IDENTIFICATION

D2 * ML F ADMS
Adams Method Integration Package (FORTRAN)
A. C. Hindmarsh, Dept. 59-24
Lockheed Missiles and Space Company
Sunnyvale, California
18 August 1961 (Revised 4-26-62)

PURPOSE

To integrate any number of simultaneous first order differential equations having a common independent variable, viz.,

$$\frac{dy_i}{dt} = f_i(t, y_1, y_2, \ldots, y_n), \quad i = 1, 2, \ldots, n.$$

A differential equation of order $K > 1$ of the form $\frac{d^K y}{dt^K} = f(t, y)$ can be solved by this program after it is reduced to K simultaneous first order equations. This package is intended to use as a part of the user's main program and is not a complete program in itself.

RESTRICTIONS

The following is a list of variables used in the package:

NDEQNS	DELTAT	ADMCF3
NUMACT	DTNMNL	STGATE
NACTVR	DTMAX	MSTIDX
YNST	DTMIN	INSDBL
YNTH	DTNEW	HAVCTE
ACTIND	PASTTM	NSUB
ADMKUL	PASTDT	NDEQEX
ADMKLL	ADMCF1	NOUTEX
TIME	ADMCF2	NHITEX

Those variables which are underlined are to be defined by the user as prescribed in this writeup. The others are defined and used entirely within the package.

The FORTRAN statement numbers used by the package range from 1 to 21. The programmer may use any number larger than 21 in the main program, subject to the restrictions (explained later) concerning the numbers 30, 40, and 50.

METHOD

The formula on which the Adams method of integration is based is

$$y^* = y + \Delta t(\dot{y} + \tfrac{1}{2}\nabla\dot{y} + \tfrac{5}{12}\nabla^2\dot{y} + \tfrac{3}{8}\nabla^3\dot{y})$$

where $\dot{y} = \frac{dy}{dt}$ and ∇ denotes the backward difference operator. From the initial

Supervisor T. Donati Systems ________

point (t_o, y_o), three starting points (required in the above formula) are obtained by using a very small nominal Δt, setting the backward difference coefficients equal to zero, and introducing each coefficient successively as a new point is computed. After this starting procedure is completed, the integration continues, using the complete quadrature formula above. The size of Δt is then controlled by a halving and doubling procedure which is dependent upon the magnitude of the estimated relative truncation error, but is in all cases kept between prescribed bounds. The tests on this error leading to halving or doubling are not performed during the start procedure, and doubling is never performed twice in succession.

Although Δt is controlled largely internally, there is a provision in the program for giving solutions of the system at any prescribed value of TIME, i.e., at any critical point. As soon as a critical time has been overtaken, the increment Δt from the previous point must be redetermined so as to hit the critical point. The package then modifies the backward differences in accordance with this Δt_{new} and integrates to the critical time. At this point any desired changes can be made in the equations. Integration is then resumed from the critical point using the starting procedure described above, as if the critical point were an initial point. Every point of finite discontinuity in the derivatives should be treated as a critical point.

Output can be effected at every computed point, or on any conditions desired (and inserted) by the user (e.g., at every tenth point plus critical points). The user must write his own output, however, as described for "Block D" below.

The computational procedures of the package itself are not performed in the main program, but in SUBROUTINE BIGSUB. Control passes from the main program to BIGSUB and back many times during each integration. Information flow between the two is accomplished by means of COMMON storage.

The following table gives the name, definition, and equations, where applicable, of each variable used in the program.

