N72-12116

FORMAC INTEGRATION PROGRAM

A SPECIAL APPLICATIONS PACKAGE USED IN DEVELOPING TECHNIQUES OF ORBITAL DECAY AND LONG-TERM EPHEMERIS PREDICTION FOR SATELLITES IN EARTH ORBIT

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

George C. Marshall Space Flight Center

Aero-Astrodynamics Laboratory

CONTRACT NAS 8-26113

NOVEMBER 1971

CSC

COMPUTER SCIENCES CORPORATION

FORMAC INTEGRATION PROGRAM

A SPECIAL APPLICATIONS PACKAGE USED IN DEVELOPING TECHNIQUES OF ORBITAL DECAY AND LONG-TERM EPHEMERIS PREDICTION FOR SATELLITES IN EARTH ORBIT

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

George C. Marshall Space Flight Center

Aero-Astrodynamics Laboratory

CONTRACT NAS 8-26113

NOVEMBER 1971

COMPUTER SCIENCES CORPORATION

8300 South Whitesburg Drive

Huntsville, Alabama 35802

Major Offices and Facilities Throughout the World

FORMAC INTEGRATION PROGRAM - A SPECIAL APPLICATIONS PACKAGE USED IN DEVELOPING TECHNIQUES OF ORBITAL DECAY AND LONG-TERM EPHEMERIS PREDICTION FOR SATELLITES IN EARTH ORBIT

November 1971

by

C. K. Rowe

Prepared For

National Aeronautics and Space Administration

George C. Marshall Space Flight Center

Aero-Astrodynamics Laboratory

Contract NAS 8-26113

REVIEWED BY: B F Barry

B. F. Barry
Project Manager

APPROVED BY: W B Twichell Jr

W. B. Twichell, Jr.
Director
Aerospace Systems Center

FOREWORD

This report presents the results of work performed by Computer Sciences Corporation while under contract to the Aero-Astrodynamics Laboratory of the George C. Marshall Space Flight Center, Contract NAS 8-26113. The program described herein evolved as a supplemental application package when developing techniques of orbital decay and long-term satellite ephemeris prediction (see Ref. 1).

The author is grateful to Mr. F. C. Boles (formerly with CSC) for his important contributions in the initial phase of the program development.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

SUMMARY

The symbolic manipulation capabilities of the FORMAC (FORmula MAnipulation Compiler) language are employed to expand and analytically evaluate integrals of the form

$$I = k \int \frac{\sin^p x \cos^q x}{(1 + e \cos x)^n} dx \qquad o < e < 1$$

where p, q are non-negative integers and n is an arbitrary integer. Basically, the program integration is effected by expanding the integral(s) into a series of subintegrals and then substituting a pre-derived and pre-coded solution for that particular subintegral. Derivation of the integral solutions necessary for precoding is included, as is a discussion of the FORMAC system limitations encountered in the programming effort.

SECTION 1 - THEORY

1.1 TECHNICAL DESCRIPTION

Differential equations involving integrands of the type

$$\frac{k \sin^p x \cos^q x}{(1 + e \cos x)^n} \qquad o < e < 1$$

(where k does not depend on x, p and q are non-negative integers, and n is an integer) arise in several areas of general perturbation theory. By properly utilizing the algebraic and analytic capabilities of a symbolic manipulation language such as FORMAC, these integrals can be analytically evaluated. The program described herein is designed for just that purpose.

The program consists of a FORMAC driver and a set of subroutines which effect the required integrations. The driver performs all required manipulations of each input integrand, determines the integration parameters p, q, n and the "constant" k, and transmits these quantities to the driver routine of the integration package (the set of routines which perform the required integrations). The integration package driver then identifies the integrand involved, makes any necessary variable transformations, and calls upon the proper subroutine to carry out the integration.

The complete solution of an integrand usually requires solving several subintegrals (special cases), and each integration package subroutine is designed to integrate a given type of subintegral. These integrated results are then transmitted back to the integration package driver where inverse transforms are performed (if necessary), and the results passed on to the FORMAC driver for simplification and output.

A detailed analysis of the required integrations is provided in the appendices, and the individual integration subroutines are structured according to this development.

1.2 EQUATIONS (OMITTING THE INTEGRATION CONSTANT)

If $n < 0$

$$\int \sin^p x \cos^q x \,(1 + e\cos x)^n\, dx = \sum_{i=0}^{m} \frac{m!\, e^i}{(m-i)!\, i!} \int \sin^p x \cos^{q+i} x\, dx \qquad (1\text{-}1)$$

where $m = -n$ and

$$\int \sin^a x \cos^b x\, dx = \begin{cases} \dfrac{\sin^{a+1} x \cos^{b-1} x}{a+b} + \dfrac{b-1}{a+b} \displaystyle\int \sin^a x \cos^{b-2} x\, dx & b \neq 0 \\[2ex] -\dfrac{\sin^{a-1} x \cos^{b+1} x}{a+b} + \dfrac{a-1}{a+b} \displaystyle\int \sin^{a-2} x \cos^b x\, dx & a \neq 0 \\[2ex] x & a + b = 0 \end{cases} \qquad (1\text{-}1a)$$

by backward recursion.

If $n \geq 0$ and p is odd, set $p = 2K + 1$ and $x = \cos x$ to obtain

$$\int \frac{\sin^p x \cos^q x}{(1 + e\cos x)^n}\, dx = -\sum_{i=0}^{K} (-1)^i \frac{K!}{(K-i)!\, i!} \left\{ (-1)^a \frac{(\ell - 1)!}{(\ell + a - 1)!} \frac{d^a I_{1\ell}}{de^a} \right\} \qquad (1\text{-}2)$$

where $a = 2(K - i) + q < n$

$$\ell = n - a \qquad (1\text{-}2a)$$

and

$$I_{1\ell} = \begin{cases} \left[1/e\,(\ell - 1)\right] (1 + ez)^{-\ell + 1} & \ell \neq 1 \\ 1/e \log (1 + ez) & \ell = 1 \end{cases} \qquad (1\text{-}2b)$$

or,

$$\int \frac{\sin^p x \cos^q x}{(1 + e\cos x)^n}\,dx = -\sum_{i=0}^{K}(-1)^i \frac{K!}{(K-i)!\,i!}\left\{\sum_{j=1}^{b}\alpha_j \int z^{b-j}\,dz + \sum_{j=0}^{n-1}\beta_{n-j}(-1)^j \frac{(\ell-1)!}{(\ell+K-1)!}\frac{d^j I_{1\ell}}{de^j}\right\} \qquad (1\text{-}3)$$

where $b = a - n + 1 \qquad a \geq n$

$\ell = n - j$

$$\alpha_1 = \frac{1}{e},\ \alpha_j = -\frac{1}{e}\left(\frac{n+j-2}{j-1}\right)\alpha_{j-1} \qquad 2 \leq j \leq b \qquad (1\text{-}3a)$$

$$\beta_{n-j} = -\sum_{\ell=0}^{j}\frac{n!\,e^{j-\ell}\,\alpha_{b-\ell}}{(n-j+\ell)!\,(K-\ell)!} \qquad 0 \leq j \leq n-1,\ \alpha_o = 0$$

If $n \geq 0$ and p is even, set $p = 2K$ and use $\sin^2 x = 1 - \cos^2 x$ to obtain

$$\int \frac{\sin^p x \cos^q x}{(1 + e\cos x)^n}\,dx = \sum_{i=0}^{K}(-1)^i \frac{K!}{(K-i)!\,i!}\left\{(-1)^a \frac{(\ell-1)!}{(\ell+a-1)!}\frac{d^a I_{2\ell}}{de^a}\right\} \qquad (1\text{-}4)$$

where $a = 2(K - i) + q < n$

$\ell = n - a$ $\qquad (1\text{-}4a)$

and

$$I_{2\ell} = \int \frac{dx}{(1 + e\cos x)^{\ell}} = \frac{R(x) + BI_{2\ell-1} - CI_{2\ell-2}}{A} \qquad (1\text{-}4b)$$

with $R(x) = \dfrac{\sin x}{(1 + e \cos x)^{\ell-1}}$

$$A = -(\ell - 1)\left(\frac{1 - e^2}{e^2}\right)$$

$$B = \frac{2n - 3}{e}$$

$$C = -\frac{(n - 2)}{e} \tag{1-4c}$$

$$I_{21} = \frac{2}{\left(1 - e^2\right)^{1/2}} \tan^{-1}\left\{\left(\frac{1 - e}{1 + e}\right)^{1/2} \tan \frac{x}{2}\right\}$$

$$I_{22} = \frac{1}{(1 - e^2)} I_{21} - \frac{e}{(1 - e^2)} \frac{\sin x}{(1 + e \cos x)}$$

or,

$$\int \frac{\sin^p x \cos^q x}{(1 + e \cos x)^n} dx = -\sum_{i=0}^{K} (-1)^i \frac{K!}{(K - i)!\, i!} \left\{ \sum_{j=1}^{b} \alpha_j \int \cos^{b-j} x \, dx + \sum_{j=0}^{n-1} \beta_{n-j} (-1)^i \frac{(\ell - 1)!}{(\ell + K - 1)!} \frac{d^j I_{2\ell}}{de^j} \right\} \tag{1-5}$$

where b , ℓ , α_0 , α_1 , α_j , β_{n-j} are defined as before.

SECTION 2 - PROGRAMMING

2.1 GENERAL

The deck consists of: (1) a main program that determines the integration parameters ($\underline{p}$, $\underline{q}$, $\underline{n}$) for the various integrals appearing in the differential equation(s); (2) an integration module driver that executes the proper logic to call the required integration routines and accumulates their results; and (3) several integration routines that actually perform the required integrations.

2.2 PROGRAM CHARACTERISTICS

The program is coded in the FORMAC language and runs on the IBM 7094 under IBSYS (Version 13) control. Program input is internal (no card or tape reads) and consists of the differential equation(s), along with parameters which indicate the initial equation set and how many equations to evaluate per set (provision is made for up to six sets of three equations each). Program output (print only) consists of the analytical integration results and certain parameters which provide a trace of the integration flow.

Program run-time varies from 3 to 20 minutes, depending on the number of equations and their individual complexity. Normal output is 0 to 5000 lines and is in FORMAC notation (as is the input). One non-system diagnostic appears in subroutine MNDVIG. This diagnostic indicates an improperly dimensioned array, terminates the current integration, and returns control to the main program for continued execution. An A5 work tape is required.

