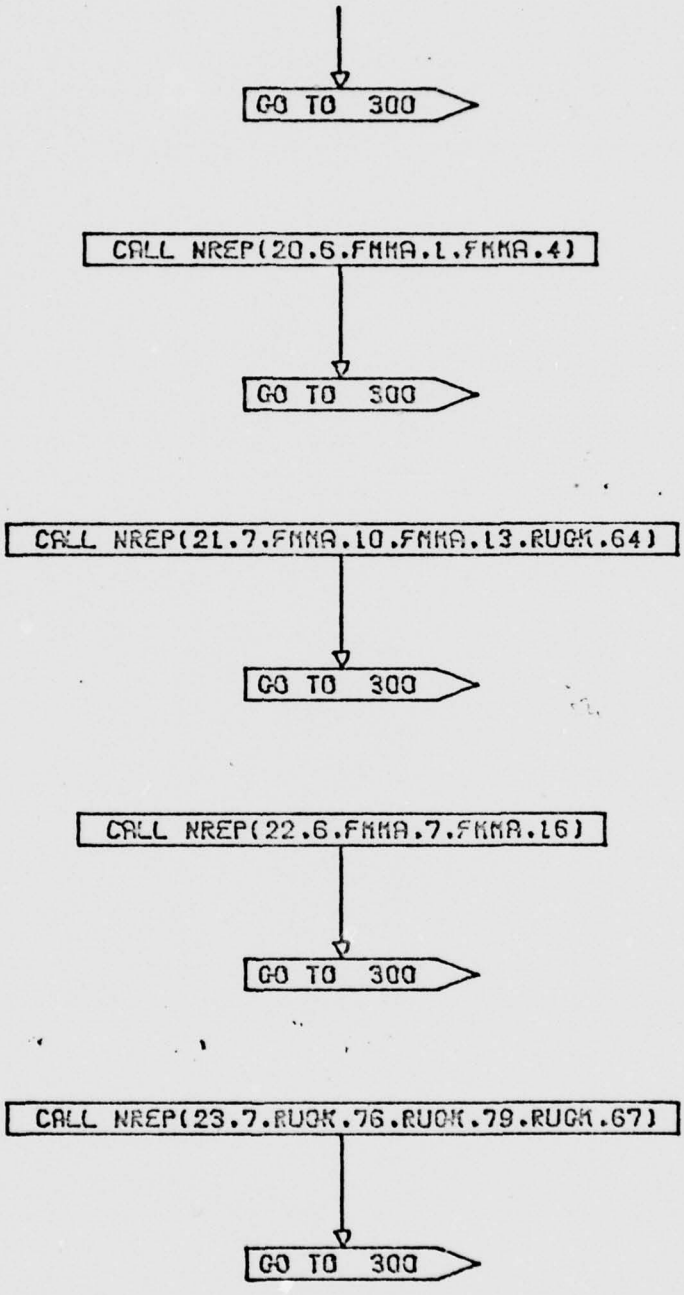
CONT. ON PG 16

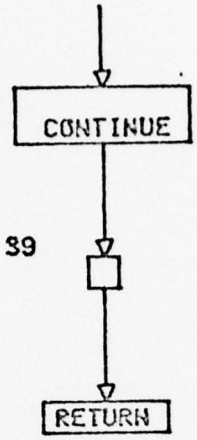

IF(IRECOV .EQ. 3 .AND. ISMSEP .NE.0)
F
T
15
GO TO 1151
15 A2
IF(IMIM .NE. 0)
F
T
GO TO 300
IF(IRECOV .EQ. 4 .AND. ISMSEP .NE. 0)
F
T
GO TO 1151
IS1=30+IUP$ IS2=27+IUP
CALL NREP(15,6,WORK,IS1,WORK,IS2)

CONT. ON PG 17

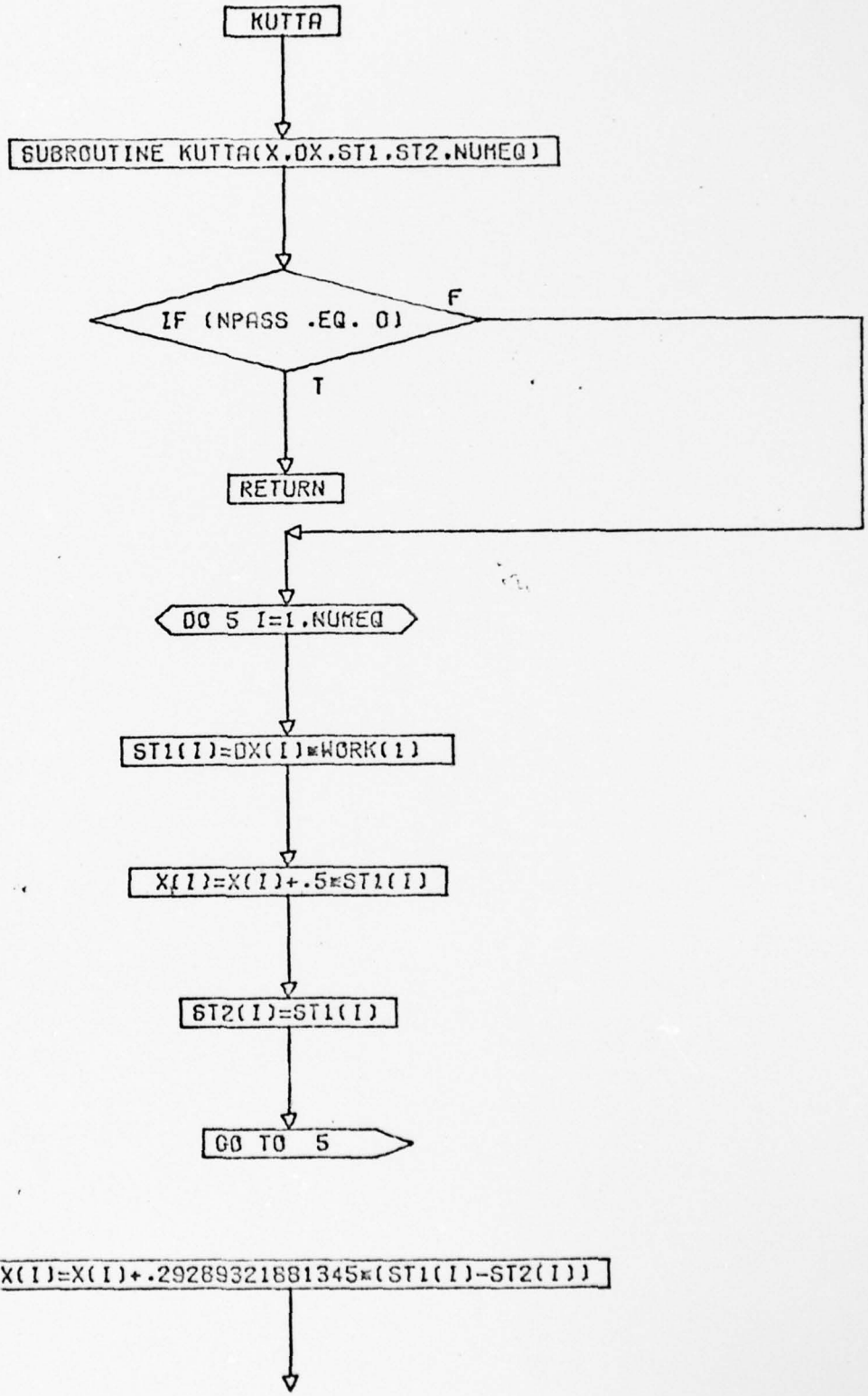

GO TO 300
CALL NREP(16,6,FORCS,22,RHOENT,22)
GO TO 300
CALL NREP(17,6,FORCS,25,RHOENT,25)
GO TO 300
ZACC = ZACC/386.088
CALL NREP(18,6,ORI,1)
GO TO 300
CALL NREP(19,7,DXCGN,1,XACCEL,1,TXFRCE,1)

CONT. ON PG 18

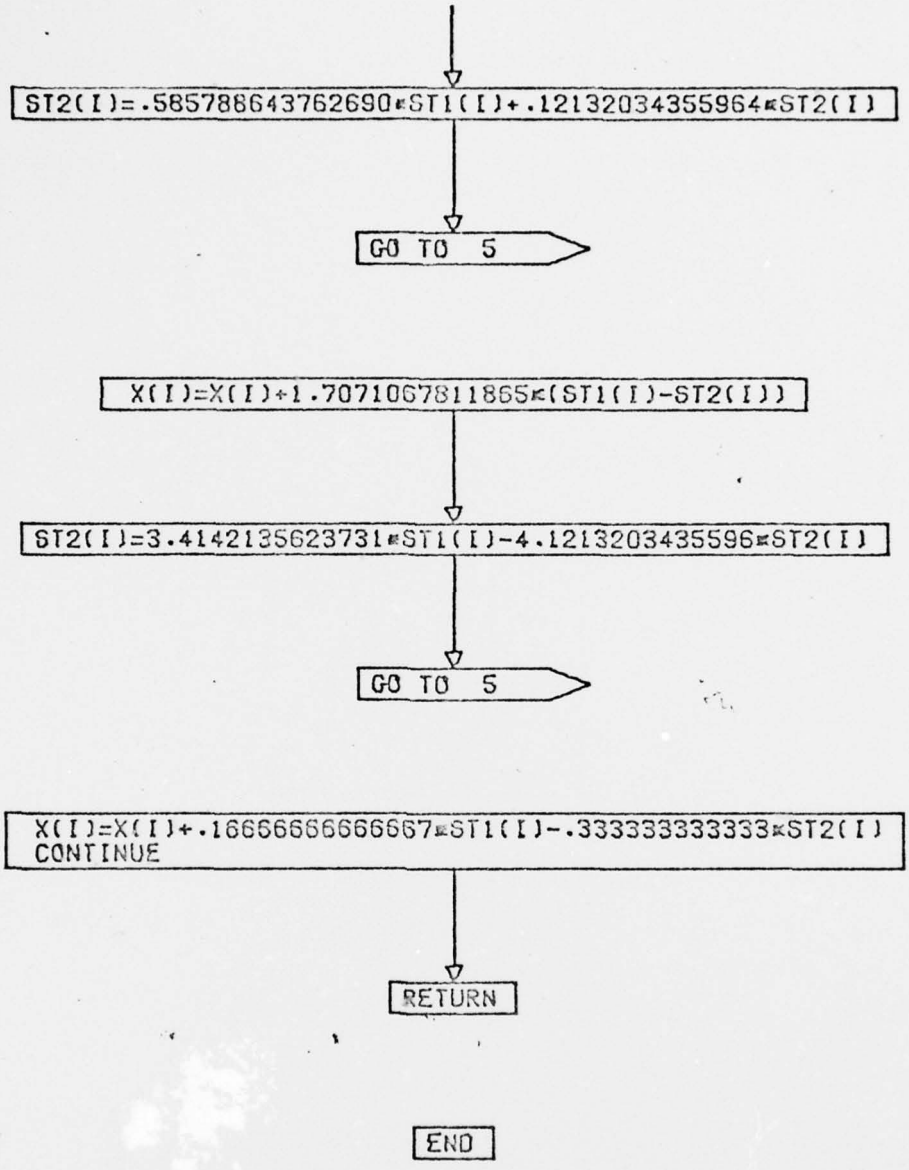

GO TO 300
CALL NREP(20,6,FMMA,1,FMMA,4)
GO TO 300
CALL NREP(21,7,FMMA,10,FMMA,13,RUCK,64)
GO TO 300
CALL NREP(22,6,FMMA,7,FMMA,16)
GO TO 300
CALL NREP(23,7,RUCK,76,RUCK,79,RUCK,67)
GO TO 300
89
CONT. ON PG 19



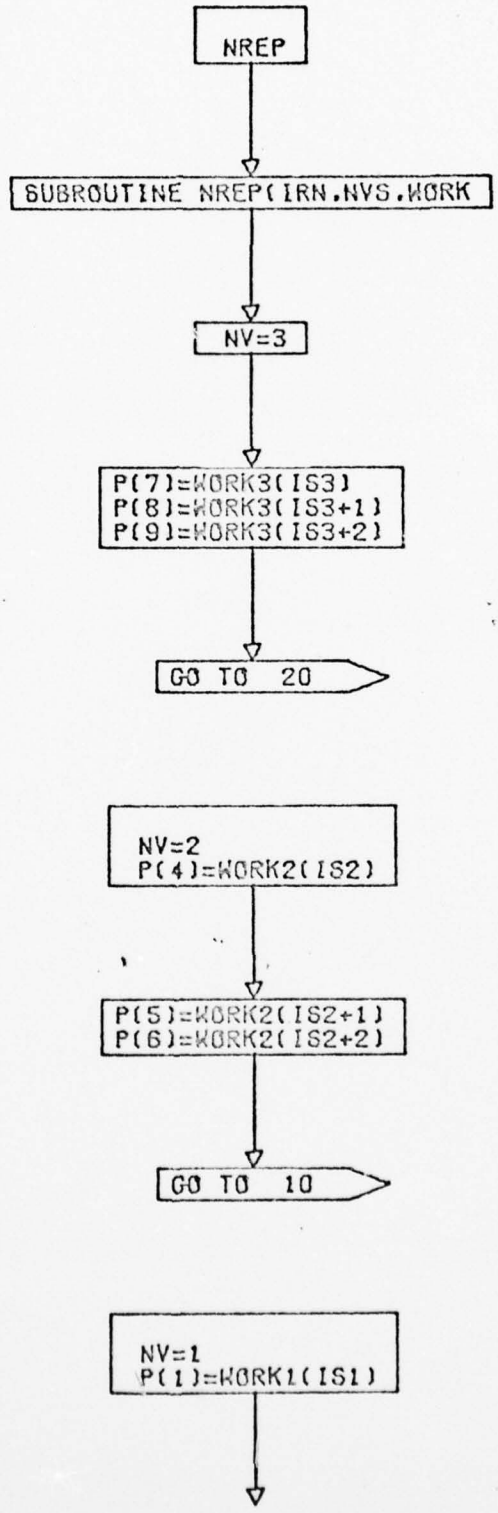
END


KUTTA
SUBROUTINE KUTTA(X,DX,ST1,ST2,NUMEQ)
IF (NPASS .EQ. 0)
F
T
RETURN
DO 5 I=1,NUMEQ
ST1(I)=DX(I)*WORK(1)
X(I)=X(I)+.5*ST1(I)
ST2(I)=ST1(I)
GO TO 5
X(I)=X(I)+.29289321881345*(ST1(I)-ST2(I))

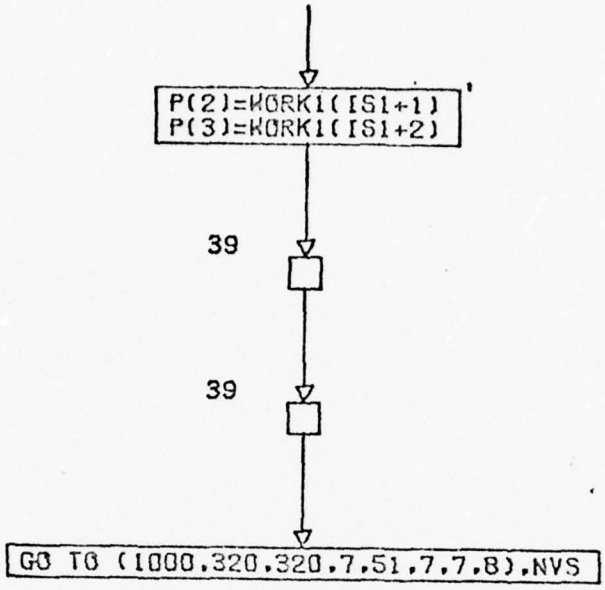
CONT. ON PG 2

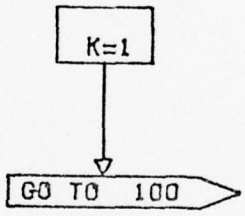

ST2(I)=.585788643762690*ST1(I)+.12132034355964*ST2(I)
GO TO 5
X(I)=X(I)+1.707106781865*(ST1(I)-ST2(I))
ST2(I)=3.4142135623731*ST1(I)-4.1213203435596*ST2(I)
GO TO 5
X(I)=X(I)+.16666666666667*ST1(I)-.33333333333*ST2(I)
CONTINUE
RETURN
END

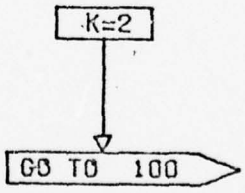
PG 2 FINAL

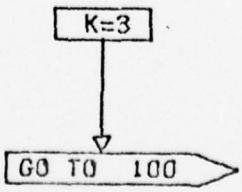

NREP
SUBROUTINE NREP(IRN,NVS,WORK
NV=3
P(7)=WORK3(IS3)
P(8)=WORK3(IS3+1)
P(9)=WORK3(IS3+2)
GO TO 20
NV=2
P(4)=WORK2(IS2)
P(5)=WORK2(IS2+1)
P(6)=WORK2(IS2+2)
GO TO 10
NV=1
P(1)=WORK1(IS1)

CONT. ON PG 2

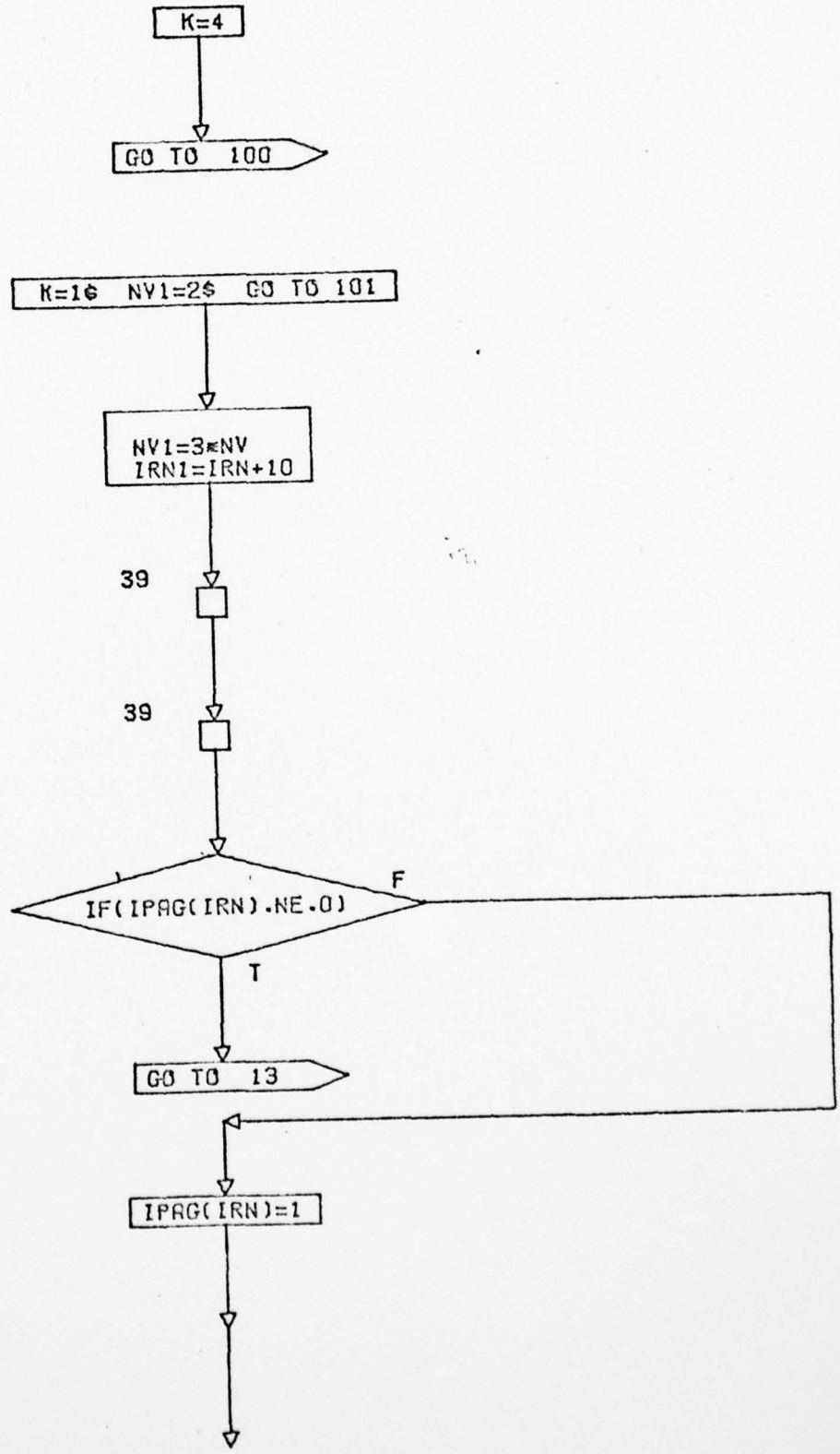

P(2)=WORK1(IS1+1)
P(3)=WORK1(IS1+2)
39
39
GO TO (1000,320,320,7,51,7,7,8),NVS


K=1
GO TO 100

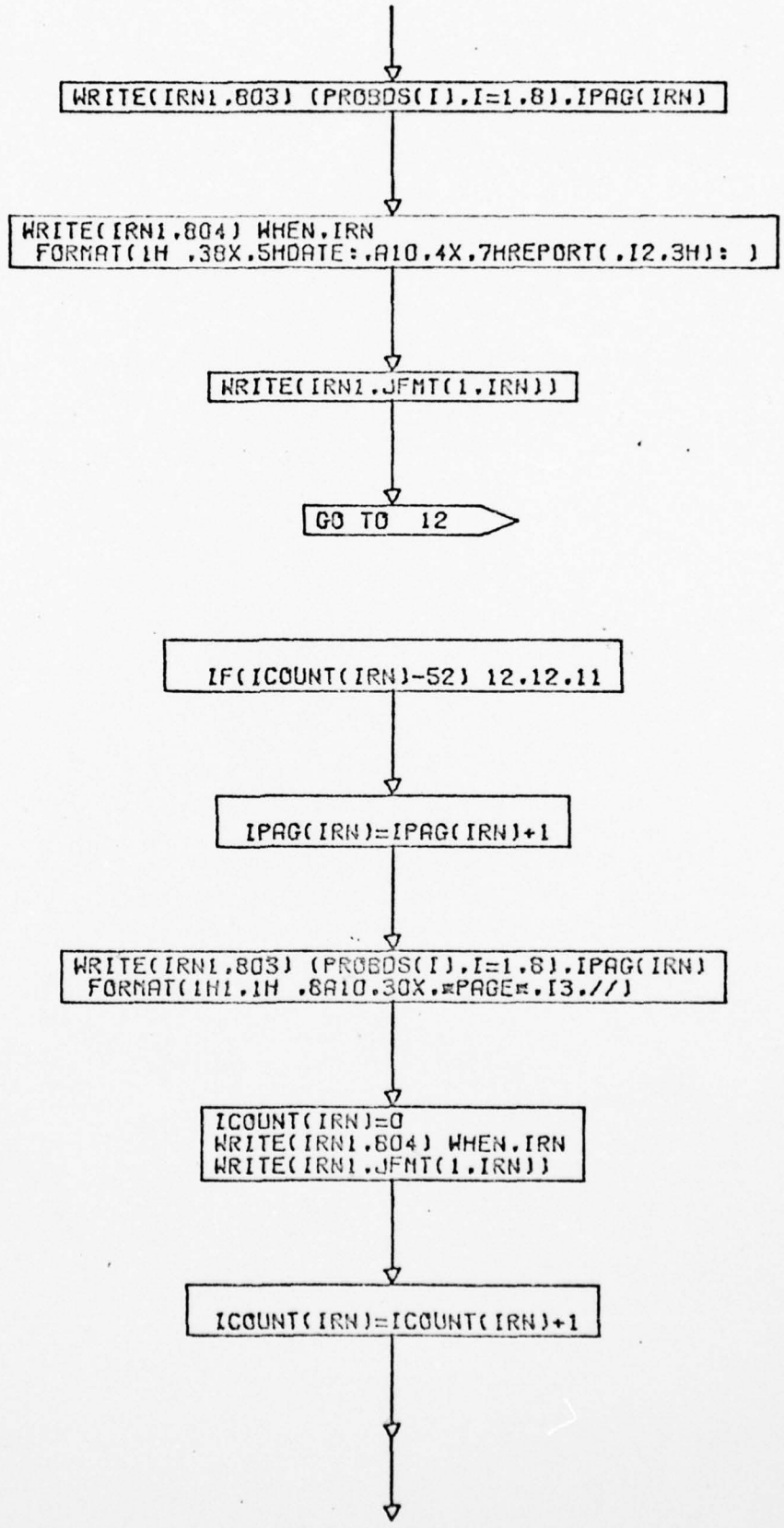

K=2
GO TO 100

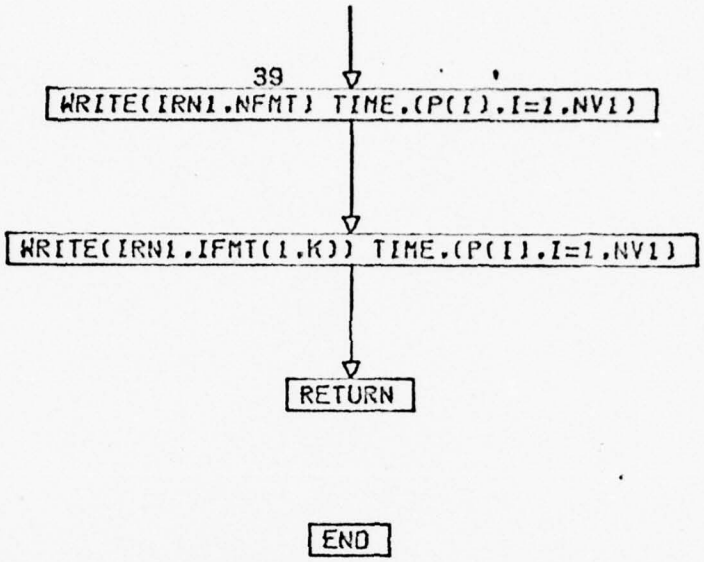

K=3
GO TO 100

CONT. ON PG 3

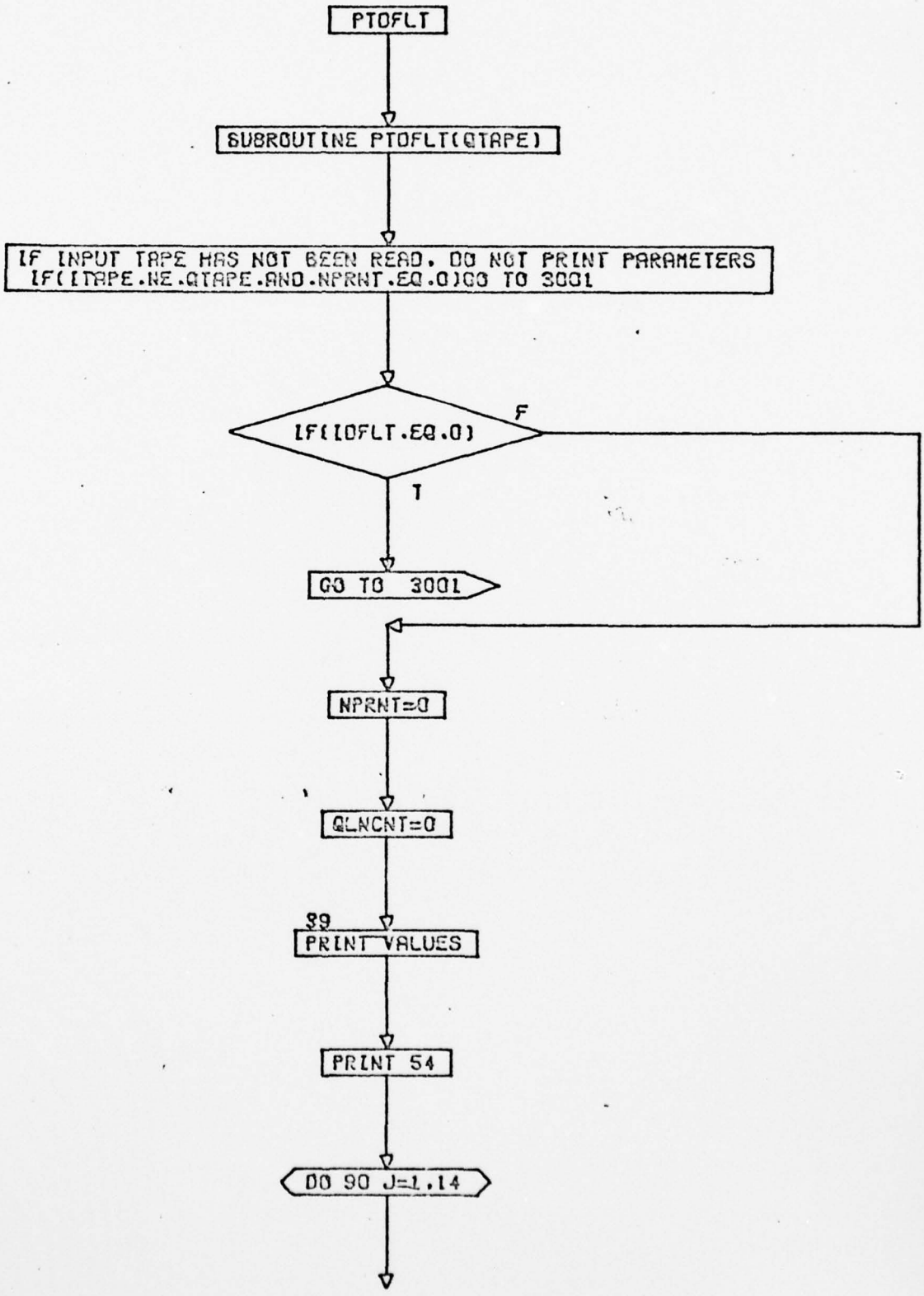

K=4
GO TO 100
K=1$ NV1=2$ GO TO 101
NV1=3*NV
IRN1=IRN+10
39
39
IF(IPAG(IRN).NE.0)
F
T
GO TO 13
IPAG(IRN)=1

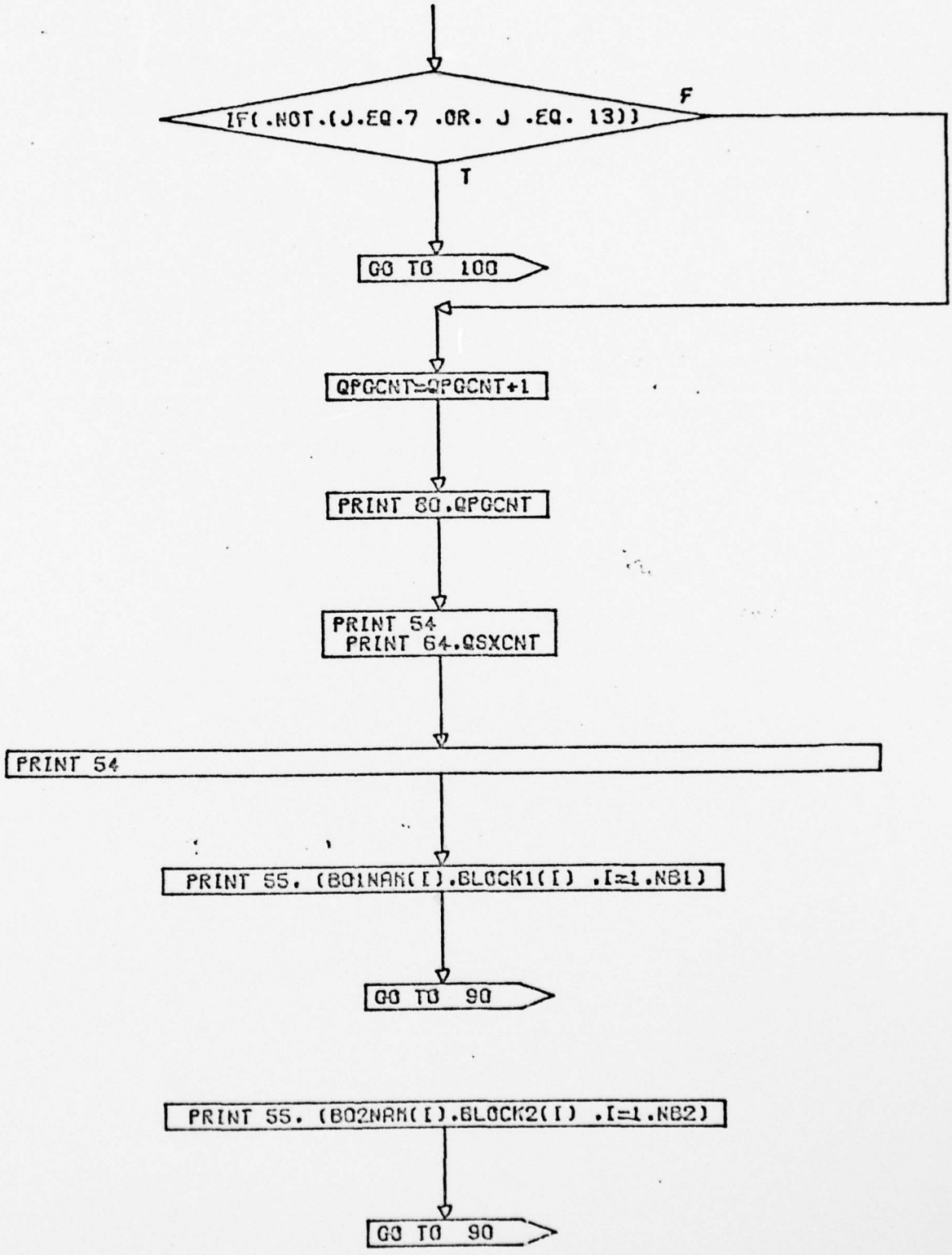
CONT. ON PG 4


WRITE(IRN1,803) (PROBDS(I),I=1,8),IPAG(IRN)
WRITE(IRN1,804) WHEN,IRN
FORMAT(1H ,38X,5HDATE:,A10,4X,7HREPORT(,I2,3H):)
WRITE(IRN1,JFMT(1,IRN))
GO TO 12
IF(ICOUNT(IRN)-52) 12,12,11
IPAG(IRN)=IPAG(IRN)+1
WRITE(IRN1,803) (PROBDS(I),I=1,8),IPAG(IRN)
FORMAT(1H1,1H ,8A10,30X,*PAGE*,I3,//)
ICOUNT(IRN)=0
WRITE(IRN1,804) WHEN,IRN
WRITE(IRN1,JFMT(1,IRN))
ICOUNT(IRN)=ICOUNT(IRN)+1
CONT. ON PG 5

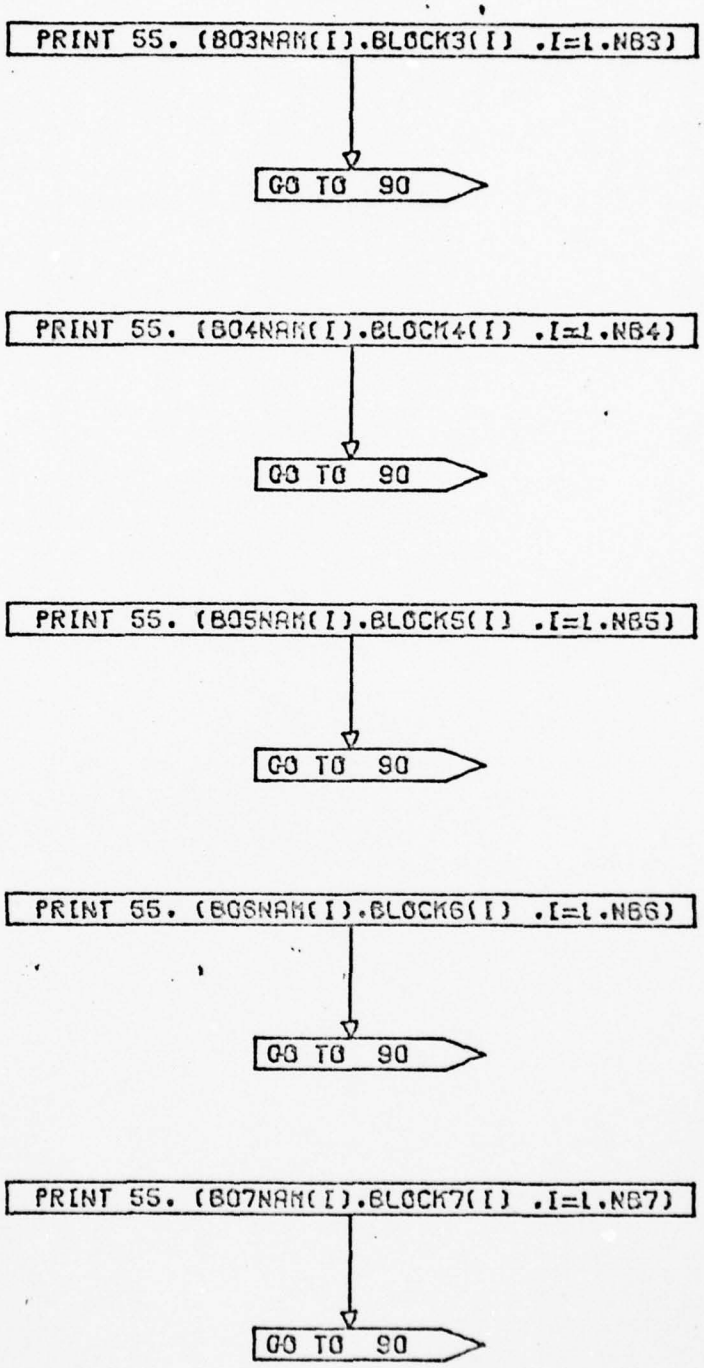

39
WRITE(IRN1,NFMT) TIME,(P(I),I=1,NV1)
WRITE(IRN1,IFMT(1,K)) TIME,(P(I),I=1,NV1)
RETURN
END

PG 5 FINAL

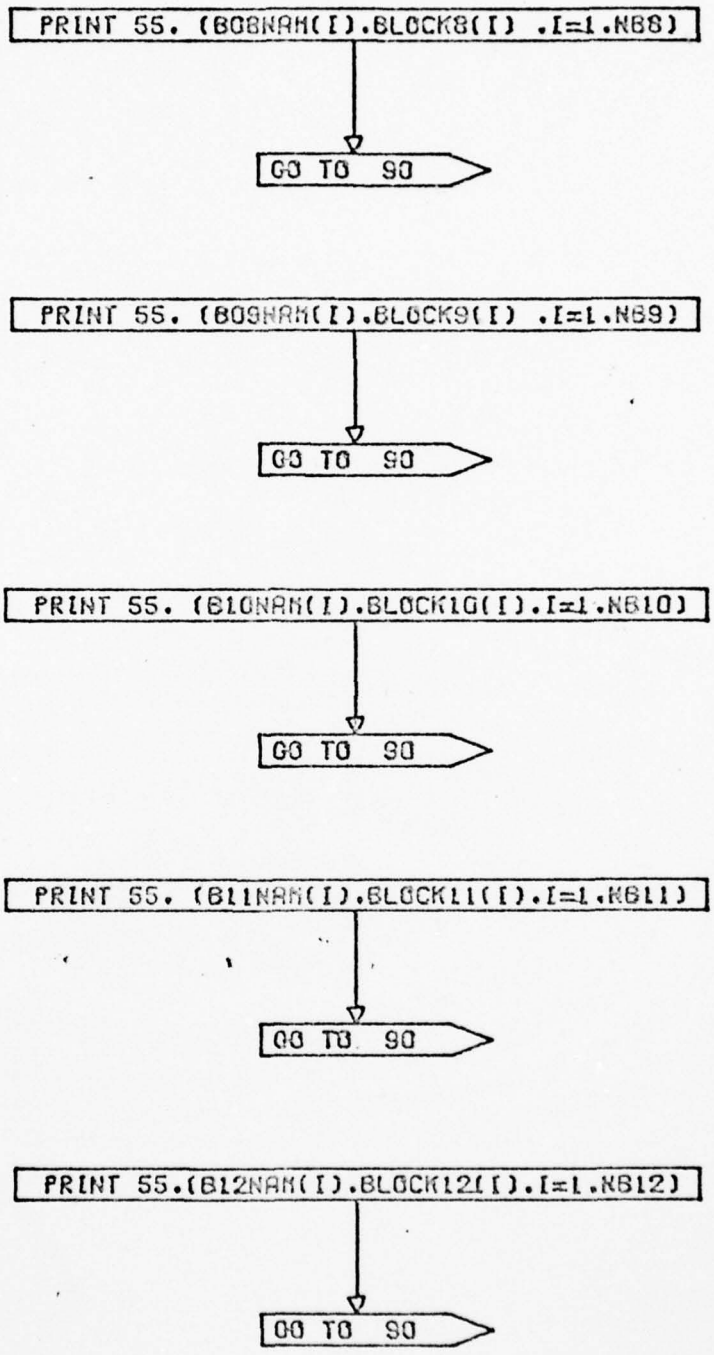

PTOFLT
SUBROUTINE PTOFLT(QTAPE)
IF INPUT TAPE HAS NOT BEEN READ, DO NOT PRINT PARAMETERS
IF(ITAPE.NE.QTAPE.AND.NPRNT.EQ.0)GO TO 3001
IF(IOFLT.EQ.0)
F
T
GO TO 3001
NPRNT=0
QLNCNT=0
99
PRINT VALUES
PRINT 54
DO 90 J=1,14
CONT. ON PG 2


IF(.NOT.(J.EQ.7 .OR. J .EQ. 13))
F
T
GO TO 100
QPGCNT=QPGCNT+1
PRINT 80,QPGCNT
PRINT 54
PRINT 64,QSXCNT
PRINT 54
PRINT 55, (BO1NAM(I),BLOCK1(I) ,I=1,NB1)
GO TO 90
PRINT 55, (BO2NAM(I),BLOCK2(I) ,I=1,NB2)
GO TO 90