Program Symbol	Writeup Symbol	Definition	Equation
NDEQNS	n	No. of differential equations	
NUMACT	m	No. of variables being tested for truncation error	
NACTVR	j	Subscript of variable being tested (a one-dimensional array)	$1 \leq j \leq n$
YNTH(I,1)	y_i	Previous value of variable i	
YNTH(I,2)	$\dot{y}_i$	Derivative of variable i	$\dot{y}_i = \frac{dy_i}{dt} = f_i(t, y_1, y_2, \ldots, y_n)$
YNTH(I,3)	$\nabla \dot{y}_i$	First order backward difference of $\dot{y}_i$	$\nabla \dot{y}_i(t) = \dot{y}_i(t) - \dot{y}_i(t - \Delta t)$
YNTH(I,4)	$\nabla^2 \dot{y}_i$	Second order backward difference of $\dot{y}_i$	$\nabla^2 \dot{y}_i(t) = \nabla \dot{y}_i(t) - \nabla \dot{y}_i(t - \Delta t)$
YNTH(I,5)	$\nabla^3 \dot{y}_i$	Third order backward difference of $\dot{y}_i$	$\nabla^3 y_i(t) = \nabla^2 \dot{y}_i(t) - \nabla^2 \dot{y}_i(t - \Delta t)$
YNST(I,K)	$y^*, \dot{y}^*, \nabla \dot{y}^*, \nabla^2 \dot{y}^*, \nabla^3 \dot{y}^*$	Two dimensional array of new values of variables, derivatives and backward difference, analogous to YNTH array	$y_i^* = y_i + \Delta t(\dot{y}_i + \frac{1}{2} \nabla \dot{y}_i + \frac{5}{12} \nabla^2 \dot{y}_i + \frac{3}{8} \nabla^3 \dot{y}_i)$ $= y_i(t + \Delta t)$ $\dot{y}_i^* = f_i(t + \Delta t, y_1^*, \ldots, y_n^*)$ $\nabla \dot{y}_i^* = \dot{y}_i^* - \dot{y}_i$ $\nabla^2 \dot{y}_i^* = \nabla \dot{y}_i^* - \nabla \dot{y}_i$ $\nabla^3 \dot{y}_i^* = \nabla^2 \dot{y}_i^* - \nabla^2 \dot{y}_i$

Program Symbol	Writeup Symbol	Definition	Equation
ACTIND(J)	Q_j	Approximate relative truncation error in variable j	$Q_j = \begin{cases} \left\lvert \Delta t \, \nabla^3 \dot{y}_j^* \right\rvert & \text{if } y_j^* \geq .000017 \\ \left\lvert \dfrac{\Delta t \, \nabla^3 \dot{y}_j^*}{y_j^*} \right\rvert & \text{if } y_j^* < .000017 \end{cases}$
ADMKUL	K_u	Upper bound on the Q_j	
ADMKLL	K	Lower bound on the Q_j	$K = \frac{1}{20} K_u$
TIME	t	Time	
DELTAT	Δt	Current increment in time	
DTNMNL	$\Delta t_{nom.}$	Small value of Δt used in starting procedure	$\Delta t_{nom.}$ = .0001 normally
DTMAX	Δt_{max}	Maximum allowable Δt	
DTMIN	Δt_{min}	Minimum allowable Δt	
DTNEW	Δt_{new}	Modified Δt used in hitting critical times	

The following variables and indices have internal significance only.

Program Symbol	Writeup Symbol	Definition	Equation
PASTTM		Previous value of t	
PASTDT		Previous value of Δt	
R	r	Ratio of new to old Δt	$r = \dfrac{\Delta t_{new}}{\Delta t}$
ADMCF1	C_1	Coefficient of $\nabla \dot{y}_1$ in quadrature formula	$C_1 = \frac{1}{2}$ except in start procedure

Program Symbol	Writeup Symbol	Definition	Equation
ADMCF2	C_2	Coefficient of $\nabla^2 \mathring{y}_1$ quadrature formula	$C_2 = \frac{5}{12}$ except in start procedure
ADMCF3	C_3	Coefficient of $\nabla^3 \mathring{y}_1$ quadrature formula	$C_3 = \frac{3}{8}$ except in start procedure
STGATE		Index indicating whether or not start procedure is in progress	= 1 during start procedure = 0 otherwise
MSTIDX		Index in start procedure indicating number of starting points that exist	
INSDBL		Index indicating whether or not Δt has been doubled since the previous integration	= 0 if doubling has occurred = 2 if doubling has not occurred
HAVGATE		Index indicating whether or not Δt has been halved since the previous integration	= 1 if halving has occurred = 0 if halving has not occurred
NSUB		Index indicating section of BIGSUB to be entered	= 1 in start procedure = 2 when taking backward differences = 3 when setting starting coefficients = 4 when computing truncation errors = 5 in halving procedure = 6 in transfer of present to past values = 7 in doubling procedure = 8 in quadrature routine = 9 in modification for critical times