2.3 BLOCK DIAGRAM/FLOWCHART

Block diagrams and flowcharts for the FORMAC main program and its subroutines are presented in Figures 2-1 through 2-6. The blocks depicted in the subroutine figures are defined as follows (see Subsection 2.4 for a complete description of the subroutines):

IM1	-	Denotes integration by SPCQ routine.
IM2	-	Denotes integration by DUIS routine.
IM3	-	Denotes integration by MNDVIG routine.
IM4	-	Denotes integration by XN routine.
T1	-	Denotes transformation given by $(\sin x)^P = (1 - \cos^2 x)^K \qquad K = \frac{P}{2}$
T2	-	Denotes transformation given by $x = \cos x$
T3	-	Denotes transformation given by $x = 1 + e \cos x$
R12	-	Denotes recursion relation 1 or 2 (depending on ITYPE) of subroutine DUIS.
IR1	-	Denotes integration by recursion relation 1 of subroutine SPCQ.
IR2	-	Denotes integration by recursion relation 2 of subroutine SPCQ.
IR3	-	Denotes integration by recursion relation 3 of subroutine SPCQ.

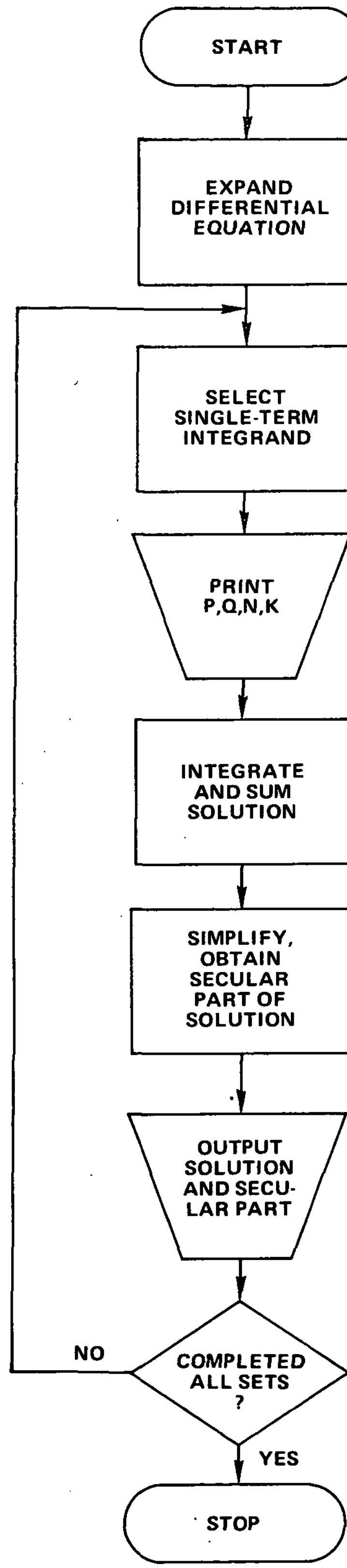


Figure 2-1. Main Program

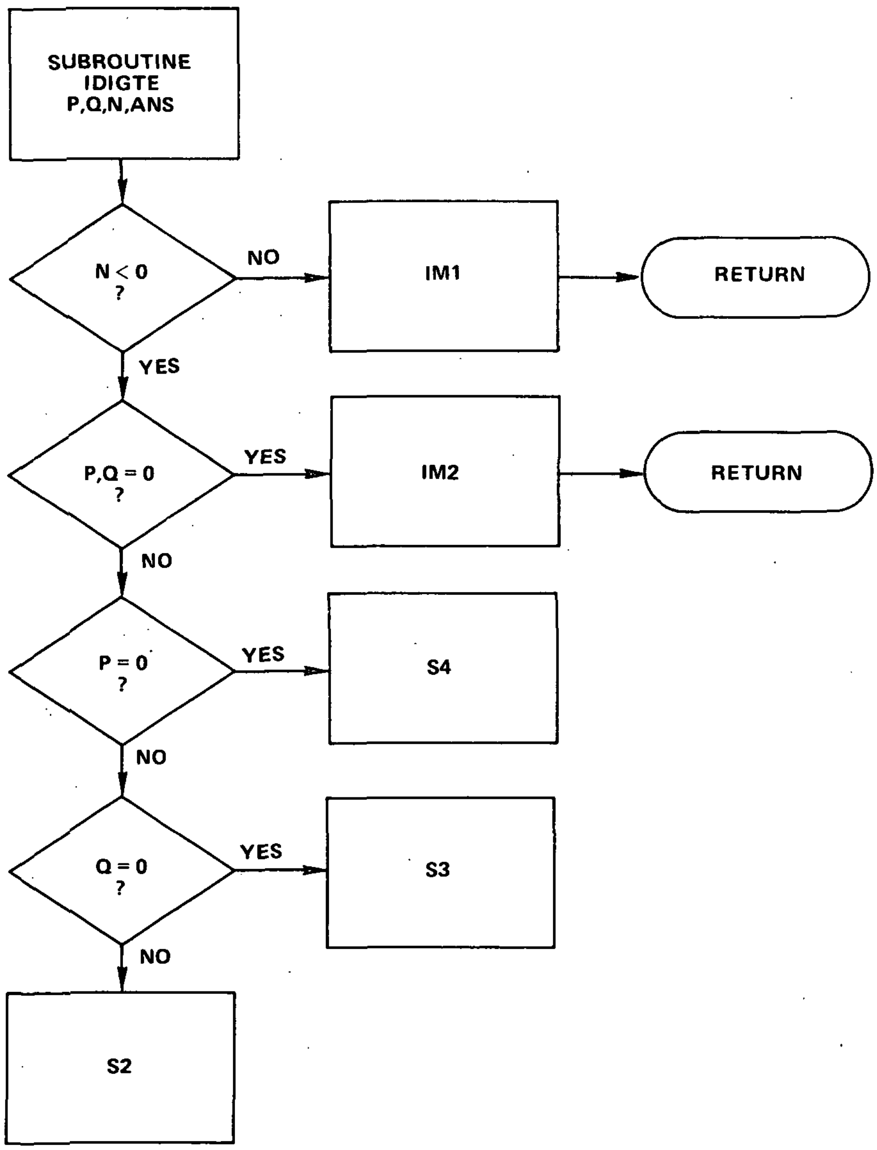
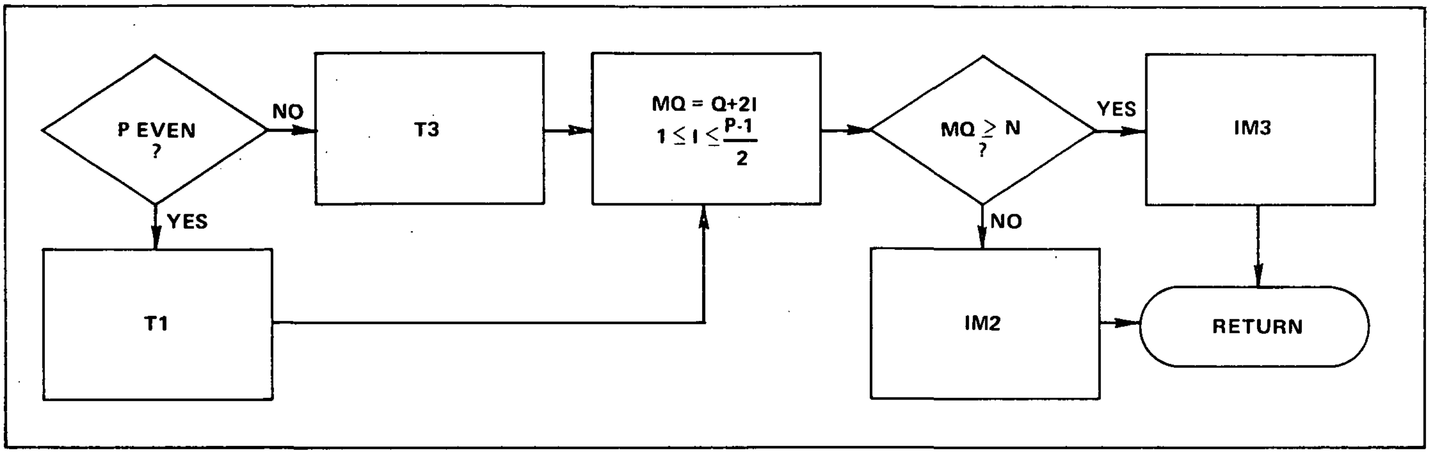
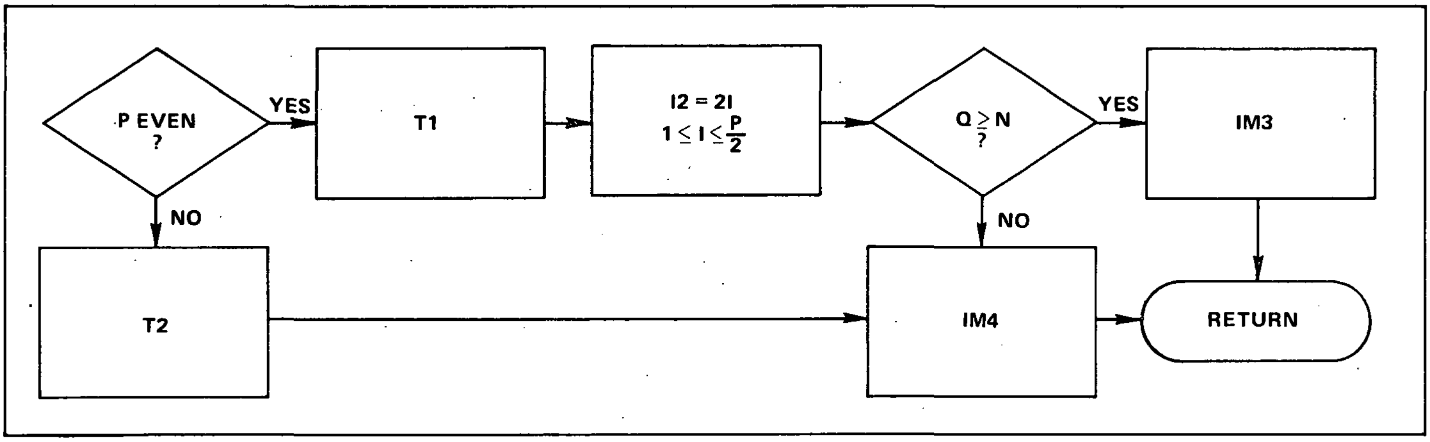