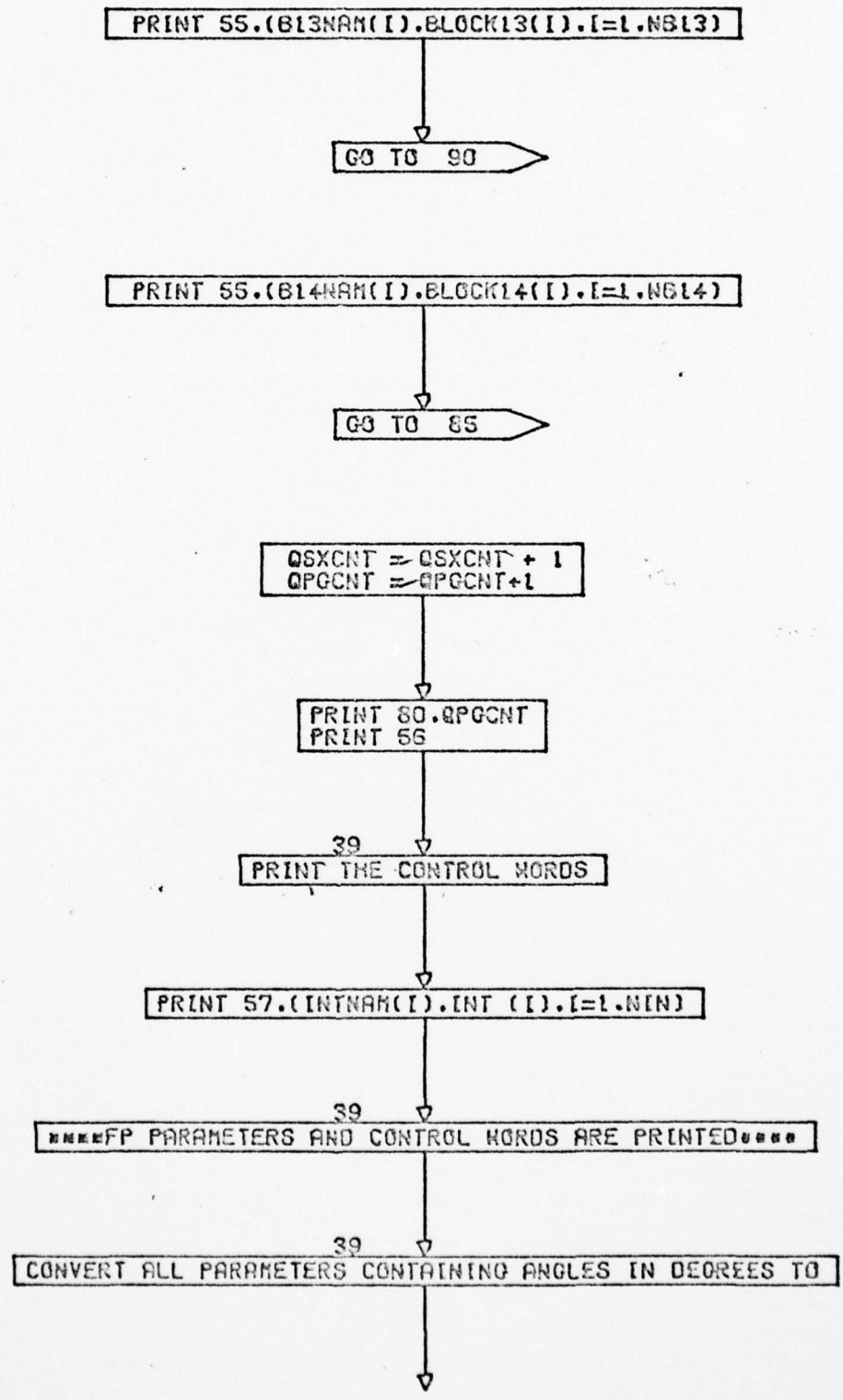
CONT. ON PG 3


PRINT 55. (B03NAM(I).BLOCK3(I) .I=1.NB3)
GO TO 90
PRINT 55. (B04NAM(I).BLOCK4(I) .I=1.NB4)
GO TO 90
PRINT 55. (B05NAM(I).BLOCK5(I) .I=1.NB5)
GO TO 90
PRINT 55. (B06NAM(I).BLOCK6(I) .I=1.NB6)
GO TO 90
PRINT 55. (B07NAM(I).BLOCK7(I) .I=1.NB7)
GO TO 90

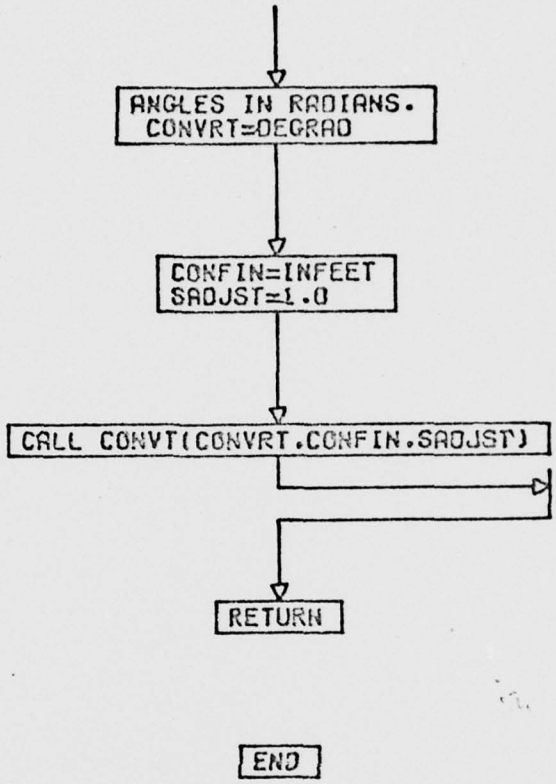
CONT. ON PG 4

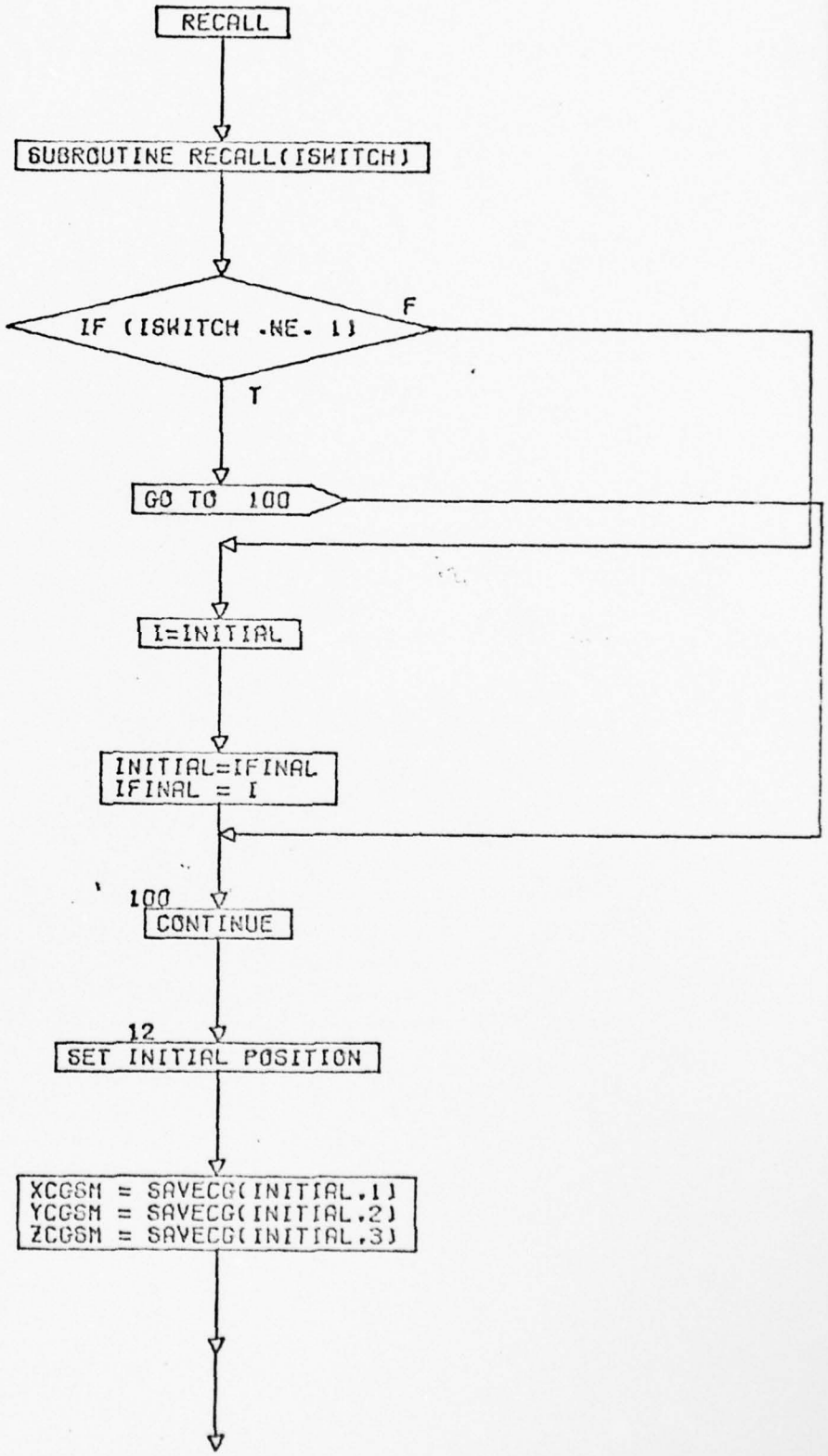

PRINT 55. (B08NAM(I).BLOCK8(I) .I=1.N88)
GO TO 90
PRINT 55. (B09NAM(I).BLOCK9(I) .I=1.N89)
GO TO 90
PRINT 55. (B10NAM(I).BLOCK10(I).I=1.NB10)
GO TO 90
PRINT 55. (B11NAM(I).BLOCK11(I).I=1.NB11)
GO TO 90
PRINT 55.(B12NAM(I).BLOCK12(I).I=1.NB12)
GO TO 90

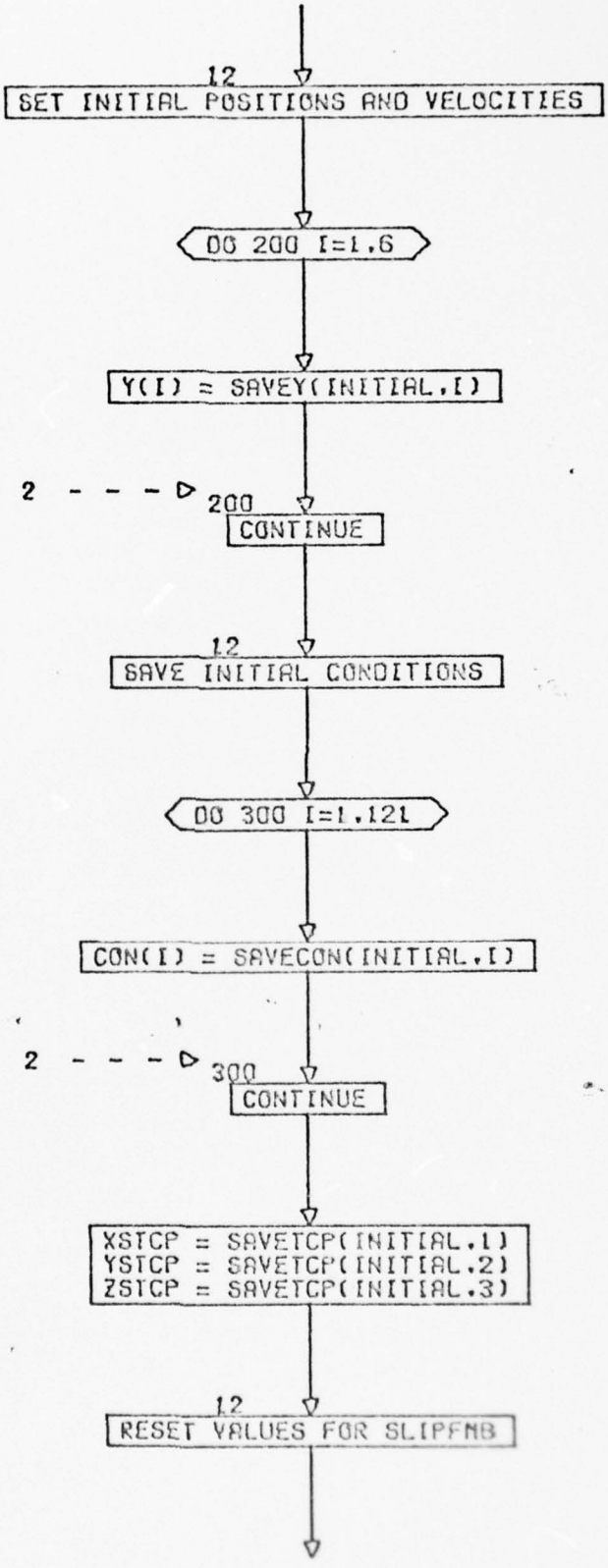
CONT. ON PG 5


PRINT 55.(B13NAM(I).BLOCK13(I).I=1.NB13)
GO TO 90
PRINT 55.(B14NAM(I).BLOCK14(I).I=1.NB14)
GO TO 85
QSXCNT = QSXCNT + 1
QPGCNT = QPGCNT+1
PRINT 80.QPGCNT
PRINT 56
39
PRINT THE CONTROL WORDS
PRINT 57.(INTNAM(I).INT (I).I=1.NIN)
39
****FP PARAMETERS AND CONTROL WORDS ARE PRINTED****
39
CONVERT ALL PARAMETERS CONTAINING ANGLES IN DEGREES TO

CONT. ON PG 6


ANGLES IN RADIANS.
CONVRT=DEGRAD
CONFIN=INFEET
SADJST=1.0
CALL CONVT(CONVRT,CONFIN,SADJST)
RETURN
END

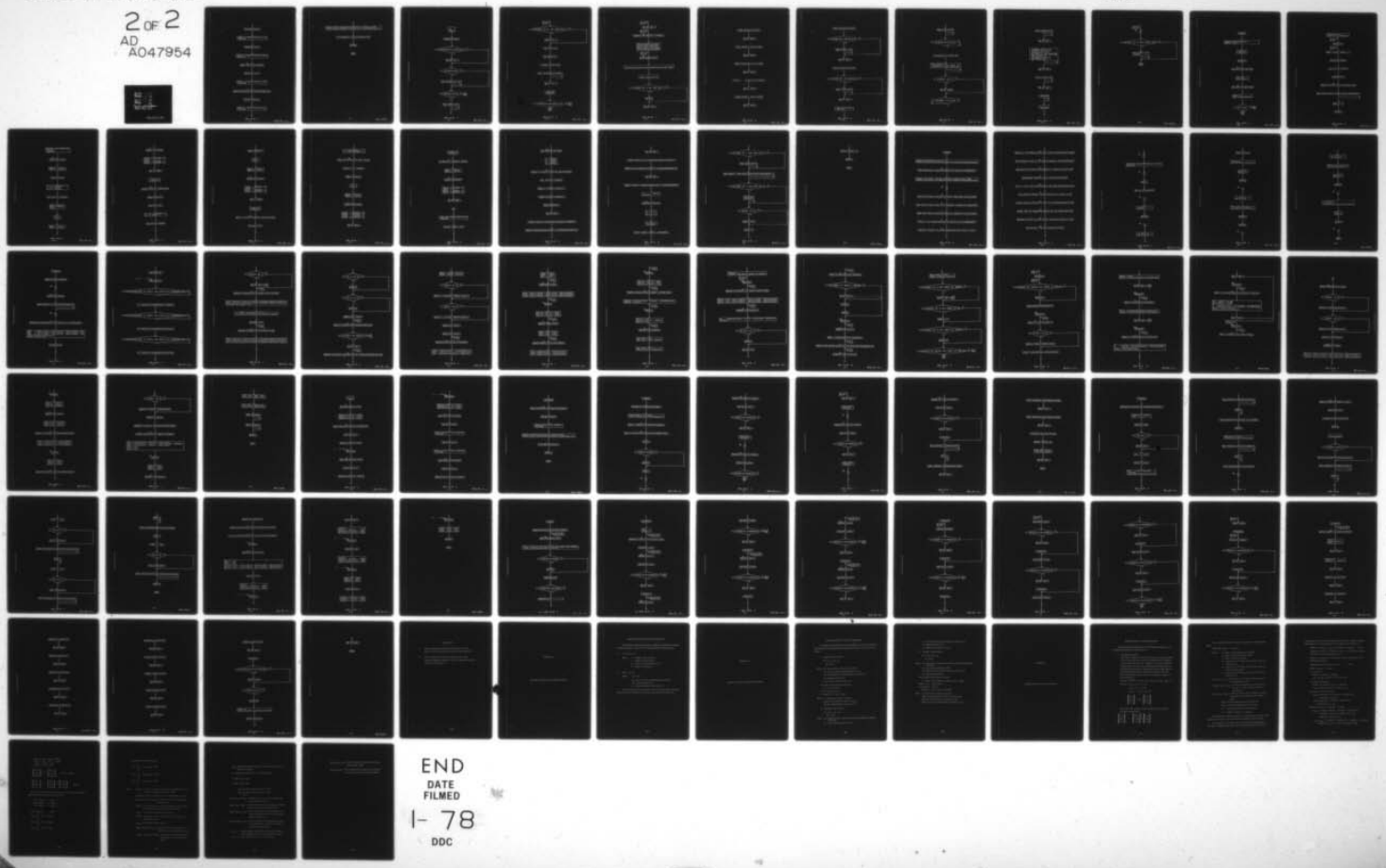

RECALL
SUBROUTINE RECALL(ISWITCH)
IF (ISWITCH .NE. 1)
F
T
GO TO 100
I=INITIAL
INITIAL=IFINAL
IFINAL = I
100
CONTINUE
12
SET INITIAL POSITION
XCGSM = SAVECG(INITIAL,1)
YCGSM = SAVECG(INITIAL,2)
ZCGSM = SAVECG(INITIAL,3)
CONT. ON PG 2


12
SET INITIAL POSITIONS AND VELOCITIES
DO 200 I=1,6
Y(I) = SAVEY(INITIAL,I)
2
200
CONTINUE
12
SAVE INITIAL CONDITIONS
DO 300 I=1,121
CON(I) = SAVECON(INITIAL,I)
2
300
CONTINUE
XSTCP = SAVETCP(INITIAL,1)
YSTCP = SAVETCP(INITIAL,2)
ZSTCP = SAVETCP(INITIAL,3)
12
RESET VALUES FOR SLIPFNB

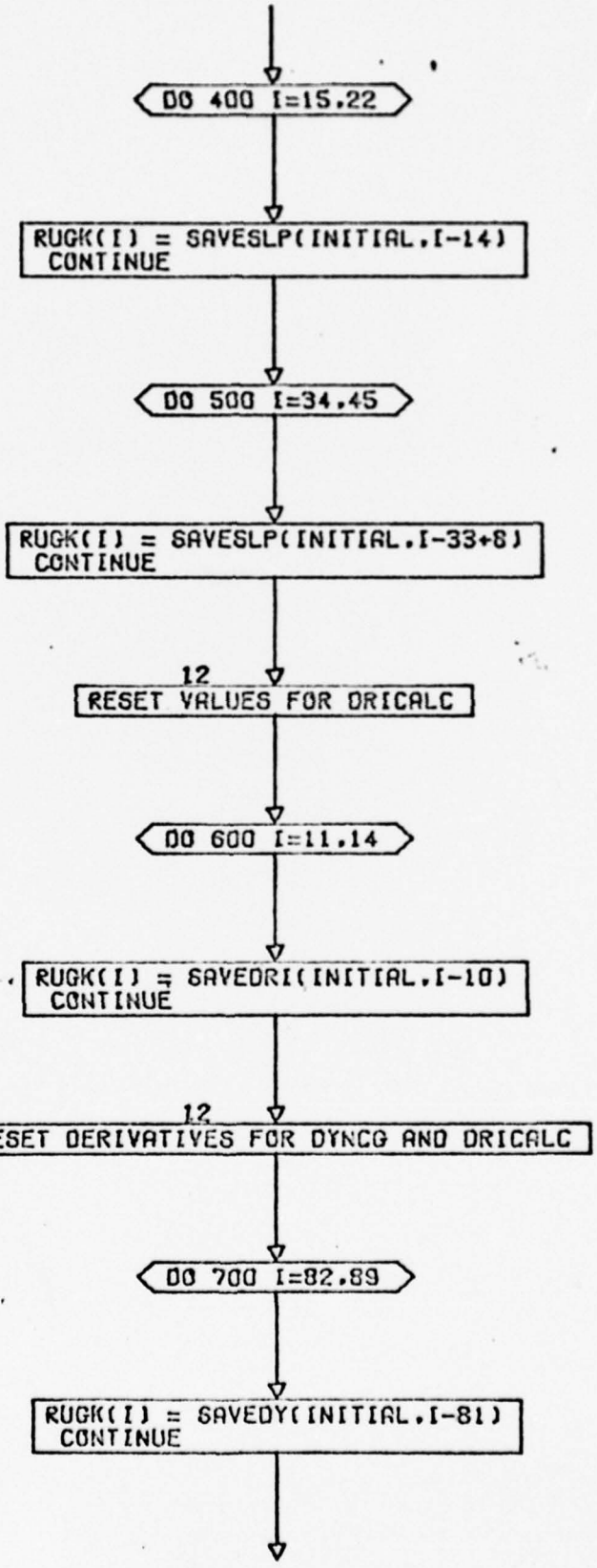
CONT. ON PG 3

AD-A047 954 COMPUTER SCIENCES CORP HUNTINGDON VALLEY PA F/G 1/3
F-18 ESCAPE SYSTEM COMPUTER SUPPORT.(U)
OCT 77
N62269-75-C-0001
UNCLASSIFIED NL

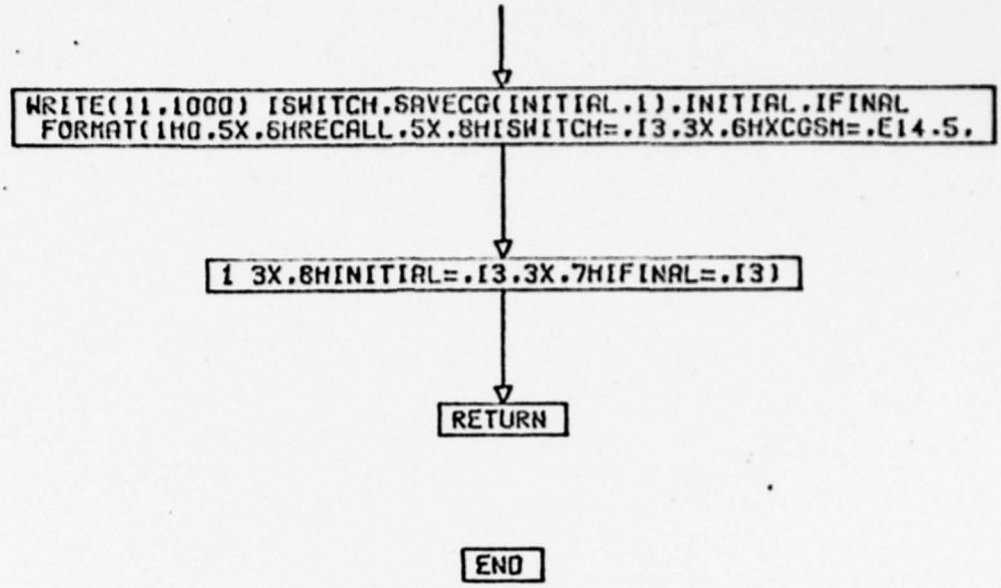
2 OF 2
AD
A047954



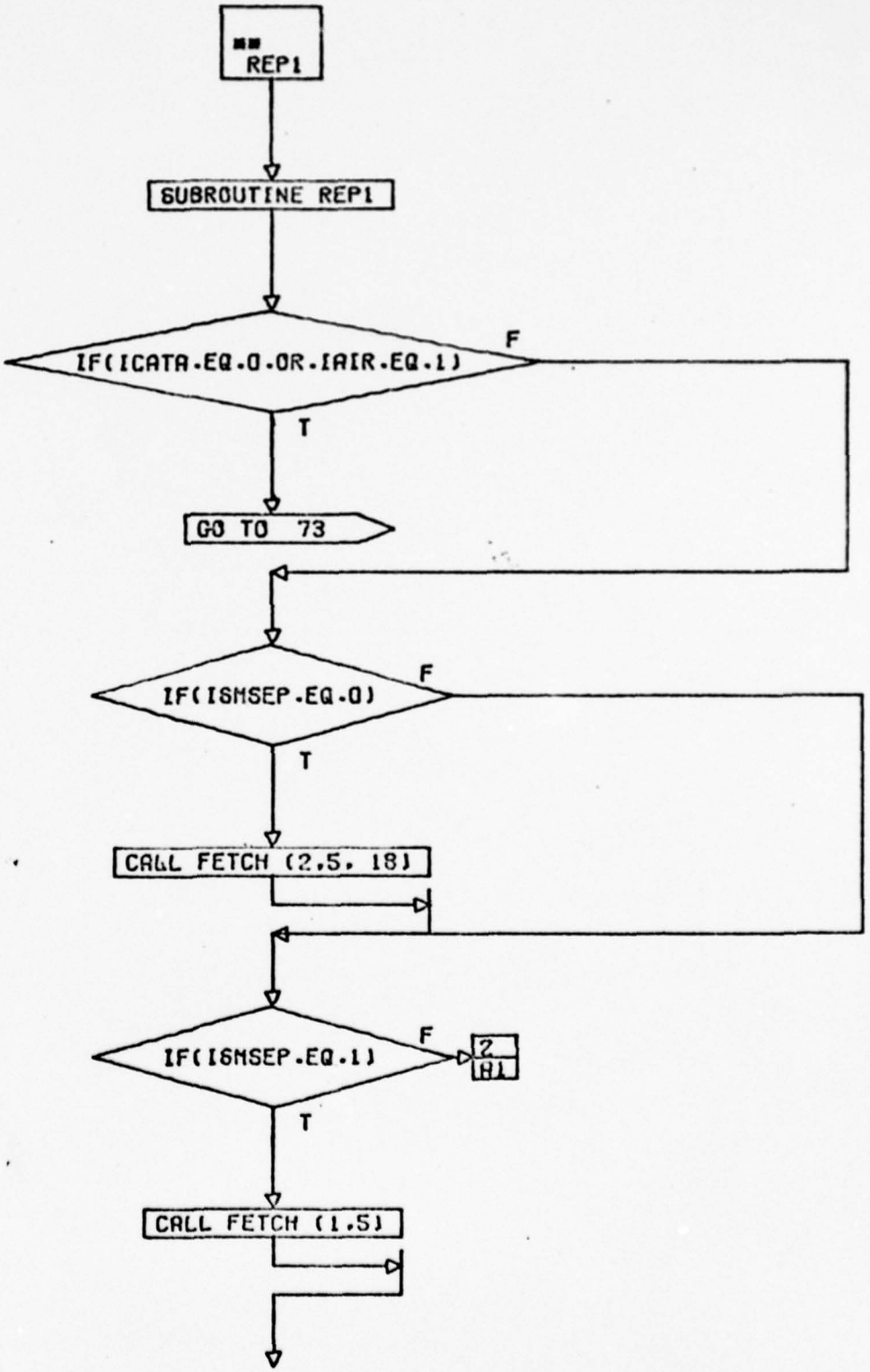
END
DATE FILMED
1-78
DDC


DO 400 I=15,22
RUGK(I) = SAVESLP(INITIAL,I-14)
CONTINUE
DO 500 I=34,45
RUGK(I) = SAVESLP(INITIAL,I-33+8)
CONTINUE
12
RESET VALUES FOR DRICALC
DO 600 I=11,14
RUGK(I) = SAVEDRI(INITIAL,I-10)
CONTINUE
12
RESET DERIVATIVES FOR DYNCG AND DRICALC
DO 700 I=82,89
RUGK(I) = SAVEDY(INITIAL,I-81)
CONTINUE

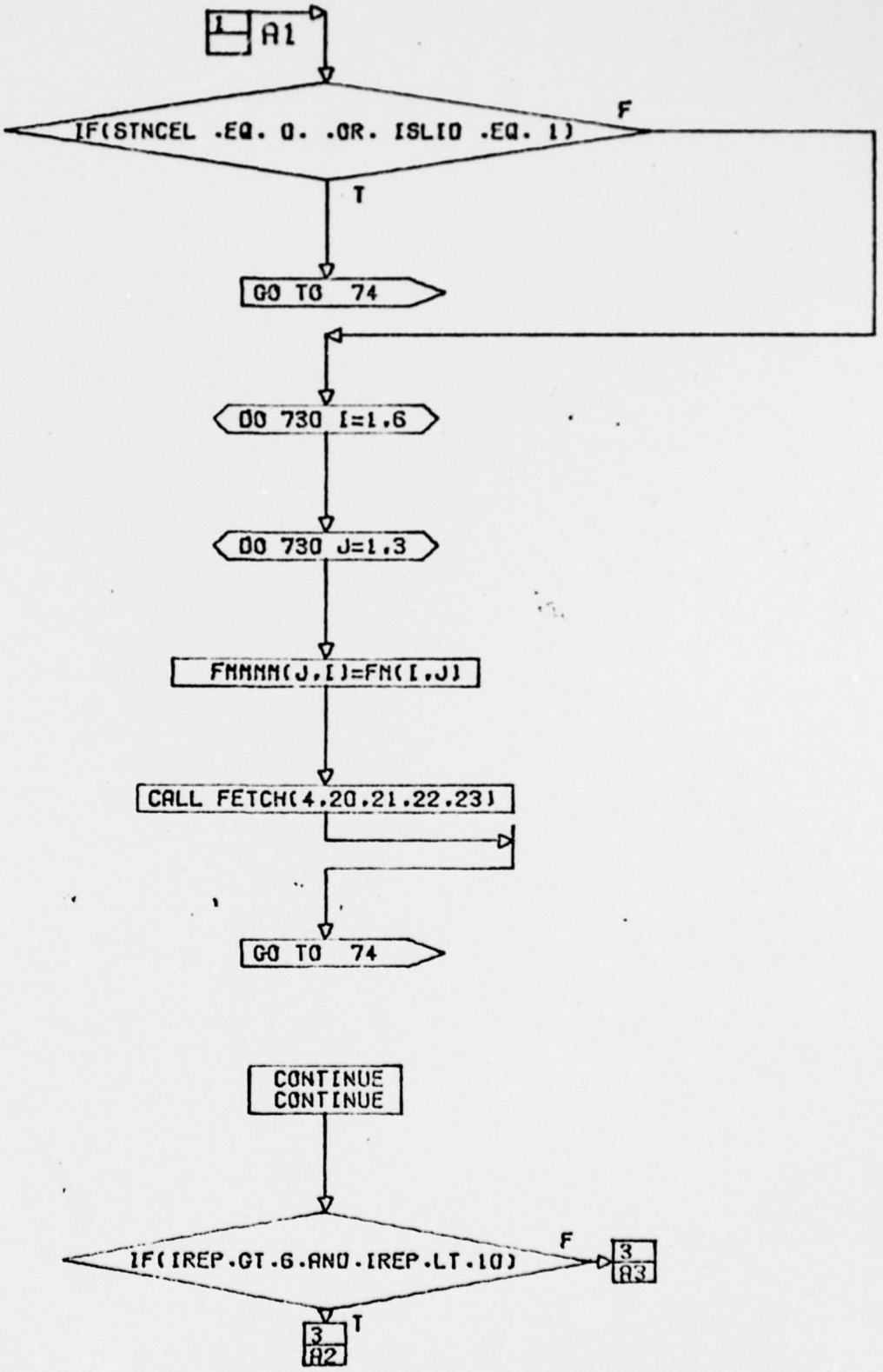
CONT. ON PG 4


WRITE(11,1000) ISWITCH,SAVECG(INITIAL,1),INITIAL,IFINAL
FORMAT(1H0,5X,6HRECALL,5X,8HISWITCH=,I3,3X,6HXCGSM=,E14.5,
1 3X,8HINITIAL=,I3,3X,7HIFINAL=,I3)
RETURN
END

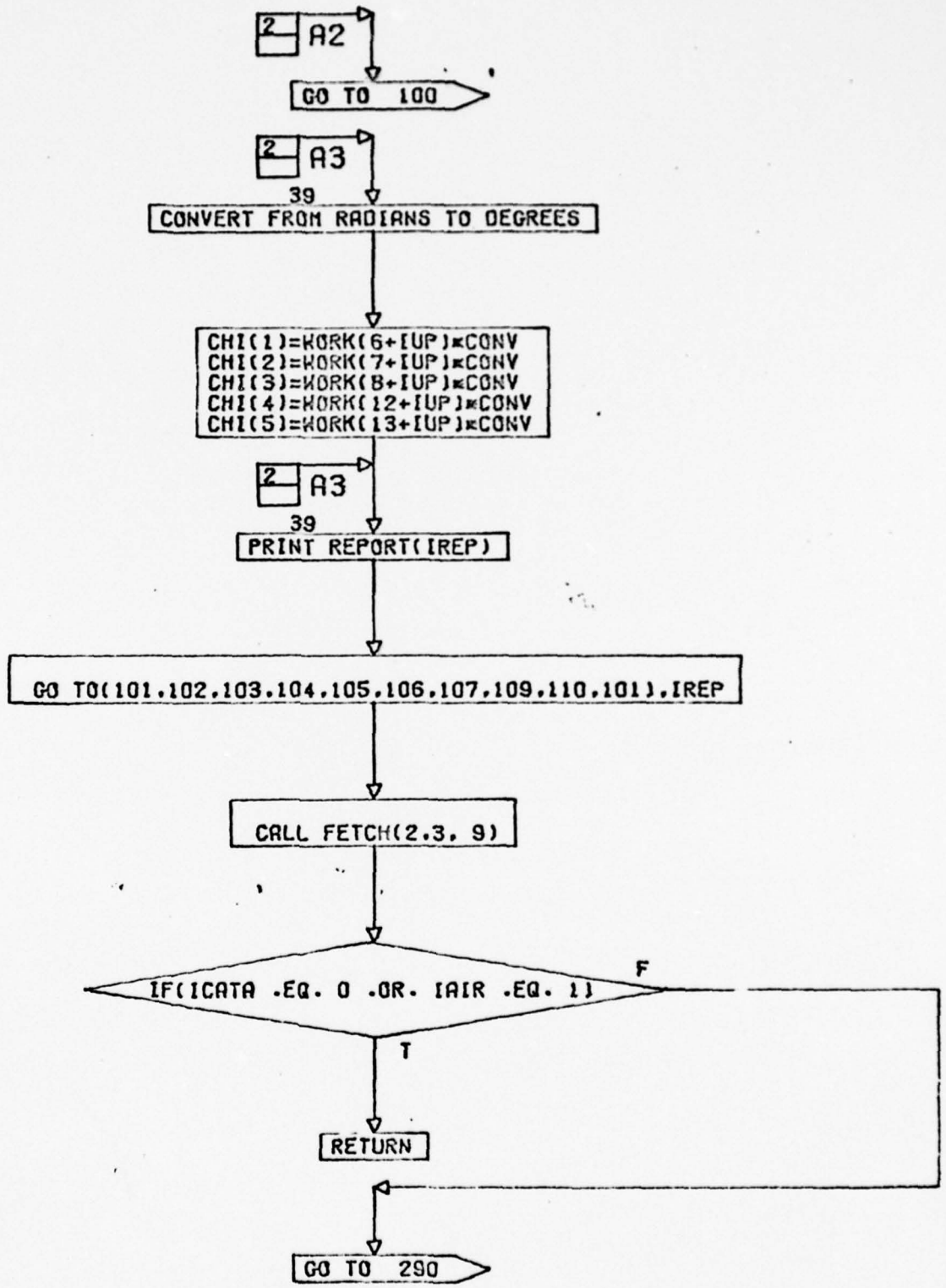
PG 4 FINAL


REP1
SUBROUTINE REP1
IF(ICATA.EQ.0.OR.IAIR.EQ.1)
F
T
GO TO 73
IF(ISMSEP.EQ.0)
F
T
CALL FETCH (2,5, 18)
IF(ISMSEP.EQ.1)
F
2
A1
T
CALL FETCH (1,5)

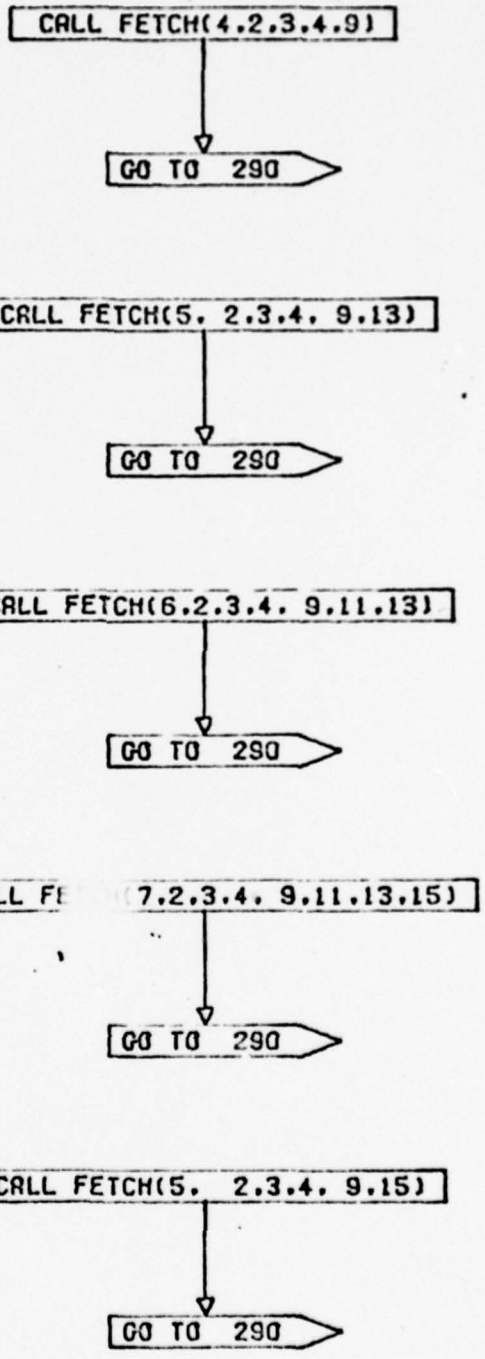
CONT. ON PG 2


1
A1
IF(STNCEL .EQ. 0. .OR. ISLID .EQ. 1)
F
T
GO TO 74
DO 730 I=1,6
DO 730 J=1,3
FMMMM(J,I)=FM(I,J)
CALL FETCH(4,20,21,22,23)
GO TO 74
CONTINUE
CONTINUE
IF(IREP.GT.6.AND.IREP.LT.10)
F
3
A3
T
3
A2

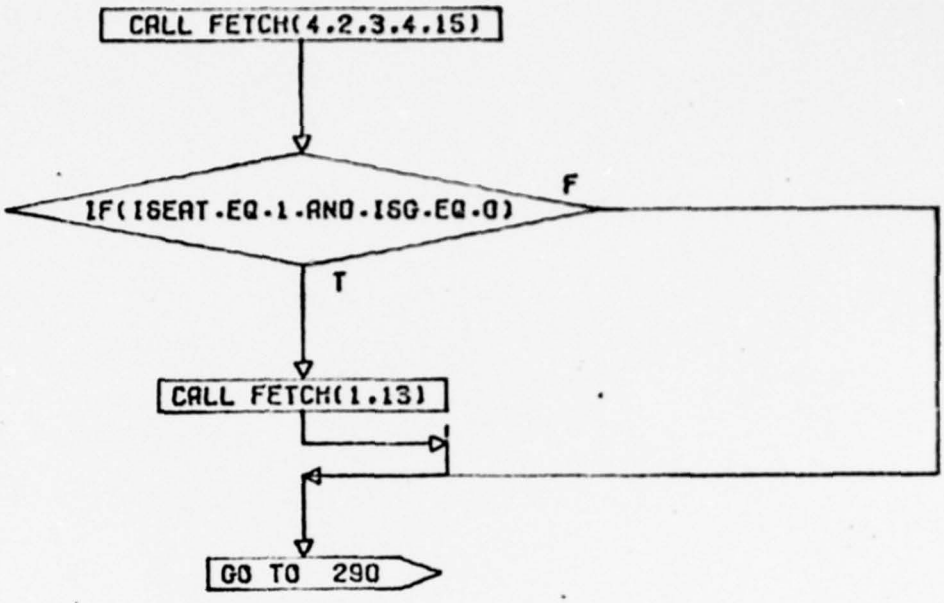
CONT. ON PG 3

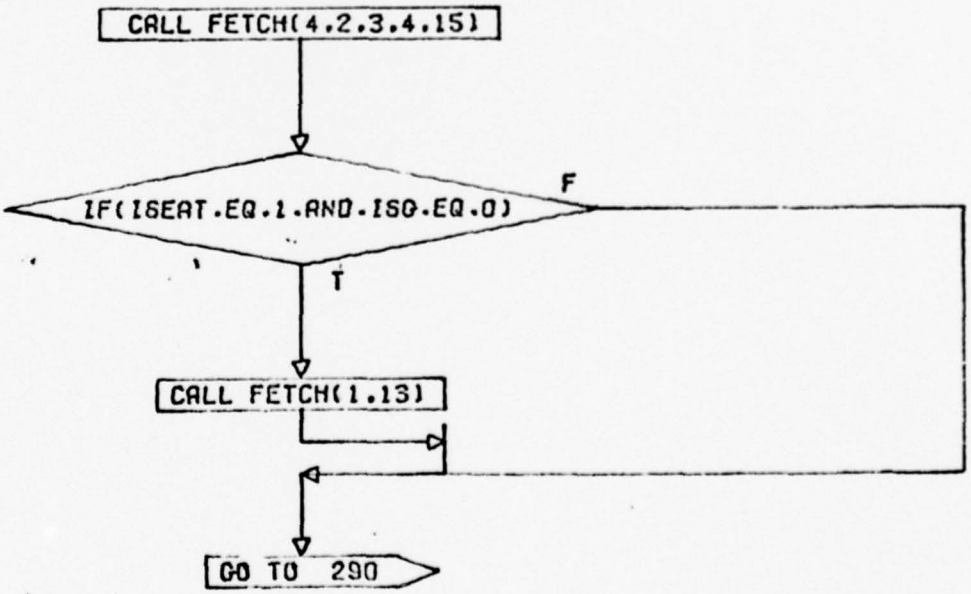

2
A2
GO TO 100
2
A3
39
CONVERT FROM RADIANS TO DEGREES
CHI(1)=WORK(6+IUP)*CONV
CHI(2)=WORK(7+IUP)*CONV
CHI(3)=WORK(8+IUP)*CONV
CHI(4)=WORK(12+IUP)*CONV
CHI(5)=WORK(13+IUP)*CONV
2
A3
39
PRINT REPORT(IREP)
GO TO(101,102,103,104,105,106,107,109,110,101),IREP
CALL FETCH(2.3, 9)
IF(ICATA .EQ. 0 .OR. IAIR .EQ. 1)
F
T
RETURN
GO TO 290