Program Symbol	Writeup Symbol	Definition	Equation
NDEQEX		Variable of assigned GØTØ exit from block B	
NOUTEX		Variable of assigned GØTØ exit from block D	See USAGE for full explanation.
NHITEX		Variable of assigned GØTØ entry into block C	

USAGE

Adams Integration Package is intended as the structural basis of the user's main integration program; i.e., certain blocks of Fortran statements must be coded and added to the package in locations designated in this writeup before the package can be utilized. These blocks include input, output, evaluation of the derivatives (the f_i mentioned under PURPOSE), tests for stop conditions, critical point routines, and any other desired routines.

There are four blocks in the main program, labeled A to D, that must be filled in by the user. Entries and exits to and from these blocks are taken care of in the package; only the actual routines prescribed below are required.

Block A. This is the first section of the program, preceding the package itself. It includes the following four parts, in addition to any preliminary routines desired by the user:

1) The DIMENSION statement must include the entries YNST(NDEQNS,5), YNTH(NDEQNS,5), NACTVR(NUMACT), ACTIND(NUMACT), where the symbols NDEQNS and NUMACT must be replaced by their numerical values before compiling. The COMMON statement is sufficient as it stands unless subroutines added by the user require augmentation of it.

2) Frequently, it is desirable to use mneumonic names for the variables and derivatives. In that case, one should write an EQUIVALENCE statement as illustrated in this example: Suppose three derivatives XDØT, YDØT, and ZDØT are to be integrated to obtain X, Y, and Z. The following statement can then be made:

```
EQUIVALENCE (X, YNST (1)), (Y, YNST (2)), (Z, YNST (3)),
            (XDØT, YNST (4)), (YDØT, YNST (5)), (ZDØT, YNST (6))
```

Note that the subscript for the derivative of any variable with subscript I is (I +NDEQNS). If an EQUIVALENCE statement is made, all the elements of the YNST(I, 1) array and YNST(I, 2) array should be given symbolic names to preserve the storage sequence established by the joint use of the COMMON and EQUIVALENCE statements.

3) Define numerically, by reading input or by inserting arithmetic statements, the following variables:

NDEQNS, NUMACT, (NACTVR(I), I=1, NUMACT), (YNST(I, 1), I=1, NDEQNS), ADMKUL, TIME, DTNMNL, DTMAX, DTMIN,

in accordance with the definitions in the table under METHOD. The array YNST(I,1) now contains the initial values of the variables corresponding to the initial value of time (normally zero) contained in TIME. The array NACTVR(I) contains the numbers (from among 1, 2,...,NDEQNS) of the variables to be tested in the halving-doubling tests. For example, if y_1 and y_3 only are to be tested, set

NUMACT=2, NACTVR(1)=1, NACTVR(2)=3.

ADMKUL is the desired relative local error in the integration. For example, if a 1/2% maximum error is sought, ADMKUL = .005. All other variables appearing in the equations of the system are initialized here also.

4) Write out any title and the initial values of variables as desired. Variables to be output during integration need not be output here, as their initial values are output on the first entry into Block D.

Block A is now complete and control must pass to statement 1, the beginning of the integration.

Block B. This block contains the differential equations of the system and must begin with statement number 30. The equations must give the present derivatives YNST(I,2) in terms of TIME, YNST(1,1), YNST(2,1),...,YNST(NDEQNS,1), and any parameters not used in the package itself but defined in Block A. The exit from this block must be GO TO NDEQEX,(4, 17).