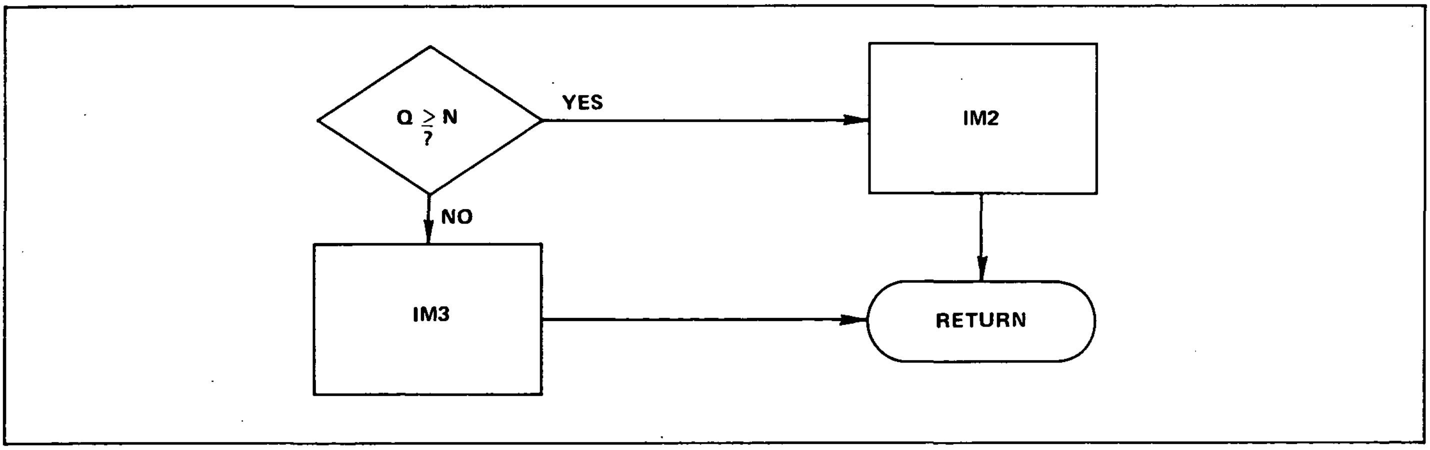
Figure 2-2. Subroutine IDIGTE (Sheet 1 of 2)



S2



S3



S4

Figure 2-2. Subroutine IDIGTE (Sheet 2 of 2)