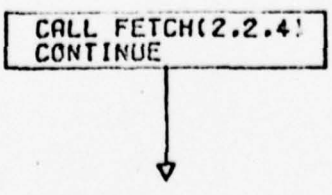
CONT. ON PG 4


CALL FETCH(4,2,3,4,9)
GO TO 290
CALL FETCH(5, 2,3,4, 9,13)
GO TO 290
CALL FETCH(6,2,3,4, 9,11,13)
GO TO 290
CALL FE 7,2,3,4, 9,11,13,15)
GO TO 290
CALL FETCH(5, 2,3,4, 9,15)
GO TO 290

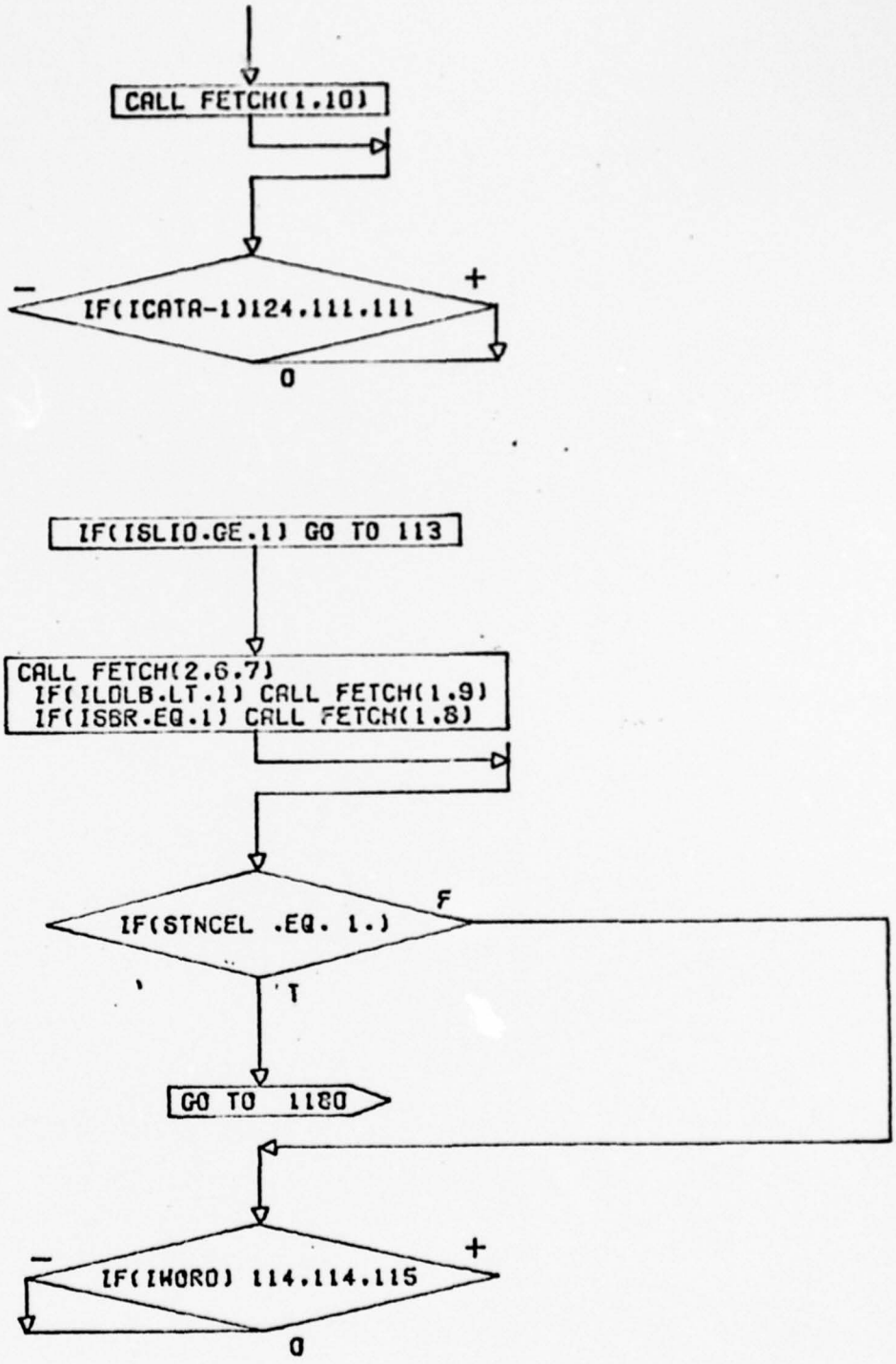
CONT. ON PG 5


CALL FETCH(4,2,3,4,15)
IF(ISEAT.EQ.1.AND.ISO.EQ.0)
F
T
CALL FETCH(1,13)
GO TO 290

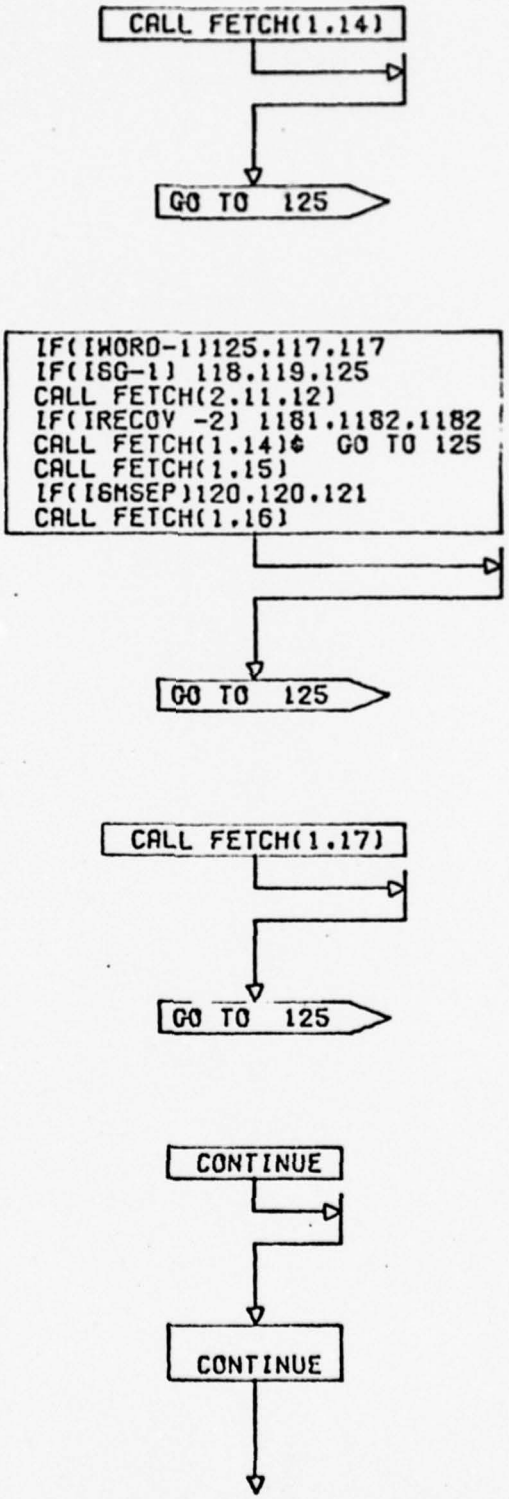

CALL FETCH(4,2,3,4,15)
IF(ISEAT.EQ.1.AND.ISO.EQ.0)
F
T
CALL FETCH(1,13)
GO TO 290


CALL FETCH(2,2,4)
CONTINUE

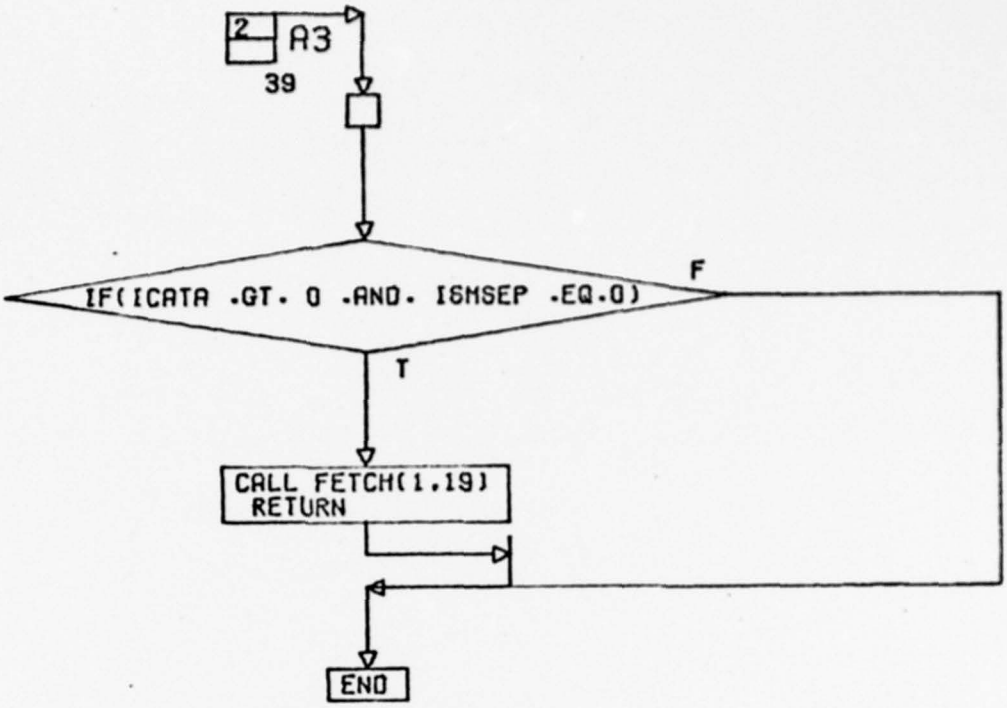
CONT. ON PG 6


CALL FETCH(1,10)
IF(ICATA-1)124,111,111
-
+
0
IF(ISLIO.GE.1) GO TO 113
CALL FETCH(2,6,7)
IF(ILOLB.LT.1) CALL FETCH(1,9)
IF(ISBR.EQ.1) CALL FETCH(1,8)
IF(STNCEL .EQ. 1.)
F
T
GO TO 1180
IF(IWORD) 114,114,115
-
+
0

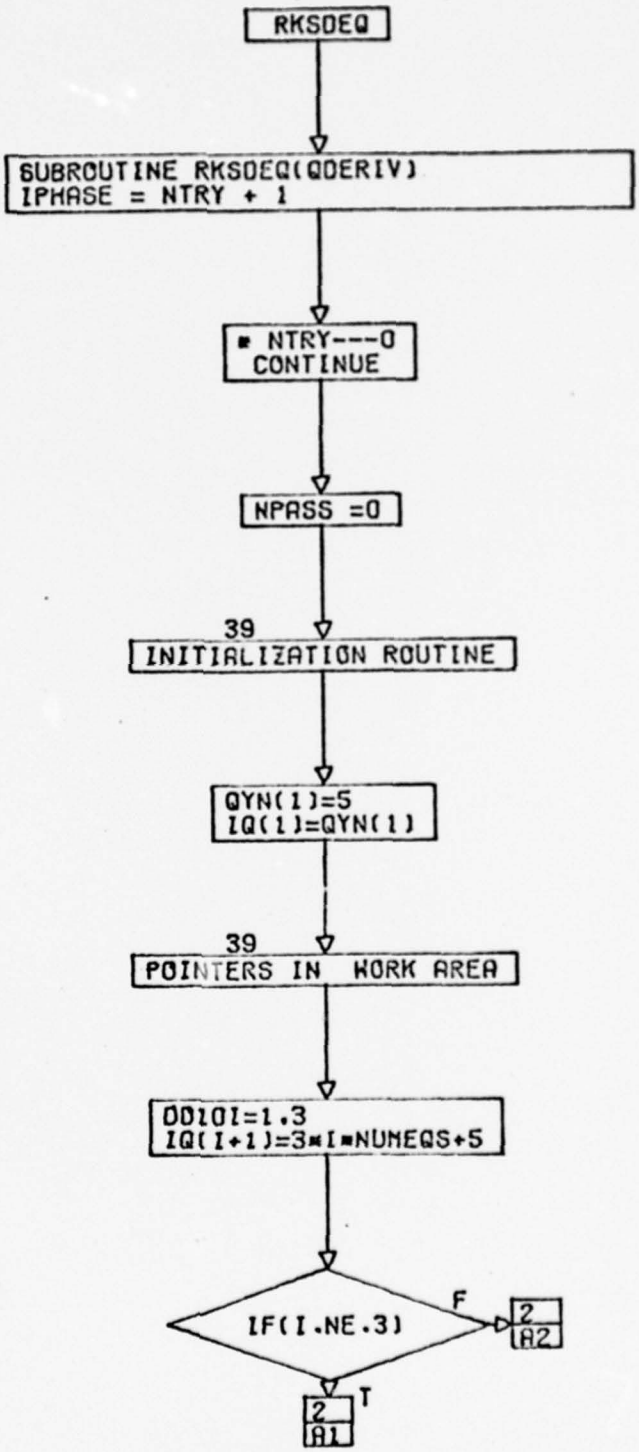
CONT. ON PG 7


CALL FETCH(1,14)
GO TO 125
IF(IWORD-1)125,117,117
IF(ISG-1) 118,119,125
CALL FETCH(2,11,12)
IF(IRECOV -2) 1181,1182,1182
CALL FETCH(1,14)$ GO TO 125
CALL FETCH(1,15)
IF(ISMSEP)120,120,121
CALL FETCH(1,16)
GO TO 125
CALL FETCH(1,17)
GO TO 125
CONTINUE
CONTINUE

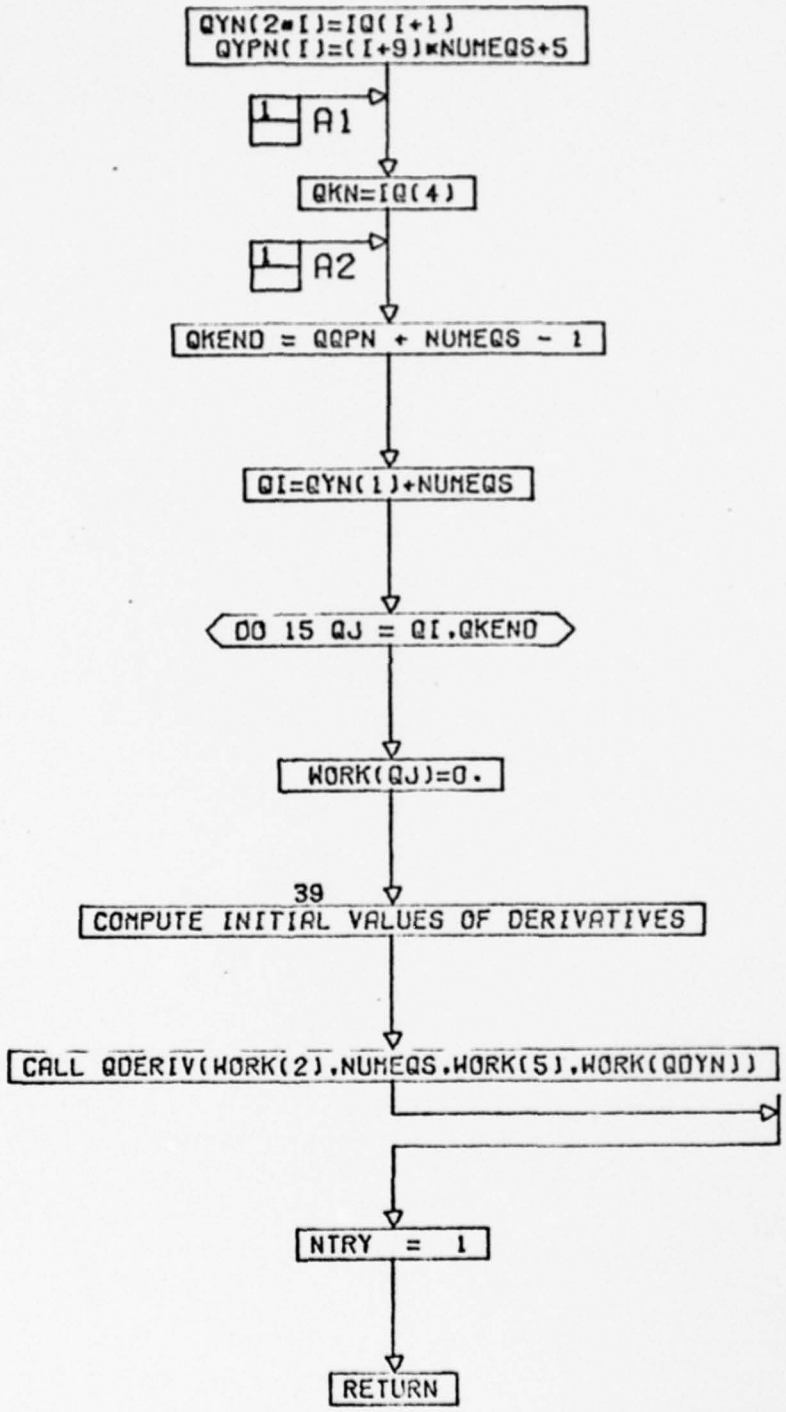
CONT. ON PG 8


2
A3
39
IF(ICATA .GT. 0 .AND. I6MSEP .EQ.0)
F
T
CALL FETCH(1,19)
RETURN
END

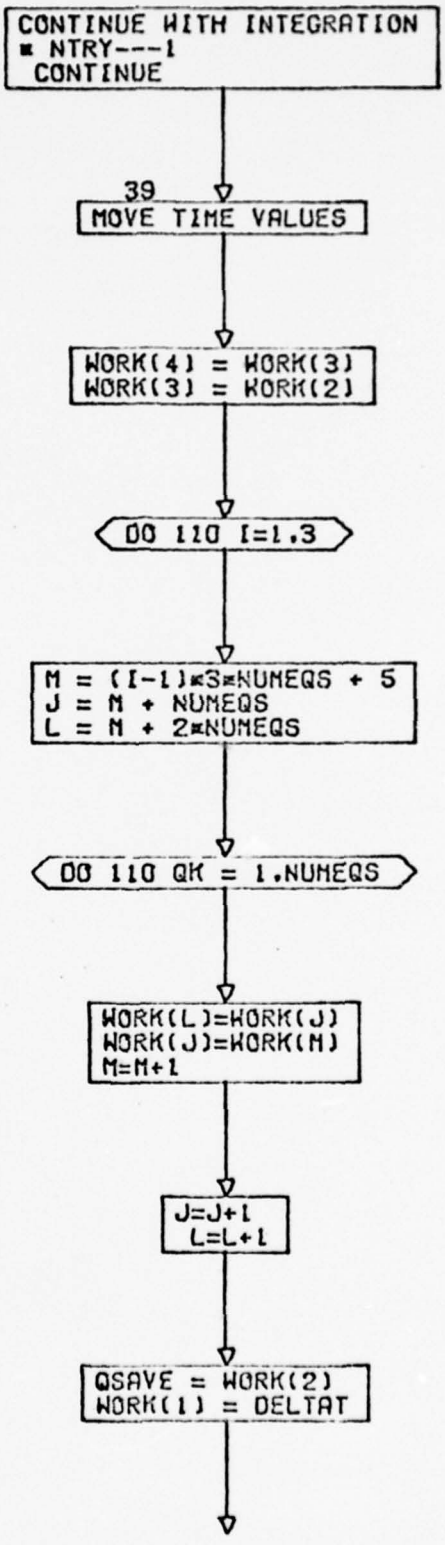
PQ B FINAL


RKSDEQ
SUBROUTINE RKSDEQ(QDERIV)
IPHASE = NTRY + 1
* NTRY---0
CONTINUE
NPASS =0
39
INITIALIZATION ROUTINE
QYN(1)=5
IQ(1)=QYN(1)
39
POINTERS IN WORK AREA
DO101=1,3
IQ(I+1)=3*I*NUMEQS+5
IF(I.NE.3)
F
2
A2
T
2
A1

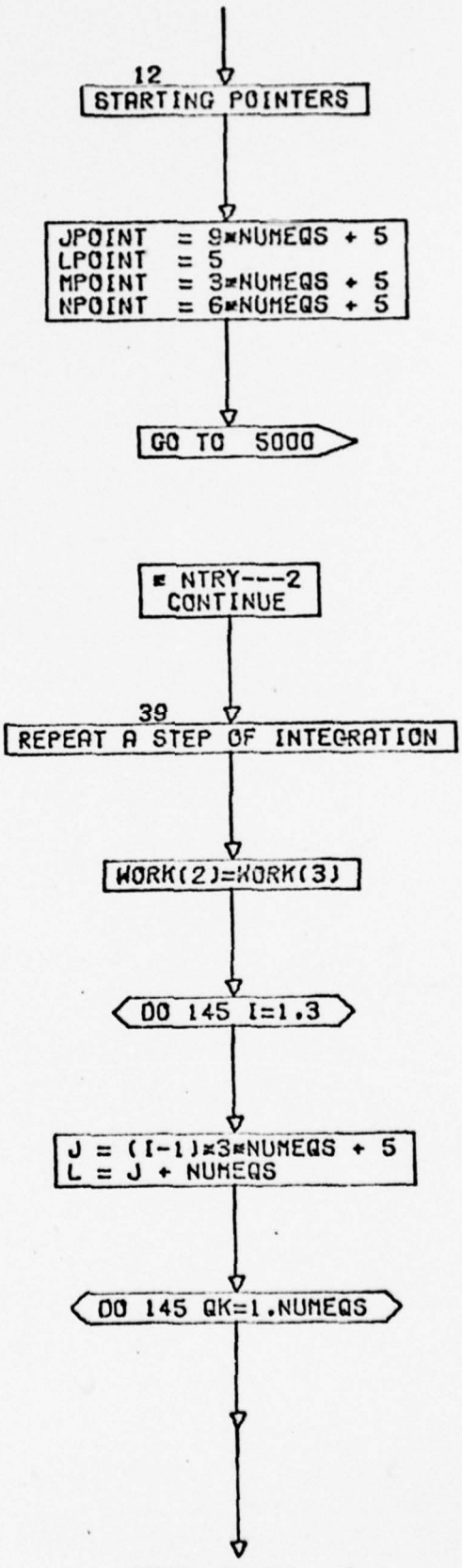
CONT. ON PG 2


QYN(2*I)=IQ(I+1)
QYPN(I)=(I+9)*NUMEQS+5
A1
QKN=IQ(4)
A2
QKEND = QQPN + NUMEQS - 1
QI=QYN(1)+NUMEQS
DO 15 QJ = QI,QKEND
WORK(QJ)=0.
39
COMPUTE INITIAL VALUES OF DERIVATIVES
CALL QDERIV(WORK(2),NUMEQS,WORK(5),WORK(QQYN))
NTRY = 1
RETURN

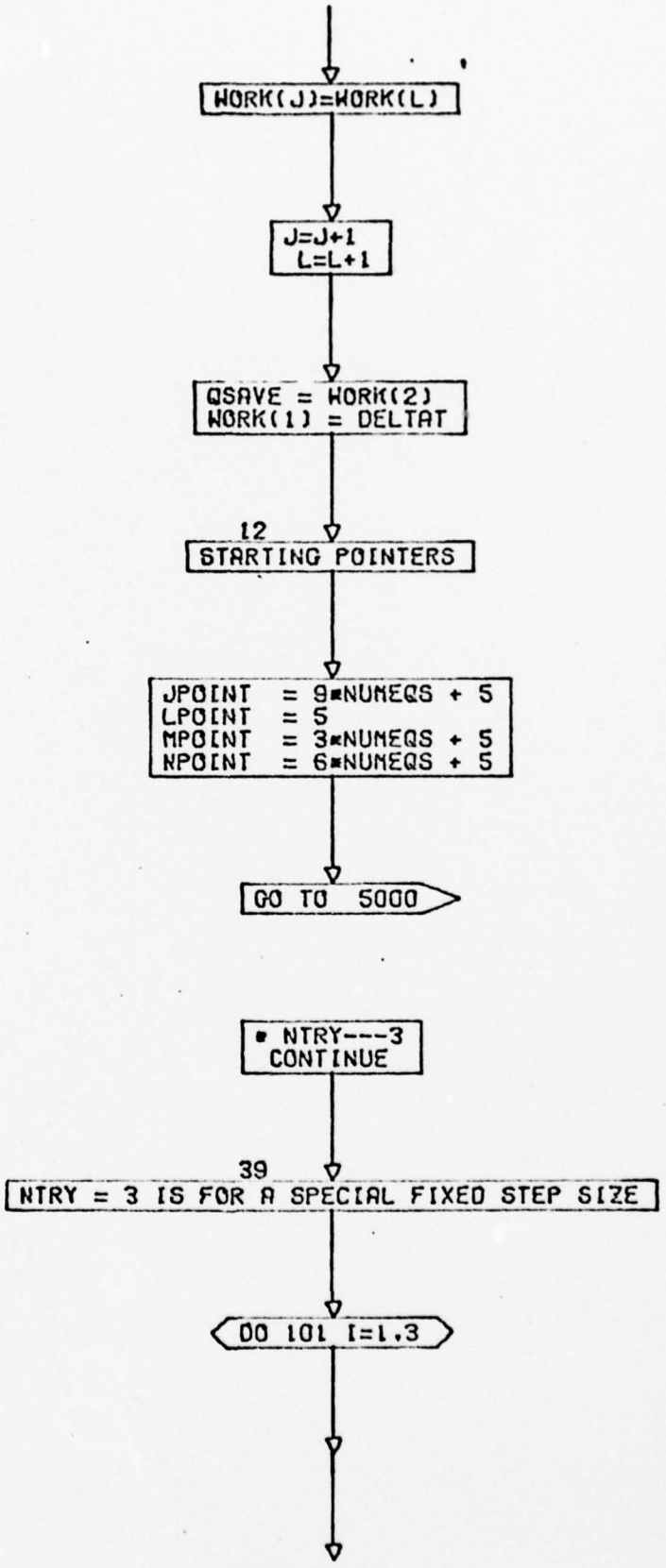
CONT. ON PG 3

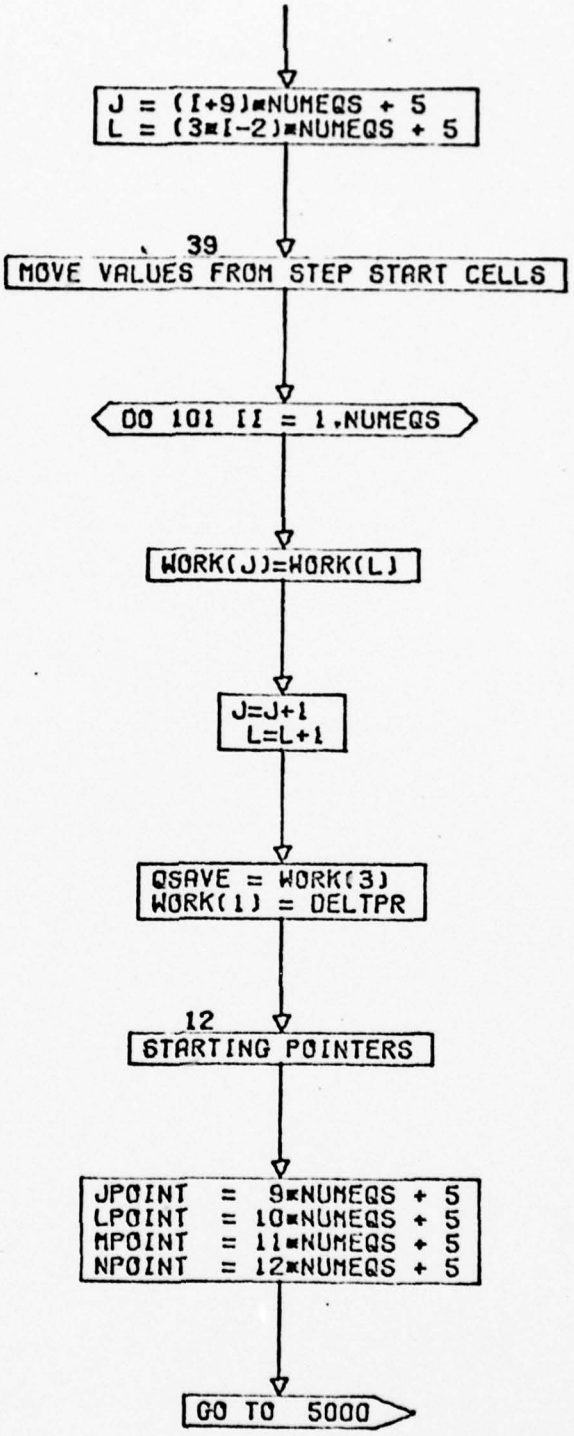

CONTINUE WITH INTEGRATION
* NTRY---1
CONTINUE
39
MOVE TIME VALUES
WORK(4) = WORK(3)
WORK(3) = WORK(2)
DO 110 I=1,3
M = (I-1)*3*NUMEQS + 5
J = M + NUMEQS
L = M + 2*NUMEQS
DO 110 QK = 1,NUMEQS
WORK(L)=WORK(J)
WORK(J)=WORK(M)
M=M+1
J=J+1
L=L+1
QSAVE = WORK(2)
WORK(1) = DELTAT

CONT. ON PG 4

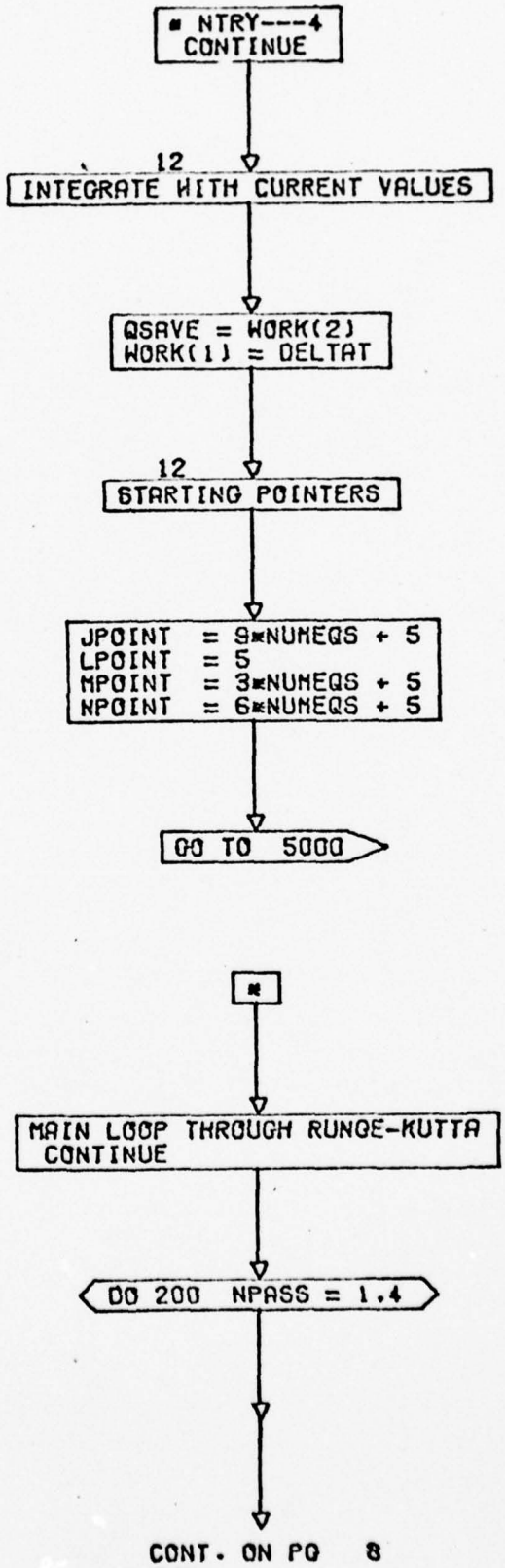

12
STARTING POINTERS
JPOINT = 9*NUMEQS + 5
LPOINT = 5
MPOINT = 3*NUMEQS + 5
NPOINT = 6*NUMEQS + 5
GO TO 5000
* NTRY---2
CONTINUE
39
REPEAT A STEP OF INTEGRATION
WORK(2)=WORK(3)
DO 145 I=1,3
J = (I-1)*3*NUMEQS + 5
L = J + NUMEQS
DO 145 QK=1,NUMEQS

CONT. ON PG 5

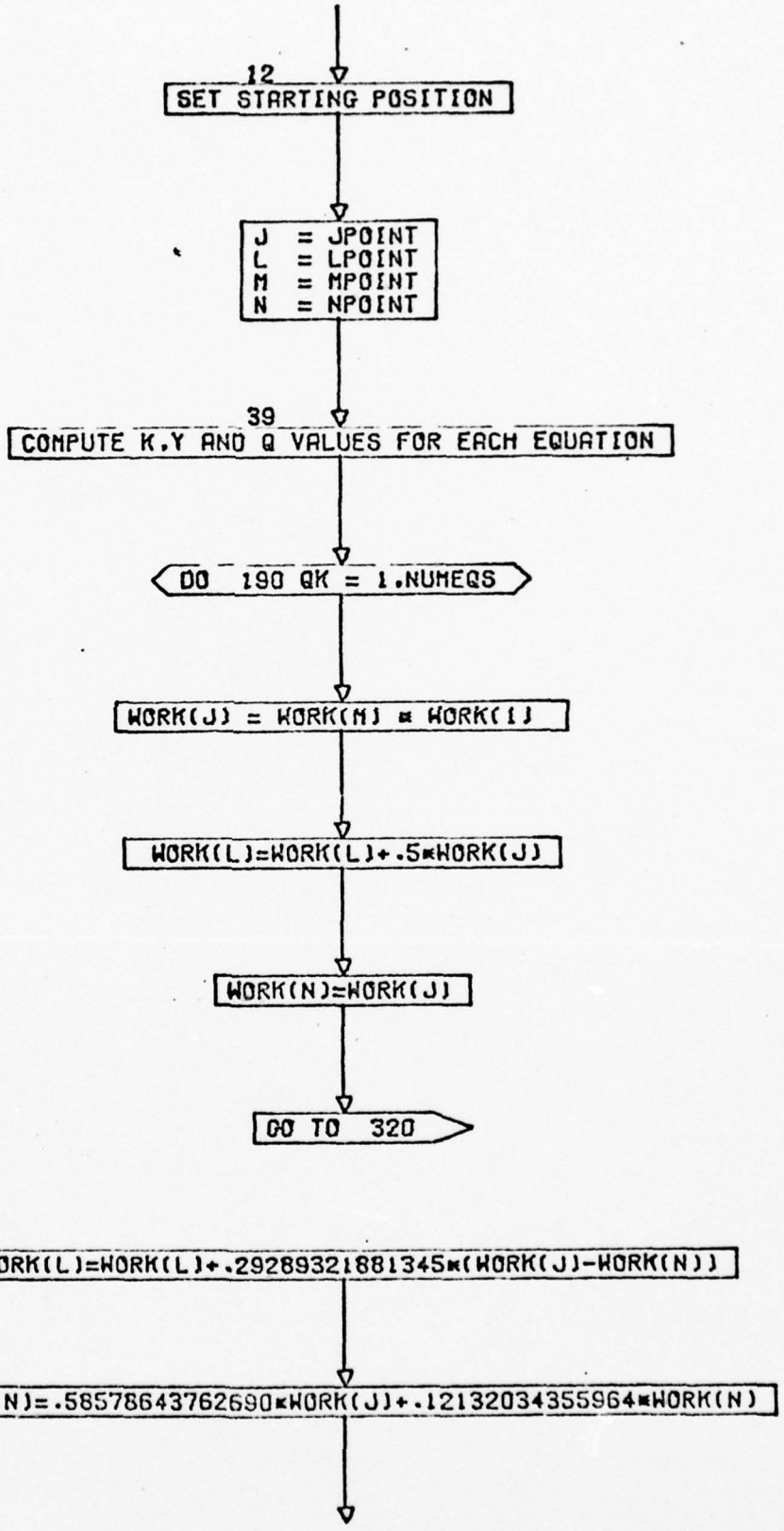

WORK(J)=WORK(L)
J=J+1
L=L+1
QSAVE = WORK(2)
WORK(1) = DELTAT
12
STARTING POINTERS
JPOINT = 9*NUMEQS + 5
LPOINT = 5
MPOINT = 3*NUMEQS + 5
NPOINT = 6*NUMEQS + 5
GO TO 5000
* NTRY---3
CONTINUE
39
NTRY = 3 IS FOR A SPECIAL FIXED STEP SIZE
DO 101 I=1,3
CONT. ON PG 6


J = (I+9)*NUMEQS + 5
L = (3*I-2)*NUMEQS + 5
39
MOVE VALUES FROM STEP START CELLS
DO 101 II = 1,NUMEQS
WORK(J)=WORK(L)
J=J+1
L=L+1
QSAVE = WORK(3)
WORK(1) = DELTPR
12
STARTING POINTERS
JPOINT = 9*NUMEQS + 5
LPOINT = 10*NUMEQS + 5
MPOINT = 11*NUMEQS + 5
NPOINT = 12*NUMEQS + 5
GO TO 5000

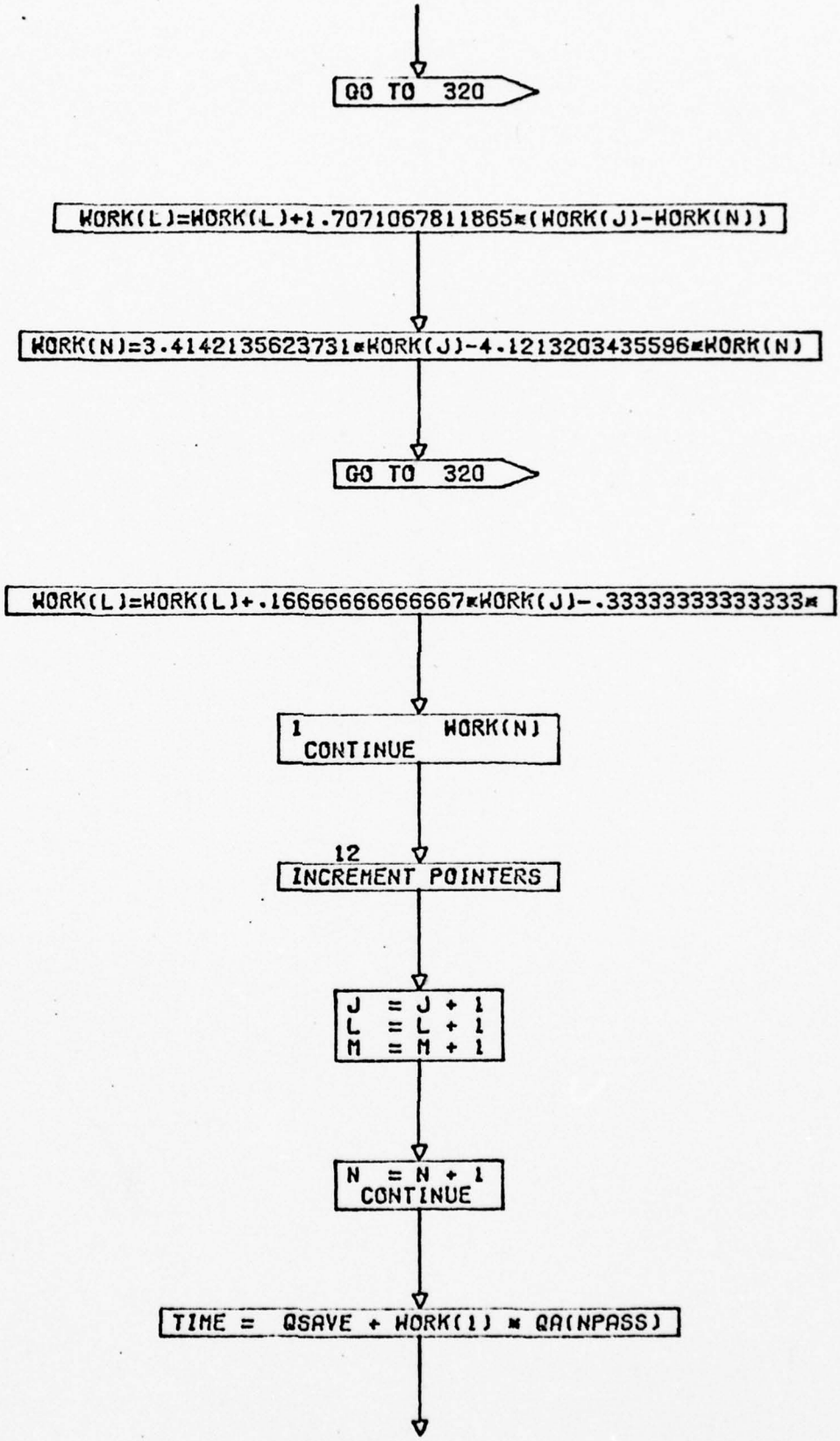
CONT. ON PG 7


* NTRY---4
CONTINUE
12
INTEGRATE WITH CURRENT VALUES
QSAVE = WORK(2)
WORK(1) = DELTAT
12
STARTING POINTERS
JPOINT = 9*NUMEQS + 5
LPOINT = 5
MPOINT = 3*NUMEQS + 5
NPOINT = 6*NUMEQS + 5
GO TO 5000
*
MAIN LOOP THROUGH RUNGE-KUTTA
CONTINUE
DO 200 NPASS = 1,4