Block C. This block deals with critical points and is divided into two main parts as follows:

1) The first part must begin with statement number 40 and detects whether a critical point (or terminal condition) has been overtaken. If it has not, control must be returned to statement number 10. If a critical point not yet encountered has been overtaken, DTNEW must be computed (generally by inverse interpolation) so as to hit the critical time exactly and control must reach a series of two statements of the following form

```
ASSIGN M TO NHITEX
GO TO 14
```

Any number of branches may be used for the computation of DTNEW, but each must end with the above two statements. If an index is being used in Block D to control output, this would be the place to set the index to that value which forces output at any or all critical points. (Alternatively, if output is desired at all critical points, the index should be set between statements 17 and 18 in the package.) Control now returns to statement 14 in the package and causes integration to the critical point and output. Finally, control returns to statement M as explained below.

2) The second part of Block E contains any changes to be made in the equations of the system or in any of those variables defined in the second part of Block A after the critical point has been attained. A change of the first type involves a switch setting that causes the desired branch to be taken in Block B, while a change of the second type involves only arithmetical statements. For example, a resetting of the variables may be necessary after reaching a discontinuity. Entry into this part of the block is effected by using statement numbers M corresponding to the numbers in the ASSIGN statements in the various branches in part (1) of Block C. This requires the user to fill out the list in statement 21 of the package with the values of M used. This statement reads

```
21    GØ TØ NHITEX, (...M...)
```

Block C ends with the statement GØ TØ 1, to which control must always pass except on run termination. This return causes a restart of the integration from the critical point just attained.

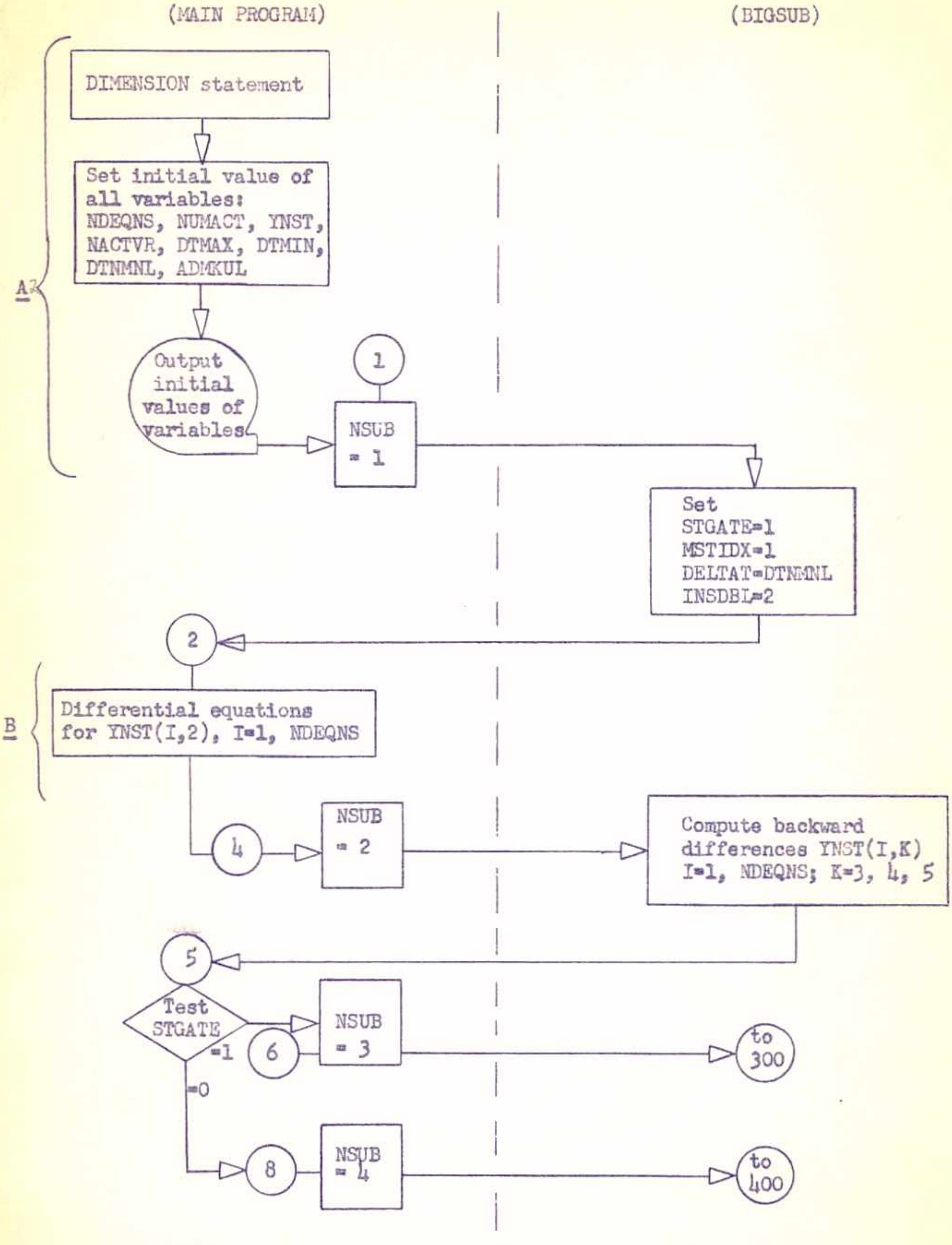
If stop conditions have been met, control must pass to a section of the main program to execute any case-termination procedures written by the programmer. Another case can be computed by in effect repeating Block A procedures and passing to statement 1. When the complete job is finished, exit by a CALL DUMP or CALL EXIT statement.