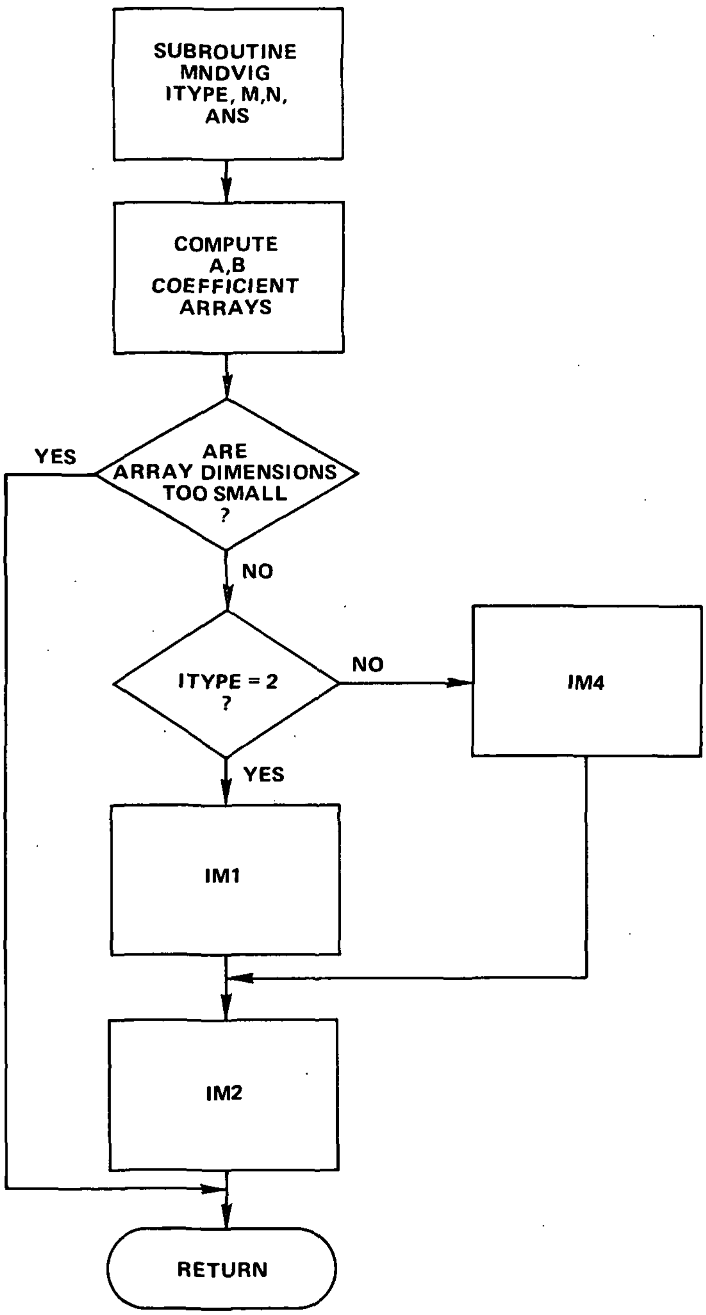


Figure 2-3. Subroutine MNDVIG

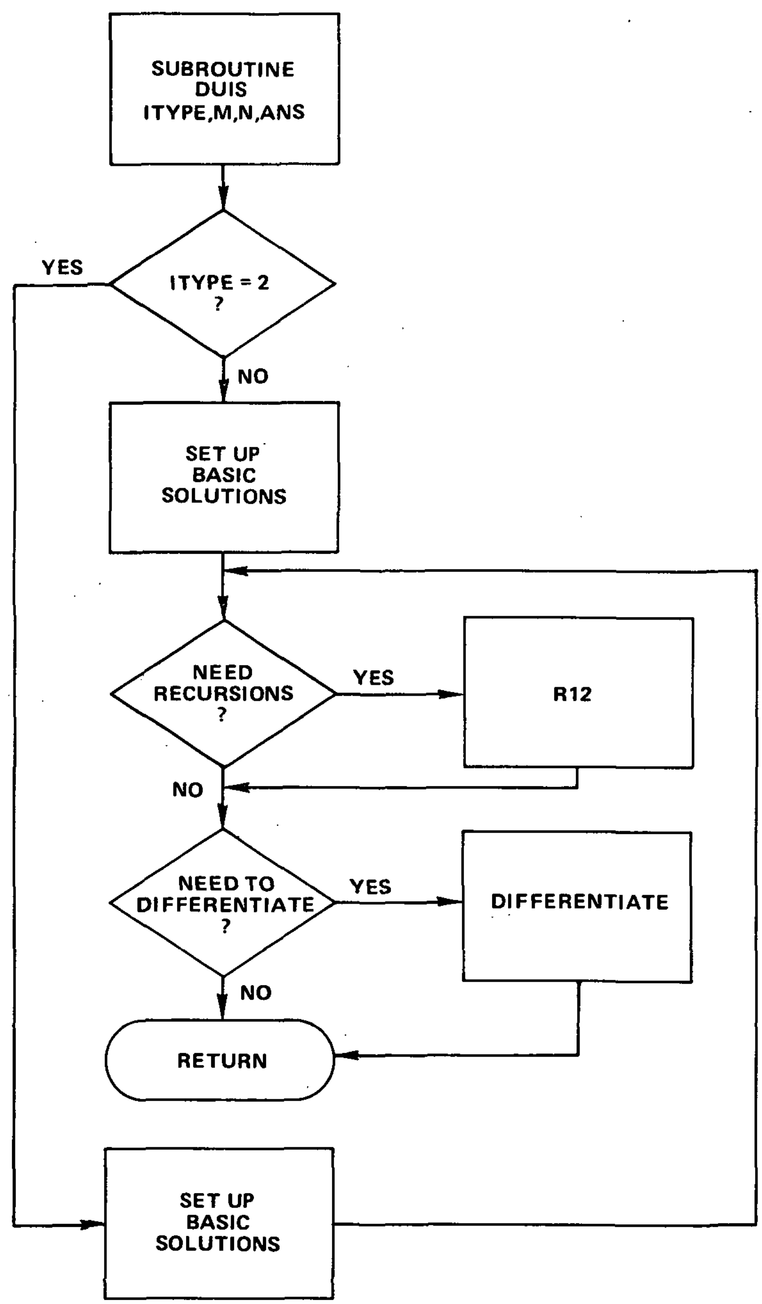


Figure 2-4. Subroutine DUIS

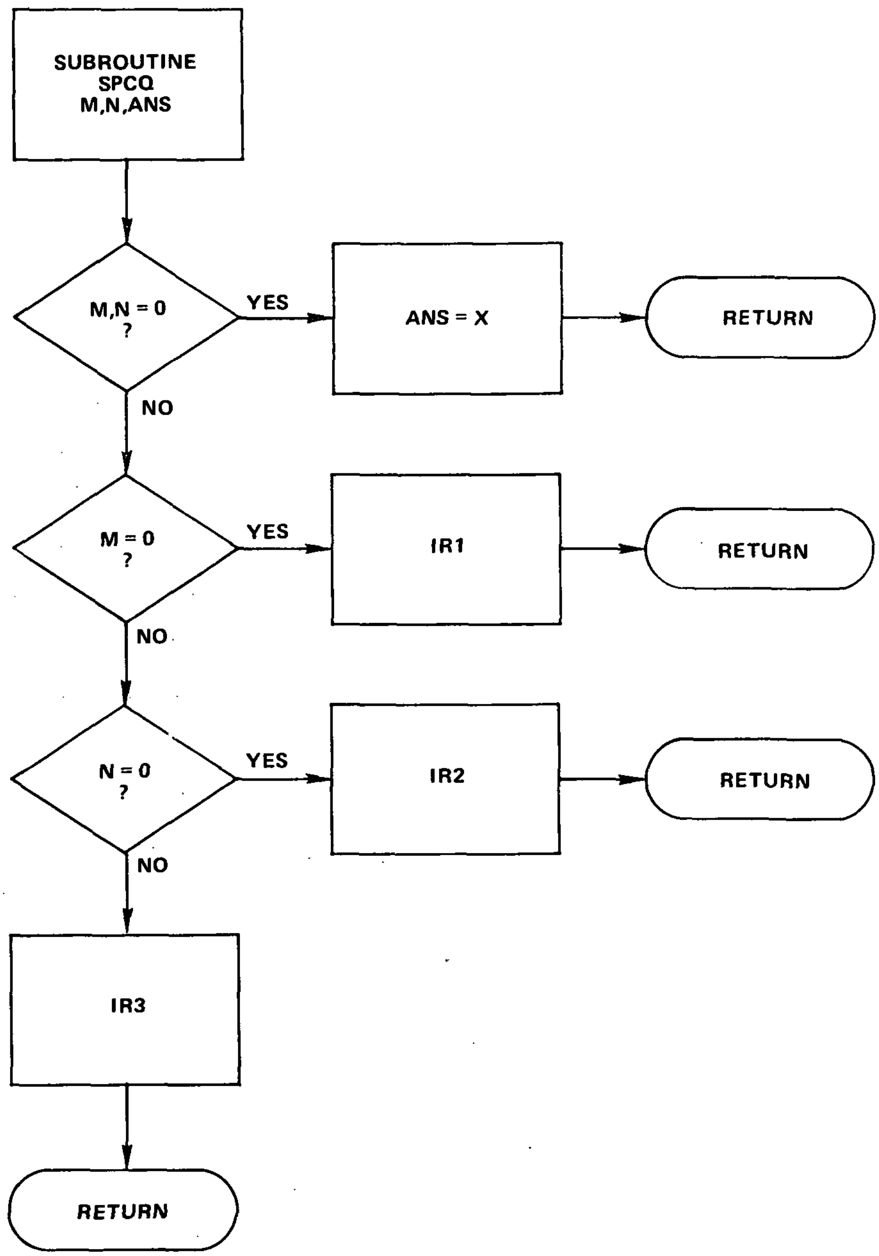


Figure 2-5. Subroutine SPCQ

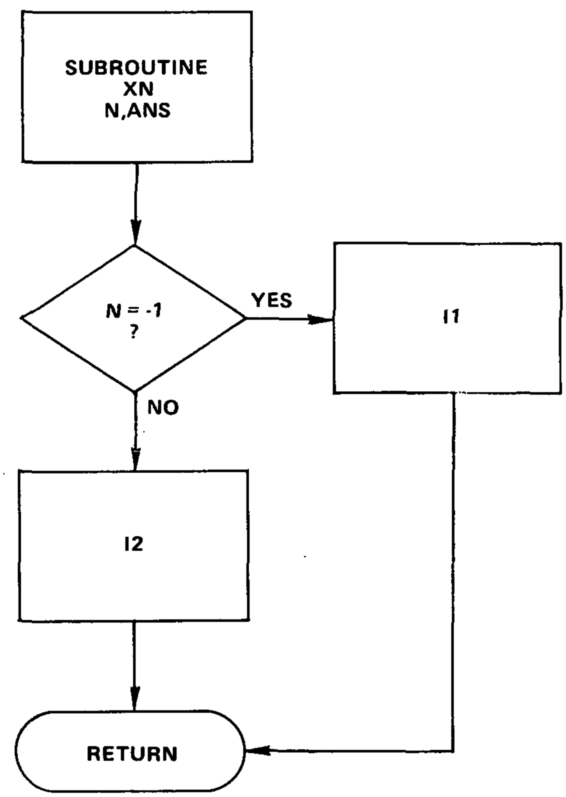


Figure 2-6. Subroutine XN

2.4 PROGRAM SUBROUTINES

All program subroutines are FORMAC routines.

2.4.1 Subroutine IDIGTE

This routine uses the integration parameters p, q, n to select the proper logic for the type integral being considered and calls the various integration routines required to effect the actual integration.

Argument List	Variable Type
M1 - m (input)	FORTRAN
M2 - n (input)	FORTRAN
ANS - I (output answer)	FORMAC

2.4.2 Subroutine MNDVIG

This routine performs a multinominal division to decompose the integrand into a quotient and remainder and then effects a termwise integration of each. (Two basic types of integrals are considered.)

Argument List		Variable Type
ITYPE	- selects integral type (input) values 1 or 2	FORTRAN
M	- exponent of numerator (input)	FORTRAN
N	- exponent of denominator (input)	FORTRAN
ANS	- integration answer (output)	FORMAC

2.4.3 Subroutine DUIS

This routine integrates by differentiating (parametrically) under the integral sign. (Two basic integral types are considered.)

Argument List		Variable Type
ITYPE	- selects integral type (input) values 1 or 2	FORTRAN
M	- exponent of numerator (input)	FORTRAN
N	- exponent of denominator (input)	FORTRAN
ANS	- integration result (output)	FORMAC

2.4.4 Subroutine SPCQ

This routine integrates various products of sin x cos x.

Argument List	Variable Type
M - exponent of sin x (input)	FORTRAN
N - exponent of cos x (input)	FORTRAN
ANS - integration result (output)	FORMAC

2.4.5 Subroutine XN

This routine integrates powers of x .

Argument List	Variable Type
N - exponent of x (input)	FORTRAN
ANS - integration result	FORMAC

2.5 MAIN PROGRAM MNEMONICS

Symbol	Mnemonics	Type	Purpose
	NTERM	FORTRAN	Number of differential equations per set
	IFLAG	FORTRAN	Specifies initial set of differential equations
dI/dx	DIDT	FORMAC	Differential equations to be integrated
(1 + e cos x)	PBYR	FORMAC	Represents expression (1 + e cos x)
	SOL	FORMAC	Solution to differential equation
	SEC	FORMAC	Secular component of solution
	NT	FORTRAN	Number of terms in differential equation
x	X	FORMAC	Integration variable
	CF	FORMAC	Coefficient of integrand
p	P	FORTRAN	Exponent of sin x
q	Q	FORTRAN	Exponent of cos x
n	N	FORTRAN	Exponent of (1+e cos x)
	QQ	FORTRAN	Flag used during expression output to signal output termination