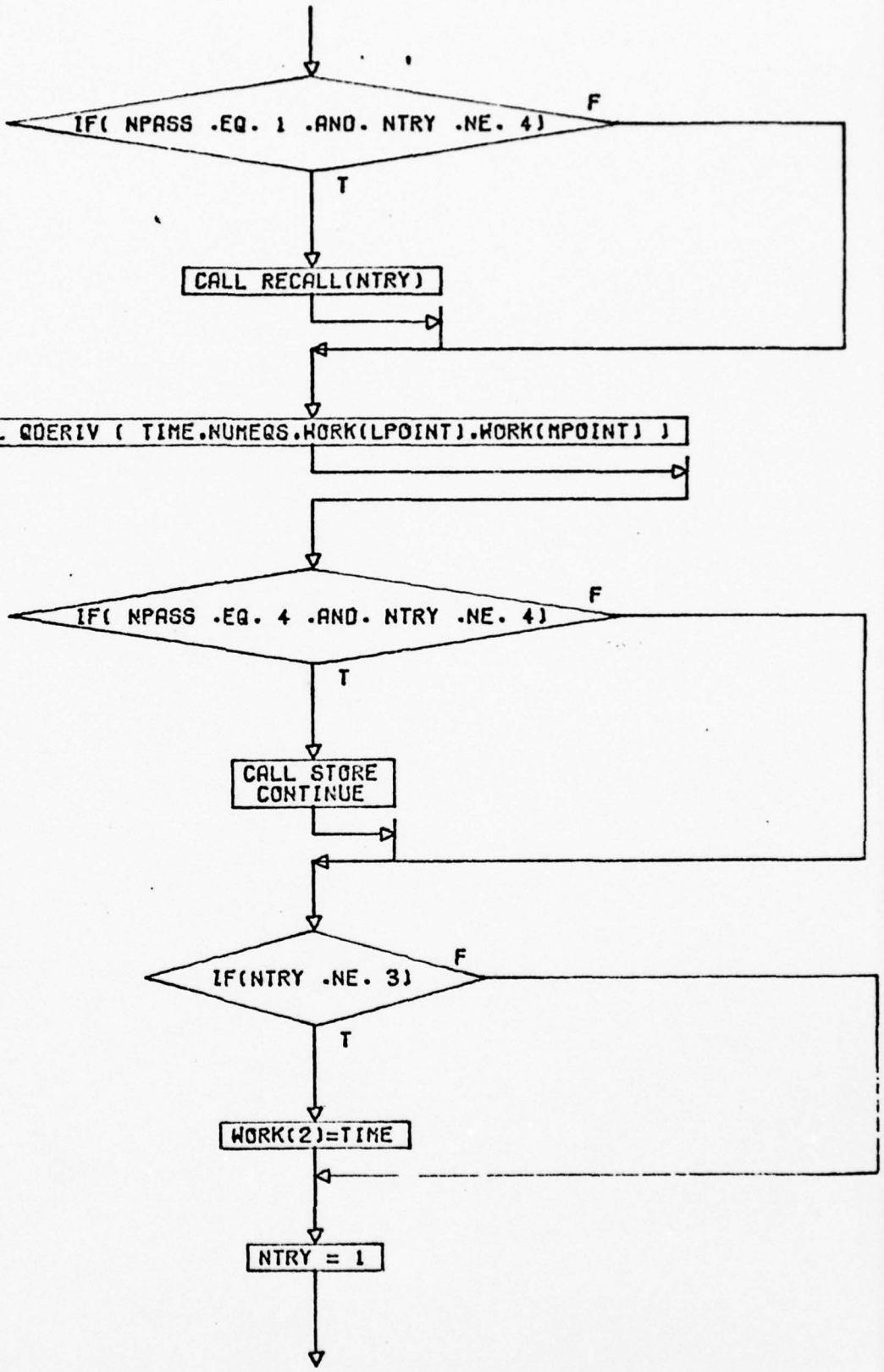
CONT. ON PG 8


12
SET STARTING POSITION
J = JPOINT
L = LPOINT
M = MPOINT
N = NPOINT
39
COMPUTE K,Y AND Q VALUES FOR EACH EQUATION
DO 190 QK = 1,NUMEQS
WORK(J) = WORK(M) * WORK(I)
WORK(L)=WORK(L)+.5*WORK(J)
WORK(N)=WORK(J)
GO TO 320
WORK(L)=WORK(L)+.29289321881345*(WORK(J)-WORK(N))
WORK(N)=.58578643762690*WORK(J)+.12132034355964*WORK(N)

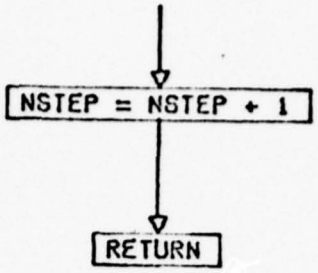
CONT. ON PG 9


GO TO 320
WORK(L)=WORK(L)+1.7071067811865*(WORK(J)-WORK(N))
WORK(N)=3.4142135623731*WORK(J)-4.1213203435596*WORK(N)
GO TO 320
WORK(L)=WORK(L)+.16666666666667*WORK(J)-.33333333333333*
1 WORK(N)
CONTINUE
12
INCREMENT POINTERS
J = J + 1
L = L + 1
M = M + 1
N = N + 1
CONTINUE
TIME = QSAVE + WORK(1) * QA(NPASS)

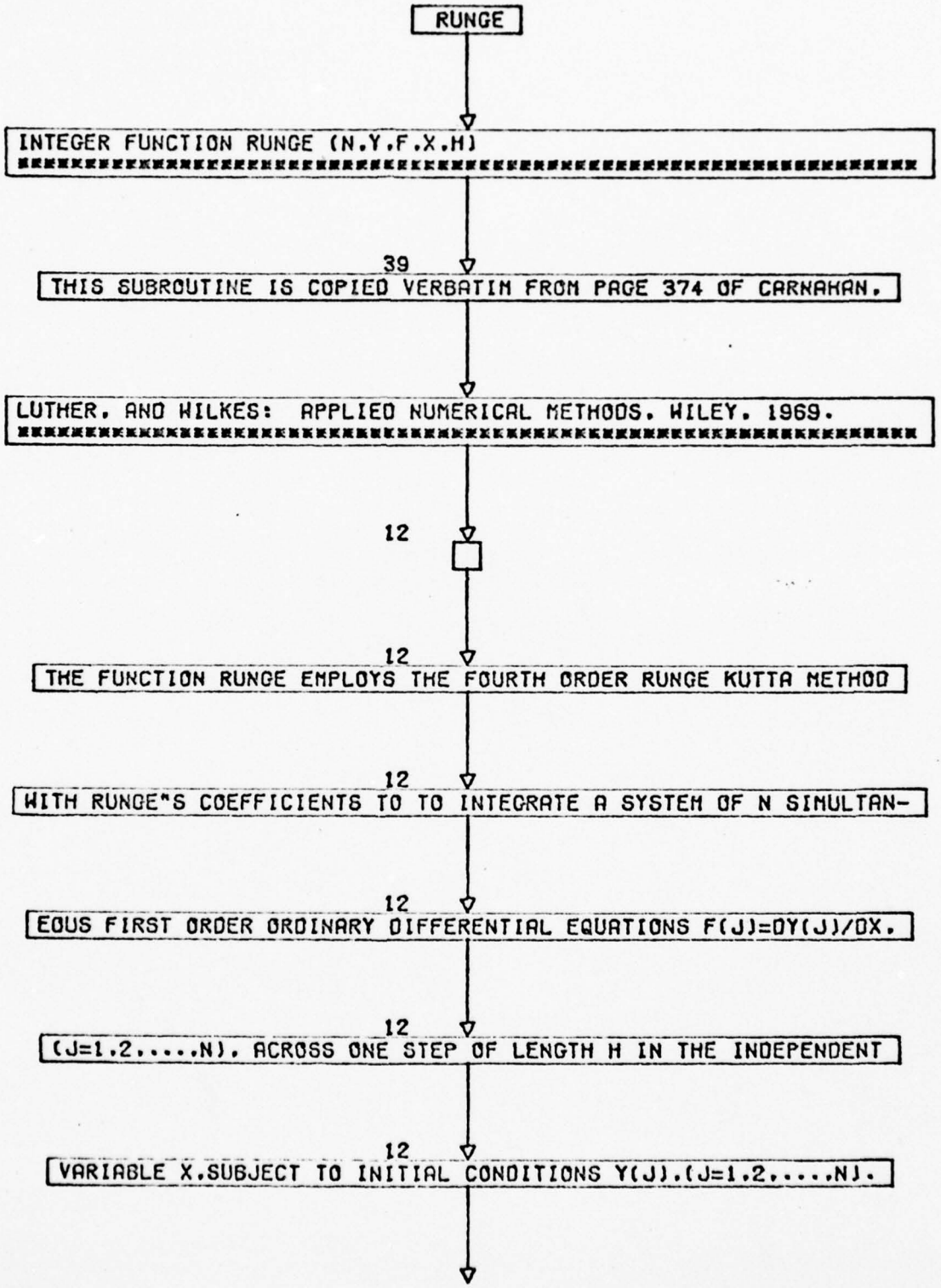
CONT. ON PG 10


IF(NPASS .EQ. 1 .AND. NTRY .NE. 4)
F
T
CALL RECALL(NTRY)
CALL QDERIV (TIME,NUMEQS,WORK(LPOINT),WORK(MPOINT))
IF(NPASS .EQ. 4 .AND. NTRY .NE. 4)
F
T
CALL STORE
CONTINUE
IF(NTRY .NE. 3)
F
T
WORK(2)=TIME
NTRY = 1

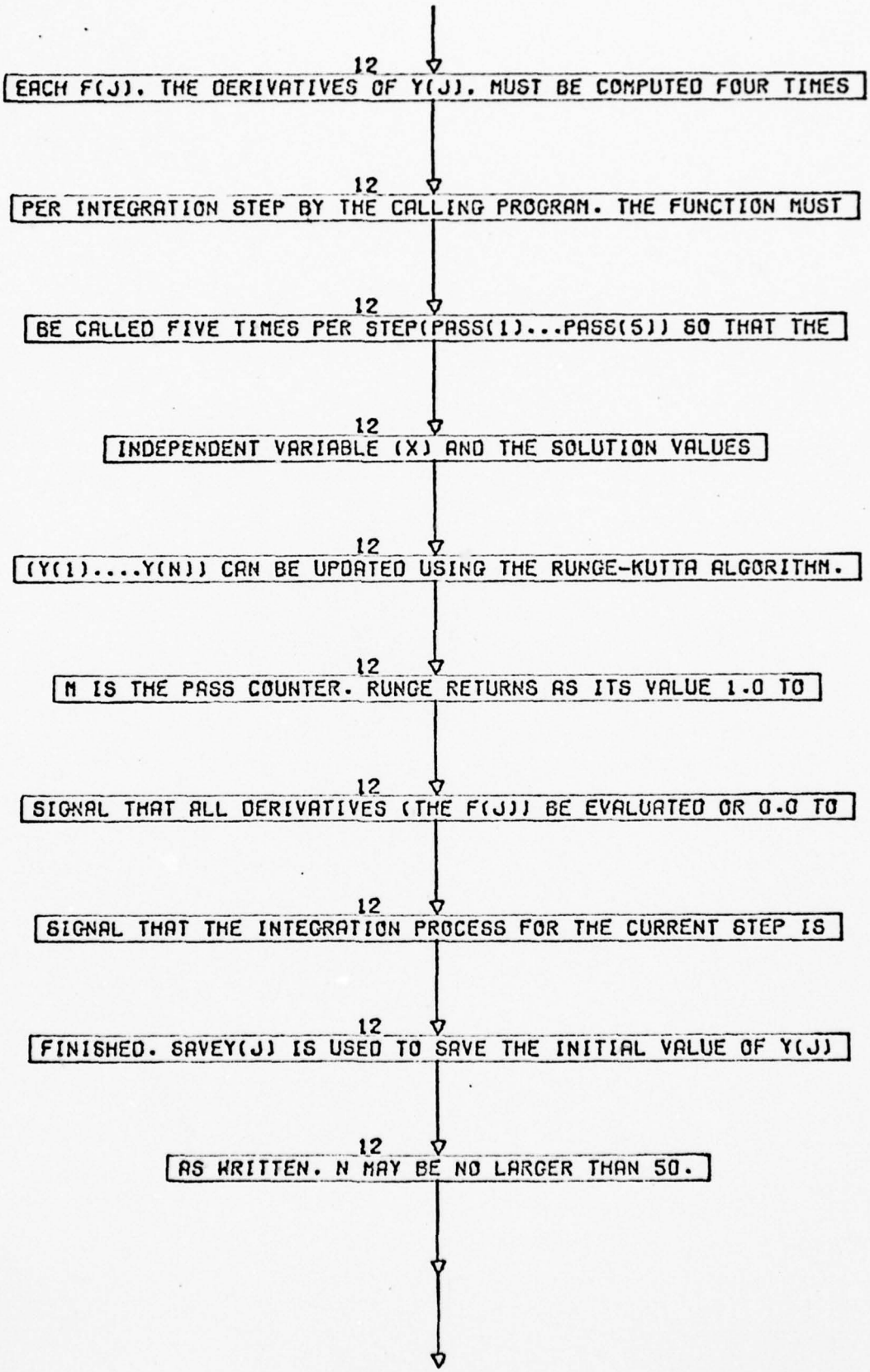
CONT. ON PG 11



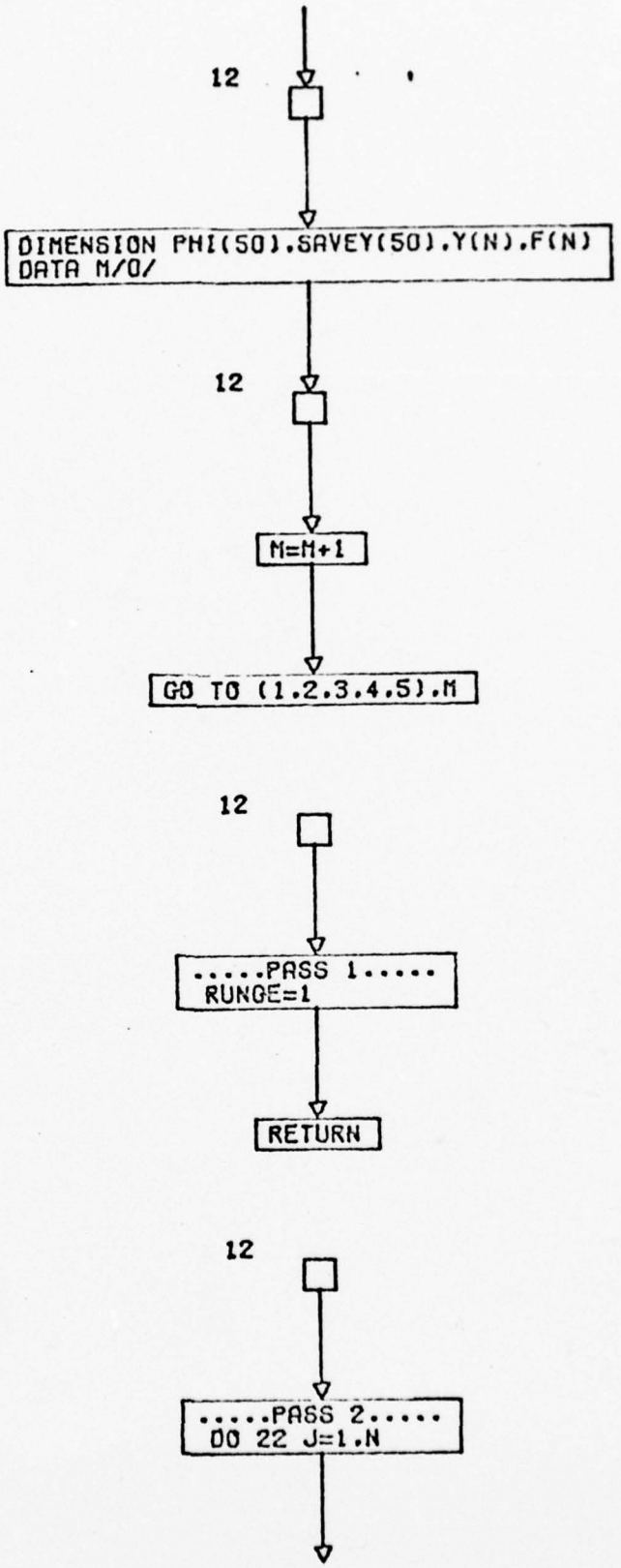
END


RUNGE
INTEGER FUNCTION RUNGE (N,Y,F,X,H)
39
THIS SUBROUTINE IS COPIED VERBATIM FROM PAGE 374 OF CARNAHAN,
LUTHER, AND WILKES: APPLIED NUMERICAL METHODS. WILEY. 1969.
12
12
THE FUNCTION RUNGE EMPLOYS THE FOURTH ORDER RUNGE KUTTA METHOD
12
WITH RUNGE"S COEFFICIENTS TO TO INTEGRATE A SYSTEM OF N SIMULTAN-
12
EOUS FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS F(J)=DY(J)/DX.
12
(J=1,2,....,N), ACROSS ONE STEP OF LENGTH H IN THE INDEPENDENT
12
VARIABLE X,SUBJECT TO INITIAL CONDITIONS Y(J),(J=1,2,....,N).

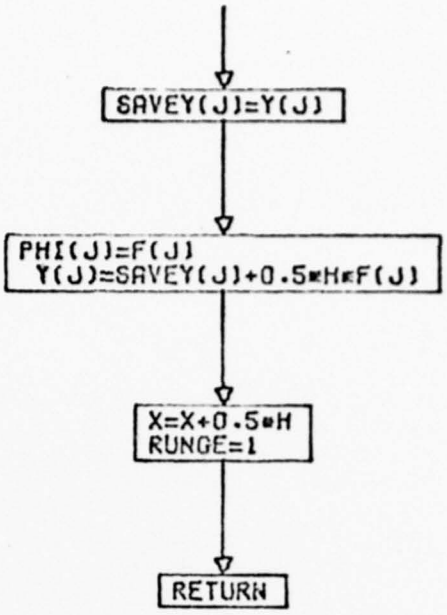
CONT. ON PG 2


12
EACH F(J). THE DERIVATIVES OF Y(J). MUST BE COMPUTED FOUR TIMES
12
PER INTEGRATION STEP BY THE CALLING PROGRAM. THE FUNCTION MUST
12
BE CALLED FIVE TIMES PER STEP(PASS(1)...PASS(5)) SO THAT THE
12
INDEPENDENT VARIABLE (X) AND THE SOLUTION VALUES
12
(Y(1)....Y(N)) CAN BE UPDATED USING THE RUNGE-KUTTA ALGORITHM.
12
N IS THE PASS COUNTER. RUNGE RETURNS AS ITS VALUE 1.0 TO
12
SIGNAL THAT ALL DERIVATIVES (THE F(J)) BE EVALUATED OR 0.0 TO
12
SIGNAL THAT THE INTEGRATION PROCESS FOR THE CURRENT STEP IS
12
FINISHED. SAVEY(J) IS USED TO SAVE THE INITIAL VALUE OF Y(J)
12
AS WRITTEN. N MAY BE NO LARGER THAN 50.

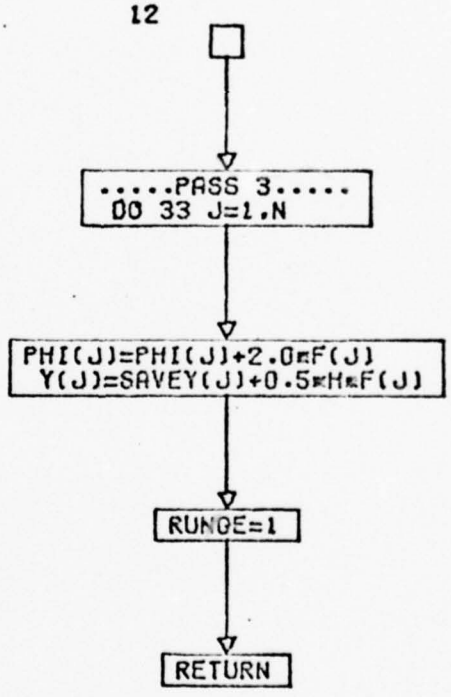
CONT. ON PG 3


12
DIMENSION PHI(50).SAVEY(50).Y(N).F(N)
DATA M/0/
12
M=M+1
GO TO (1.2.3.4.5).M
12
.....PASS 1.....
RUNGE=1
RETURN
12
.....PASS 2.....
DO 22 J=1.N

CONT. ON PG 4



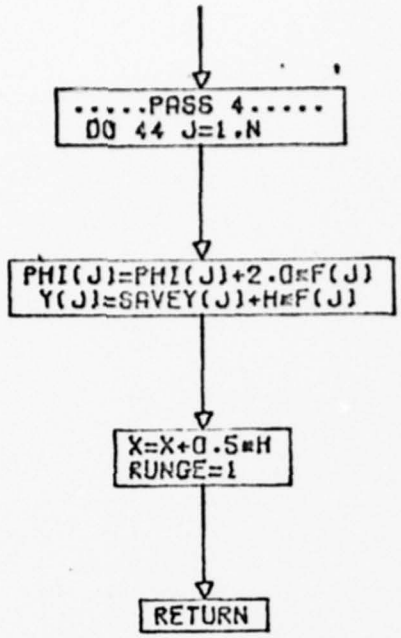
12

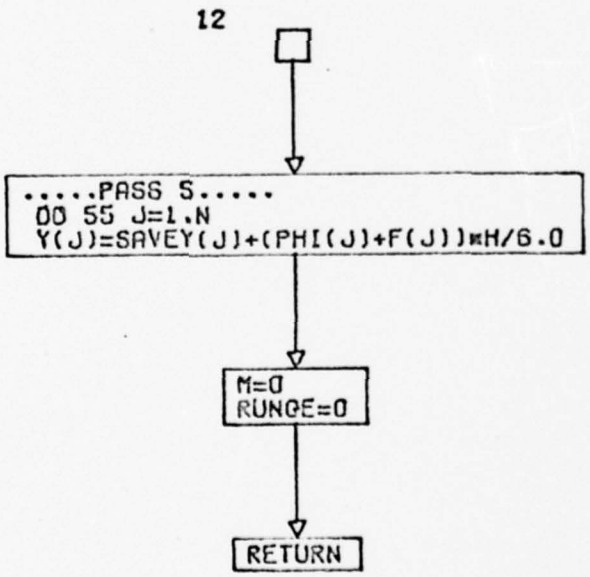


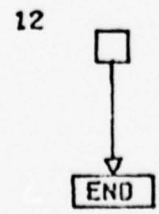
12



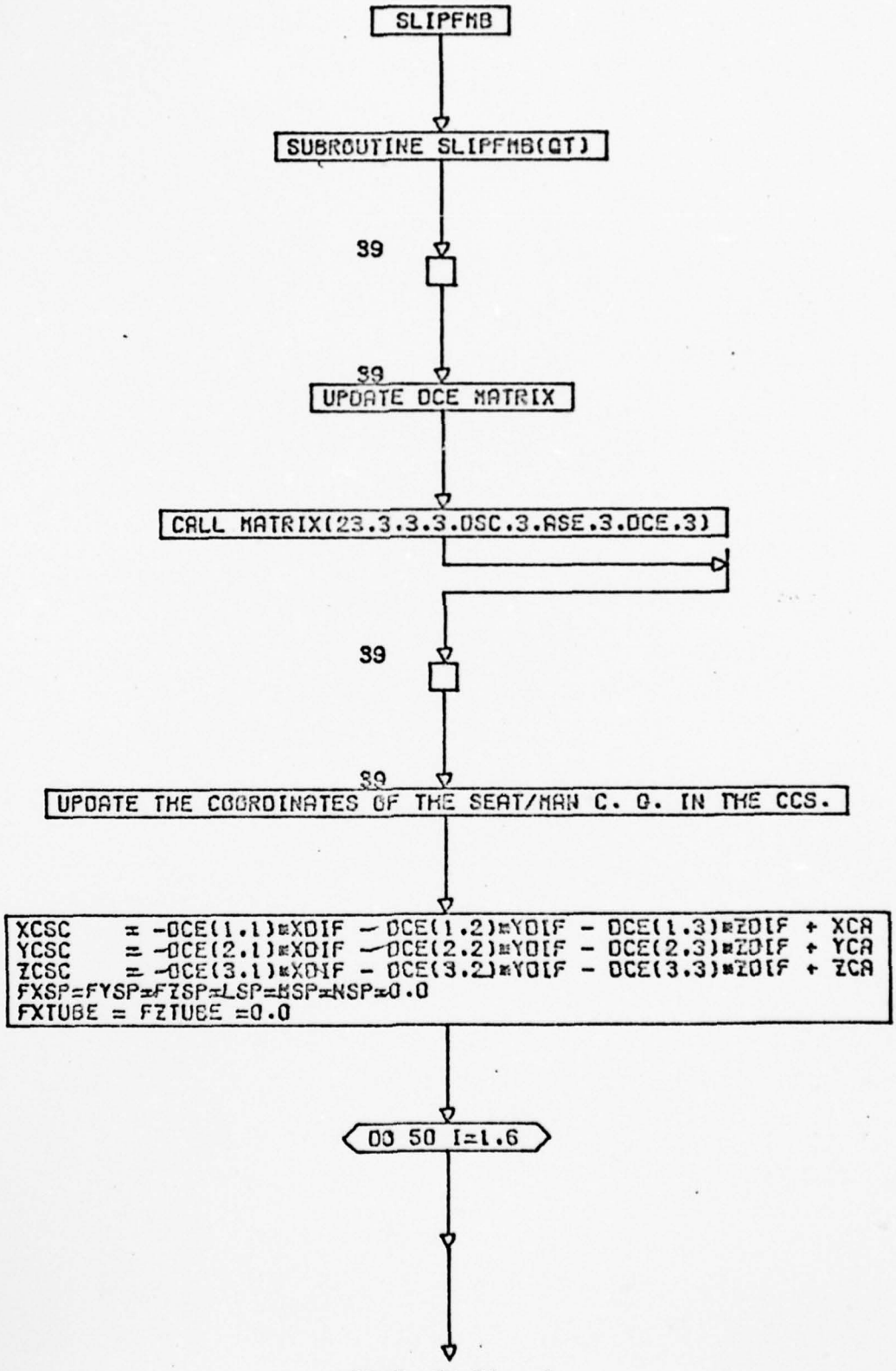
CONT. ON PG 5

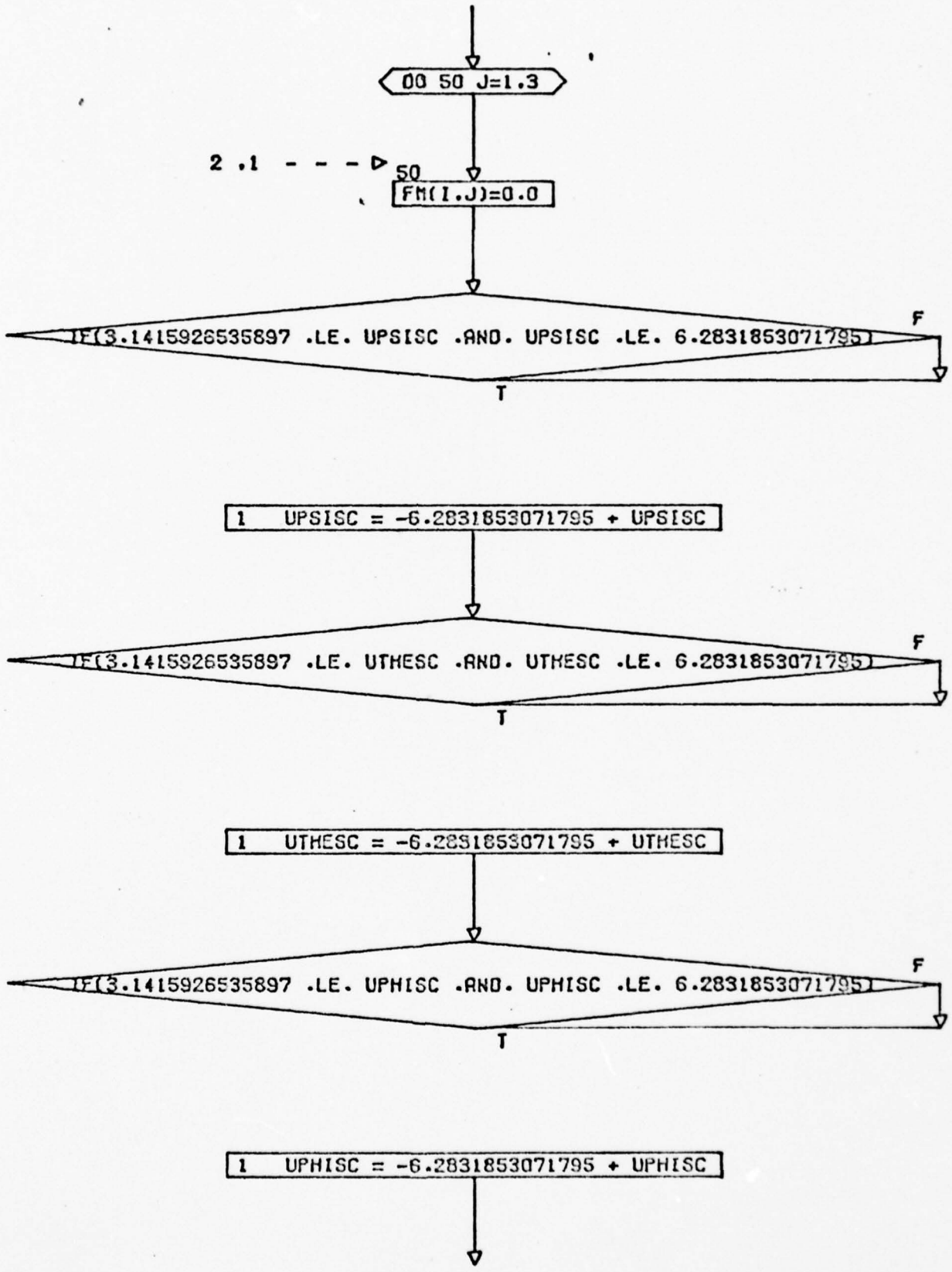

.....PASS 4.....
DO 44 J=1,N
PHI(J)=PHI(J)+2.0*F(J)
Y(J)=SAVEY(J)+H*F(J)
X=X+0.5*H
RUNGE=1
RETURN


12
.....PASS 5.....
DO 55 J=1,N
Y(J)=SAVEY(J)+(PHI(J)+F(J))*H/6.0
M=0
RUNGE=0
RETURN

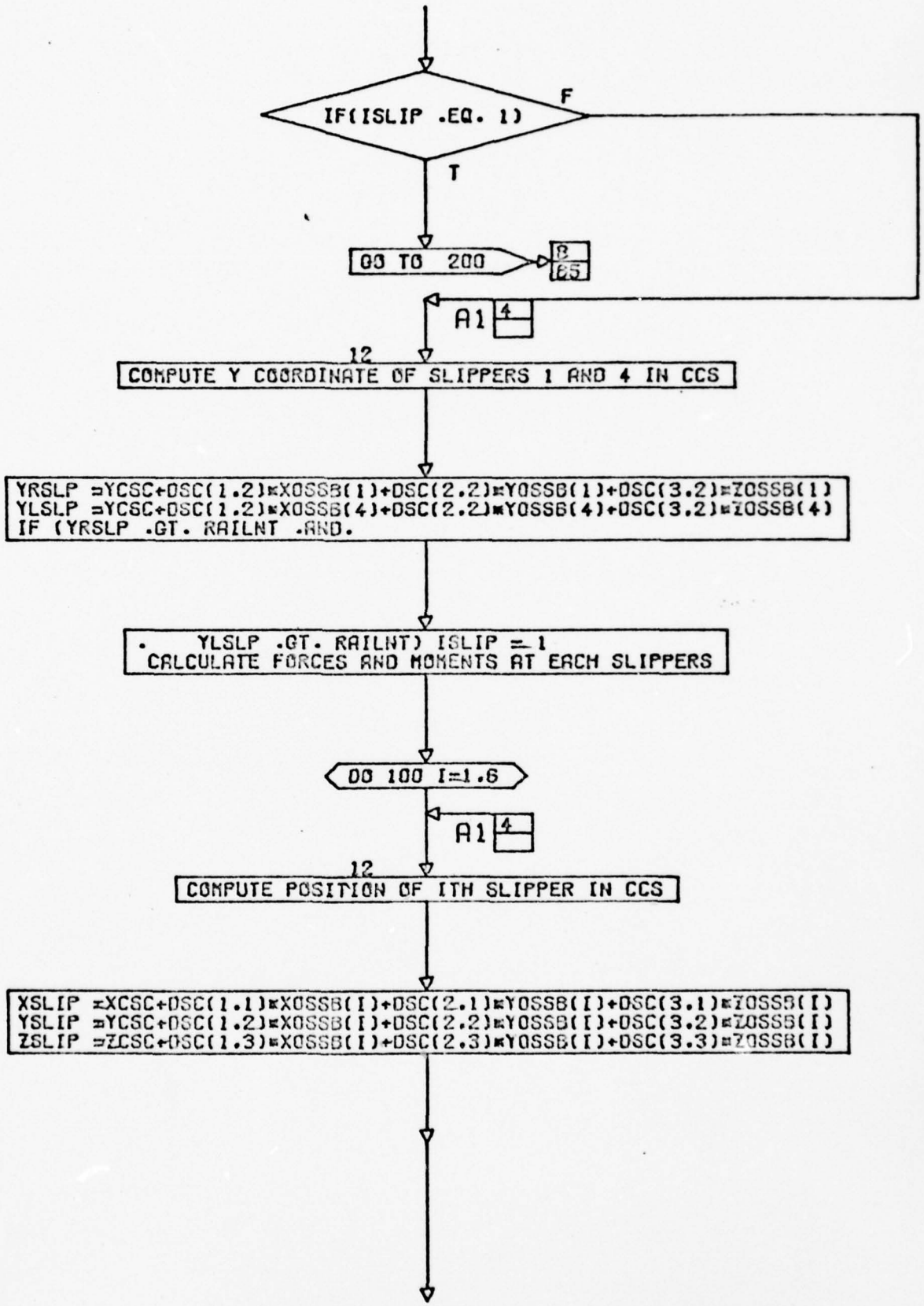

12
END

PG 5 FINAL

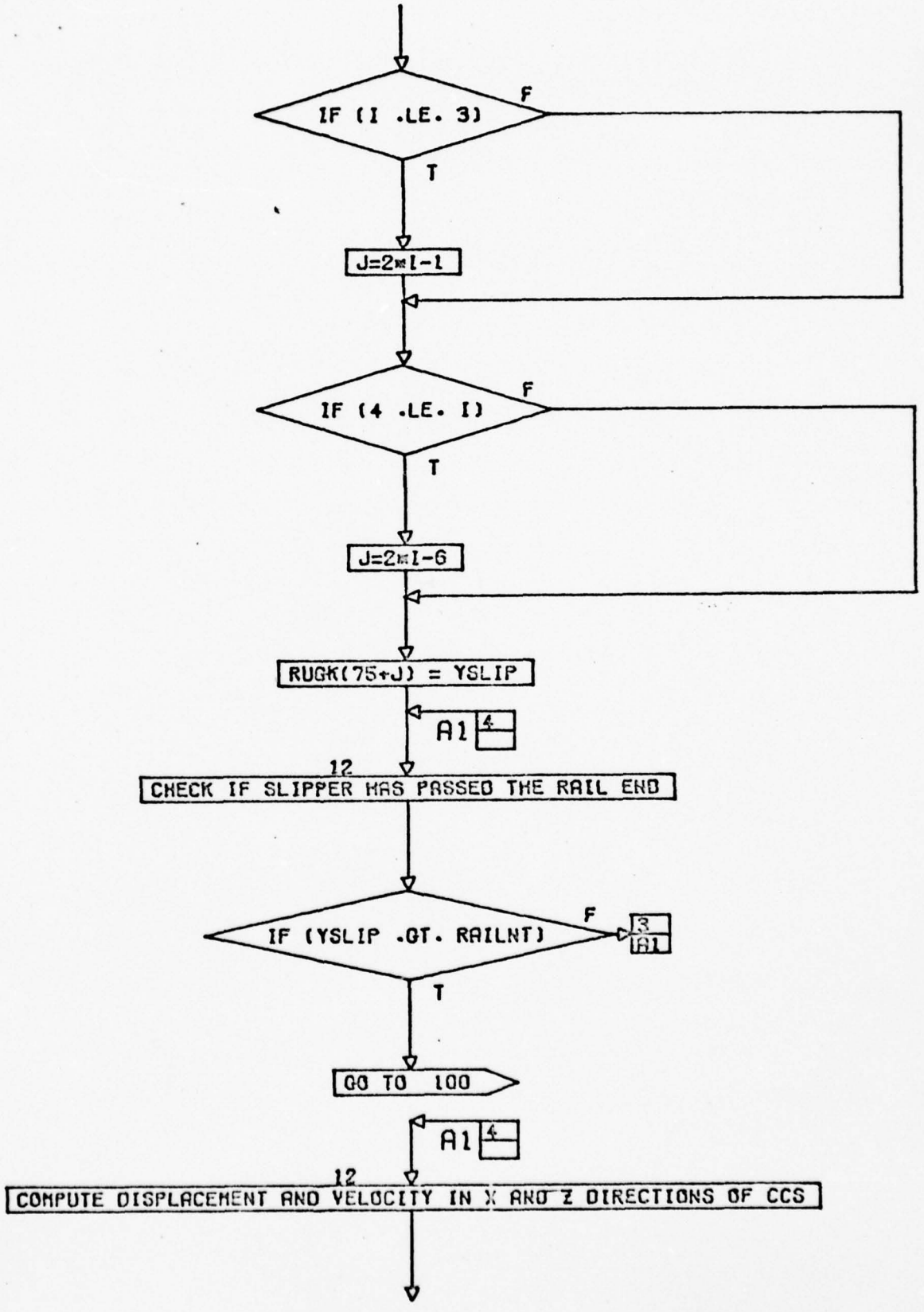

SLIPFMB
SUBROUTINE SLIPFMB(QT)
39
39
UPDATE DCE MATRIX
CALL MATRIX(23.3.3.3.DSC.3.ASE.3.DCE.3)
39
39
UPDATE THE COORDINATES OF THE SEAT/MAN C. G. IN THE CCS.
XCSC = -DCE(1.1)*XDIF - DCE(1.2)*YDIF - DCE(1.3)*ZDIF + XCA
YCSC = -DCE(2.1)*XDIF - DCE(2.2)*YDIF - DCE(2.3)*ZDIF + YCA
ZCSC = -DCE(3.1)*XDIF - DCE(3.2)*YDIF - DCE(3.3)*ZDIF + ZCA
FXSP=FYSP=FZSP=LSP=MSP=NSP=0.0
FXTUBE = FZTUBE =0.0
DO 50 I=1,6
CONT. ON PG 2


DO 50 J=1,3
2 ,1 - - - ▷ 50
FM(I,J)=0.0
IF(3.1415926535897 .LE. UPSISC .AND. UPSISC .LE. 6.2831853071795)
F
T
1 UPSISC = -6.2831853071795 + UPSISC
IF(3.1415926535897 .LE. UTHESC .AND. UTHESC .LE. 6.2831853071795)
F
T
1 UTHESC = -6.2831853071795 + UTHESC
IF(3.1415926535897 .LE. UPHISC .AND. UPHISC .LE. 6.2831853071795)
F
T
1 UPHISC = -6.2831853071795 + UPHISC