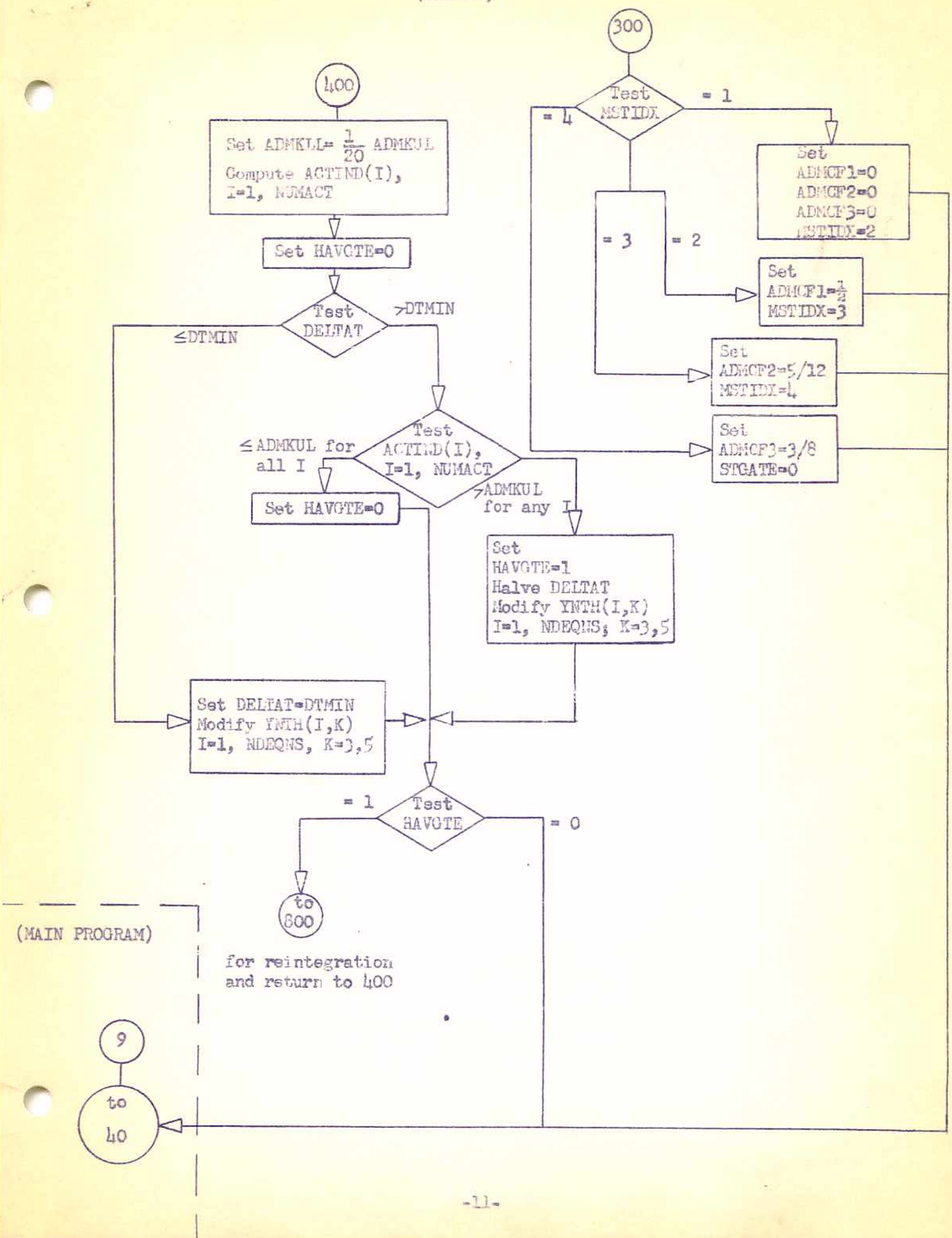
Block D. This block contains any output routines desired. It must begin with statement number 50 and end with GØ TØ NØUTEX, (12, 20), to which control must always pass.