2.6 PROGRAM LISTING

The following is a sample of the program listing.

```
$JOB            C113 MEYER   BIN 406,310030,10,12,14CF
$PAUSE          RESET ALL CONSOLE KEYS
$EXECUTE        IBJOB
$IBJOB          FORMAC,FIOCS,MAP
$IBFMC DRIVE    M94/2
      INTEGER OUTPUT(20)
      INTEGER DUMMYO
      INTEGER DELDT,E,X,ANS,ANSCK,W,DUMMYO,DUMMYF
      INTEGER P,Q,DIDXT,DUMMY
      INTEGER PBYR
      SYMARG
      ATOMIC M
      ATOMIC PBYR
      ATOMIC E,X,W,I
 LBL  PARAM (X,M),(FMCSIN(X),FMCSIN(M)D,(FMCCOS(X),FMCCOS(M))
      LET ANS=0
      NN=1
1     LET DUMMYO=5/2*E*FMCSIN(W)*FMCCOS(I)**2
      LET DELDT=DUMMYO*FMCSIN(X)**2*FMCCOS(X)**3*PBYR**2
      LET DUMMYO=DUMMYO
      GO TO 1001
10    CALL IDIGTE(2,3,2,ANS,LBL)
      LET ANS=DUMMYO*ANS
      LET ANS=EXPAND ANS
      WRITE(6,50)
      WRITE(6,103)
      QQ=0.
104   LET QQ=BCDCON ANS,OUTPUT,19
      WRITE(6,101) (OUTPUT(KK),KK=2,19D
      IF(QQ.NE.0.) GO TO 104
      LET ANS=SUBST ANS,(FMCSIN(X),FMCSIN(M))
      LET ANS=SUBST ANS,(FMCCOS(X),FMCCOS(M))
      LET ANS=SUBST ANS,(FMCCOS(M)**2,U1-FMCSIN(M)**2))
      GO TO 98
2     WRITE(6,50)
      LET DUMMYO=(1+E**2)**(1/2)
      LET DELDT=DUMMYO*FMCSIN(X)**2*FMCCOS(X)**3/PBYR**2
      LET DUMMYO=DUMMYO
      GO TO 1001
20    CALL IDIGTE(DELDT,2,3,-2,ANS,DUMMYO)
      GO TO 98
3     WRITE(6,50)
      LET DUMMYO=1
      LET DELDT=DUMMYO*FMCSIN(X)**3*FMCCOS(X)**2/PBYR**3
      LET DUMMYO=DUMMYO
      GO TO 1001
30    CALL IDIGTE(DELDT,3,2,-3,ANS,DUMMYO)
      GO TO 98
4     WRITE(6,50)
      LET DUMMYO=1
      LET DELDT=FMCSIN(X)**2/PBYR**3
      LET DUMMYO=DUMMYO
      GO TO 1001
40    CALL IDIGTE(DELDT,2,0,-3,ANS,DUMMYO)
      GO TO 98
```

```
5       WRITE(6,50)
        LET DUMMYO=1
        LET DELDT=FMCSIN(X)**3/PBYR**2
        LET DUMMYO=DUMMYO
        GO TO 1001
501     CALL IDIGTE(DELDT,3,0,-2,ANS,DUMMYO)
        GO TO 98
6       WRITE(6,50)
        LET DUMMYO=1
        LET DELDT=FMCCOS(X)**3/PBYR**3
        LET DUMMYO=DUMMYO
        GO TO 1001
60      CALL IDIGTE(DELDT,0,3,-3,ANS,DUMMYO)
        GO TO 98
98      QQ=0.
        WRITE(6,50)
50      FORMAT(1H1,///)
        WRITE(6,102)
102     FORMAT(///13H  COEFFICIENT/)
99      LET QQ=BCDCON DUMMYO,OUTPUT,1
        WRITE(6,101) (OUTPUT(II),II=2,19)
        IF(QQ.NE.0.) GO TO 99
        QQ=0.
        WRITE(6,103)
103     FORMAT(///5H  ANS/)
100     LET QQ=BCDCON ANS,OUTPUT,19
        WRITE(6,101) (OUTPUT(II),II=2,19)
        IF(QQ.NE.0.) GO TO 100
        QQ=0.
        NN=NN+1
        NN=7
        GO TO(1,2,3,4,5,6,7),NN
1001    CONTINUE
C   THE FOLLOWING LOGIC IS DESIGNED TO ACCEPT A GIVEN INTEGRAND OF THE
C   FORM
C       DIDX=K*SIN(X)**P*COS(X)**Q*(1+E*COS(X))**N
C   WHERE K IS INDEPENDENT OF X.
C   ISOLATE K,AND DETERMINE P,Q,N FO- PURPOSES OF INTEGRAL IDENTIFICAT
C   ION.
C   START LOGIC
        LET DIDXT=DELDT
        LET DUMMY=COEFF DIDXT,FMCSIN(X)*M0,V1,V2
        P=V2
        LET DUMMY=COEFF DIDXT,FMCCOS(X)**0,V1,V2
        Q=V2
        LET DUMMY=COEFF DIDXT,PBYR**0,V1,V2
        N=V2
        LET DUMMY=DIDXT/(FMCSIN(X)**P*FMCCOS(X)**Q*PBYR**N)
        LET DUMMY=EXPAND DUMMY
        WRITE(6,804)
804     FORMAT(1H0,5HDUMMY,///)
        QQ=0.
1026    LET QQ=BCDCON DUMMY,OUTPUT,19
        WRITE(6,101) (OUTPUT(KK),KK=2,19)
        IF(QQ.NE.0.) GO TO 1026
```

```
      WRITE(6,105) P,Q,N
105   FORMAT(1H0,1X,2HP=,I2,1X,2HQ=,I2T1X,2HN=,I2/)
C  END LOGIC
      GO TO (10,20,30,40,501,60,7),NN
101   FORMAT(1H0,19A6/)
7     STOP
      END
$IEDIT          SYSCK1
$IBFTC DRIVE    M94/2
$IEDIT
$IBFMC IPM00    M94/2
C  SUBROUTINE IDIGTE IS DESIGNED TO INTEGRATE THE FOLLOWING DIFFERENT
C  IAL FORM
C              DI/DX=SIN(X)**P*COS(X)**-*(1+E*COS(X))**N
C  P,Q,N INTEGRAL,P,Q NONNEGATIVE
C  THE SOLUTION METHOD SELECTED DEPENDS ON P,Q,N
C  THE INTEGRATION IS DONE VIA INTEGRATION SUBROUTINES AS REQUIRED
C  THE INTEGRATION IS DONE VIA INTEGRATION SUBROUTINES AS RQUIRED
C CALLING SEQUENCE DEFINITIONS
C INPUT VARIABLE
C     M1***EXPONENT OF SIN(X)
C     M2***EXPONENT OF COS(X)
C     M3***EXPONENT OF (1+E*COS(X))
C OUTPUT VARIABLE
C     ANS***INTEGRAL SOLUTION
C
      SUBROUTINE IDIGTE(M1,M2,M3,ANS,LBL)
      INTEGER ANS,ANSI,ANSII,ANSIII,E,P,Q,QI,X
      SYMARG ANS
      ATOMIC E,X
      AUTSIM QINT
      LET ANS=0
      P=M1
      Q=M2
      N=M3
C TEST N FOR INTEGRAND TYPE
      IF(N.LT.0) GO TO 2
C START LOGIC BLOCK FOR N.GE.0
      CALL SPCQ(P,Q,ANS)
      IF(N.EQ.0) GO TO 10
      LET ANSI=0
      DO 1 I=1,N
      QI=Q+I
      LET ANSII=0
      CALL SPCQ(P,QI,ANSII)
      LET ANSI=FMCFAC(N)/(FMCFAC(I)*FMCFAC(N-I))*E**I*ANSII+ANSI
1     CONTINUE
      LET ANS = ANS+ANSI
10    LET ANS=EXPAND ANS
      LET ANS=SUBST ANS,LBL
      RETURN
C END LOGIC BLOCK FOR N.GE.0
C START LOGIC BLOCK FOR N.LT.0
2     N=-N
C  PARTICULAR ATTENTION MUST BE PAID TO ALL POSSIBLE CASES FOR THIS TYPE
```

```
C   INTEGRAND AS THESE CASES REQUIRE DIFFERENT INTEGRATION METHODS
        IF((P.EQ.0).AND.(Q.EQ.0)) GO TO 5
        IF(P.EQ.0) GO TO 4
        IF(Q.EQ.0) GO TO 3
C DETERMINE IF P IS EVEN
        IF((MOD(P,2)).EQ.0) GO TO 24
C  SINCE P IS ODD TRANSFORM THE INTEGRAND BY X=COS(X) TO
C  DI/DX=-(1-X**2)**K*X**Q/U1+E*X)**N        K=(P-1)/2
C EACH TERM IN THE EXPANSION OF THE INTEGRAND IS OBTAINED,
C THEN INTEGRATED EITHER BY DIFFERIENTATION UNDER AN
C INTEGRAL SIGN OR BY MULTINOMIAL DIVISION
       K=(P-1)/2
       LET ANSI=0
       MQ=Q
       IF(MQ.GE.N) GO TO 21
       CALL DUIS(2,MQ,N,ANSI)
       GO TO 22
21     CALL MNDVIG(2,MQ,N,ANSI)
22     LET ANS=-ANSI
       IF(K.EQ.0) GO TO 232
       LET ANSI=0
       DO 23 I=1,K
       LET ANSII=0
       MQ=2*I+Q
       IF(MQ.GE.N) GO TO 230
       CALL DUIS(2,MQ,N,ANSII)
       GO TO 231
230    CALL MNDVIG(2,MQ,N,ANSII)
231    LET ANSI = -(-1)**I*FMCFAC(K)/(FMCFAC(I)*FMCFAC(K-I))*ANSII+ANSI
23     CONTINUE
       LET ANS=ANS+ANSI
232    LET ANS=EXPAND ANS
       LET ANS=DUMMY*ANS
       LET ANS=EXPAND ANS
       RETURN
24     CONTINUE
C P IS EVEN. USE SIN(X)**P = (1-COS(X)M*2)**K,K=P/2
C TO GET DI/DX = (1-COS(X)**2)**K*COS(X)**Q/(1+E*COS(X))**N
       K=P/2
       MQ=Q
       LET ANSI=0
       IF(MQ.GE.N) GO TO 25
       CALL DUIS(1,MQ,N,ANSI)
       GO TO 26
25     CALL MNDVIG(1,MQ,N,ANSI)
26     LET ANS=ANSI
       IF(K.EQ.0) GO TO 272
       LET ANSI=0
       DO 27 I=1,K
       LET ANSII=0
       I2=2*I
       MQ=I2+Q
       IF(MQ.GE.N) GO TO 270
       CALL DUIS(1,MQ,N,ANSII)
       GO TO 271
```

```
270    CALL MNDVIG(1,MQ,N,ANSII)
271    LET ANSI=(-1)**I*FMCFAC(K)/(FMCFAC(I)*FMCFAC(K-I))*ANSII+ANSI
27     CONTINUE
       LET ANS=ANS+ANSI
272    LET ANS=EXPAND ANS
       LET ANS=DUMMY*ANS
       LET ANS=EXPAND ANS
       RETURN
3      CONTINUE
C INTEGRAND IS FORM SIN(X)**P/(1+E*COS(X))**N
C DETERMINE IF P IS EVEN
       IF((MOD(P,2)).EQ.0) GO TO 32
C SINCE P IS ODD TRANSFORM THE INTEGRAND BY X=1+E*COS(X)
C TO DI/DX = (-1/E**P)*(E**2-(X-1)**2)M*K/X**N
C WHERE K=(P-1)/2
C EACH TERM IN THE EXPANSION OF THE INTEGRAND IS OBTAINED
C THEN INTEGRATED AS AN INTEGRAL POWER OF X.
       K=(P-1)/2
       LET ANSI=0
       CALL XN(-N,ANSI)
       LET ANS=(-1/E)*ANSI
       IF(K.EQ.0) GO TO 310
       LET ANSI=0
       DO 31 I=1,K
       LET ANSII=0
       I2=2*I
       MM=I2-N
       CALL XN(MM,ANSII)
       DO 30 J=1,I2
       MM=I2-J-N
       LET ANSIII=0
       CALL XN(MM,ANSIII)
       LET ANSII=(-1)**J*FMCFAC(I2)/(FMCFAC(J)*FMCFAC(I2-J))*
      *ANSIII+ANSII
30     CONTINUE
       LET ANSI=(-1)**(I-1)*E**(-1-I2)*FMCFAC(K)/(FMCFAC(I)*FMCFAC(K-I))*
      *ANSII+ANSI
31     CONTINUE
       LET ANS=ANS+ANSI
310    LET ANS=EXPAND ANS
       LET ANS=DUMMY*ANS
       LET ANS=EXPAND ANS
       RETURN
32     CONTINUE
C P IS EVEN. USE SIN(X)**P=(1-COS(X)**2)**K,WHERE
C K=P/2 TO GET DI/DX = (1-COS(X)**2)**K/(1+E*COS(X))**N
       LET ANSI=0
       K=P/2
       CALL DUIS(1,0,N,ANSI)
       LET ANS=ANSI
       IF(K.EQ.0) GO TO 352
       LET ANSI=0
       DO 35 I=1,K
       LET ANSII=0
       I2=2*I
```

```
      IF(I2.GE.N) GO TO 350
      CALL DUIS(1,I2,N,ANSII)
      GO TO 351
350   CALL MNDVIG(1,I2,N,ANSII)