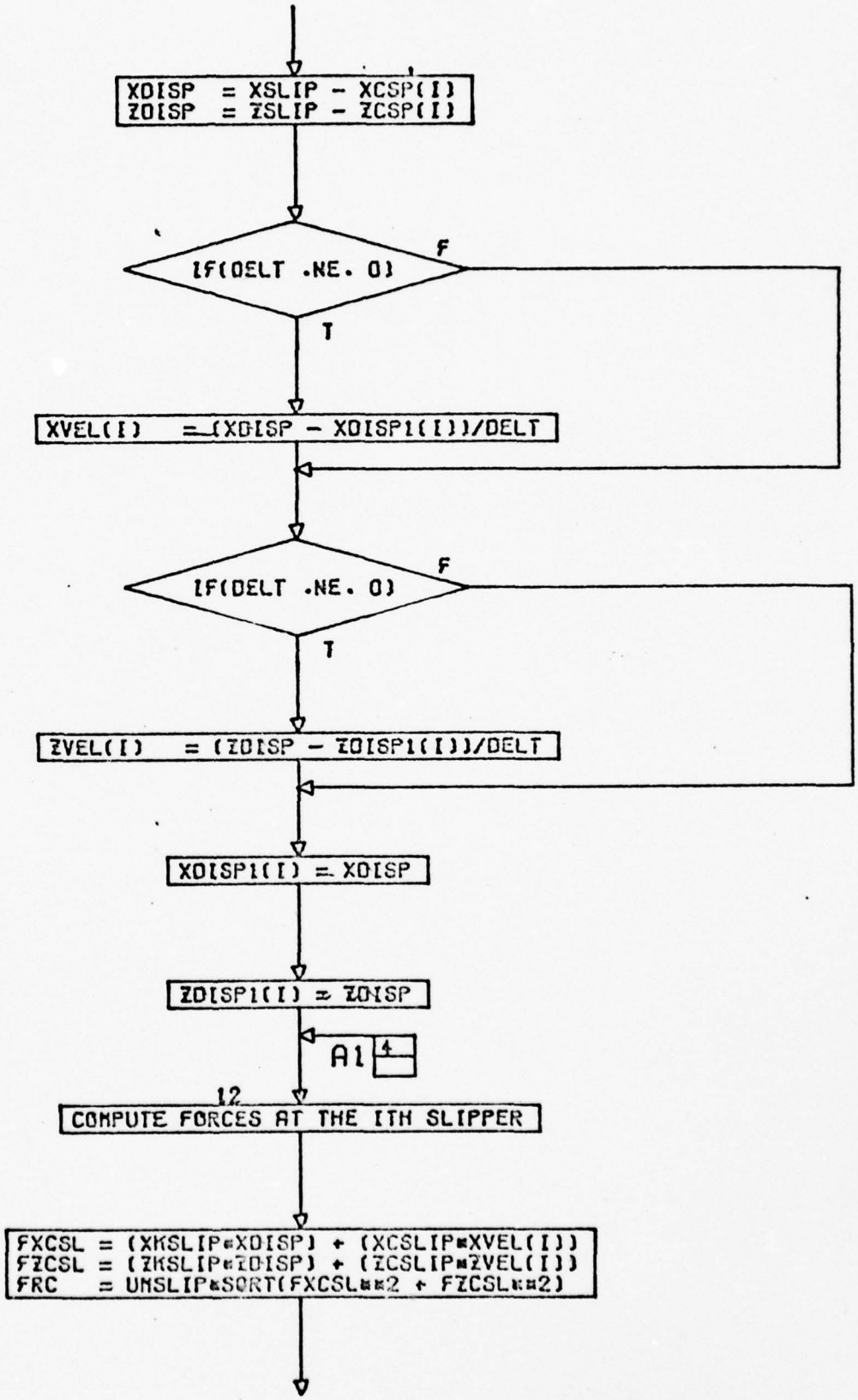
CONT. ON PG 3


IF(ISLIP .EQ. 1)
F
T
GO TO 200
8
85
A1
4
12
COMPUTE Y COORDINATE OF SLIPPERS 1 AND 4 IN CCS
YRSLP =YCSC+DSC(1,2)*XOSSB(1)+DSC(2,2)*YOSSB(1)+DSC(3,2)*ZOSSB(1)
YLSLP =YCSC+DSC(1,2)*XOSSB(4)+DSC(2,2)*YOSSB(4)+DSC(3,2)*ZOSSB(4)
IF (YRSLP .GT. RAILNT .AND.
YLSLP .GT. RAILNT) ISLIP = 1
CALCULATE FORCES AND MOMENTS AT EACH SLIPPERS
DO 100 I=1,6
A1
4
12
COMPUTE POSITION OF ITH SLIPPER IN CCS
XSLIP =XCSC+DSC(1,1)*XOSSB(I)+DSC(2,1)*YOSSB(I)+DSC(3,1)*ZOSSB(I)
YSLIP =YCSC+DSC(1,2)*XOSSB(I)+DSC(2,2)*YOSSB(I)+DSC(3,2)*ZOSSB(I)
ZSLIP =ZCSC+DSC(1,3)*XOSSB(I)+DSC(2,3)*YOSSB(I)+DSC(3,3)*ZOSSB(I)

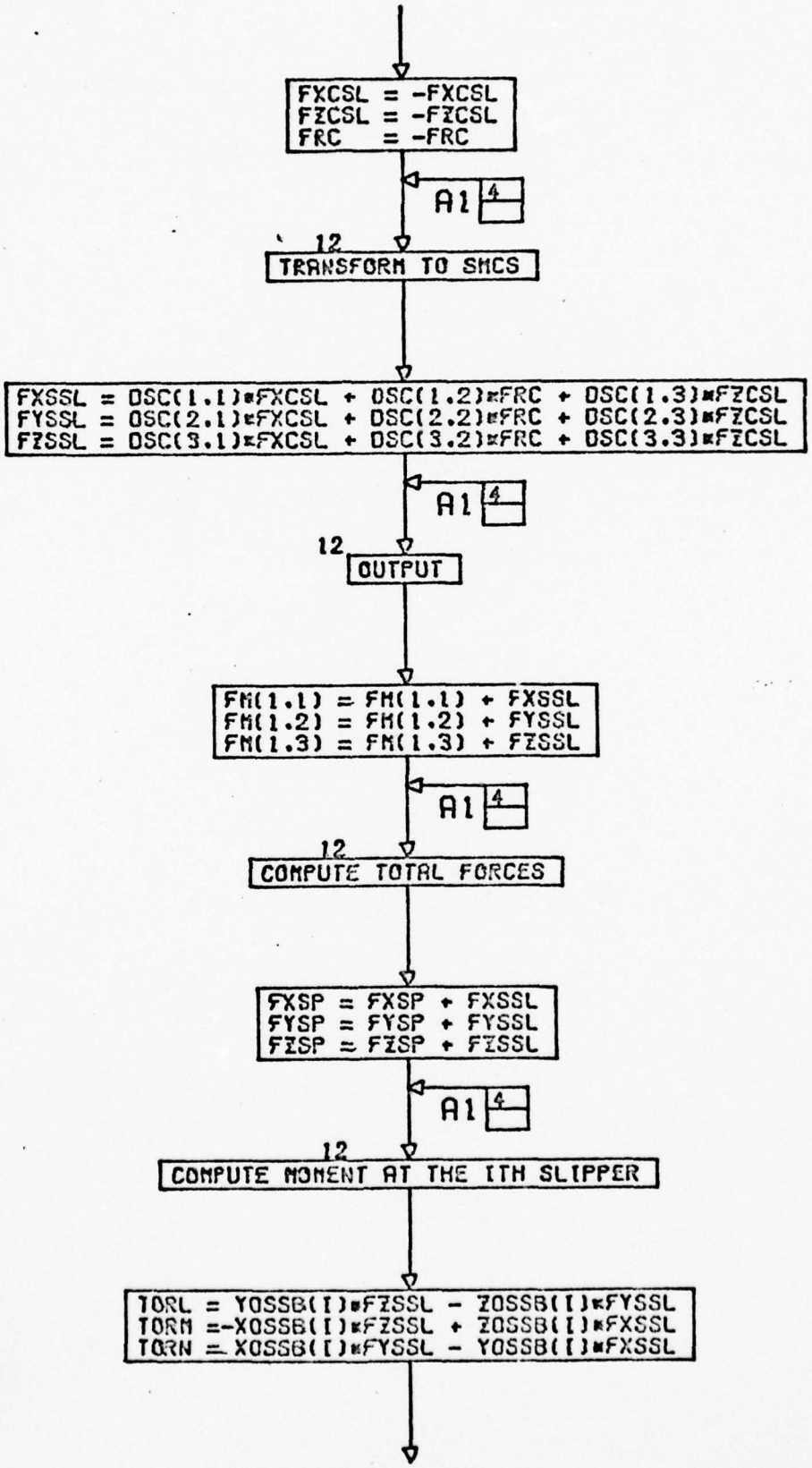
CONT. ON PG 4


IF (I .LE. 3)
F
T
J=2*I-1
IF (4 .LE. I)
F
T
J=2*I-6
RUGK(75+J) = YSLIP
A1
4
12
CHECK IF SLIPPER HAS PASSED THE RAIL END
IF (YSLIP .GT. RAILNT)
F
3
A1
T
GO TO 100
A1
4
12
COMPUTE DISPLACEMENT AND VELOCITY IN X AND Z DIRECTIONS OF CCS

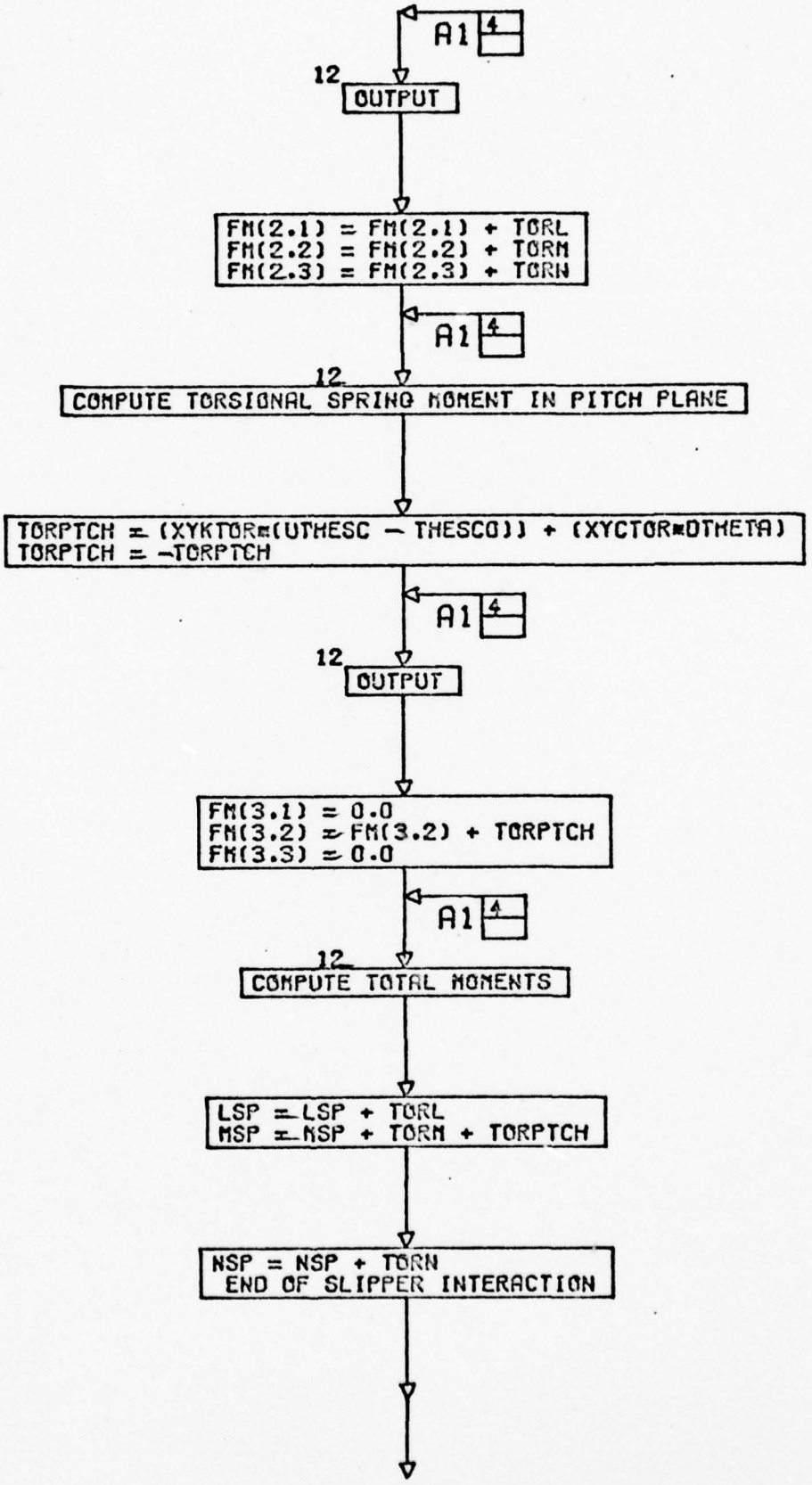
CONT. ON PG 5

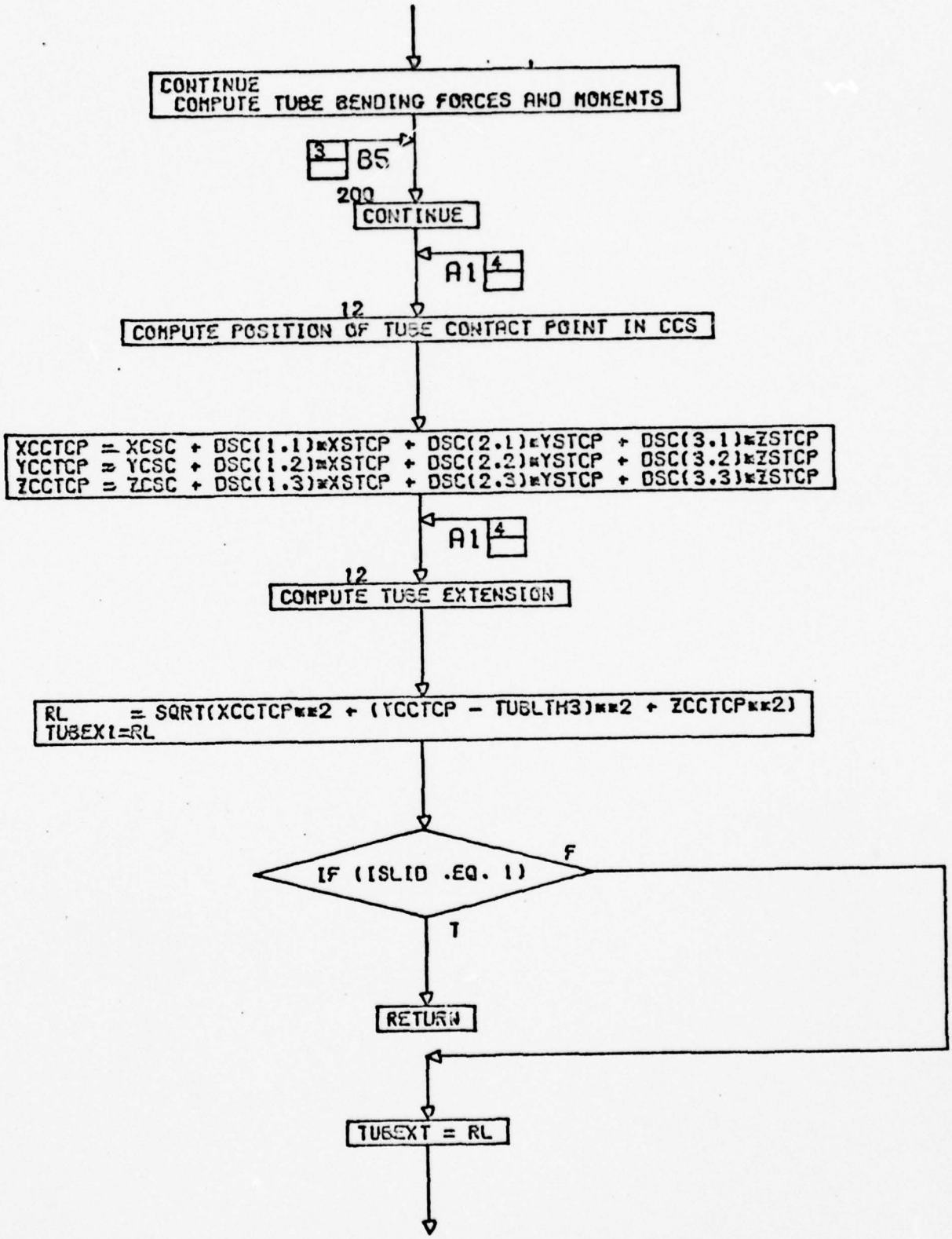

XDISP = XSLIP - XCSP(I)
ZDISP = ZSLIP - ZCSP(I)
IF(DELT .NE. 0)
F
T
XVEL(I) = (XDISP - XDISP1(I))/DELT
IF(DELT .NE. 0)
F
T
ZVEL(I) = (ZDISP - ZDISP1(I))/DELT
XDISP1(I) = XDISP
ZDISP1(I) = ZDISP
A1
4
12
COMPUTE FORCES AT THE ITH SLIPPER
FXCSL = (XKSLIP*XDISP) + (XCSLIP*XVEL(I))
FZCSL = (ZKSLIP*ZDISP) + (ZCSLIP*ZVEL(I))
FRC = UNSLIP*SQRT(FXCSL**2 + FZCSL**2)

CONT. ON PG 6

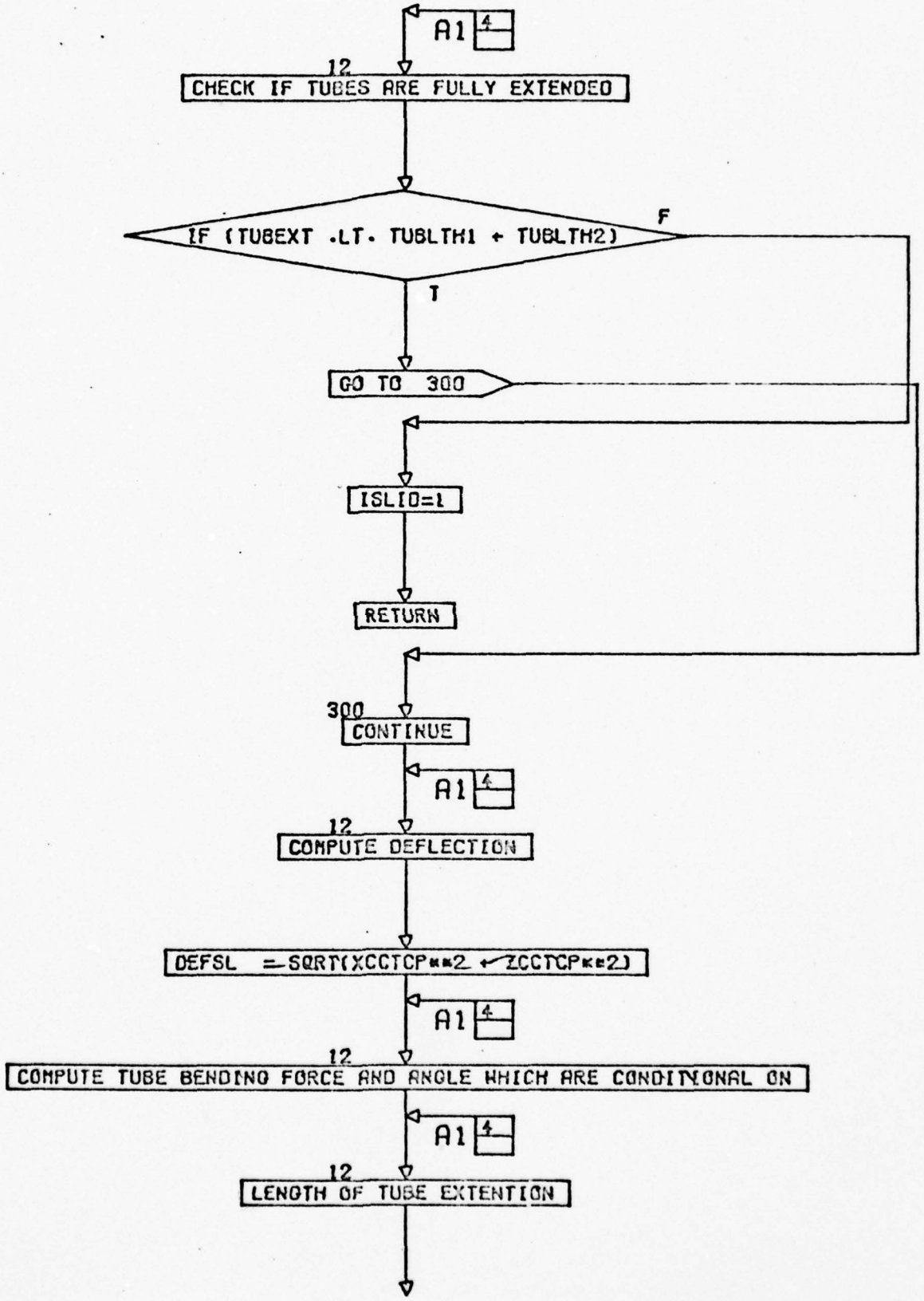

FXCSL = -FXCSL
FZCSL = -FZCSL
FRC = -FRC
A1 4
12
TRANSFORM TO SMCS
FXSSL = OSC(1.1)*FXCSL + OSC(1.2)*FRC + OSC(1.3)*FZCSL
FYSSL = OSC(2.1)*FXCSL + OSC(2.2)*FRC + OSC(2.3)*FZCSL
FZSSL = OSC(3.1)*FXCSL + OSC(3.2)*FRC + OSC(3.3)*FZCSL
A1 4
12
OUTPUT
FM(1.1) = FM(1.1) + FXSSL
FM(1.2) = FM(1.2) + FYSSL
FM(1.3) = FM(1.3) + FZSSL
A1 4
12
COMPUTE TOTAL FORCES
FXSP = FXSP + FXSSL
FYSP = FYSP + FYSSL
FZSP = FZSP + FZSSL
A1 4
12
COMPUTE MOMENT AT THE ITH SLIPPER
TORL = YOSSB(I)*FZSSL - ZOSSB(I)*FYSSL
TORM =-XOSSB(I)*FZSSL + ZOSSB(I)*FXSSL
TORN = XOSSB(I)*FYSSL - YOSSB(I)*FXSSL

CONT. ON PG 7

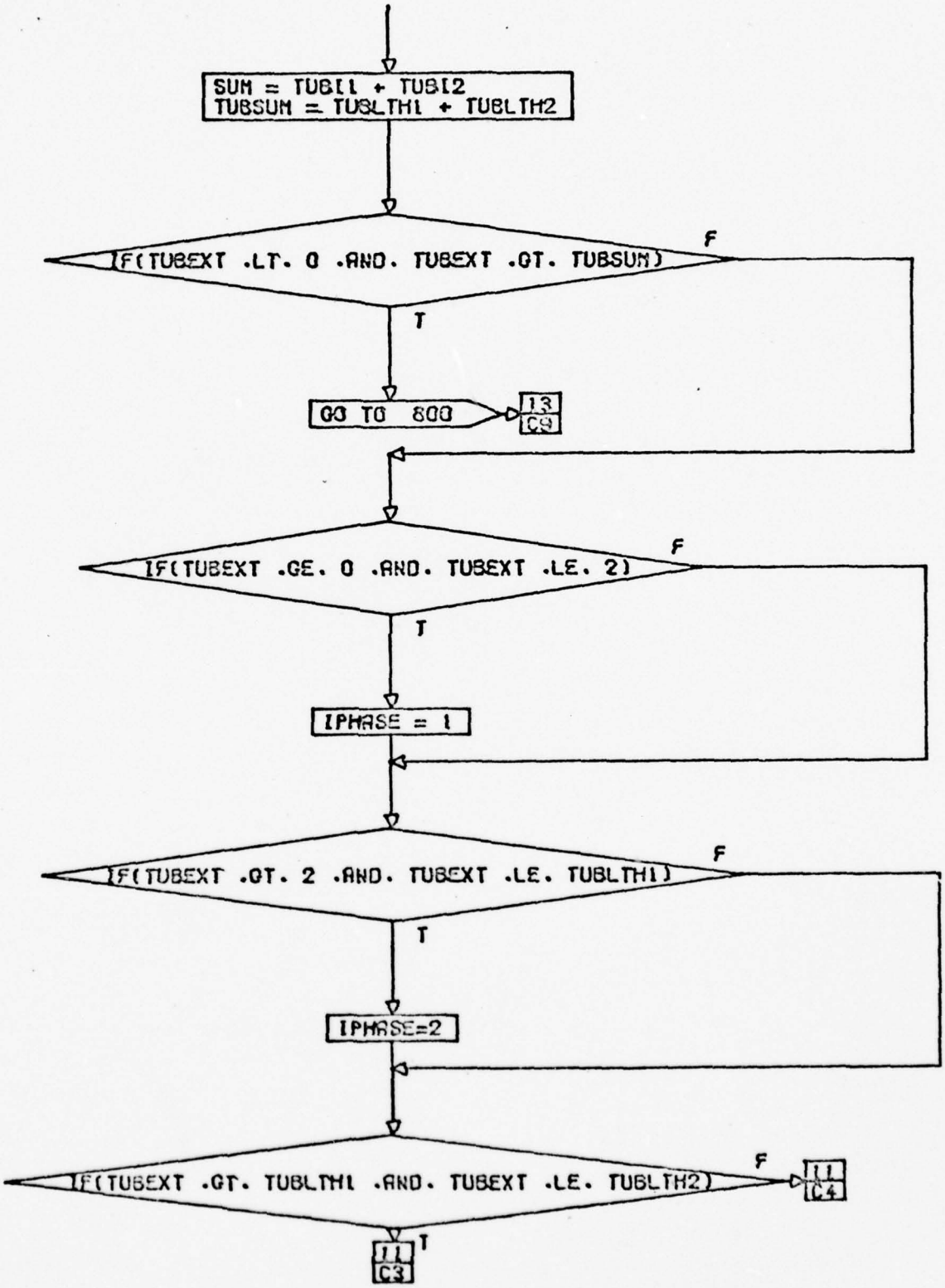

A1 4
12
OUTPUT
FM(2.1) = FM(2.1) + TORL
FM(2.2) = FM(2.2) + TORM
FM(2.3) = FM(2.3) + TORN
A1 4
12
COMPUTE TORSIONAL SPRING MOMENT IN PITCH PLANE
TORPTCH = (XYKTOR*(UTHESC - THESCO)) + (XYCTOR*DTHETA)
TORPTCH = -TORPTCH
A1 4
12
OUTPUT
FM(3.1) = 0.0
FM(3.2) = FM(3.2) + TORPTCH
FM(3.3) = 0.0
A1 4
12
COMPUTE TOTAL MOMENTS
LSP = LSP + TORL
MSP = MSP + TORM + TORPTCH
NSP = NSP + TORN
END OF SLIPPER INTERACTION
CONT. ON PG 8


CONTINUE
COMPUTE TUBE BENDING FORCES AND MOMENTS
3
B5
200
CONTINUE
A1
4
12
COMPUTE POSITION OF TUBE CONTACT POINT IN CCS
XCCTCP = XCSC + DSC(1,1)*XSTCP + DSC(2,1)*YSTCP + DSC(3,1)*ZSTCP
YCCTCP = YCSC + DSC(1,2)*XSTCP + DSC(2,2)*YSTCP + DSC(3,2)*ZSTCP
ZCCTCP = ZCSC + DSC(1,3)*XSTCP + DSC(2,3)*YSTCP + DSC(3,3)*ZSTCP
A1
4
12
COMPUTE TUBE EXTENSION
RL = SQRT(XCCTCP**2 + (YCCTCP - TUBLTH3)**2 + ZCCTCP**2)
TUBEX1=RL
IF (ISLID .EQ. 1)
F
T
RETURN
TUBEXT = RL

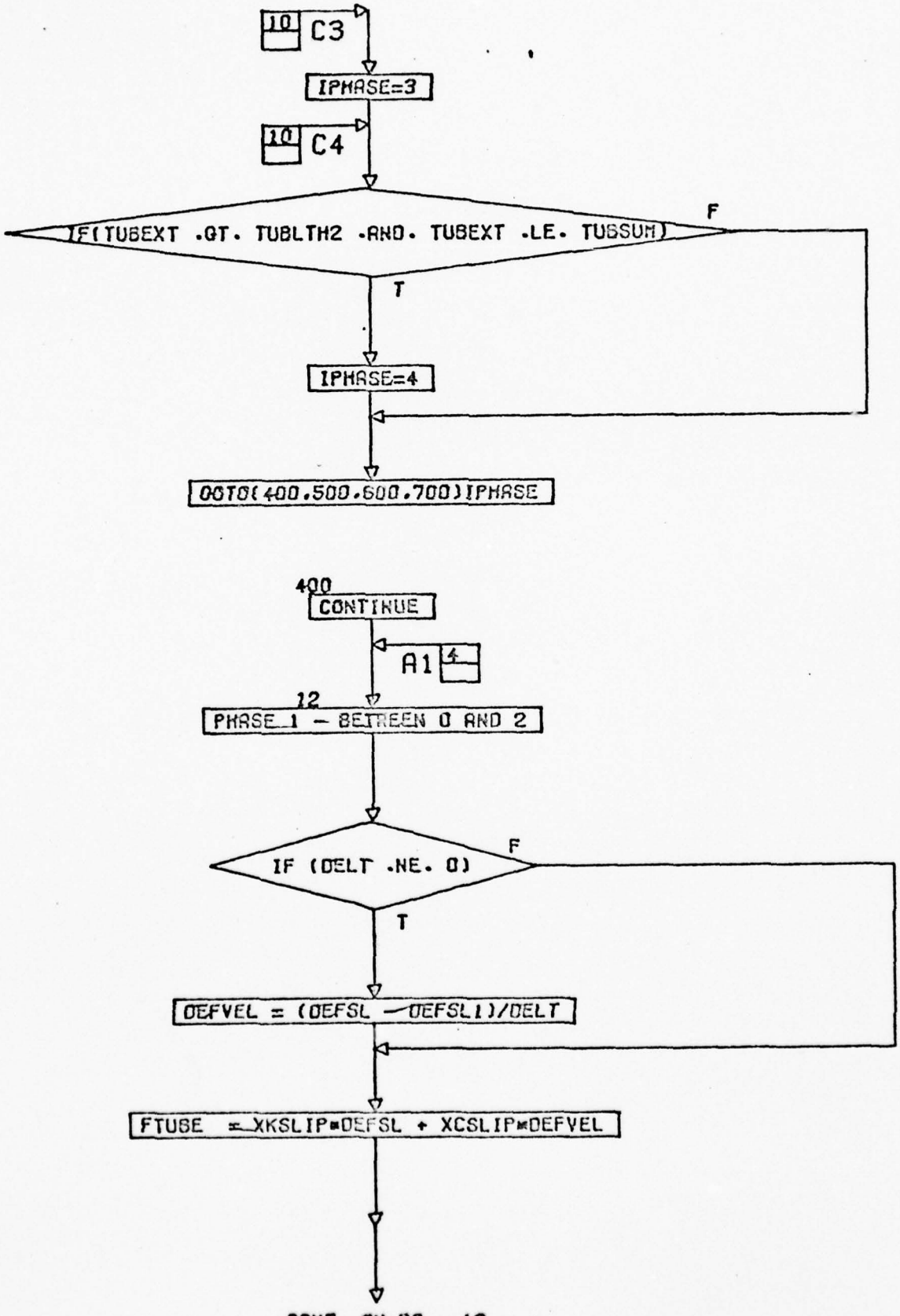
CONT. ON PG 9

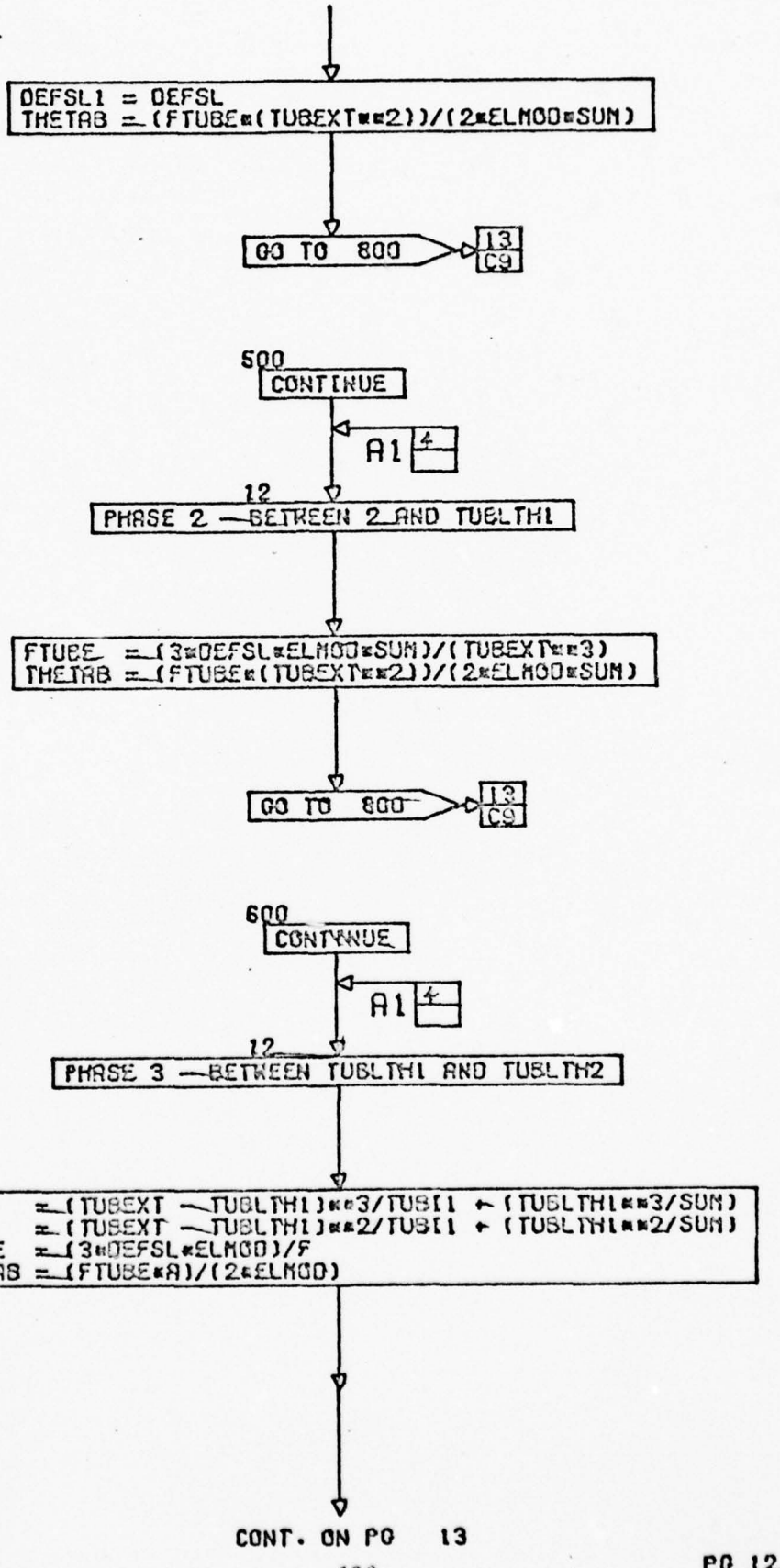

A1 4
12
CHECK IF TUBES ARE FULLY EXTENDED
IF (TUBEXT .LT. TUBLTH1 + TUBLTH2)
F
T
GO TO 300
ISLID=1
RETURN
300
CONTINUE
A1 4
12
COMPUTE DEFLECTION
DEFSL =SQRT(XCCTCP**2 +ZCCTCP**2)
A1 4
12
COMPUTE TUBE BENDING FORCE AND ANGLE WHICH ARE CONDITIONAL ON
A1 4
12
LENGTH OF TUBE EXTENTION

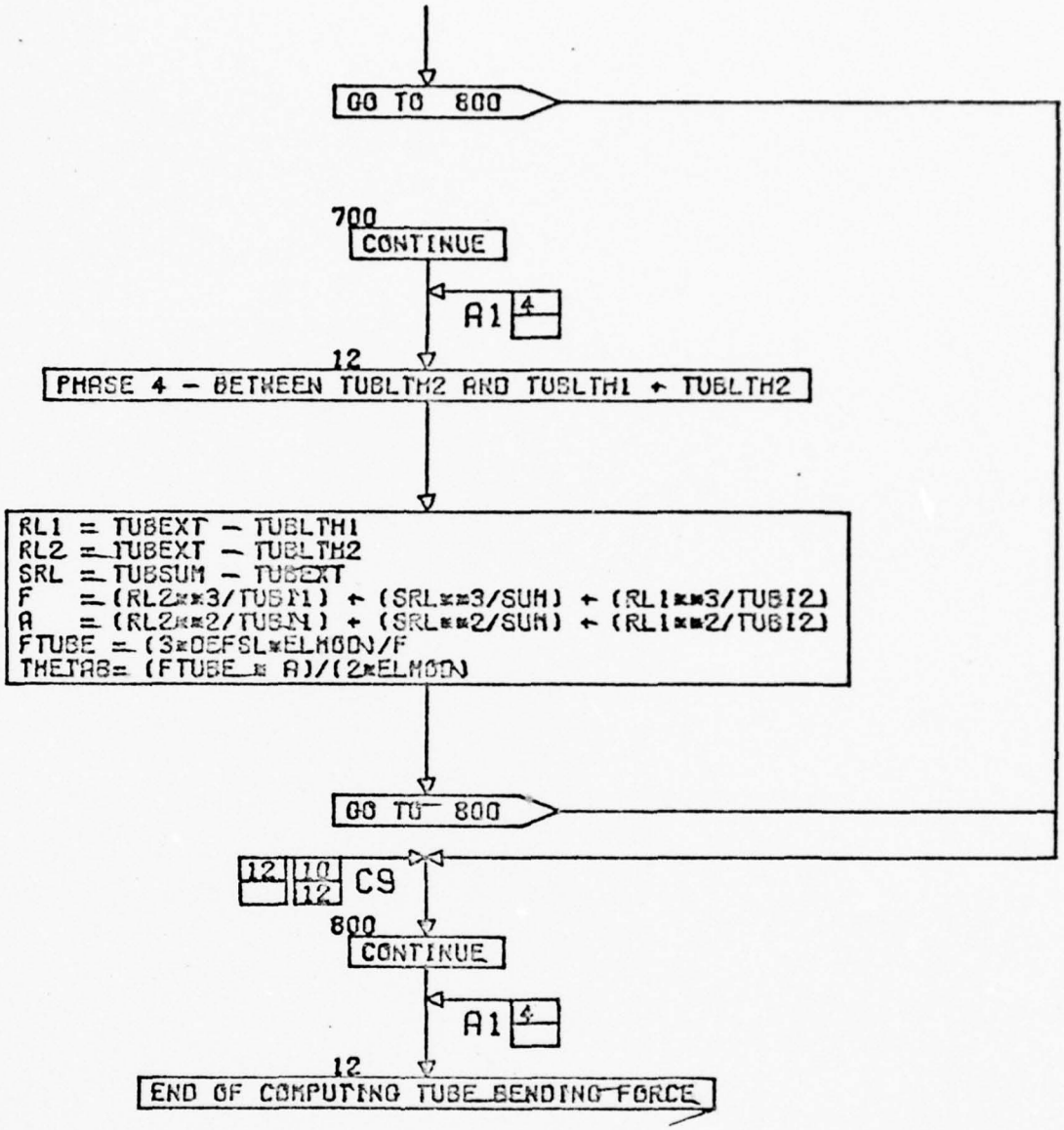
CONT. ON PG 10

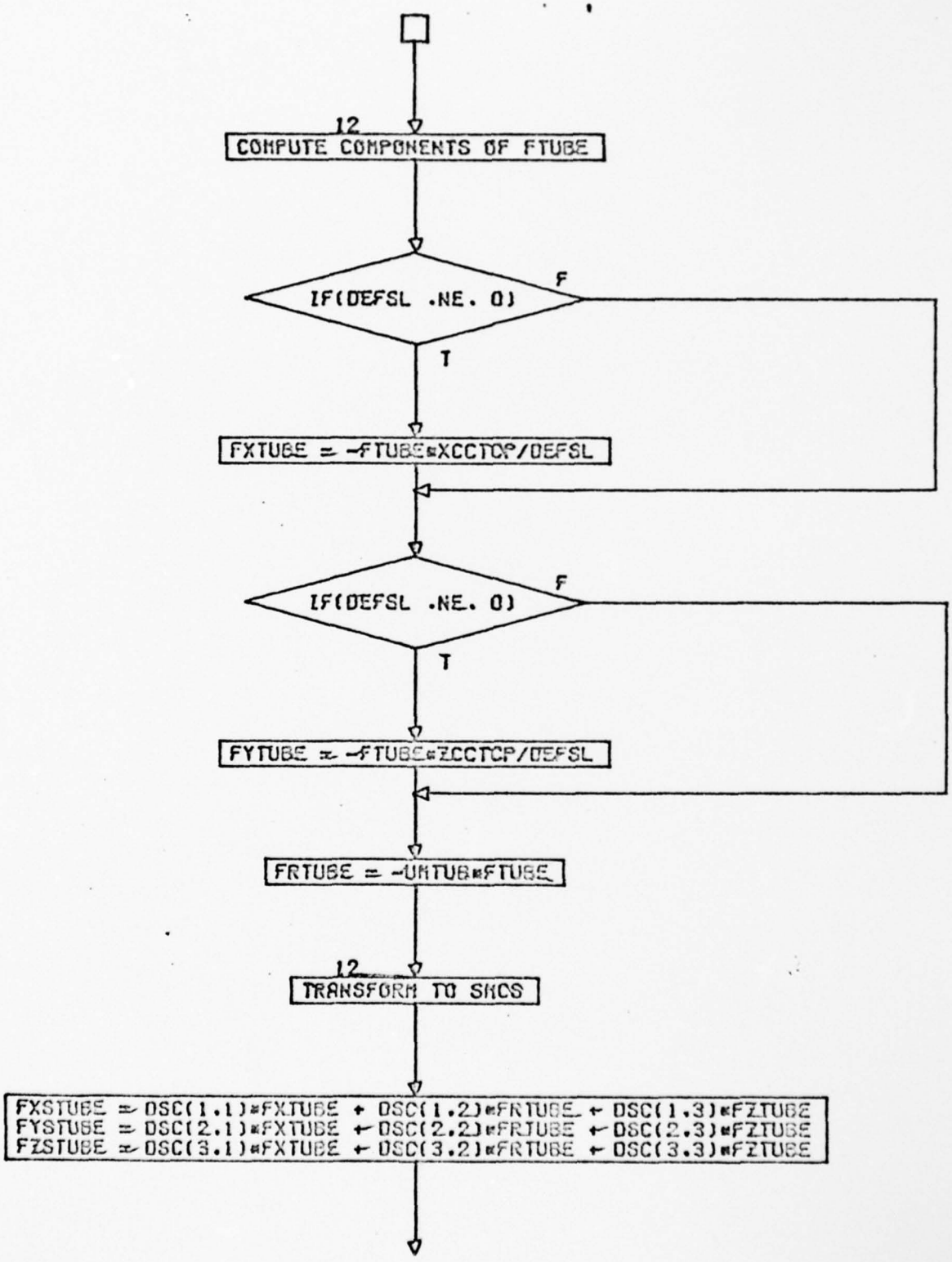

SUM = TUBI1 + TUBI2
TUBSUM = TUBLTH1 + TUBLTH2
IF(TUBEXT .LT. 0 .AND. TUBEXT .GT. TUBSUM)
F
T
GO TO 800
13
C9
IF(TUBEXT .GE. 0 .AND. TUBEXT .LE. 2)
F
T
IPHASE = 1
IF(TUBEXT .GT. 2 .AND. TUBEXT .LE. TUBLTH1)
F
T
IPHASE=2
IF(TUBEXT .GT. TUBLTH1 .AND. TUBEXT .LE. TUBLTH2)
F
11
C4
T
11
C3