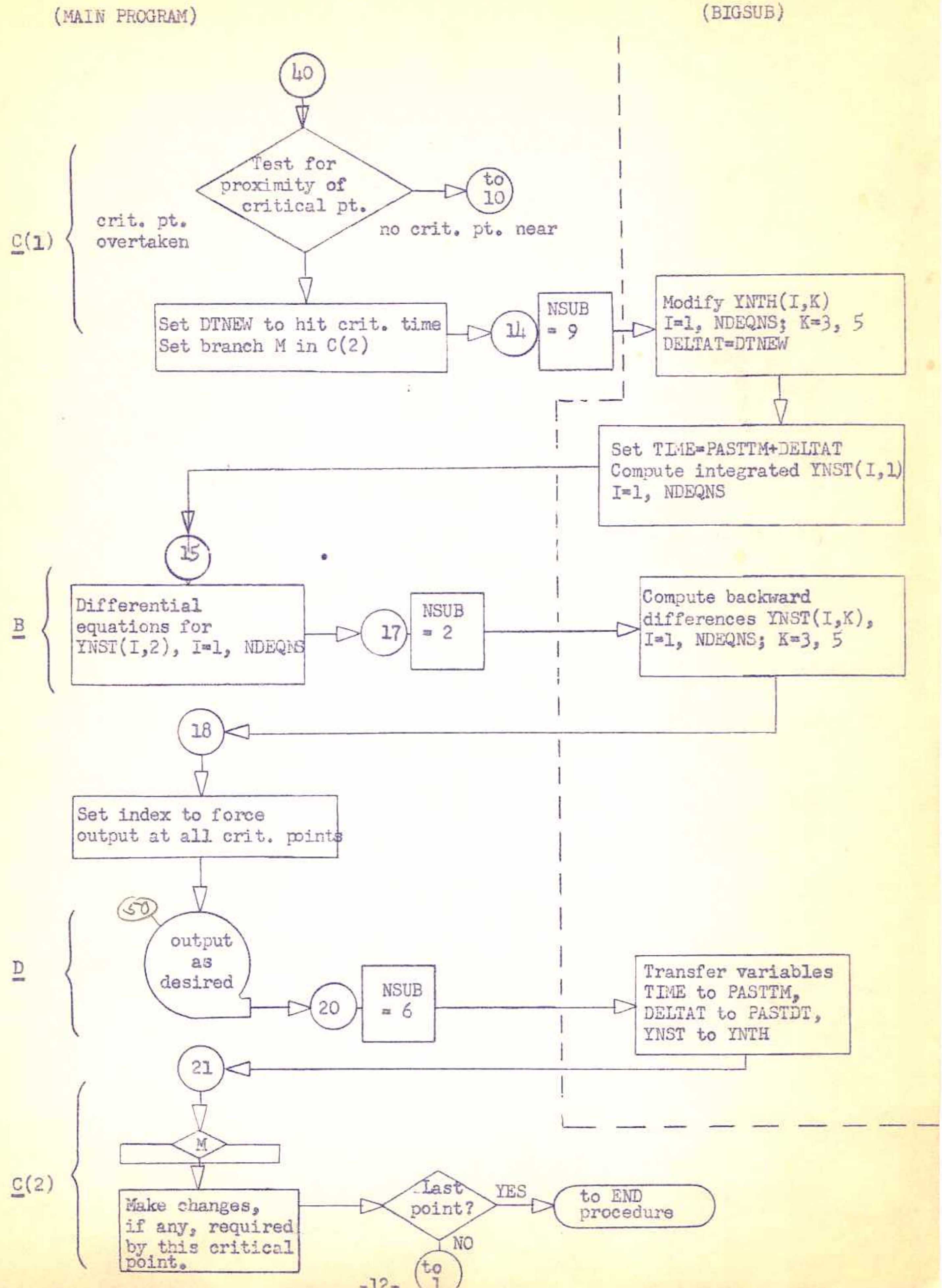
With the completion of Block D the user is free to continue the program as desired. The only restriction is that no other statements may have numbers less than 22 or equal to 30, 40, 50, or the values "M" used in Block C.