351   LET ANSI=(-1)**I*FMCFAC(K)/(FMCFAC(I)*FMCFAC(K-I))*ANSII+ANSI
35    CONTINUE
      LET ANS=ANS+ANSI
352   LET ANS=EXPAND ANS
      LET ANS=DUMMY*ANS
      LET ANS=EXPAND ANS
      RETURN
4     CONTINUE
C INTEGRAND IS FORM DI/DX = CIS(X)**Q/U1+E*COS(X))*N
      LET ANSI=0
      IF(Q.GE.N) GO TO 40
      CALL DUIS(1,Q,N,ANSI)
      GO TO 41
40    CALL MNDVIG(1,Q,N,ANSI)
41    LET ANS=ANSI
      LET ANS=EXPAND ANS
      LET ANS=DUMMY*ANS
      LET ANS=EXPAND ANS
      RETURN
5     CONTINUE
C     INTEGRAND IS FORM DI/DX = 1/(1+E*COS(X))**N
      LET ANSI=0
      CALL DUIS(1,0,N,ANSI)
      LET ANS=EXPAND ANSI
      LET ANS=DUMMY*ANS
      LET ANS=EXPAND ANS
      RETURN
C END LOGIC FOR N.LT.0
      END
$IEDIT             SYSCK1
$IBFTC IPM00       M94/2
$IEDIT
$IBFMC IPM01       M94/2
C SUBROUTINE MNDVIG INTEGRATES TWO DIFFERENT INTEGRAND
C FORMS BY MULTINOMIAL DIVISION, FOLLOWED BY TERM
C WISE INTEGRATION OF A FINITE SUM OF BASIC INTEGRAND
C THE INTEGRAND FORMS ARE
C       DI/DX=-COS(X)**M/(1.+E*COS(X))*MN M.GE.N
C       DI/DX = X**M/(1.+E*X)**N    M.GECN
C THE DIVISION YIELDS A QUOTIENT WH+CH IS A
C SUM OF (M-N+1) BASIC INTEGRANDS OF T+E FORM
C       A(J)*COS(X)**(M-N+1-J)
C OR    A(J)*X**(M-N+1-J)
C WITH THE A(J) CONSTANT AND 1.LE.J.LEC(M-N+1),
C AND A REMAINDER WHICH IS A SUM OF N BASIC
C INTEGRANDS OF THE FORM
C       B(N-K)*COS(X)**K/(1.+E*COS(X))*MN
C OR    B(N-K)*X**K/(1.+E*X)**N
C WITH THE B(N-K) CONSTANT AND 0.LE.K.LE.(N-1)
C CALLING SEQUENCE DEFINITIONS
C INPUT VARIABLES
```

```
C     ITYPE *** INDICATES INTEGRAND TYPE,
C               ITYPE.EQ.1 FOR TYPE 1, 2 FOR TYPE2.
C     M     *** EXPONENT OF NUMERATOR
C     N     *** EXPONENT OF DENOMINATOR
C     ANS    ** INTEGRAL SOLUTION
C
      SUBROUTINE MNDVIG(ITYPE,M,N,ANS)
      INTEGER A,ANS,ANSI,ANSII,B,BSUM,F,X
      SYMARG ANS
      ATOMIC F,X
      DIMENSION A(20),B(20)
      AUTSIM QINT
      LET ANS=0
      MM=M
      NN=N
      MAXA=MM-NN+1
      ITYP=ITYPE
C ARE A,B COEFFICIENT ARRAYS LARGE ENOUGH
      IF(MAXA.GT.20) GO TO 6
      IF(NN.GT.20) GO TO 7
C SET UP A COEFFICIENT ARRAY
      LET A(1)=1/E**NN
      IF(MAXA.EQ.1) GO TO 10
      DO 1 I=2,MAXA
      LET A(I)=(-1/F)*(NN+I-2)/(I-1)*AUI-1)
1     CONTINUE
   10 CONTINUE
C SET UP B COEFFICIENT ARRAY
      LET B(NN)=-A(MAXA)
      N1=NN-1
      DO 2 K=1,N1
      LET BSUM=FMCFAC(NN)/(FMCFAC(NN-KD*FMCFAC(K))*F**K*A(MAXA)
      DO 3 L=1,K
      IF(MAXA.LE.L) GO TO 3
      LET BSUM=BSUM+FMCFAC(NN)/(FMCFACUNN-K+L)*FMCFAC(K-L))*F**(K-L)*A(M
     *AXA-L)
3     CONTINUE
      NNK=NN-K
      LET B(NNK )=-BSUM
      LET BSUM=0
2     CONTINUE
C INTEGRATE QUOTIENT TERMWISE
      LET ANSI=0
      DO 4 I=1,MAXA
      MAXAI=MAXA-I
      IF(ITYP.EQ.2) GO TO 40
      CALL SPCQ(0,MAXAI,ANSI)
      GO TO 41
40    CALL XN(MAXAI,ANSI)
41    LET ANS=ANS+A(I)*ANSI
      LET ANSI=0
4     CONTINUE
C INTEGRATE REMAINDER TERMWISE
      LET ANSI=0
      CALL DUIS(ITYP,0,NN,ANSI)
```

```
      LET ANS=ANS+B(NN)*ANSI
      IF(N1.EQ.0) RETURN
      DO 5 K=1,N1
      LET ANSI=0
      KI=K
      CALL DUIS(ITYP,KI,NN,ANSI)
      NNK=NN-K
      LET ANS=ANS+B(NNK)*ANSI
5     CONTINUE
      RETURN
6     WRITE(6,60)
60    FORMAT(///17H A ARRAY TO SMALL)
      RETURN
7     WRITE(6,70)
70    FORMAT(///17H B ARRAY TO SMALL)
      RETURN
      END
$IEDIT          SYSCK1
$IBFTC IPM01    M94/2
$IEDIT
$IBFMC IPM02    M94/2
C SUBROUTINE DUIS INTEGRATES THE FOLLOWING FORMS
C        DI/DX= COS(X)**M/(1+E*COS(X))**N      M.LT.N
C        DI/DX= X**M/(1+E*X)**N                M.LT.N
C THE INTEGRAL EVALUATIONS REQUIRE THE USE OF RECURSIVE
C       RELATIONS AND/OR DIFFERENTIAT+ON UNDER AN INTEGRAL
C       SIGN.
C       CALLING SEQUENCE DEFINITIONS
C INPUT VARIABLES
C    ITYPE *** SELECTS INTEGRAND TYPE
C    M     ***    EXPONENT OF NUMERATOR
C    N     ***    EXPONENT OF DENOMINATOR
C OUTPUT VARIABLE
C    ANS ***    INTEGRAL SOLUTION
C
      SUBROUTINE DUIS(ITYPE,M,N,ANS)
      INTEGER A,ARG,ANS11,ANS12,ANS11ITANS12I,ANS13,ANS2,B,C,E,X
      INTEGER ANS
      SYMARG ANS
      ATOMIC E,X
      AUTSIM QINT
      LET ANS=0
      LET ANS11I=0
      LET ANS12I=0
      LET ANS2=0
      MM=M
      NN=N
      ITYP=ITYPE
      K=NN-MM
C DETERMINE INTEGRAND TYPE
      IF(ITYP.EQ.2) GO TO 2
C SET UP BASIC SOLUTIONS FOR TYPE 1
      LET ARG=((1-E)/(1+E))**(1/2)*FMCSIN(X/2)/FMCCOS(X/2)
      LET ANS11=2/(1-E**2)**(1/2)*FMCATN(ARG)
      LET ANS12=1/(1-E**2)*ANS11-E*FMCSIN(X)/((1-E**2)*(1+E*FMCCOS(X)))
```

```
      IF(K.GT.2) GO TO 12
      IF(K.EQ.2) GO TO 10
      IF(MM.GT.0) GO TO 1
      LET ANS=ANS11
      RETURN
1     LET ANS=(-1)**MM*FMCFAC(K-1)/FMCFAC(K+MM-1)*FMCDIF(ANS11,F,MM)
      RETURN
10    IF(MM.GT.0) GO TO 11
      LET ANS=ANS12
      RETURN
11    LET ANS=(-1)**MM*FMCFAC(K-1)/FMCFAC(K+MM-1)*FMCDIF(ANS12,F,MM)
      RETURN
12    LET A=-(NN-1)*(1-E**2)/E
      LET B=(2*NN-3)/E
      LET C=-(NN-2)/E
      LET ANS11I=ANS11
      LET ANS12I=ANS12
      LET ANS13=0
      DO 13 I=3,K
      LET ANS13=ANS13+FMCSIN(X)/(1+E*FMCCOS(X))**(NN-1)
     *-B*ANS12I-C*ANS11I
      LET ANS13=ANS13/A
      LET ANS11I=ANS12I
      LET ANS12I=ANS13
13    CONTINUE
      IF(MM.EQ.0) GO TO 14
      LET ANS=(-1)**MM*FMCFAC(K-1)/FMCFAC(K+MM-1)*FMCDIF(ANS13,F,MM)
      RETURN
14    LET ANS=ANS13
      RETURN
2     CONTINUE
C CHECK FOR K.EQ.1
      IF(K.EQ.1) GO TO 21
      LET ANS2=1/((1-K)*E)*(1+E*X)**(1-K)
      IF(MM.EQ.0) GO TO 20
      LET ANS=(-1)**MM*FMCFAC(K-1)/FMCFAC(K+MM-1)*FMCDIF(ANS2,F,MM)
      LET ANS=SUBST ANS,(X,FMCCOS(X))
      RETURN
20    LET ANS=ANS2
      LET ANS=SUBST ANS,(X,FMCCOS(X))
      RETURN
21    LET ANS2=1/E*FMCLOG(1+E*X)
      IF(MM.EQ.0) GO TO 22
      LET ANS=(-1)**MM*FMCFAC(K-1)/FMCFAC(K+MM-1)*FMCDIF(ANS2,F,MM)
      LET ANS=SUBST ANS,(X,FMCCOS(X))
      RETURN
22    LET ANS=ANS2
      LET ANS=SUBST ANS,(X,FMCCOS(X))
      RETURN
      END
$IEDIT          SYSCK1
$IBFTC IPM02    M94/2
$IEDIT
$IBFMC IPM03    M94/2
C SUBROUTINE SPCQ INTEGRATES THE FOLLOWING FORM
```

```
C       DI/DX = SIN(X)**M*COS(X)**N   M,N.GE.0 (INTEGER)
C STANDARD RECURSIVE RELATIONS ARE USED IN THE EVALUATION
C CALLING SEQUENCE DEFINITIONS
C INPUT VARIABLE
C   M       *** EXPONENT OF SIN(X)
C   N       *** EXPONENT OF COS(X)
C OUTPUT VARIABLE
C   ANS     *** INTEGRAL SOLUTION
C
      SUBROUTINE SPCQ(M,N,ANS)
      INTEGER A,ANS,ANSI,B,BSAVE,X
      SYMARG ANS
      ATOMIC X
      MM=M
      NN=N
      LET ANS=0
      IF((MM.EQ.0).AND.(NN.EQ.0)) GO TO 4
      IF(MM.EQ.0) GO TO 2
      IF(NN.EQ.0) GO TO 3
      LET B=1
1     LET ANS=ANS+B/(MM+NN)*FMCSIN(X)*M(MM+1)*FMCCOS(X)**(NN-1)
      LET A=(NN-1)/(MM+NN)
      LET B=B*A
      NN=NN-2
      IF(NN.GT.0) GO TO 1
      IF(NN.LT.0) RETURN
      LET BSAVE=B
      LET B=1
      LET ANSI=0
10    LET ANSI=ANSI-B/MM*FMCSIN(X)**(MM-1)*FMCCOS(X)
      LET A=(MM-1)/(MM)
      LET B=B*A
      MM=MM-2
      IF(MM.GT.0) GO TO 10
      IF(MM.LT.0) GO TO 11
      LET ANSI=ANSI+B*X
11    LET ANS=ANS+BSAVE*ANSI
      RETURN
2     LET B=1
20    LET ANS=ANS+B/NN*FMCSIN(X)*FMCCOS(X)**(NN-1)
      LET A=(NN-1)/(NN)
      LET B=B*A
      NN=NN-2
      IF(NN.GT.0) GO TO 20
      IF(NN.LT.0) RETURN
      LET ANS=ANS+B*X
      RETURN
3     LET B=1
30    LET ANS=ANS-B/MM*FMCSIN(X)**(MM-1)*FMCCOS(X)
      LET A=(MM-1)/MM
      LET B=B*A
      MM=MM-2
      IF(MM.GT.0) GO TO 30
      IF(MM.LT.0) RETURN
      LET ANS=ANS+B*X
```

```
      RETURN
4     LET ANS=ANS+X
      RETURN
      END
$IEDIT          SYSCK1
$IBFTC IPM03    M94/2
$IEDIT
$IBFMC IPM04    M94/2
C SUBROUTINE XN INTEGRATES ALL INTEGRAL POWERS OF X
C CALLING SEQUENCE DEFINITIONS
C INPUT VARIABLE
C     N***        EXPONENT OF X
C OUTPUT VARIABLE
C     ANS***----INTEGRAL SOLUTION
C
      SUBROUTINE XN(N,ANS)
      INTEGER ANS,X,F
      SYMARG ANS
      ATOMIC X,F
      LET ANS=0
      NN=N
      IF((NN+1).EQ.0) GO TO 1
      NP1=NN+1
      LET ANS=X**NP1/NP1
      LET ANS=SUBST ANS,(X,(1+F*FMCCOSUX)))
      RETURN
1     LET ANS=FMCLOG(X)
      LET ANS=SUBST ANS,(X,(1+F*FMCCOS(X)))
      RETURN
      END
$IEDIT          SYSCK1
$IBFTC IPM04    M94/2
$IEDIT
```