CONT. ON PG 11

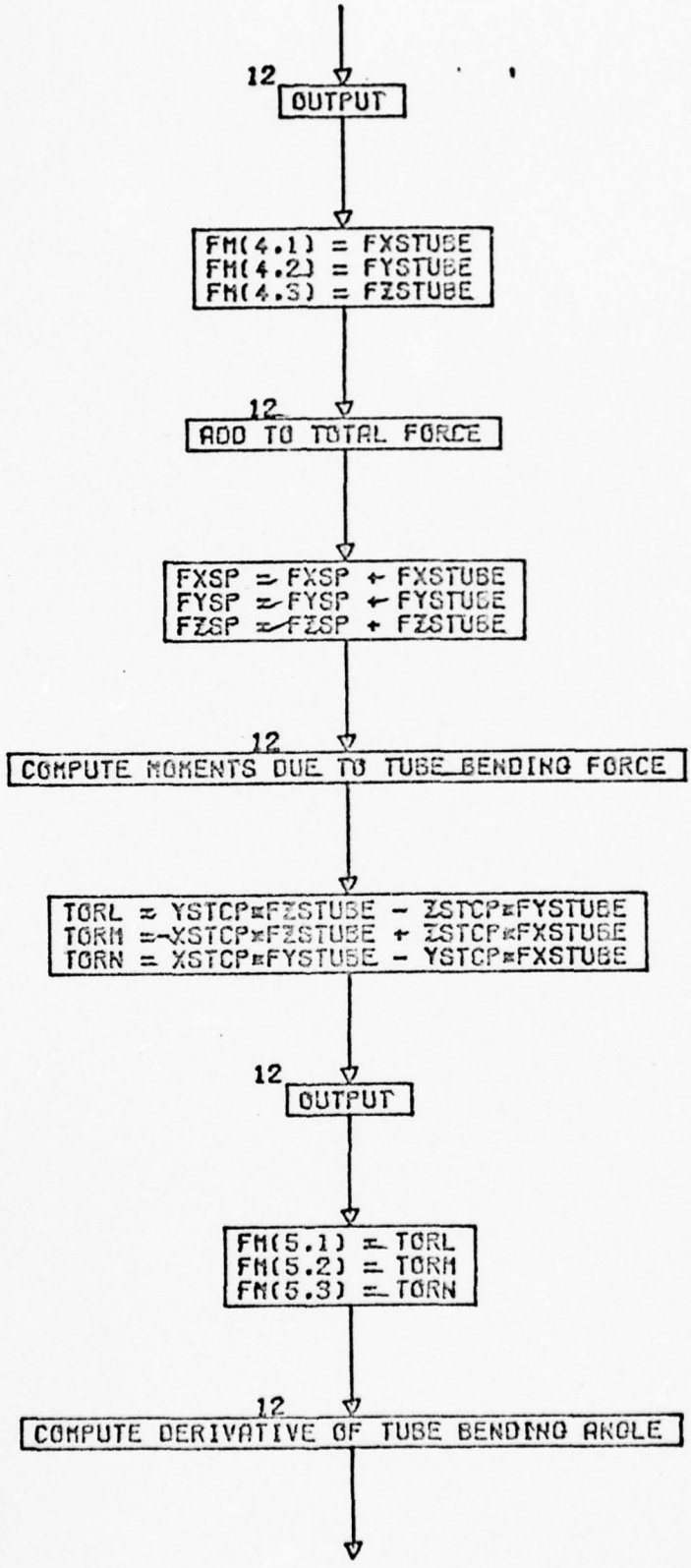

10 C3
IPHASE=3
10 C4
IF(TUBEXT .GT. TUBLTH2 .AND. TUBEXT .LE. TUBSUM)
F
T
IPHASE=4
GOTO(400,500,600,700)IPHASE
400
CONTINUE
A1 4
12
PHASE_1 = BETWEEN 0 AND 2
IF (DELT .NE. 0)
F
T
DEFVEL = (DEFSL - DEFSL1)/DELT
FTUBE = XKSLIP*DEFSL + XCSLIP*DEFVEL
CONT. ON PG 12


DEFSL1 = DEFSL
THETAB = (FTUBE**(TUBEXT**2))/(2*ELMOD*SUM)
GO TO 800
13
C9
500
CONTINUE
A1
4
12
PHASE 2 —BETWEEN 2 AND TUBLTH1
FTUBE = (3*DEFSL*ELMOD*SUM)/(TUBEXT**3)
THETAB = (FTUBE**(TUBEXT**2))/(2*ELMOD*SUM)
GO TO 800
13
C9
600
CONTINUE
A1
4
12
PHASE 3 —BETWEEN TUBLTH1 AND TUBLTH2
F = (TUBEXT —TUBLTH1)**3/TUB(1 + (TUBLTH1**3/SUM)
A = (TUBEXT —TUBLTH1)**2/TUB(1 + (TUBLTH1**2/SUM)
FTUBE = (3*DEFSL*ELMOD)/F
THETAB = (FTUBE*A)/(2*ELMOD)
CONT. ON PG 13

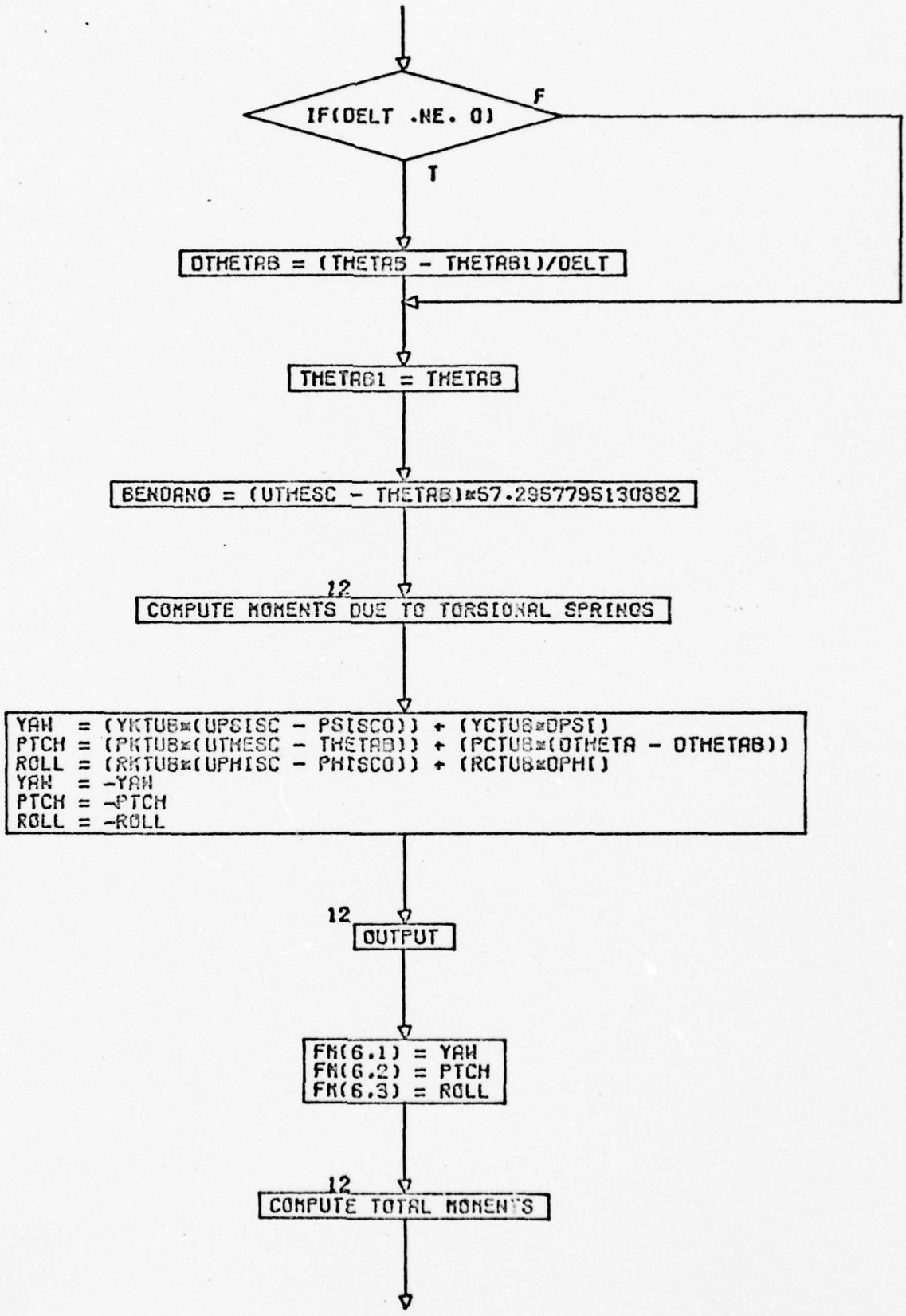

GO TO 800
700
CONTINUE
A1
4
12
PHASE 4 - BETWEEN TUBLTH2 AND TUBLTH1 + TUBLTH2
RL1 = TUBEXT - TUBLTH1
RL2 = TUBEXT - TUBLTH2
SRL = TUBSUM - TUBEXT
F = (RL2**3/TUBI1) + (SRL**3/SUM) + (RL1**3/TUBI2)
A = (RL2**2/TUBI1) + (SRL**2/SUM) + (RL1**2/TUBI2)
FTUBE = (3*DEFSL*ELMODN/F
THETAB= (FTUBE * A)/(2*ELMODN)
GO TO 800
12
10
12
C9
800
CONTINUE
A1
4
12
END OF COMPUTING TUBE BENDING FORCE


12
COMPUTE COMPONENTS OF FTUBE
IF(DEFSL .NE. 0)
F
T
FXTUBE = -FTUBE*XCCTOP/DEFSL
IF(DEFSL .NE. 0)
F
T
FYTUBE = -FTUBE*ZCCTOP/DEFSL
FRTUBE = -UNTUB*FTUBE
12
TRANSFORM TO SHCS
FXSTUBE = DSC(1,1)*FXTUBE + DSC(1,2)*FRTUBE + DSC(1,3)*FZTUBE
FYSTUBE = DSC(2,1)*FXTUBE + DSC(2,2)*FRTUBE + DSC(2,3)*FZTUBE
FZSTUBE = DSC(3,1)*FXTUBE + DSC(3,2)*FRTUBE + DSC(3,3)*FZTUBE

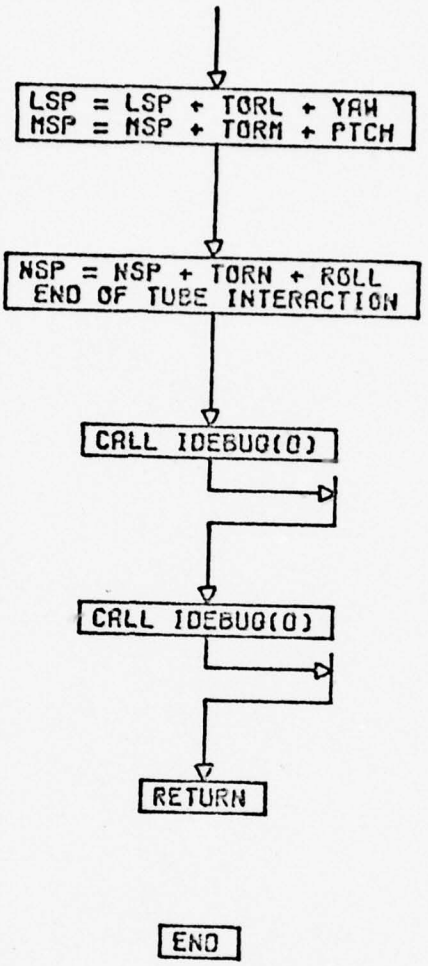
CONT. ON PG 2

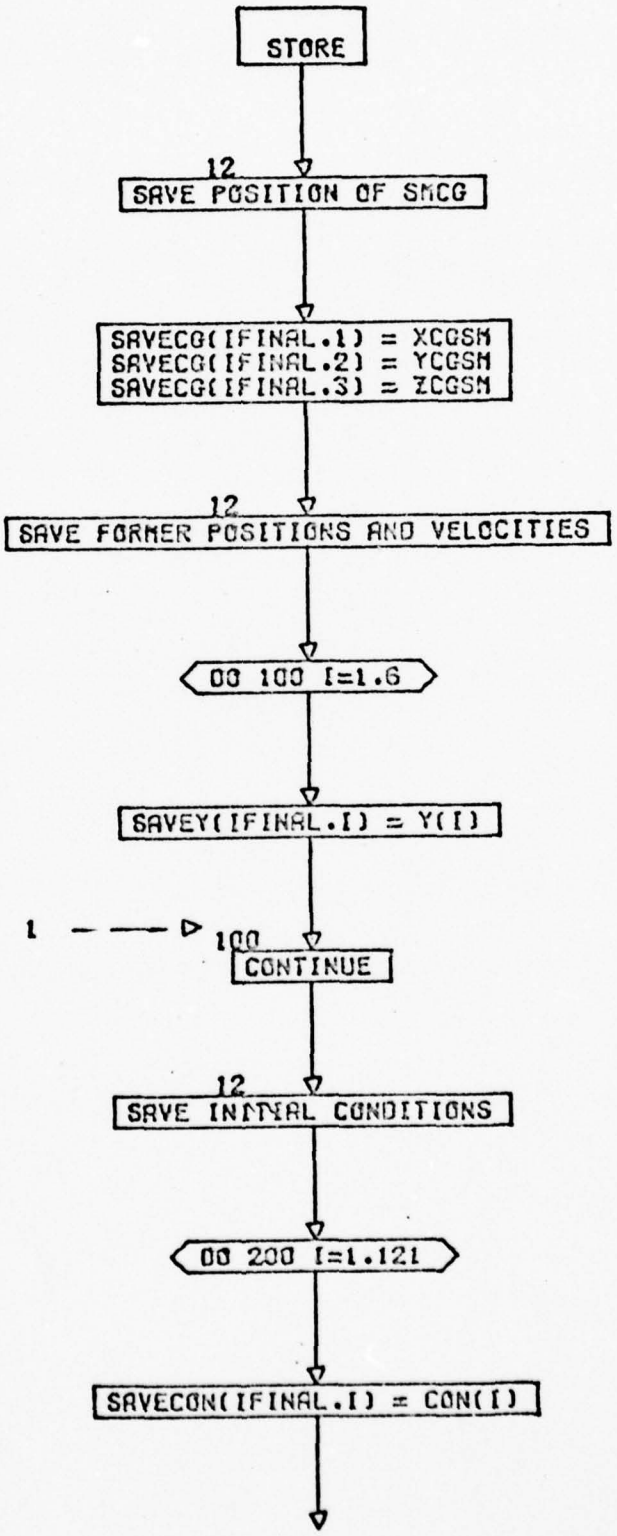

12
OUTPUT
FM(4.1) = FXSTUBE
FM(4.2) = FYSTUBE
FM(4.3) = FZSTUBE
12
ADD TO TOTAL FORCE
FXSP = FXSP + FXSTUBE
FYSP = FYSP + FYSTUBE
FZSP = FZSP + FZSTUBE
12
COMPUTE MOMENTS DUE TO TUBE BENDING FORCE
TORL = YSTCP*FZSTUBE - ZSTCP*FYSTUBE
TORM = -XSTCP*FZSTUBE + ZSTCP*FXSTUBE
TORN = XSTCP*FYSTUBE - YSTCP*FXSTUBE
12
OUTPUT
FM(5.1) = TORL
FM(5.2) = TORM
FM(5.3) = TORN
12
COMPUTE DERIVATIVE OF TUBE BENDING ANGLE

CONT. ON PG 3

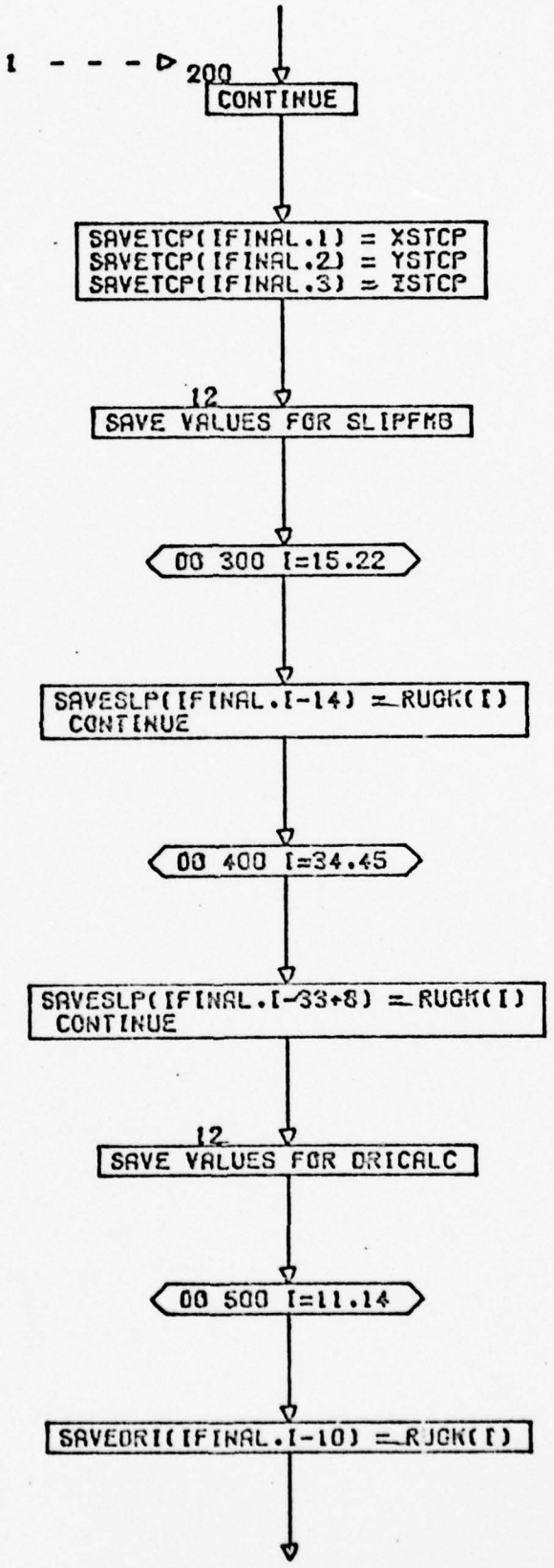

IF(DELT .NE. 0)
F
T
DTHETAB = (THETAB - THETAB1)/DELT
THETAB1 = THETAB
BENDANG = (UTHESC - THETAB)*57.2957795130862
12
COMPUTE MOMENTS DUE TO TORSIONAL SPRINGS
YAW = (YKTUB*(UPSISC - PSISCO)) + (YCTUB*DPSI)
PTCH = (PKTUB*(UTHESC - THETAB)) + (PCTUB*(DTHETA - DTHETAB))
ROLL = (RKTUB*(UPHISC - PHISCO)) + (RCTUB*DPHI)
YAW = -YAW
PTCH = -PTCH
ROLL = -ROLL
12
OUTPUT
FM(6,1) = YAW
FM(6,2) = PTCH
FM(6,3) = ROLL
12
COMPUTE TOTAL MOMENTS

CONT. ON PG 4

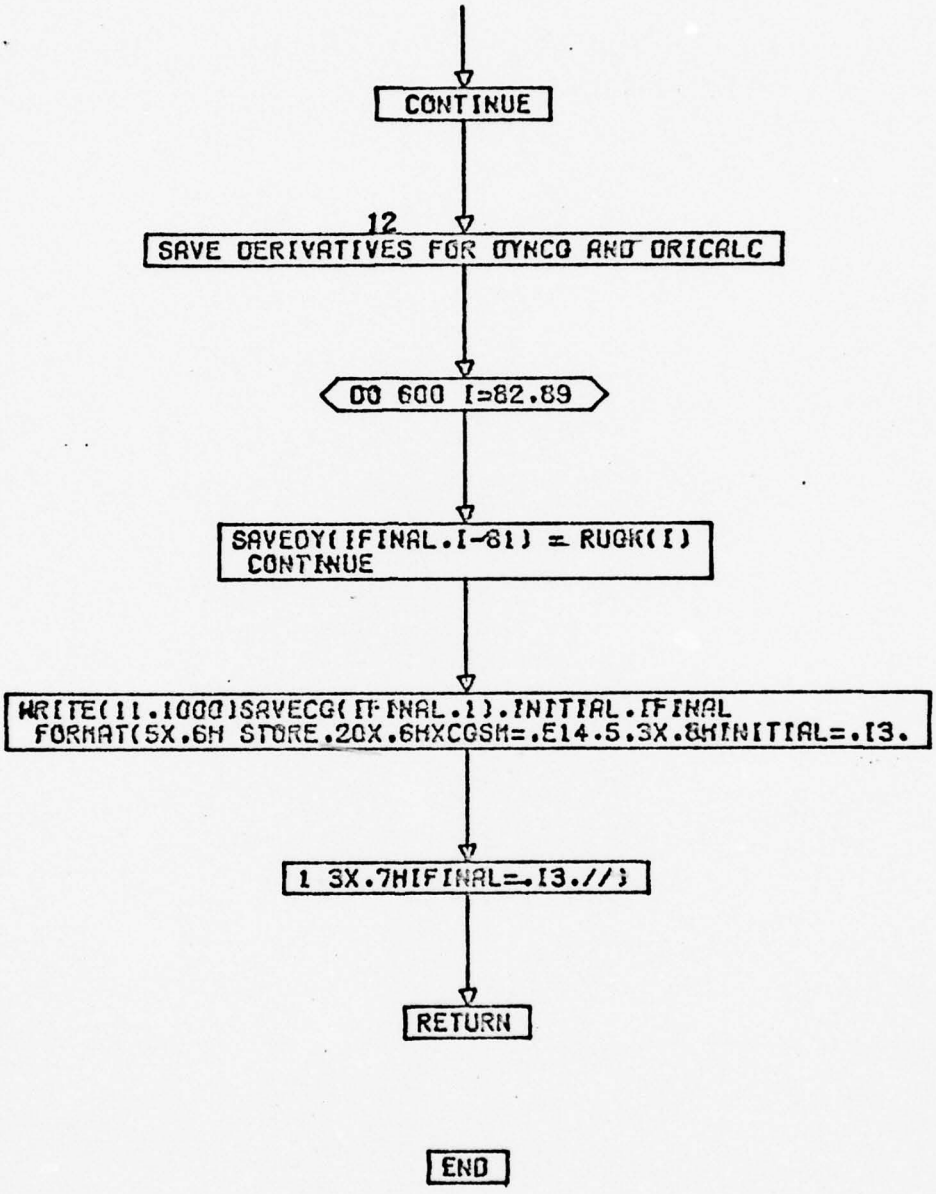

LSP = LSP + TORL + YAW
MSP = MSP + TORM + PTCH
NSP = NSP + TORN + ROLL
END OF TUBE INTERACTION
CALL IDEBUG(0)
CALL IDEBUG(0)
RETURN
END

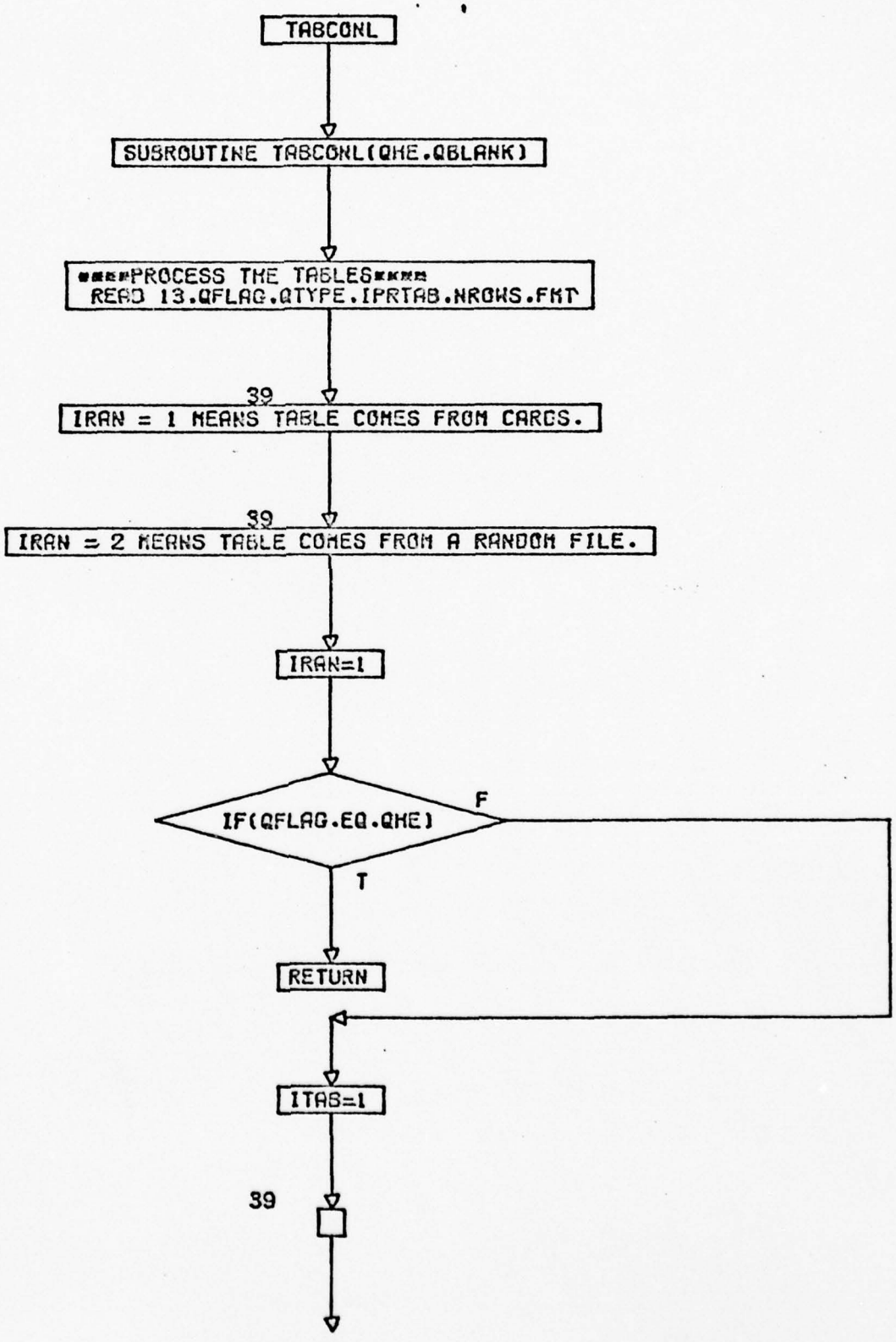

STORE
12
SAVE POSITION OF SMCG
SAVECG(IFINAL,1) = XCGSM
SAVECG(IFINAL,2) = YCGSM
SAVECG(IFINAL,3) = ZCGSM
12
SAVE FORMER POSITIONS AND VELOCITIES
DO 100 I=1,6
SAVEY(IFINAL,I) = Y(I)
1
100
CONTINUE
12
SAVE INITIAL CONDITIONS
DO 200 I=1,121
SAVECON(IFINAL,I) = CON(I)

CONT. ON PG 2

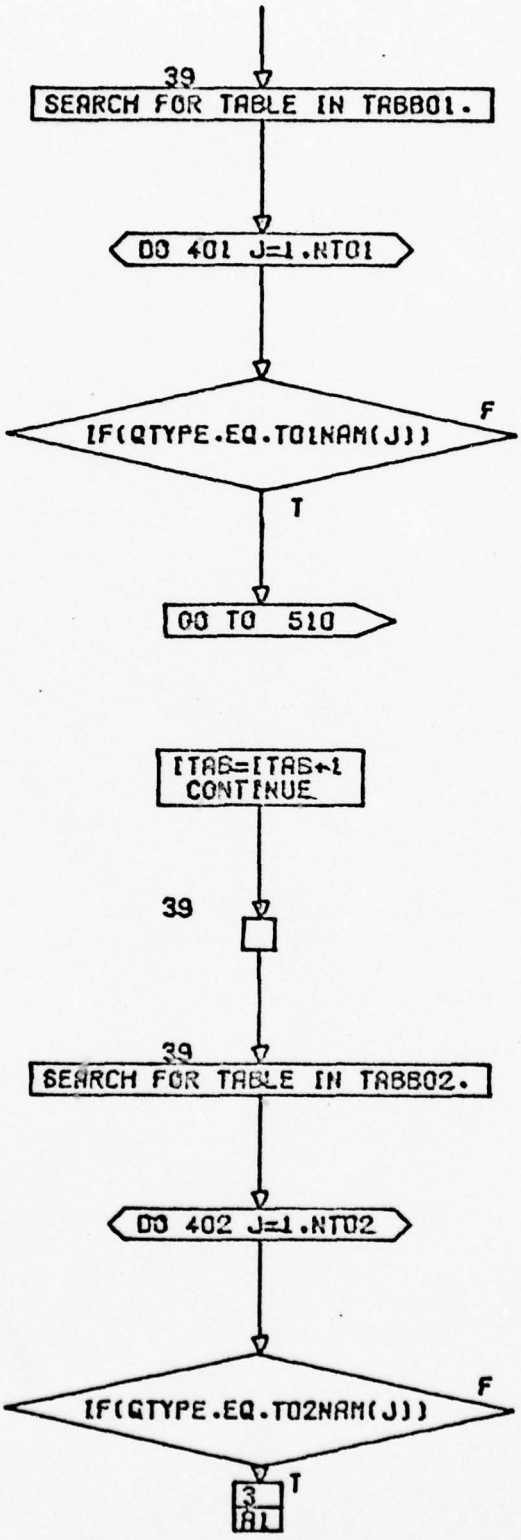

1
200
CONTINUE
SAVETCP(IFINAL.1) = XSTCP
SAVETCP(IFINAL.2) = YSTCP
SAVETCP(IFINAL.3) = ZSTCP
12
SAVE VALUES FOR SLIPFMB
DO 300 I=15.22
SAVESLP(IFINAL.I-14) = RUGK(I)
CONTINUE
DO 400 I=34.45
SAVESLP(IFINAL.I-33+8) = RUGK(I)
CONTINUE
12
SAVE VALUES FOR DRICALC
DO 500 I=11.14
SAVEDRI(IFINAL.I-10) = RUGK(I)

CONT. ON PG 3

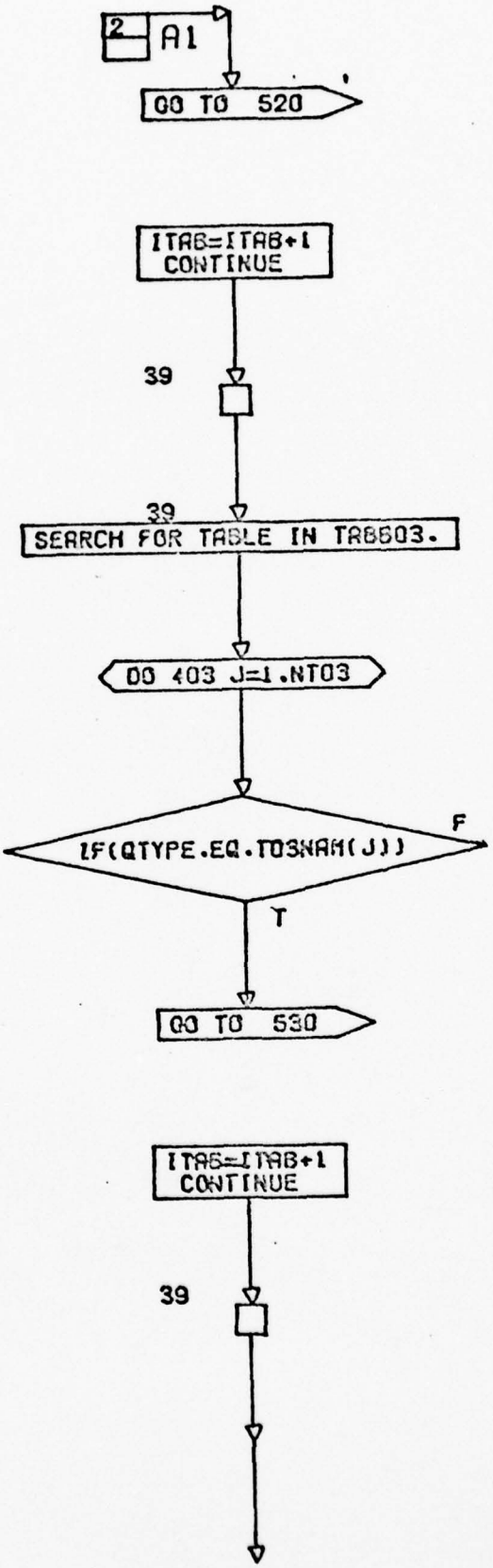

CONTINUE
12
SAVE DERIVATIVES FOR DYNCG AND ORICALC
DO 600 I=82,89
SAVEDY(IFINAL,I-81) = RUGK(I)
CONTINUE
WRITE(11,1000)SAVECG(IFINAL,1),INITIAL,IFINAL
FORMAT(5X,6H STORE,20X,6HXCGSM=,E14.5,3X,8HINITIAL=,I3,
1 3X,7HIFINAL=,I3,//)
RETURN
END

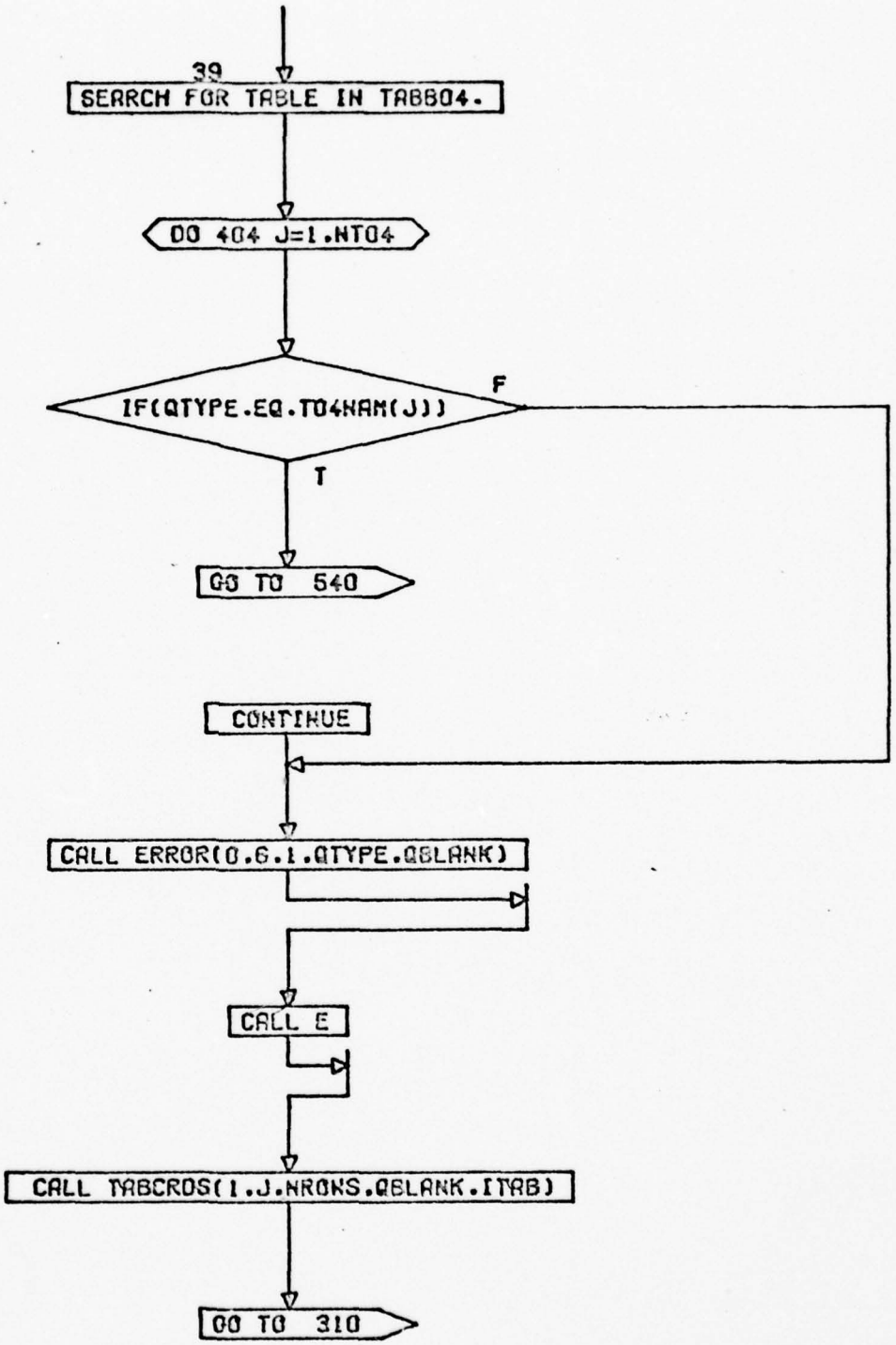

TABCONL
SUBROUTINE TABCONL(QHE.QBLANK)
****PROCESS THE TABLES****
READ 13.QFLAG.QTYPE.IPRTAB.NROWS.FMT
39
IRAN = 1 MEANS TABLE COMES FROM CARDS.
39
IRAN = 2 MEANS TABLE COMES FROM A RANDOM FILE.
IRAN=1
IF(QFLAG.EQ.QHE)
F
T
RETURN
ITAB=1
39

CONT. ON PG 2

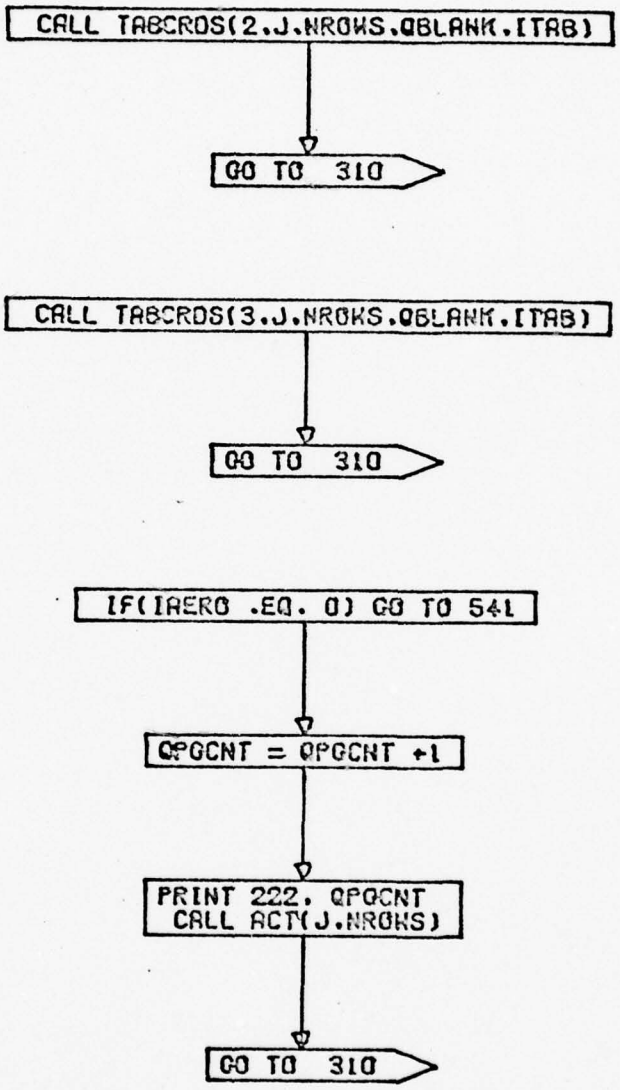

39
SEARCH FOR TABLE IN TABB01.
DO 401 J=1,NT01
IF(QTYPE.EQ.T01NAM(J))
F
T
GO TO 510
ITAB=ITAB+1
CONTINUE
39
39
SEARCH FOR TABLE IN TABB02.
DO 402 J=1,NT02
IF(QTYPE.EQ.T02NAM(J))
F
T
3
81