When the coding of the main program is complete, with the package, blocks A to D, other computations and monitor control cards all included, the deck is prepared for compilation by placing BIGSUB behind the main program along with any other subroutines used. BIGSUB is ready for compilation as received by the user except that the symbols in the DIMENSION statement must first be replaced with numerical values.

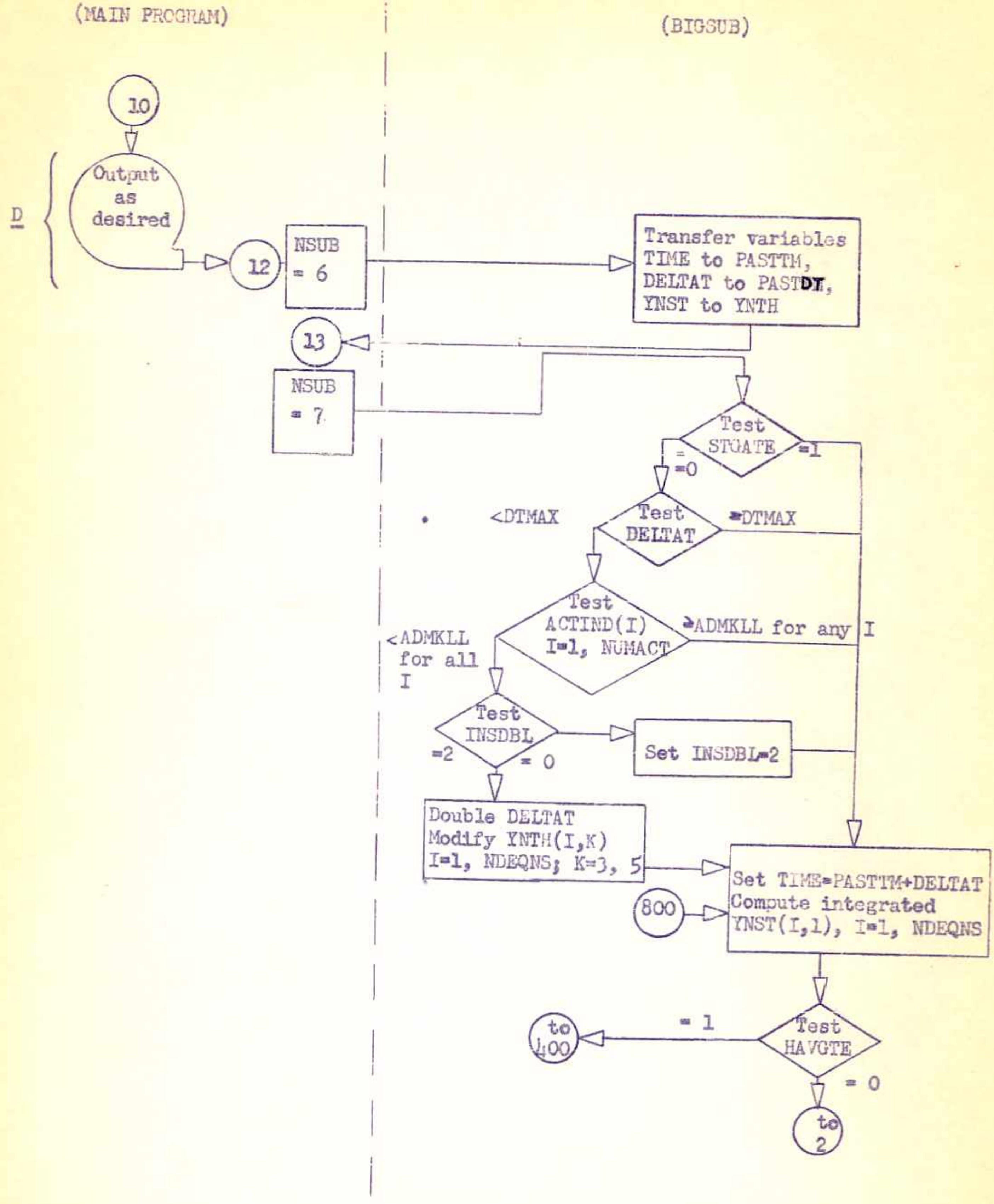
ADAMS INTEGRATION PACKAGE
FLOW CHART




400
Set ADMKLL= 1/20 ADMKUL
Compute ACTIND(I),
I=1, NUMACT
Set HAVGTE=0
Test DELTAT
>DTMIN
≤DTMIN
Test ACTIND(I), I=1, NUMACT
≤ADMKUL for all I
>ADMKUL for any I
Set HAVGTE=0
Set HAVGTE=1
Halve DELTAT
Modify YNTH(I,K)
I=1, NDEQNS; K=3,5
Set DELTAT=DTMIN
Modify YNTH(I,K)
I=1, NDEQNS, K=3,5
Test HAVGTE
= 1
= 0
to 800
for reintegration
and return to 400
300
Test MSTIDX
= 1
= 2
= 3
= 4
Set
ADMCF1=0
ADMCF2=0
ADMCF3=0
MSTIDX=2
Set
ADMCF1=1/2
MSTIDX=3
Set
ADMCF2=5/12
MSTIDX=4
Set
ADMCF3=3/8
STGATE=0
(MAIN PROGRAM)
9
to
40



(MAIN PROGRAM)
(BIGSUB)
40