2.7 DECK SETUP

Figure 2-7 represents the deck setup of the program and Figure 2-8 is a sample instruction card.

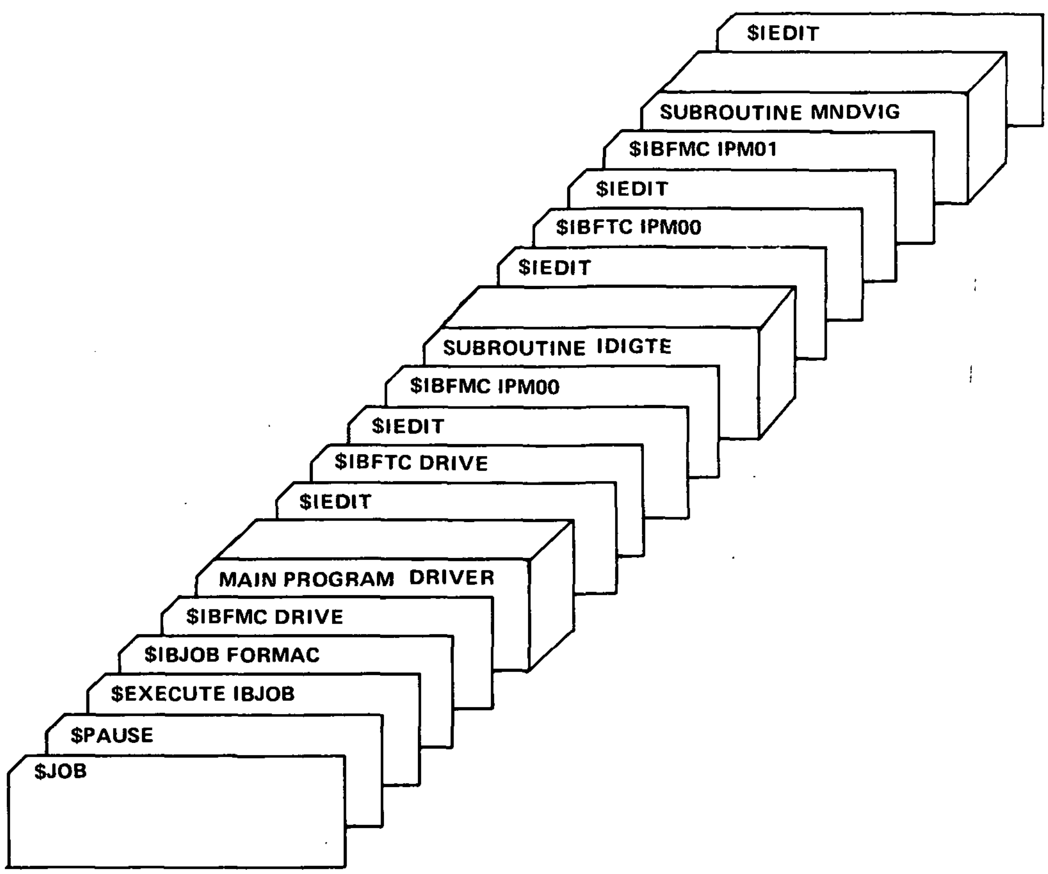


Figure 2-7. Deck (Sheet 1 of 2)

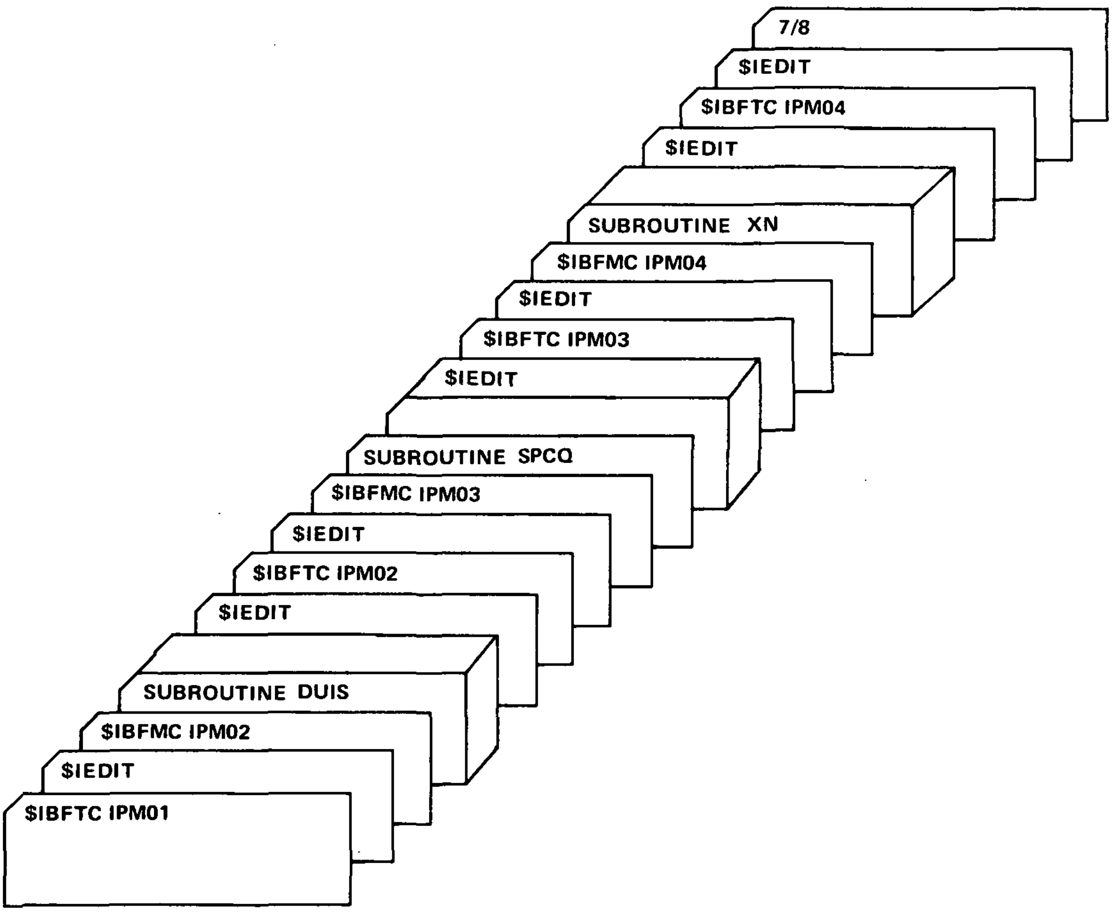


Figure 2-7. Deck (Sheet 2 of 2)

7094-__________ INSTRUCTIONS

NAME:	OP CODE: 12	STACK #
BIN # / LOC:	JOB: # 310030	

IF EXCEEDS MAX:

☐ STR ☐ STZ ☐ DMP ☐ RETSY

☒ IBSYS
☐ SPOOK
☐ OTHER

☒ COMPL /ASSMBL
☒ EXECUTE
☐ PUNCH (BCD BIN)

☒ 4 FTRN
☐ 2 FTRN
☐ APT
☐ PERT

☐ MAP
☐ FAP
☐ SCAT
☒ OTHER

FAST TAPES: A B C D

INPUT TAPES			WORK
LOGIC	REEL NO.	DEN	LOGIC
			A5

LINES OF OUTPUT (1000'S)

☒ 0-5 ☐ 5-15 ☐ 15-30 ☐ OVER

MAXIMUM TIME:

HOURS ____ MINUTES 10

NUMBER OF CASES ____

(TYPICAL)

PROGRAMMER COMMENTS:

FORMAC

OVER: ____

☐ SEE ON-LINE
☐ SEE TECHNIQUE:
☐ MAX EXCEEDED
☐ RETURN TO SYS
☐ LINE MAX

OPERATOR COMMENTS:

OPER INIT: ____
OVER: ____

OUTPUT TAPES ONLY						4020
REEL NO.	LOGIC	DEN.	UNIT	NO OF CPYS	SAVE	TAPE
	B-1	8				

NO FILES	NO FRAMES	COPIES		DENSITY		COPY-FLO		KALVAR
		P	F	5	8	P	F	

MSFC - Form 533 (Rev February 1966)

Figure 2-8. Sample Instruction Card

2.8 LIMITATIONS ENCOUNTERED

The application of FORMAC, as described herein, revealed several system limitations. The first, and most severe, of these limitations was the lack of sufficient free list storage. On a number of occasions, program execution was terminated while expression manipulation was being carried out. This required breaking down sequences of analytical computations to obtain partial solutions and, on some occasions, running the program "piecemeal."