CONT. ON PG 3

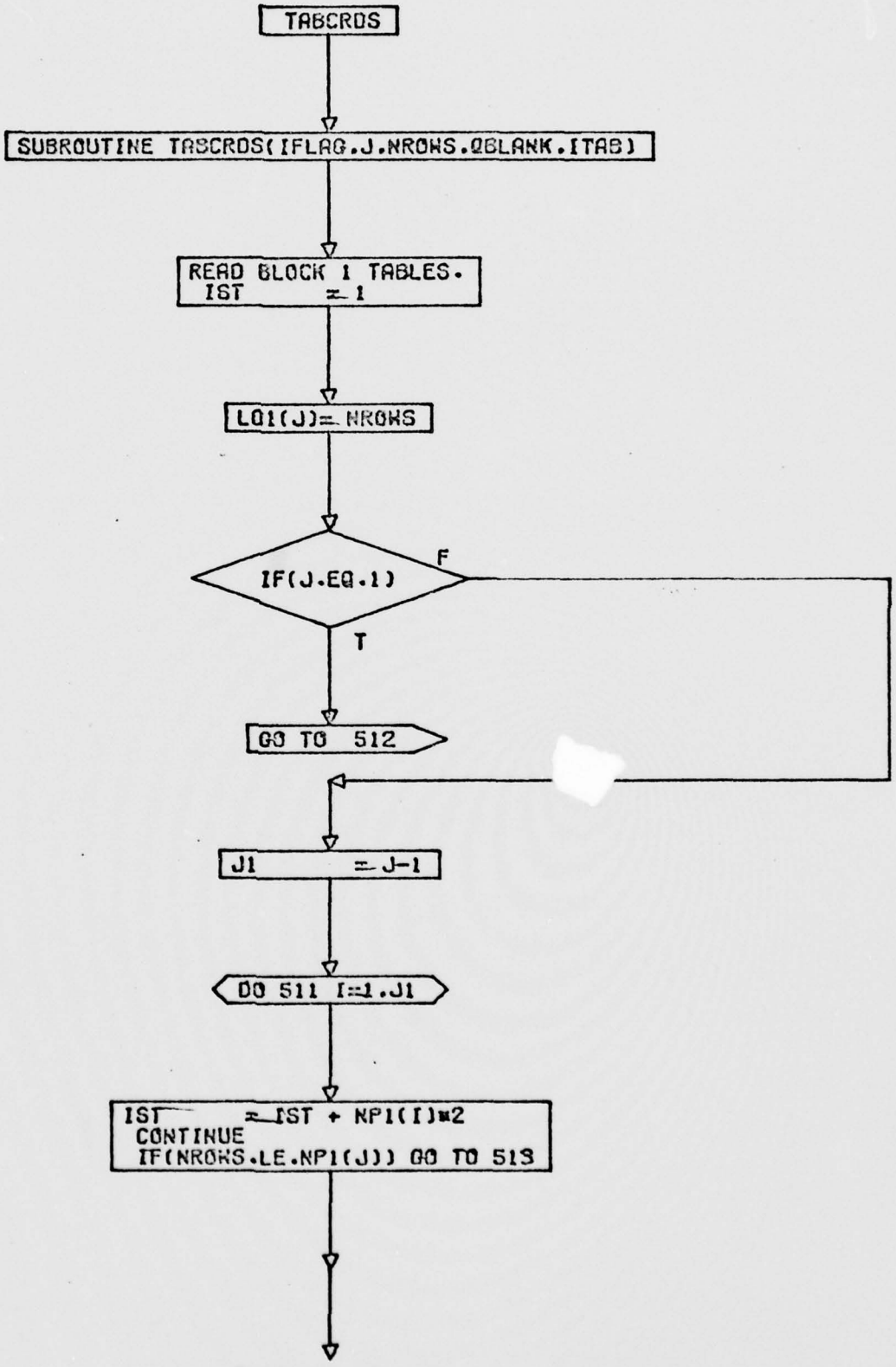

2
A1
GO TO 520
ITAB=ITAB+1
CONTINUE
39
39
SEARCH FOR TABLE IN TR8803.
DO 403 J=1,NT03
IF(QTYPE.EQ.TOSNAM(J))
F
T
GO TO 530
ITAB=ITAB+1
CONTINUE
39
CONT. ON PG 4

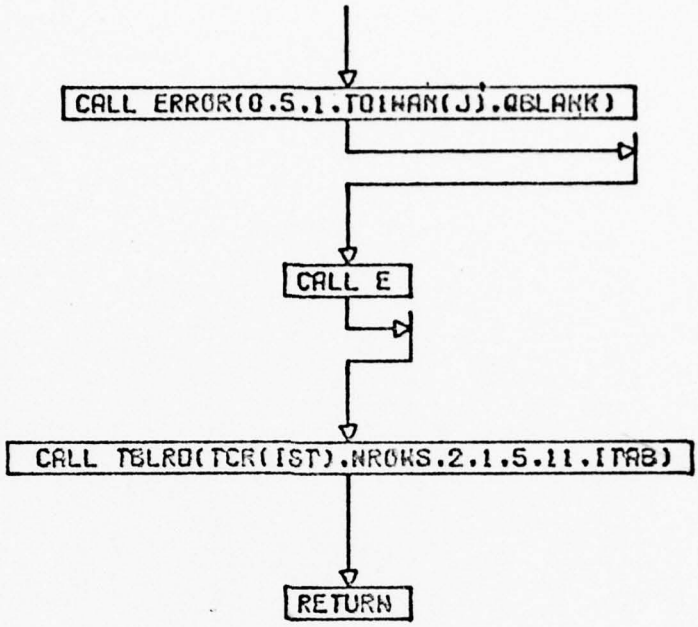

39
SEARCH FOR TABLE IN TAB804.
DO 404 J=1,NT04
IF(QTYPE.EQ.T04NAM(J))
F
T
GO TO 540
CONTINUE
CALL ERROR(0,6,1,QTYPE,QBLANK)
CALL E
CALL TABCROS(1,J,NROWS,QBLANK,ITAB)
GO TO 310

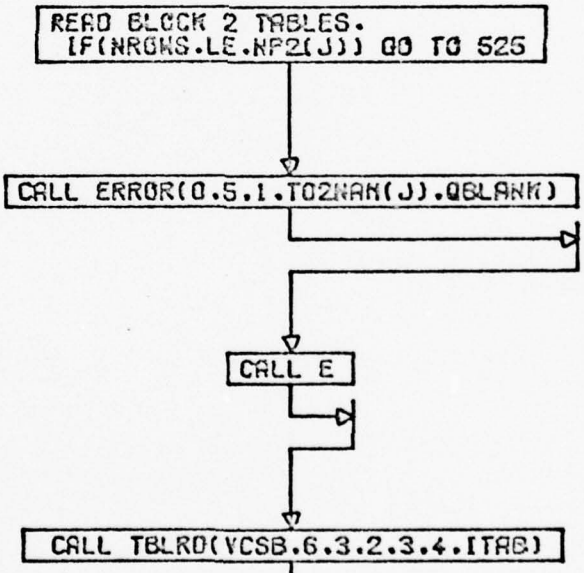
CONT. ON PG 5


CALL TABCRDS(2,J,NROWS,QBLANK,ITAB)
GO TO 310
CALL TABCRDS(3,J,NROWS,QBLANK,ITAB)
GO TO 310
IF(IAERO .EQ. 0) GO TO 541
QPGCNT = QPGCNT +1
PRINT 222, QPGCNT
CALL ACT(J,NROWS)
GO TO 310
END

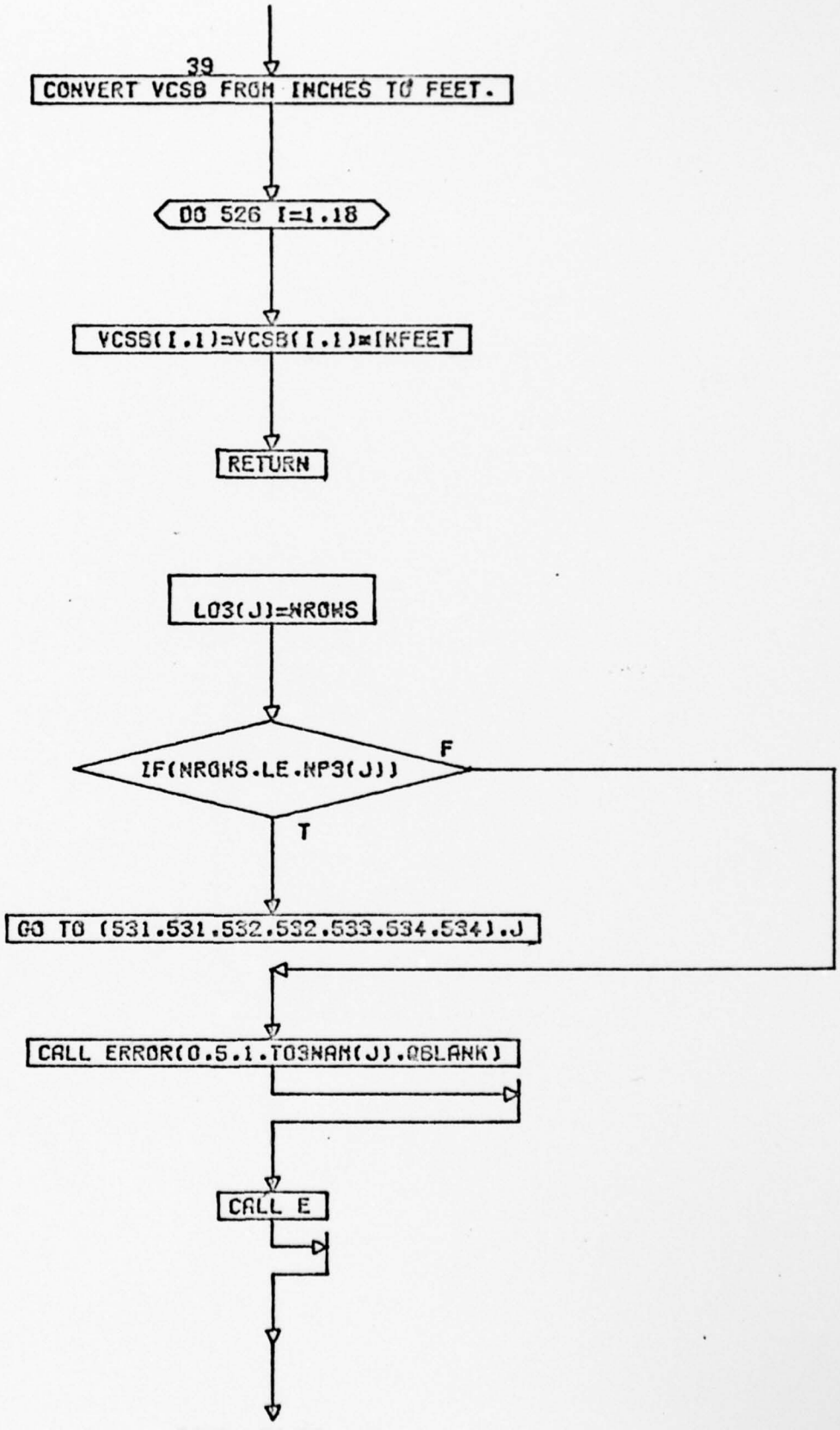
PG 5 FINAL

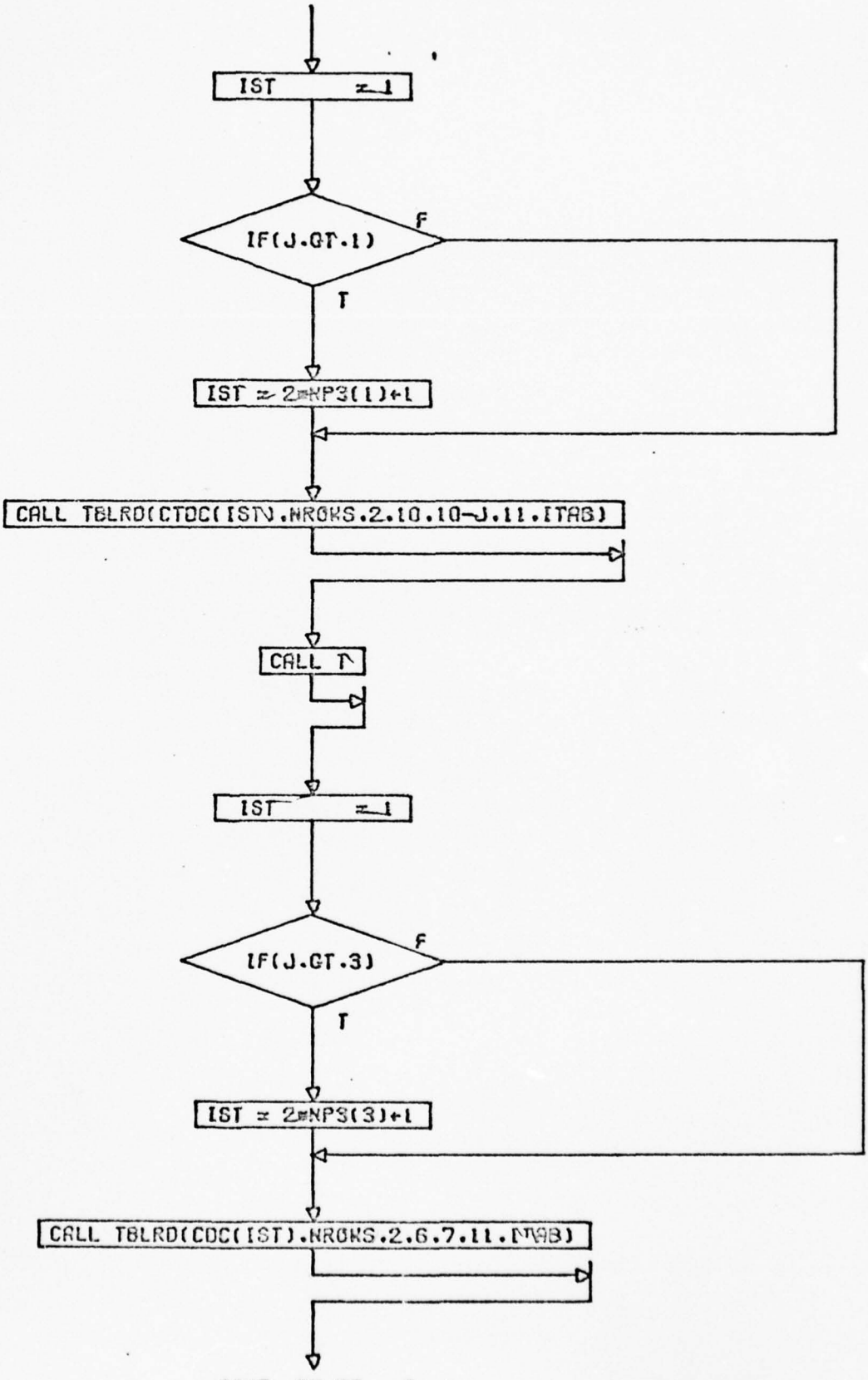

TABCRDS
SUBROUTINE TABCRDS(IFLAG,J,NROWS,QBLANK,ITAB)
READ BLOCK 1 TABLES.
IST = 1
LQ1(J)= NROWS
IF(J.EQ.1)
F
T
GO TO 512
J1 = J-1
DO 511 I=1,J1
IST = IST + NP1(I)*2
CONTINUE
IF(NROWS.LE.NP1(J)) GO TO 513
CONT. ON PG 2

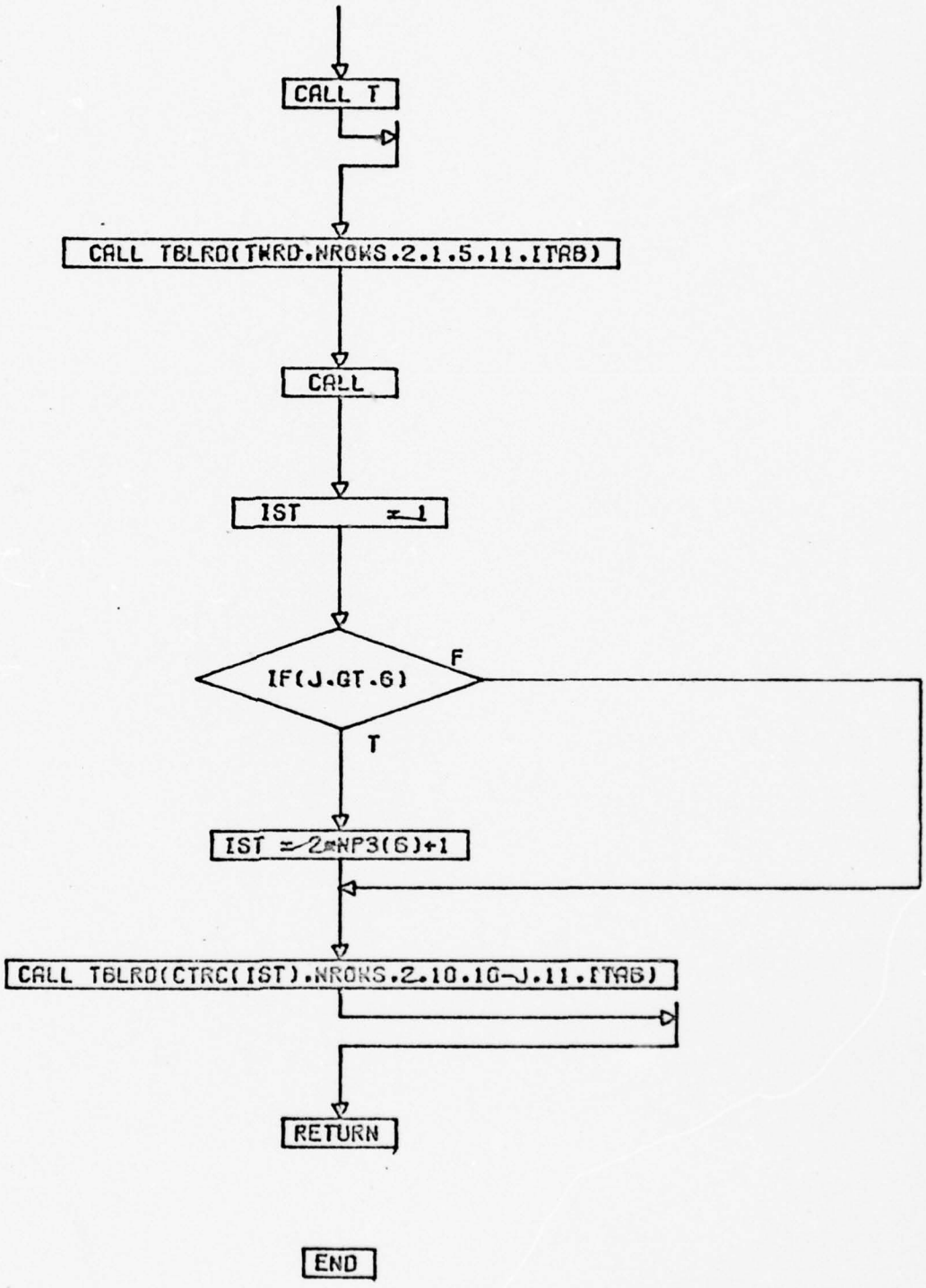

CALL ERROR(0.5.1.TO1NAM(J).QBLANK)
CALL E
CALL TBLRD(TCR(IST).NROWS.2.1.5.11.ITAB)
RETURN


READ BLOCK 2 TABLES.
IF(NROWS.LE.NP2(J)) GO TO 525
CALL ERROR(0.5.1.TO2NAM(J).QBLANK)
CALL E
CALL TBLRD(VCSB.6.3.2.3.4.ITAB)
39

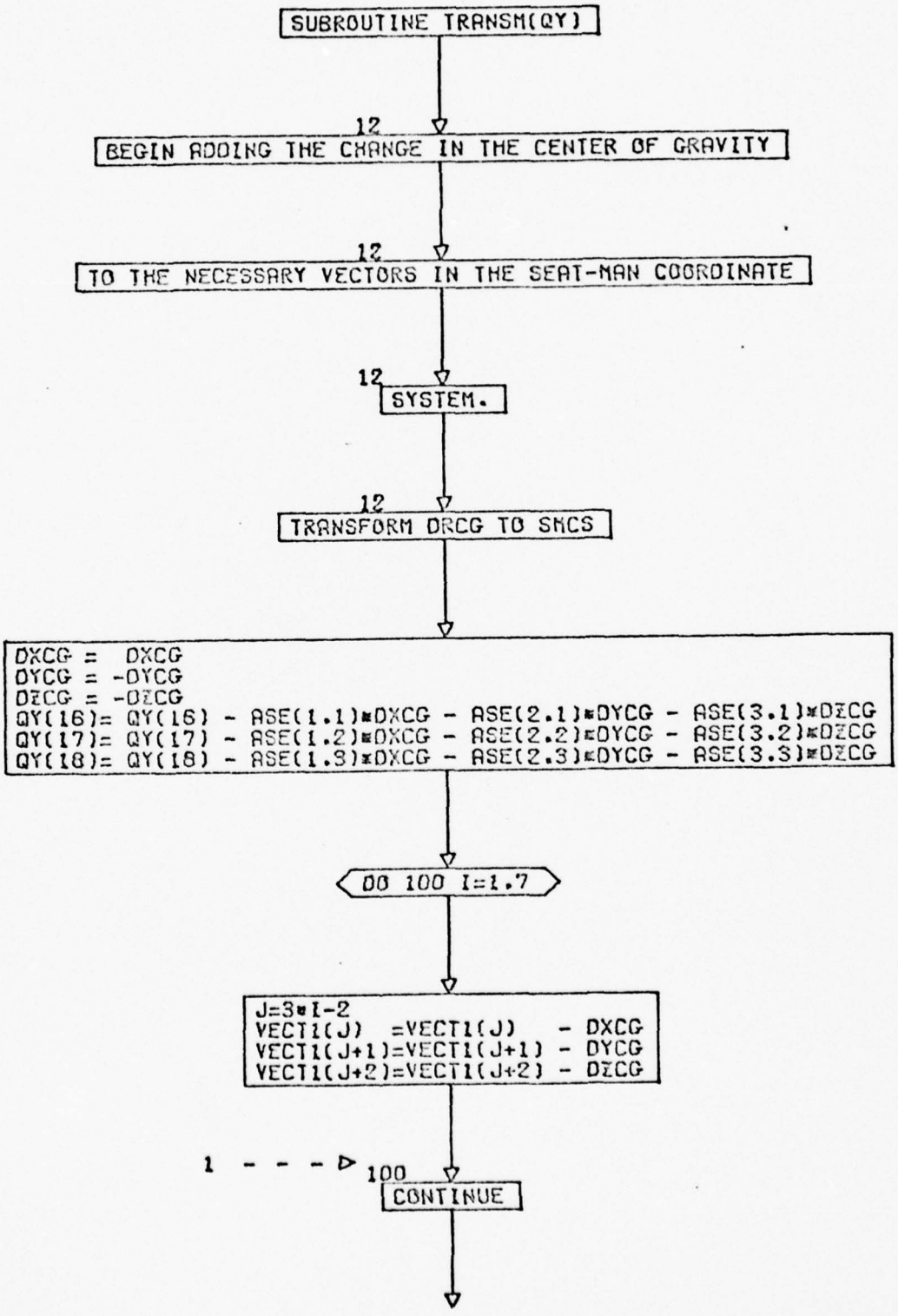
CONT. ON PG 3


39
CONVERT VCSB FROM INCHES TO FEET.
DO 526 I=1,18
VCSB(I,1)=VCSB(I,1)*INFEET
RETURN
LO3(J)=NROWS
IF(NROWS.LE.NP3(J))
F
T
GO TO (531,531,532,532,533,534,534),J
CALL ERROR(0,5,1,TO3NAM(J),QBLANK)
CALL E
CONT. ON PG 4

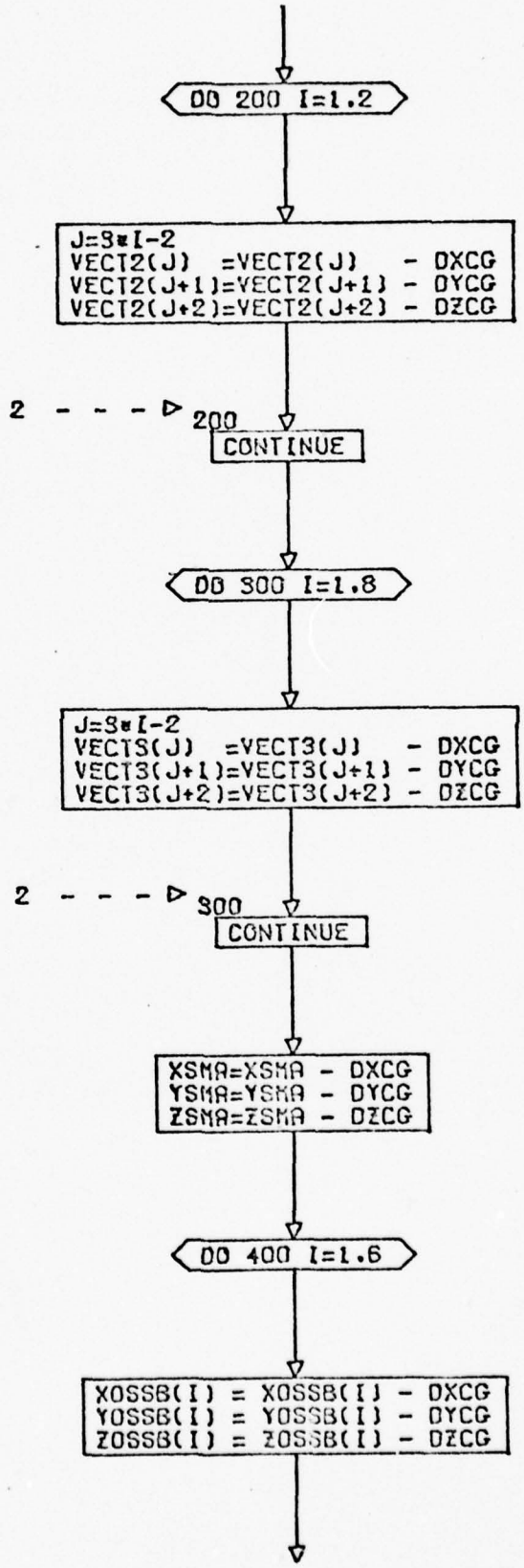

IST = 1
IF(J.GT.1)
F
T
IST = 2*NP3(1)+1
CALL TBLRD(CTOC(IST),NROWS,2,10,10-J,11,ITAB)
CALL T
IST = 1
IF(J.GT.3)
F
T
IST = 2*NP3(3)+1
CALL TBLRD(CDC(IST),NROWS,2,6,7,11,ITAB)
CONT. ON PG 5


CALL T
CALL TBLRD(TWRD,NROWS,2,1,5,11,ITAB)
CALL
IST =1
IF(J.GT.6)
F
T
IST =2*NP3(6)+1
CALL TBLRD(CTRC(IST),NROWS,2,10,16-J,11,ITAB)
RETURN
END

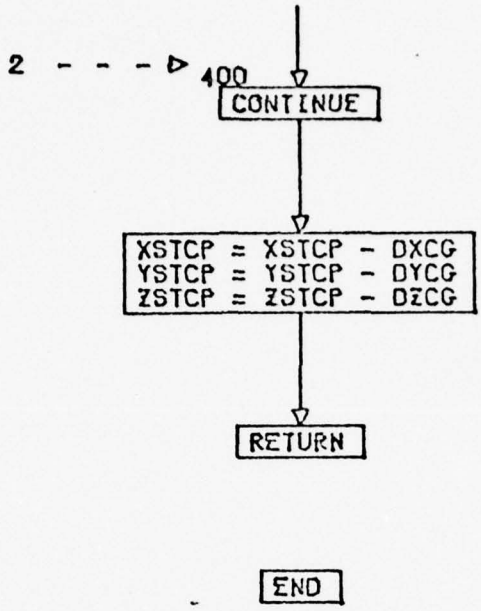
PQ 5 FINAL


SUBROUTINE TRANSM(QY)
12
BEGIN ADDING THE CHANGE IN THE CENTER OF GRAVITY
12
TO THE NECESSARY VECTORS IN THE SEAT-MAN COORDINATE
12
SYSTEM.
12
TRANSFORM DRCG TO SMCS
DXCG = DXCG
DYCG = -DYCG
DZCG = -DZCG
QY(16)= QY(16) - ASE(1.1)*DXCG - ASE(2.1)*DYCG - ASE(3.1)*DZCG
QY(17)= QY(17) - ASE(1.2)*DXCG - ASE(2.2)*DYCG - ASE(3.2)*DZCG
QY(18)= QY(18) - ASE(1.3)*DXCG - ASE(2.3)*DYCG - ASE(3.3)*DZCG
DO 100 I=1,7
J=3*I-2
VECT1(J) =VECT1(J) - DXCG
VECT1(J+1)=VECT1(J+1) - DYCG
VECT1(J+2)=VECT1(J+2) - DZCG
1
100
CONTINUE

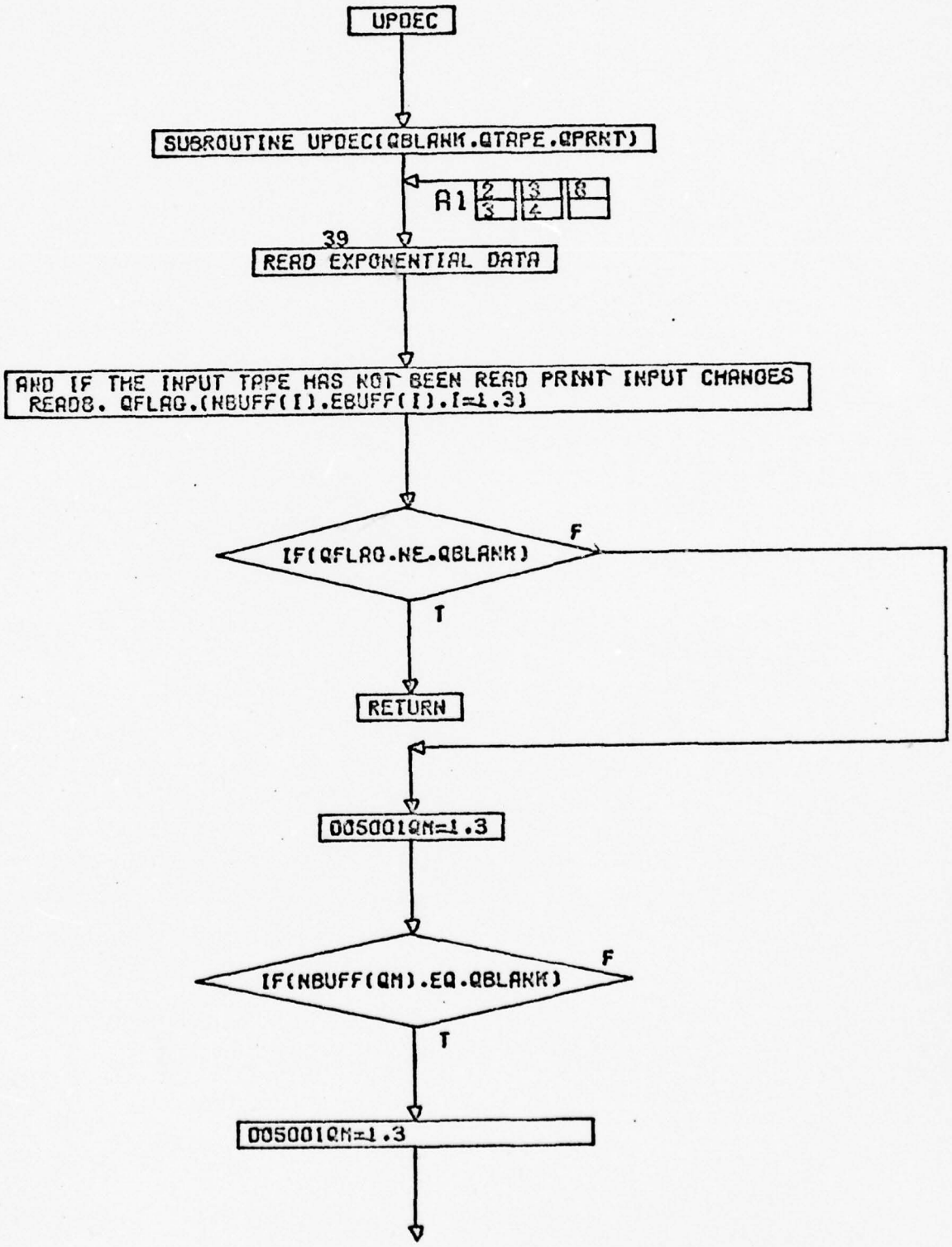
CONT. ON PG 2


DO 200 I=1,2
J=3*I-2
VECT2(J) =VECT2(J) - DXCG
VECT2(J+1)=VECT2(J+1) - DYCG
VECT2(J+2)=VECT2(J+2) - DZCG
2
200
CONTINUE
DO 300 I=1,8
J=3*I-2
VECT3(J) =VECT3(J) - DXCG
VECT3(J+1)=VECT3(J+1) - DYCG
VECT3(J+2)=VECT3(J+2) - DZCG
2
300
CONTINUE
XSMA=XSMA - DXCG
YSMA=YSMA - DYCG
ZSMA=ZSMA - DZCG
DO 400 I=1,6
XOSSB(I) = XOSSB(I) - DXCG
YOSSB(I) = YOSSB(I) - DYCG
ZOSSB(I) = ZOSSB(I) - DZCG