Test for proximity of critical pt.
to 10
no crit. pt. near
C(1)
crit. pt. overtaken
Set DTNEW to hit crit. time
Set branch M in C(2)
14
NSUB = 9
Modify YNTH(I,K)
I=1, NDEQNS; K=3, 5
DELTAT=DTNEW
Set TIME=PASTTM+DELTAT
Compute integrated YNST(I,1)
I=1, NDEQNS
15
B
Differential equations for YNST(I,2), I=1, NDEQNS
17
NSUB = 2
Compute backward differences YNST(I,K), I=1, NDEQNS; K=3, 5
18
Set index to force output at all crit. points
D
50
output as desired
20
NSUB = 6
Transfer variables TIME to PASTTM, DELTAT to PASTDT, YNST to YNTH
21
C(2)
M
Make changes, if any, required by this critical point.
Last point?
YES
to END procedure
NO
to 1


(MAIN PROGRAM)
(BIGSUB)
10
Output as desired
D
12
NSUB = 6
Transfer variables TIME to PASTTM, DELTAT to PASTDT, YNST to YNTH
13
NSUB = 7
Test STGATE
=0
=1
Test DELTAT
<DTMAX
=DTMAX
Test ACTIND(I) I=1, NUMACT
<ADMKLL for all I
≥ADMKLL for any I
Test INSDBL
=2
= 0
Set INSDBL=2
Double DELTAT Modify YNTH(I,K) I=1, NDEQNS; K=3, 5
Set TIME=PASTTM+DELTAT Compute integrated YNST(I,1), I=1, NDEQNS
800
Test HAVGTE
= 1
to 400
= 0
to 2