Problems were also encountered in communicating between the various program elements, i.e., between the main program and a subroutine, or between two subroutines. For example, a result obtained in a subroutine could be communicated to the main program for I/O purposes, but could not be differentiated.

The isolation of the integration parameters $\underline{p}$, $\underline{q}$, $\underline{n}$ was unduly complicated since there was no convenient way to separate the numerator of a fraction from its denominator. In conjunction with this, it was discovered that the automatic simplification transformations did not work as expected. For example, one such transformation is stated as

$$\log (x \cdot y) = \log x + \log y$$

Since expressions are stored in internal Polish delimeter form (Ref. 2), it was expected that application of the transformation to a product expression such as

$$\log \left[\sin^3 x/(1 + e \cos x)^2\right]$$

would yield

$$3 \log \sin x - 2 \log (1 + e \cos x)$$

However, it did not; in fact, no transformation was made at all. The only explanation for this is that $\underline{x}$ and $\underline{y}$ must be separate expressions before the transformation can be applied.

These limitations, while not entirely prohibitive, were an inconvenience and were circumvented only by special programming efforts. Such efforts required additional programming time and detracted slightly from the overall program efficiency.

REFERENCES

1. Barry, B. F., Pimm, R. S., and Rowe, C. K., "Techniques of Orbital Decay and Long-Term Ephemeris Prediction for Satellites in Earth Orbit," Computer Sciences Corporation, November 1971.

2. "FORMAC Programmer's Guide," Common Research Computer Facility, Houston, Texas, 1967.

3. Sokolnikoff, I. S., "Advanced Calculus," McGraw-Hill Book Company, New York, 1939.

APPENDIX A - INTEGRAL SOLUTION FORMULATION

This appendix contains the formulation of the solution for integrals of the form

$$\int \frac{\sin^p x \cos^q x \, dx}{(1 + e \cos x)^n} \qquad \text{(A-1)}$$

where p, q, n are integers (p, $q \geq 0$) and $0 < e < 1$. Two basic cases arise, depending on whether n is negative or non-negative.

Case 1 (n negative). If n is negative the integral becomes

$$\int \sin^p x \cos^q x \, (1 + e \cos x)^{-n} \, dx \qquad \text{(A-2)}$$

and application of the binomial expansion transforms this integral to

$$\sum_{i=0}^{m} \frac{m! \, e^i}{(m - i)! \, i!} \int \sin^p x \cos^{q+i} x \, dx \qquad m = -n \qquad \text{(1-1)}$$

Each integral appearing in this summation can be evaluated by repeated application (backward recursion) of

$$\int \sin^a x \cos^b x \, dx = \frac{\sin^{a+1} x \cos^{b-1} x}{a + b} + \frac{b - (2K - 1)}{a + b} \int \sin^a x \cos^{b-2K} x \, dx \qquad \text{(1-1a)}$$

$K = 1, 2, \cdots$, until $b - 2K = 1$ or 0.

Case 2 (n non-negative). If n is non-negative, two subcases arise as p is even or odd. If p is odd, transform the integrand by $z = \cos x$ and employ the binomial expansion to yield

$$-\sum_{i=0}^{K} (-1)^i \frac{K!}{(K - 1)! \, i!} \int \frac{z^{2(K-i)+q}}{(1 + ez)^n} dz \qquad (p = 2K + 1) \qquad \text{(A-3)}$$

If $\underline{p}$ is even, the binomial expansion (using $\cos^2 x = 1 - \sin^2 x$) will yield

$$\sum_{i=0}^{K} (-1)^i \frac{K!}{(K-i)!\,i!} \int \frac{\cos^{2(K-i)+q} x}{(1 + e \cos x)^n}\, dx \qquad (p = 2K) \tag{A-4}$$

In either case, the required integrals can be obtained (see Appendix B) by parametrically differentiating (Ref. 3) one or the other of two fundamental integrals, or by a multinomial division followed by a termwise integration of the quotient (integrating powers of z or $\cos x$) and a termwise integration of the remainder by parametric differentiation of these same fundamental integrals.

APPENDIX B - DERIVATION OF THE FUNDAMENTAL INTEGRALS

This appendix contains the derivations of the solutions for the two integrals

$$I_1 = \int \frac{z^a \, dz}{(1 + ez)^n} \qquad (z = \cos x)$$

and

$$I_2 = \int \frac{\cos^a x \, dx}{(1 + e \cos x)^n}$$

required in Case 2 of Appendix A. The solution procedures are essentially the same in both instances.

Solution for I_1. If $a < n$, define the fundamental integral $I_{1\ell}$ by

$$I_{1\ell} = \int \frac{dz}{(1 + ez)^{\ell}} \qquad \ell = n - a \tag{B-1}$$

and differentiate $I_{1\ell}$ a times with respect to the parameter e to obtain

$$I_1 = (-1)^a \left[\frac{(\ell - 1)!}{(\ell + a - 1)!}\right] \frac{d^a I_{1\ell}}{de^a} \tag{B-2}$$

$I_{1\ell}$ is easily obtained as

$$I_{1\ell} = \begin{cases} -\dfrac{1}{e(\ell - 1)} (1 + ez)^{-\ell+1} & \ell \neq 1 \\ \dfrac{1}{e} \log (1 + ez) & \ell = 1 \end{cases} \tag{1-2b}$$

If $a \geq n$, expand the denominator and divide to obtain

$$\frac{z^a}{(1 + ez)^n} = \sum_{j=1}^{a-n+1} \alpha_j \, z^{a-n+1-j} + \sum_{K=0}^{n-1} \beta_{n-K} \frac{z^K}{(1 + ez)^n} \tag{B-3}$$

where

$$\alpha_1 = \frac{1}{e}, \quad \alpha_j = -\frac{1}{e}\left(\frac{n+j-2}{j-1}\right)\alpha_{j-1} \qquad 2 \le j \le a-n+1$$

(1-3a)

$$\beta_{n-K} = -\left(\sum_{\ell=0}^{K} \frac{n!\,e^{K-\ell}\,\alpha_{a-n+1-\ell}}{(K-\ell)!\,(n-K+\ell)!}\right) \qquad 0 \le K \le n-1, \ \alpha_0 = 0$$

Then

$$I_1 = \sum_{j=1}^{a-n+1} \alpha_j \int z^{a-n+1-j}\,dz + \sum_{K=0}^{n-1} \beta_{n-K} \int \frac{z^K\,dz}{(1+ez)^n}$$

$$= \sum_{j=1}^{a-n+1} \frac{\alpha_j}{a-n+2-j} z^{a-n+2-j} \tag{B-4}$$

$$+ \sum_{K=0}^{n-1} \beta_{n-K} (-1)^K \left[\frac{(\ell-1)!}{(\ell+K-1)!} \frac{d^K I_{1\ell}}{de^K}\right]$$

where $\ell = n - K$ and $I_{1\ell}$ is given above.

Solution for I_2. The solution procedure here is entirely analagous to that employed for I_1, but slightly more complicated. If $a < n$ define the fundamental integral

$$I_{2\ell} = \int \frac{dx}{(1 + e\cos x)^{\ell}} \qquad \ell = n - a \tag{B-5}$$

and differentiate $I_{2\ell}$ a times with respect to the parameter e to obtain

$$I_2 = (-1)^a \left[\frac{(\ell-1)!}{(\ell+a-1)!}\right] \frac{d^a I_{2\ell}}{de^a} \tag{B-6}$$

Here, the solution for $I_{2\ell}$ is not quite as easily obtained as for $I_{1\ell}$. In deriving this solution define

$$R(x) = \frac{\sin x}{(1 + e \cos x)^{\ell-1}}$$

Then

$$\frac{dR(x)}{dx} = \frac{A + B\,(1 + e \cos x) + C\,(1 + e \cos x)^2}{(1 + e \cos x)^{\ell}}$$

where

$$A = -(\ell - 1)\left(\frac{1 - e^2}{e^2}\right)$$

$$B = \frac{2n - 3}{e} \tag{1-4c}$$

$$C = -\frac{(n - 2)}{e}$$

Hence

$$R(x) = AI_{2\ell} + BI_{2\ell-1} + CI_{2\ell-2} \tag{B-7}$$

from which

$$I_{2\ell} = \frac{R(x) - BI_{2\ell-1} - CI_{2\ell-2}}{A} \tag{1-4b}$$

yielding $I_{2\ell}$ by forward recursion.

By elementary means

$$I_{21} = \frac{2}{\left(1 - e^2\right)^{1/2}} \tan^{-1}\left\{\left(\frac{1 - e}{1 + e}\right)^{1/2} \tan\frac{x}{2}\right\} \tag{1-4c}$$

and

$$I_{22} = \frac{1}{(1 - e^2)} I_{21} - \frac{e}{(1 - e^2)} \frac{\sin x}{(1 + e \cos x)} \tag{1-4c}$$

so that $I_{2\ell}$ is known for all ℓ.

If $a \geq n$

$$\begin{aligned} I_2 &= \sum_{j=1}^{a-n+1} \alpha_j \int \cos^{a-n+1-j} x \, dx + \sum_{K=0}^{n-1} \beta_{n-K} \int \frac{\cos^K x \, dx}{(1 + e \cos x)^n} \\ &= \sum_{j=1}^{a-n+1} \alpha_j \int \cos^{a-n+1-j} x \, dx + \sum_{K=0}^{n-1} \beta_{n-K} (-1)^K \left[\frac{(\ell - 1)!}{(\ell + K - 1)!} \frac{dI^K_{2\ell}}{de^K} \right] \end{aligned} \tag{1-5}$$

where α_j, β_{n-K} are defined as before and

$$\int \cos^m x \, dx = \frac{1}{m} \cos^{m-K} x \sin x + \frac{m-K}{m} \int \cos^{m-2K} x \, dx \tag{B-8}$$

$K = 1, 2, \cdots$, by backward recursion until $m - 2K = 1$ or 0.

CSC

COMPUTER SCIENCES CORPORATION

Major Offices and Facilities Throughout the World