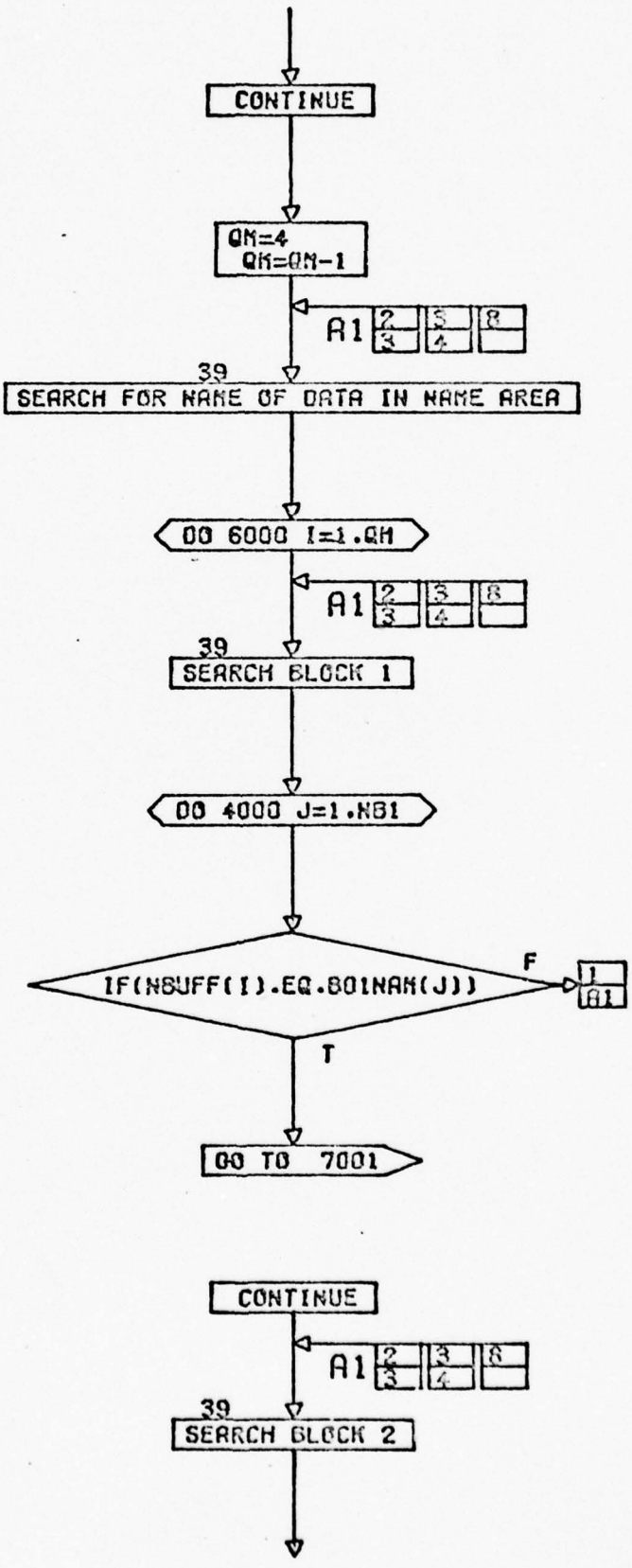
CONT. ON PG 3


2
400
CONTINUE
XSTCP = XSTCP - DXCG
YSTCP = YSTCP - DYCG
ZSTCP = ZSTCP - DZCG
RETURN
END

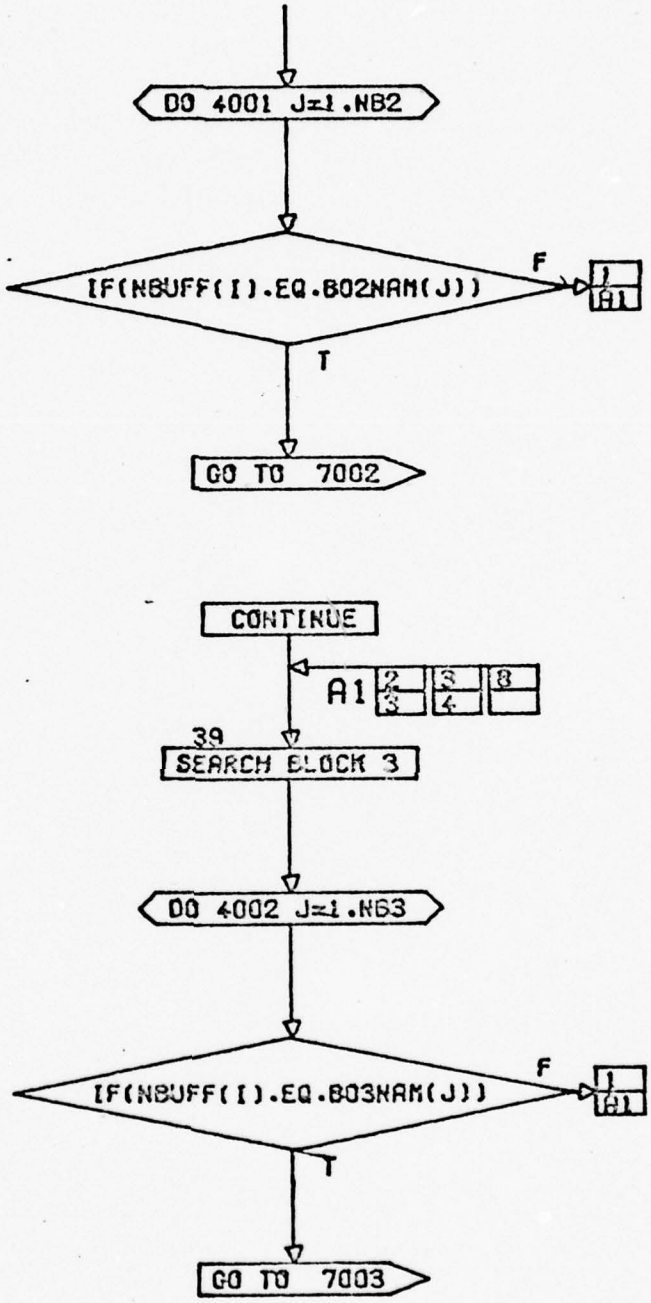
PG 3 FINAL

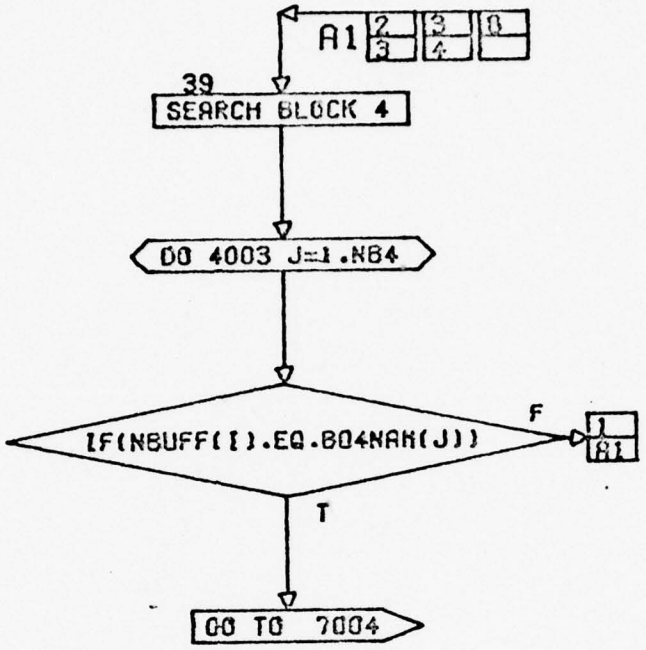


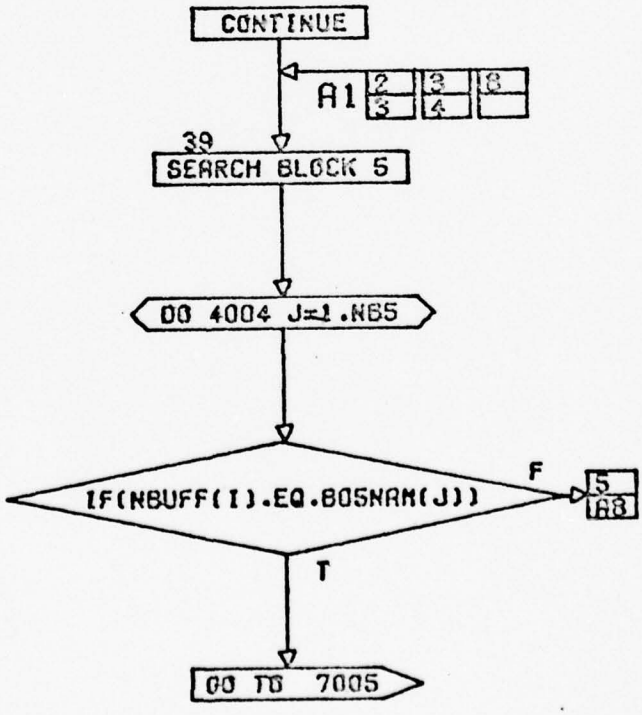
-152- CONT. ON PG 2



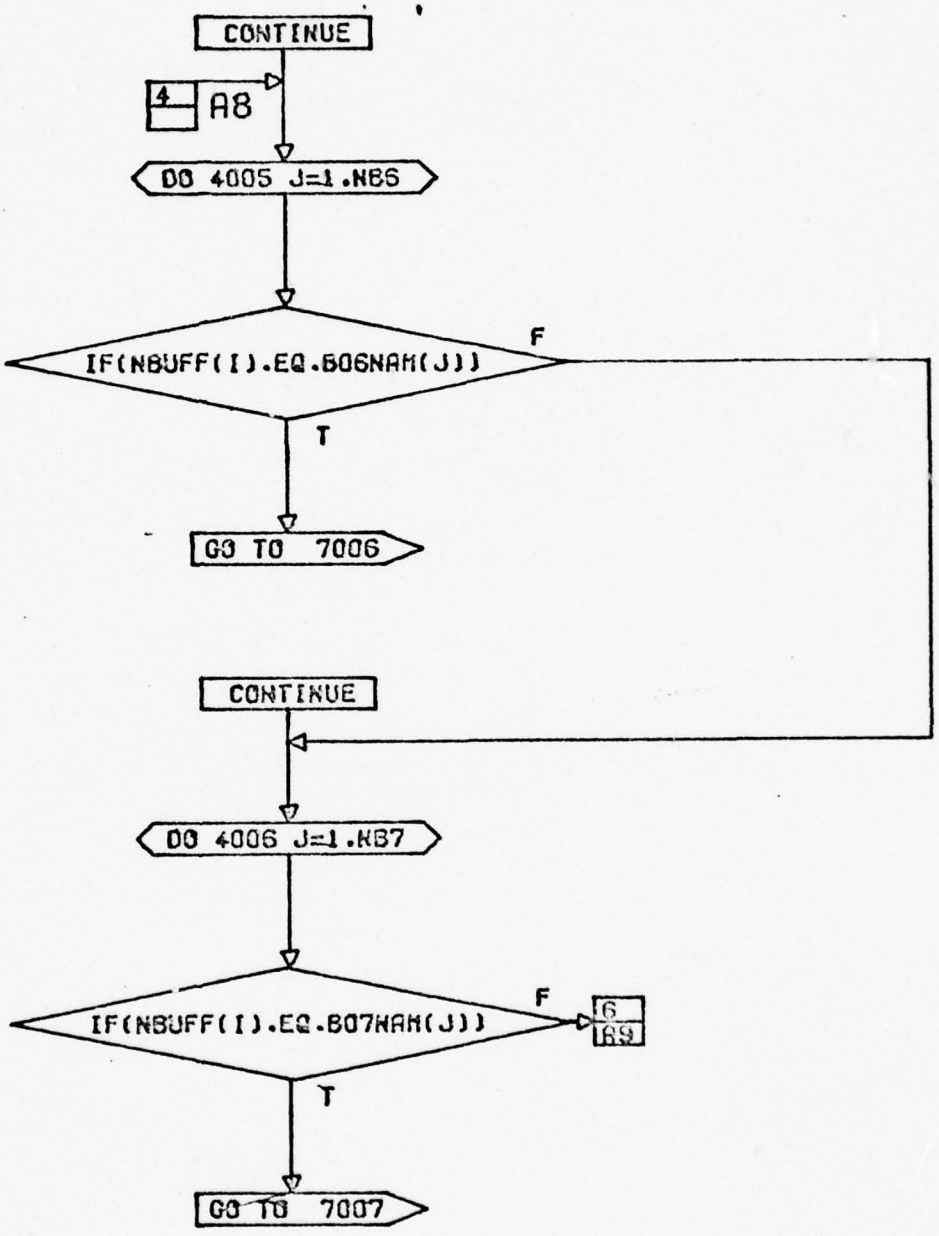
-153- CONT. ON PG 3

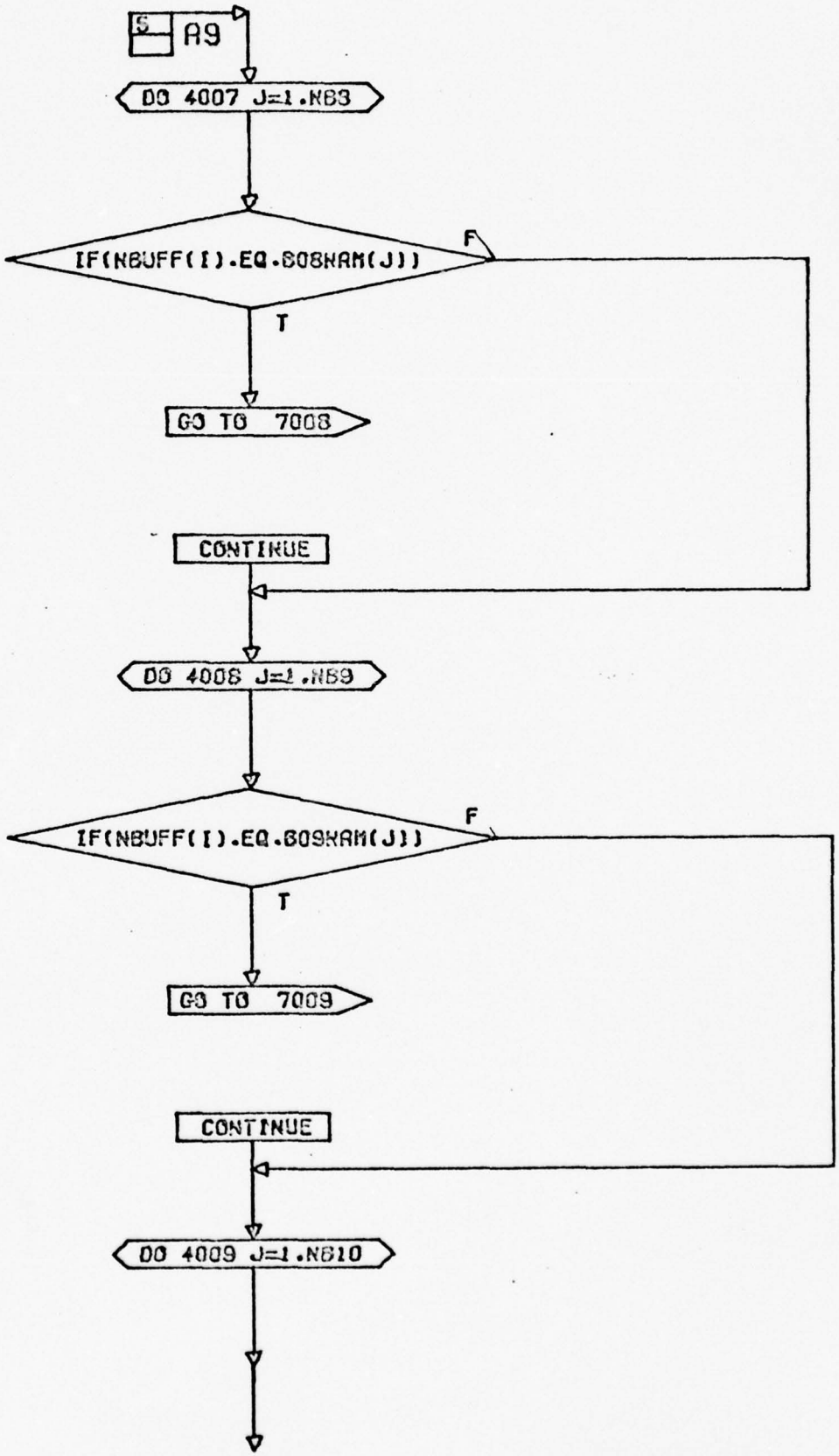

DO 4001 J=1,NB2
IF(NBUFF(I).EQ.B02NAM(J))
F
T
GO TO 7002
CONTINUE
A1
39
SEARCH BLOCK 3
DO 4002 J=1,NB3
IF(NBUFF(I).EQ.B03NAM(J))
F
T
GO TO 7003
CONTINUE
CONT. ON PG 4

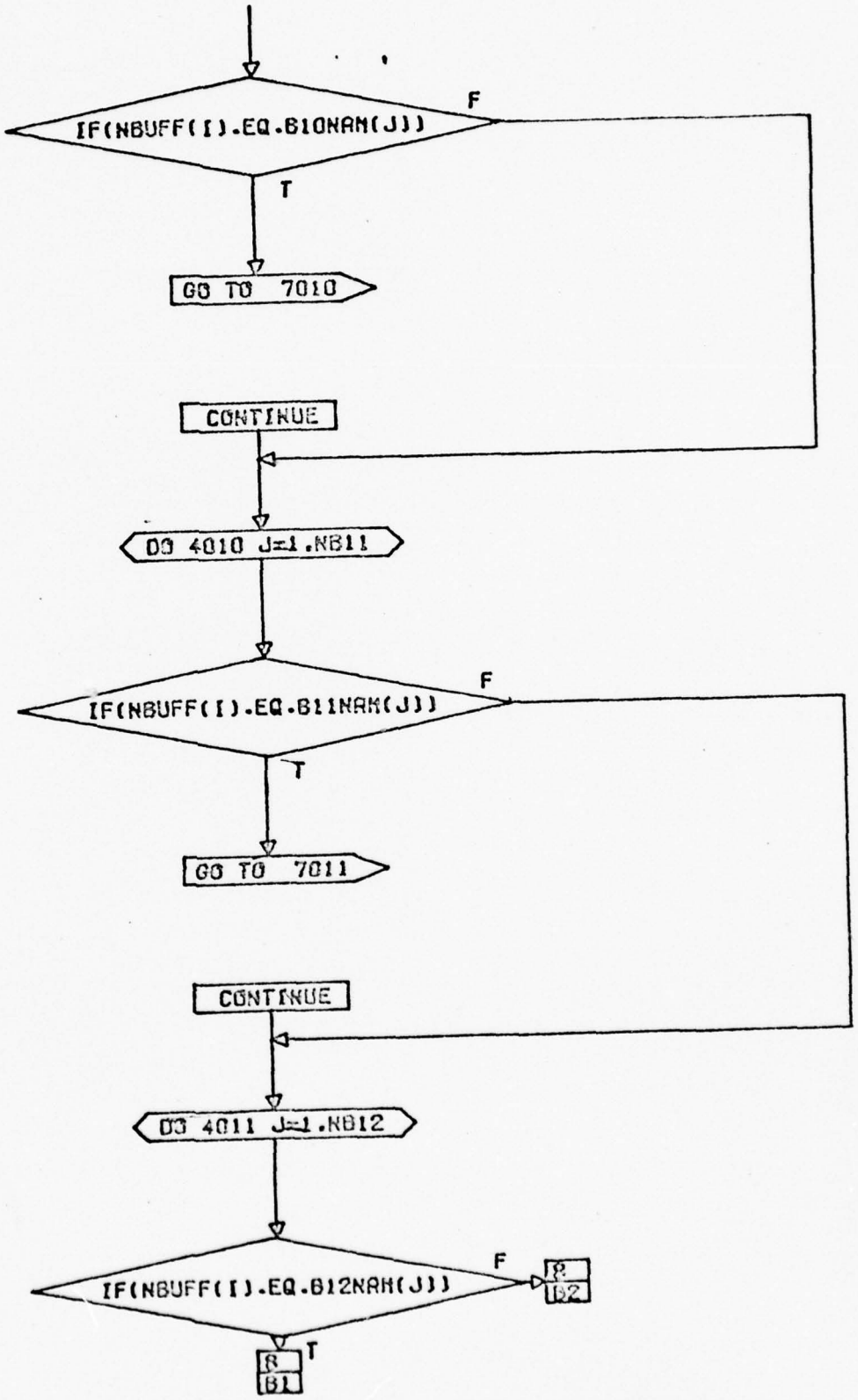

A1
2
3
4
8
39
SEARCH BLOCK 4
DO 4003 J=1,NB4
IF(NBUFF(I).EQ.B04NAM(J))
F
1
81
T
GO TO 7004


CONTINUE
A1
2
3
4
8
39
SEARCH BLOCK 5
DO 4004 J=1,NB5
IF(NBUFF(I).EQ.B05NAM(J))
F
5
83
T
GO TO 7005

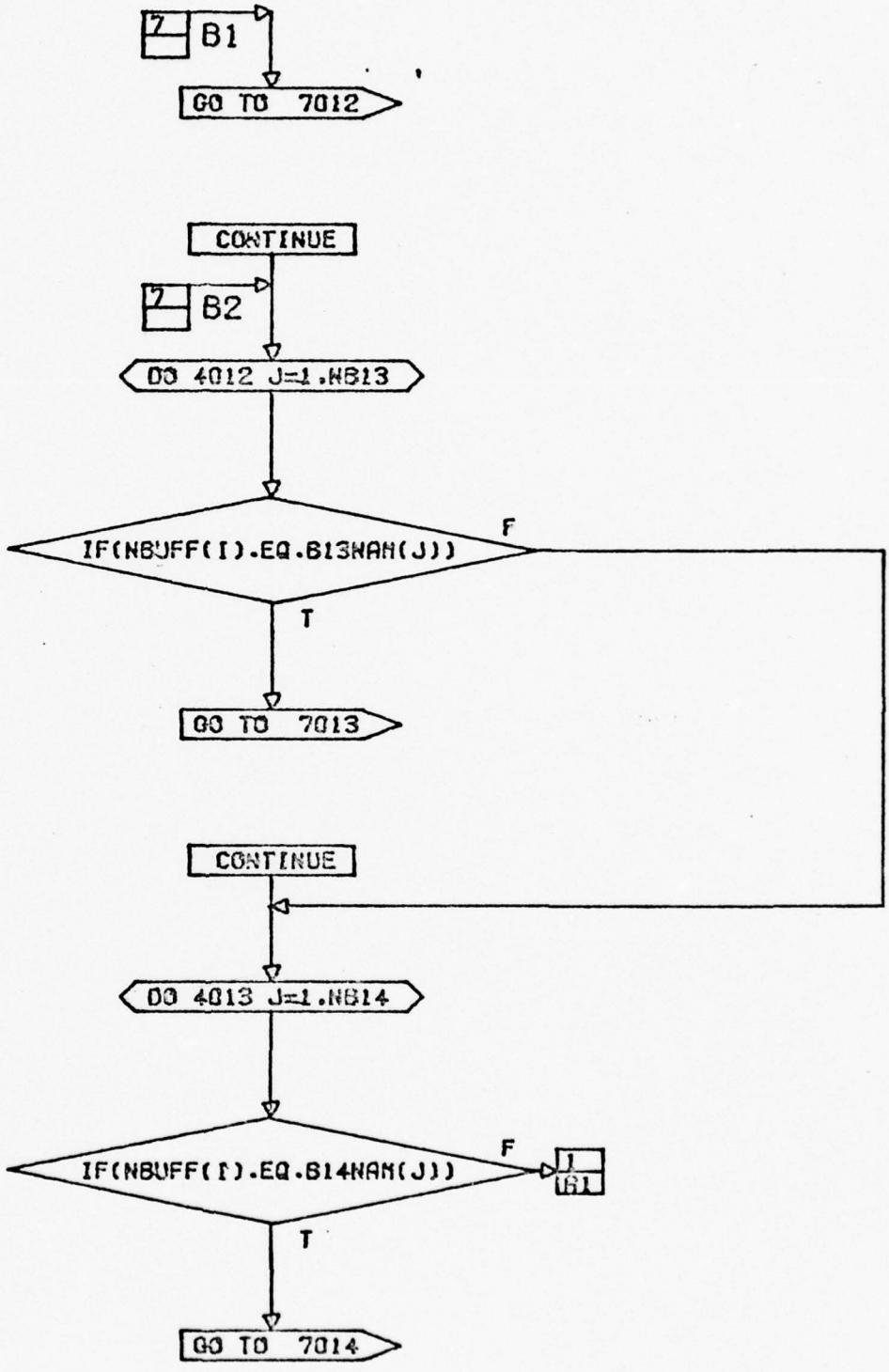
CONT. ON PG 5


CONTINUE
4
A8
DO 4005 J=1,NB6
IF(NBUFF(I).EQ.B06NAM(J))
F
T
GO TO 7006
CONTINUE
DO 4006 J=1,NB7
IF(NBUFF(I).EQ.B07NAM(J))
F
6
A9
T
GO TO 7007
CONTINUE
CONT. ON PG 6

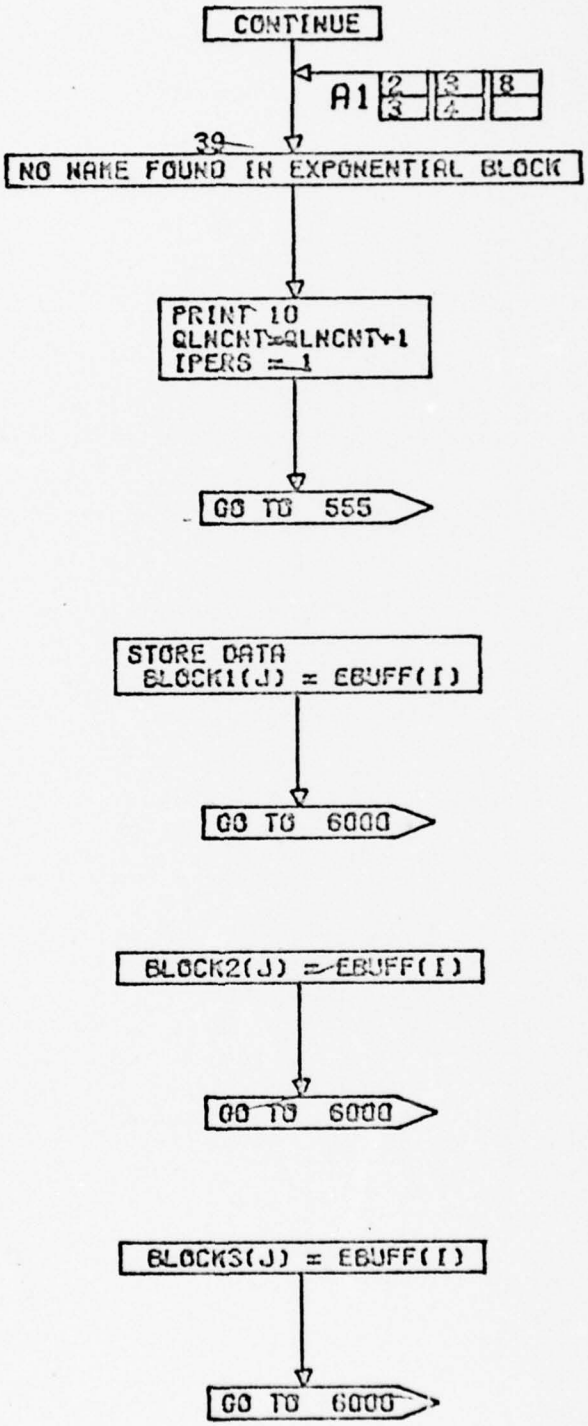

5
A9
DO 4007 J=1,NB3
IF(NBUFF(I).EQ.BOBNAM(J))
F
T
GO TO 7008
CONTINUE
DO 4008 J=1,NB9
IF(NBUFF(I).EQ.BO9NAM(J))
F
T
GO TO 7009
CONTINUE
DO 4009 J=1,NB10
CONT. ON PG 7


IF(NBUFF(I).EQ.B10NAM(J))
F
T
GO TO 7010
CONTINUE
DO 4010 J=1,NB11
IF(NBUFF(I).EQ.B11NAM(J))
F
T
GO TO 7011
CONTINUE
DO 4011 J=1,NB12
IF(NBUFF(I).EQ.B12NAM(J))
F
8
B2
T
8
B1

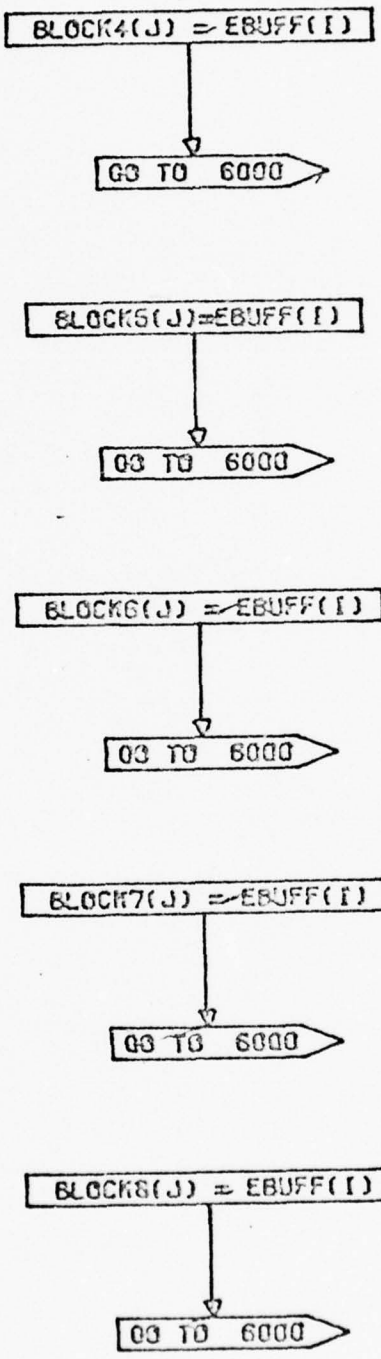
CONT. ON PG 8


7
B1
GO TO 7012
CONTINUE
7
B2
DO 4012 J=1,NB13
IF(NBUFF(I).EQ.B13NAN(J))
F
T
GO TO 7013
CONTINUE
DO 4013 J=1,NB14
IF(NBUFF(I).EQ.B14NAN(J))
F
J
161
T
GO TO 7014

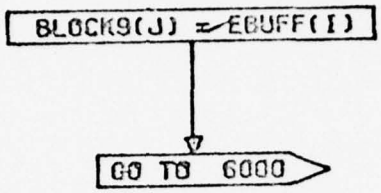
CONT. ON PG 9

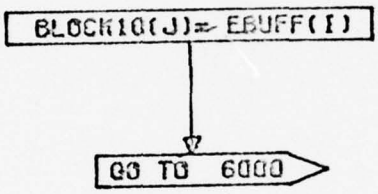

CONTINUE
A1
2
3
8
39
NO NAME FOUND IN EXPONENTIAL BLOCK
PRINT 10
QLNCNT=QLNCNT+1
IPERS = 1
GO TO 555
STORE DATA
BLOCK1(J) = EBUFF(I)
GO TO 6000
BLOCK2(J) = EBUFF(I)
GO TO 6000
BLOCK3(J) = EBUFF(I)
GO TO 6000

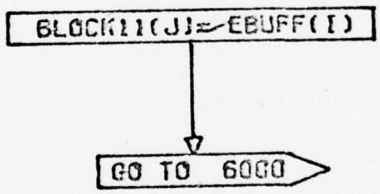
CONT. ON PG 10

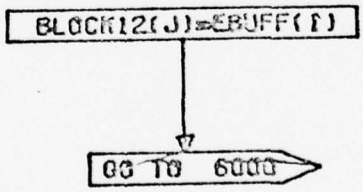

BLOCK4(J) = EBUFF(I)
GO TO 6000
BLOCK5(J)=EBUFF(I)
GO TO 6000
BLOCK6(J) = EBUFF(I)
GO TO 6000
BLOCK7(J) = EBUFF(I)
GO TO 6000
BLOCK8(J) = EBUFF(I)
GO TO 6000

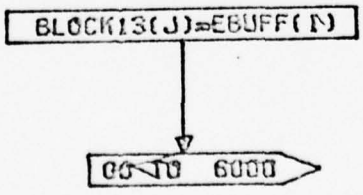
CONT. ON PG 11


BLOCK9(J) = EBUFF(I)
GO TO 6000

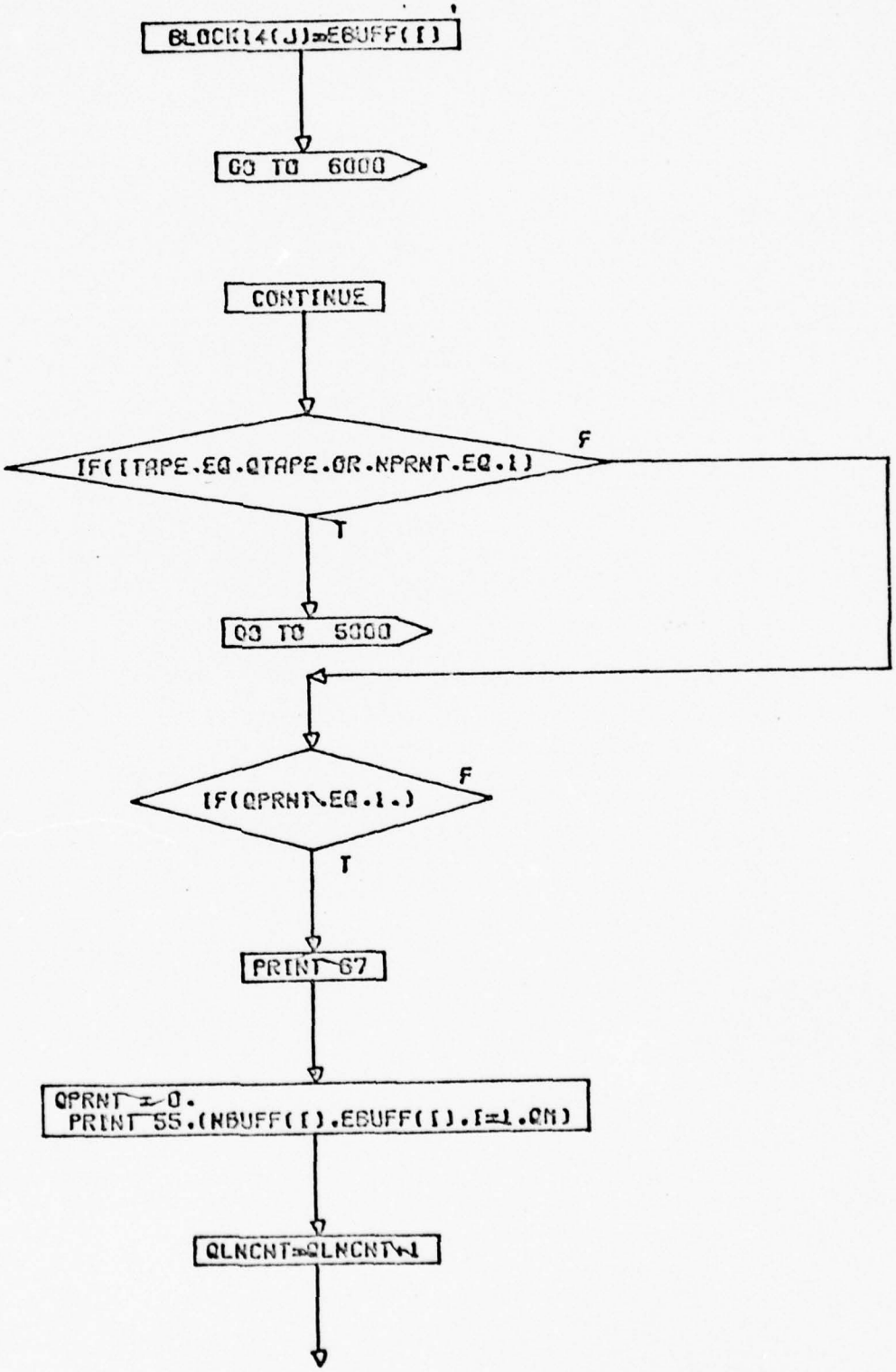

BLOCK10(J)= EBUFF(I)
GO TO 6000


BLOCK11(J)=EBUFF(I)
GO TO 6000

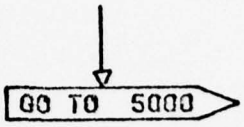

BLOCK12(J)=EBUFF(I)
GO TO 6000


BLOCK13(J)=EBUFF(I)
GO TO 6000

CONT. ON PG 12


BLOCK14(J)=EBUFF(I)
GO TO 6000
CONTINUE
IF(ITAPE.EQ.QTAPE.OR.NPRNT.EQ.1)
F
T
GO TO 5000
IF(QPRNT.EQ.1.)
F
T
PRINT 67
QPRNT=0.
PRINT 55,(NBUFF(I),EBUFF(I),I=1,QN)
QLNCNT=QLNCNT+1

CONT. ON PG 13



END

REFERENCES

1. NWL Technical Report Number TR-3098, February, 1974, titled: 'Air Crew Automated Escape System Simulation Model.'

2. NWL Technical Note TN-K7-7/74, February 1974, titled: 'Program Maintenance Manual for ICARUS a Computer Program to Simulate Escape Systems.'

APPENDIX A

DYNAMIC RESPONSE INDEX (DRI) EQUATIONS

DYNAMIC RESPONSE INDEX (DRI) EQUATIONS

The following spring damper equation is utilized in calculating the dynamic response index (DRI), a measure of the compressibility of the pilot's spine:

1. $\ddot{Y} + c\dot{Y} + kY = A_y$

Where: c = damping coefficient (/sec)

k = stiffness constant (lbs/in)

A_y = vertical acceleration (ins/sec ^{2})

Y = displacement in inches

2. $DRI = (\omega_n^2/g)\, Y$

Where: $\omega_n^2 = k/m$

m = mass of seat/man combination (lbs-sec ^{2}/in)

ω_n^2 = natural freq (rad/sec)

g = acceleration due to gravity (in/sec ^{2})

The above equation (1) is numerically integrated and the DRI is calculated using the displacement obtained at each time step of the ejection trajectory.

APPENDIX B

DYNAMIC CENTER OF GRAVITY EQUATIONS

DYNAMIC CENTER OF GRAVITY EQUATIONS

The displacement of the pilot's center of gravity from its static position has been modelled using 3 independent spring-damper equations, one for each axis of motion.

1. Along the x-axis of motion

$$\ddot{X} + c_x \dot{X} + F(x) = A_x$$

$$A_x = F_x/M$$

Where: F_x = impressed force along the x-axis (lbs)

M = mass of seat/man combination (lbs-sec ^{2}/ft)

A_x = acceleration of seat/man along the x-axis (ft/sec ^{2})

c_x = damping constant (/sec)

$F(x)$ = elastic acceleration (ft/sec ^{2})

$F(x)$ subject to the following constraints:

$x < 0$, $F(x) = k_n(x)$

$0 \leq x \leq x_{slack}$, $F(x) = 0$

$x > x_{slack}$, $F(x) = k_p(x - x_{slack})$

Where: x = displacement along the x-axis (ft)

x_{slack} = slack constant along the x-axis (ft)

k_n, k_p = spring modulus constants (/sec ^{2})

2. Along the y-axis of motion

$$\ddot{Y} + c_y \dot{Y} + s_y Y = A_y$$

$$A_y = F_y/M$$

Where: F_y = impressed force acting on the seat/man combination along the y-axis (lbs)

M = mass of seat/man (lbs/sec^2 /ft)

A_y = acceleration of seat/man along the y-axis (ft/sec ^{2})

c_y = damping constant (/sec)

s_y = spring modulus constant (/sec ^{2})

3. Along the z-axis of motion

$$\ddot{Z} + c_Z \dot{Z} = F(z) = A_Z$$

$$A_Z = F_Z/M$$

Where: F_Z = impressed force acting on the seat/man combination along the z-axis (lbs)

M = mass of seat/man (lbs-sec ^{2}/ft)

A_Z = acceleration of seat/man along the y-axis (ft/sec ^{2})

c_Z = damping constant (/sec)

F (z) subject to the following constraints:

$Z < ZBOT$, $F(Z) = SZN2 * (Z - ZBOT) + SZN1 * (ZBOT)$

$ZBOT < Z < 0$, $F(Z) = SZN1 * Z$

$0 \leq Z \leq SLACK$, $F(Z) = 0$

$ZSLACK < Z$, $F(Z) = SZP * (Z - ZSLACK)$

Where: Z = displacement along the z-axis (ft)

ZBOT = bottoming distance in the (-) Z direction (ft)

ZSLACK = slack distance in (+) Z direction (ft)

SZN1, SZN2, SZP= spring modulus constants (/sec ^{2})

APPENDIX C

MARTIN-BAKER CATAPULT EQUATIONS

MARTIN-BAKER CATAPULT EQUATIONS

The following linear spring-damper and tube bending equations are used to simulate the Martin-Baker catapult system.

1. Rail/Slipper Interaction

There are six slippers attached to the seat, three on each side of the seat, starting at the bottom of the seat (right hand side) - slippers 1, 2, and 3, (on the left hand side) - slippers 4, 5, and 6. During the catapult phase of the trajectory, the right and left slippers move up and off the right and left rails (respectively) of the catapult (tube 3). The following spring-damper equations are used to arrive at the forces and moments acting on the seat/man during this segment of the catapult phase:

If $Y_{slip_i} >$ RAILNT then F_{sx_i}, F_{sz_i}, F_{sy_i}, M_{sx_i}, M_{sz_i}, $M_{sy_i} = 0$

otherwise $F_{csx_i} = k_s \Delta x_i + c_s \dot{\Delta} x_i$

$$F_{csz_i} = k_s \Delta z_i + c_s \dot{\Delta} x_i$$

$$F_{csy_i} = \mu_s (F_{x_i}{}^2 + F_{z_i}{}^2)^{\frac{1}{2}}$$

$$\begin{bmatrix} F_{ssx_i} \\ F_{ssy_i} \\ F_{ssz_i} \end{bmatrix} = DSC \begin{bmatrix} F_{csx_i} \\ F_{csy_i} \\ F_{csz_i} \end{bmatrix}$$

Where DSC is the rotation from the catapult (CCS) the seat/man coordinate system (SMCS)

$$\begin{bmatrix} L_{ss_i} \\ M_{ss_i} \\ N_{ss_i} \end{bmatrix} = \begin{bmatrix} X_{sscg_i} \\ Y_{sscg_i} \\ Z_{sscg_i} \end{bmatrix} X \begin{bmatrix} F_{ssx_i} \\ F_{ssy_i} \\ F_{ssz_i} \end{bmatrix}$$

Torque due to the torsional spring-damper equations acting in the pitch plane.

$$TORPTCH = k_{TOR} \Delta\theta + c_{TOR} \dot{\Delta\theta}$$

Where: k_s = slipper spring modules constant (lbs/ft)

c_s = slipper damping constant (lbs-sec/ft)

μ_s = slipper coefficient of friction

Δx_i = displacement of i^{th} slipper (CCS) from the rail in the x-direction (ft)

Δy_i = displacement of i^{th} slipper (CCS) from the rail in the y-direction (upward) (ft)

Δz_i = displacement of i^{th} slipper (CCS) from the rail in the z-direction (ft)

F_{csx_i}, F_{csy_i}, F_{csz_i} = forces in x, y, and z-direction (lbs) in CCS due to rail/slipper interaction

X_{sscg_i}, Y_{sscg_i}, Z_{sscg_i} = distance from slipper$_i$ to seat/man center of gravity, x, y, and z directions (SMCS) respectively (ft)

L_{ssx_i}, M_{ssx_i}, N_{ssx_i} = moments produced by the rail/slipper interaction around the seat/man center of gravity (SMCS) (ft-lbs)

k_{TOR} = torsional spring modules constant (lbs/ft)

c_{TOR} = torsional damping constant (lbs-sec/ft)

$\Delta\theta$ = angle between rails and seat back (rad)

$\dot{\Delta\theta}$ = angular velocity of $\Delta\theta$ (rad/sec)

The following tube extension equations are used to arrive at the forces and moments acting on the seat/man during the catapult phase of the trajectory.

Tube extension - vector from top of tube 3 to top of telescoping tubes 2 and 1 (tube 2 contained inside tube 3, tube 1 contained inside tube 2 at start of

catapult phase). Tube 2 starts moving up and out of tube 3. When the maximum length of tube 2 is reached then tube 1 starts moving up and out of tube 2.

$$TUBEXT = \left[X_{cctcp}^{\ 2} + (Y_{cctcp} - TUBLTH3)^{\ 2} + Z_{cctcp}^{\ 2}\right] \qquad (CCS)$$

If TUBEXT > TUBLTH1 + TUBLTH2 set all forces and moments acting on seat/man due to the catapult equal to zero.

Otherwise, compute forces and moments acting on seat/man due to the tube bending as follows:

Compute Deflection of tube in X-Z plane (CCS)

$$DEFSL = (X_{cctcp}^{\ 2} + Z_{cctcp}^{\ 2})$$

$0 \leq TUBEXT \leq 2$

$$FTUBE = k_S * DEFSL + c_S * \dot{DEFSL}$$

$$\theta_{AB} = (F_{TUBE} * TUBEXT^{\ 2}) / (2*E*(I_1 + I_2))$$

$2 < TUBEXT \leq TUBLTH1$

$$FTUBE = \left[3*DEFSL*E*(I_1 + I_2)\right] / TUBEXT^3$$

$$\theta_{AB} = \left[FTUBE*TUBEXT^{\ 2}\right] / \left[2*E*(I_1 + I_2)\right]$$

$TUBLTH_1 < TUBEXT \leq TUBLTH_2$

$$F_{TUBE} = (3*DEFLSL*E) / \left[(TUBEXT - TUBLTH_1)^{\ 3} / I_1 + (TUBLTH_1^{\ 3} / (I_1 + I_2))\right]$$

$$\theta_{AB} = \left[F_{TUBE}/(2E)\right] \left[(TUBEXT - TUBLTH_1)^{\ 2}/I_1 + TUBLTH_1^{\ 2} / (I_1 + I_2)\right]$$

$TUBLTH_2 < TUBEXT \leq TUBLTH_1 + TUBLTH_2$

$$F_{TUBE} = (3*DEFSL*ELMOD) / \left[(TUBEXT - TUBLTH_2)^{\ 3}/I_1 + (TUBLTH_1 + TUBLTH_2 - TUBEXT)^{\ 3} / (I_1 + I_2) + (TUBEXT - TUBLTH_1)^{\ 3}/I_2\right]$$

$$\theta_{AB} = (F_{TUBE} * \left[(TUBEXT - TUBLTH_2)^{\ 2}/I_1 + (TUBLTH_1 + TUBLTH_2 - TUBEXT)^{\ 2}/ (I_1 + I_2) + (TUBEXT - TUBLTH_1)^{\ 2}/I_2\right]$$

$$F_{xtube} = - F_{tube} * X_{cctcp} / DEFSL$$
$$F_{ztube} = - F_{tube} * Z_{cctcp} / DEFSL$$
$$F_{ytube} = - UM_{tube} * F_{tube}$$

$$\begin{bmatrix} F_{sxtube} \\ F_{sytube} \\ F_{sztube} \end{bmatrix} = DSC \begin{bmatrix} F_{xtube} \\ F_{ztube} \\ F_{ytube} \end{bmatrix} \qquad (CCS \longrightarrow SMCS)$$

$$\begin{bmatrix} L_{tube} \\ M_{tube} \\ N_{tube} \end{bmatrix} = \begin{bmatrix} X_{stcp} \\ Y_{stcp} \\ Z_{stcp} \end{bmatrix} X \begin{bmatrix} F_{sxtube} \\ F_{sytube} \\ F_{sztube} \end{bmatrix} \qquad (SMCS)$$

Moments due to torsional spring-damper equations at the tube attachment point acting in the pitch, roll and yaw directions:

$$yaw = K_{ytube} * \Delta\psi + C_{ytube} * \Delta\dot{\psi}$$
$$ptch = K_{ptube} * \Delta\theta + C_{ptube} * \Delta\dot{\theta}$$
$$roll = K_{rtube} * \Delta\phi + C_{rtube} * \Delta\dot{\phi}$$

Total forces due to the catapult:

$$F_{xcat} = \sum_{i=1}^{6} F_{ssx_i} + F_{sxtube}$$

$$F_{ycat} = \sum_{i=1}^{6} F_{ssy_i} + F_{sytube}$$

$$F_{zcat} = \sum_{i=1}^{6} F_{ssz_i} + F_{sztube}$$

Total moments due to the catapult:

$$L_{cat} = \sum_{i=1}^{6} L_{ss_i} + L_{tube} + YAW$$

$$M_{cat} = \sum_{i=1}^{6} M_{ss_i} + M_{tube} + PTCH$$

$$N_{cat} = \sum_{i=1}^{6} N_{ss_i} + N_{tube} + ROLL$$

Where: TUBEXT = is the vector (in feet) from the top of catapult tube 3 to the top of the telescoping tubes 2 and 1 (CCS).

TUBELTH1, 2 and 3 = lengths, in feet, of catapult tubes 1, 2 and 3.

X_{cctcp}, Y_{cctcp} and Z_{cctcp} = coordinates (CCS) of the extending tubes 2 and 1 in feet.

DEFSL = the deflection, in feet, of the extending tubes 1 and 2, from the catapult initial position in the x-z plane, CCS.

$\dot{DEFSL}$ = rate of change of DEFSL in ft/sec (CCS).

FTUBE = resultant force in lbs, acting on the seat/man due to the tube bending (CCS).

UM_{tube} = tube bending friction coefficient.

F_{xtube}, F_{ytube} and F_{ztube} = components of the tube bending force (lbs) along the x, y and z directions (CCS).

F_{sxtube}, F_{sytube} and F_{sztube} = components of the tube bending force (lbs) along the x, y and z directions (SMCS).

θ_{AB} = angle between tangent to the tube end and vertical axis (Y) of of the CCS, radians.

E = elacticity modulus of tubes 1, 2 and 3, (lbs/ft ^{2}).

$$I_1 = \frac{\pi}{64} (D_{10}^4 - D_{1i}^4)$$

$$I_1 = \frac{\pi}{64} (D_{20}^4 - D_{2i}^4)$$

D_{10}, D_{20} outside diameters of tubes 1 and 2

D_{1i}, D_{2i} inside diameters of tubes 1 and 2 (ft ^{4}).

X_{stcp}, Y_{stcp} and Z_{stcp} = coordinates, in ft, or seat contact point with telescoping tubes (SMCS).

L_{tube}, M_{tube}, N_{tube} = moments produced by the tube bending, around the seat/man center of gravity (SMCS) ft-lbs.

K_{ytube}, K_{ptube}, K_{rtube} = tube attachment torsional spring modulus constants, ft-lbs/rad, in the yaw, pitch and roll directions respectively.

C_{ytube}, C_{ptube}, C_{rtube} = tube attachment torsional spring-damper constants, ft-lbs-sec/rad, in the yaw, pitch and roll directions respectively.

$\Delta\psi$, $\Delta\theta$, $\Delta\phi$ = angular displacements (rad), of the seat back relative to the catapult tube in the yaw, pitch and roll directions.

$\dot{\Delta\psi}$, $\dot{\Delta\theta}$, $\dot{\Delta\phi}$ = rate of change of the $\Delta\psi$, $\Delta\theta$, $\Delta\phi$ (rad/sec).

F_{xcat}, F_{ycat}, F_{zcat} = total forces (lbs) acting on the seat/man due to the catapult, SMCS.

L_{cat}, M_{cat}, N_{cat} = total moments (ft-lbs) produced by the catapult around the seat/man center of gravity (SMCS).