

F/G 9/2

F33615-79-C-3601

F33615-79-C-3601

NL

24

2005

AFWAL-TR-80-3101
Volume III

LEVEL

1097550



AD A 097864

**MULTI-RATE DIGITAL CONTROL SYSTEMS WITH
SIMULATION APPLICATIONS
Volume III: Source Listings**

DENNIS G. J. DIDALEUSKY
FLIGHT DYNAMICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433

DTIC
ELECTR
APR 17 1981

SEPTEMBER 1980

TECHNICAL REPORT AFWAL-TR-80-3101 VOL III
Final Report — January 1979-May 1980

Approved for public release; distribution unlimited.

DTIC FILE COPY

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

81 4 8 038

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.


DENNIS G. J. DIDALEUSKY, Captain, USAF
Project Engineer
Control Dynamics Branch
Flight Control Division


R. O. ANDERSON, Chief
Control Dynamics Branch
Flight Control Division

FOR THE COMMANDER


ROBERT C. ETTINGER, Colonel, USAF
Chief, Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIGC, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(12) 116

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
18 1. REPORT NUMBER AFWAL-TR-80-3101-VOL III	2. GOVT ACCESSION NO. AD-A097864	3. RECIPIENT'S CATALOG NUMBER
6 4. TITLE (and Subtitle) MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS. Volume III: Source Listings.		9 5. TYPE OF REPORT & PERIOD COVERED Final Report. 29 Jan 79 - 29 May 80
10 7. AUTHOR(s) Dennis J. Didaleusky		14 8. PERFORMING ORG. REPORT NUMBER STI-TR-1142-1-3
9 9. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Technology, Inc. 13766 South Hawthorne Boulevard Hawthorne, California 90250		15 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 61102F, Project 2304, Mathematics, Task N3, Math. of Flt. Control
11 11. CONTROLLING OFFICE NAME AND ADDRESS Flight Dynamics Laboratory Air Force Wright Aeronautical Laboratories Wright-Patterson AFB, Ohio 45433		12 12. REPORT DATE September 1980
14 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 2304		17 15. SECURITY CLASS. (of this report) Unclassified
16 16. DISTRIBUTION STATEMENT (of this Report) Distribution limited to U.S. Government Agencies only; Test and Evaluation; September 1980. Other requests for this document must be referred to AFWAL/FICC, Wright-Patterson AFB, OH 45433.		15a 15a. DECLASSIFICATION DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Digital Control Systems Frequency Response Computational Delays Servo Analy- Sampled Data Switch Decomposition Tustin Transform sis z-Transforms Closed-Loop Systems Simulation Error Closed-Loop w-Domain Linear Systems Analysis Analysis Multi-Rate Sampling Root Locus		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The report is organized in three volumes. Volume I contains the theoretical developments as well as illustrative examples and case studies. Volume II describes two algorithms useful in the analysis of multi-rate systems, the DISCRET and TXCONV computer programs. Volume III contains the listings of all programs and subroutines which comprise DISCRET and TXCONV.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

340425

Handwritten signature/initials

FOREWORD

The research described in this report was performed by Systems Technology, Inc., Hawthorne, California, under Air Force Contract F33615-79-C-3601. The Task Number N3, Mathematics of Flight Control, was under Project Number 2304, Mathematics. This work was directed by the Control Dynamics Branch, Flight Control Division, Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Paterson Air Force Base, Ohio. The work was administered by Captain Dennis G. J. Didaleusky.

Richard F. Whitbeck was the Systems Technology, Inc., Project Engineer under the direction of Duane McRuer.

The authors wish to express their appreciation to the Systems Technology publication staff for their efforts in preparing this three-volume report.

The authors also wish to express their thanks to Ms. Susan Riedel at Systems Technology, Inc., and to Captain Stanley Larimer and Dr. Robert Schwanz at the Flight Dynamics Laboratory for their appreciable efforts in reviewing the technical report.

This report is organized in three volumes. Volume I contains the theoretical developments as well as illustrative examples and case studies. Volume II describes two algorithms useful in the analysis of multi-rate systems, the DISCRET and TXCONV computer programs. Volume III contains the FORTRAN listings for these computer programs.

This report covers work performed from January 1979 through May 1980. The report was submitted by the authors in August 1980.

TABLE OF CONTENTS

	<u>Page</u>
Source Listing — DISCRET Computer Program	1
Source Listing — TXCONV Computer Program	34

Accession For	
NTIS GPO&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A	

RE: AFWAL-TR-80-3101, Volume III, Distribution Statement-
 Approved for Public Release per Ms. Martha Kline, AFFDL/STINFO

```

000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

```

OVERLAY(DISCRET,0,0)
PROGRAM DISCRET(INPUT OUTPUT TAPES-INPUT TAPE7-OUTPUT)
COMMON/TOTL12/CLAPOLY(S1),CLDPOLY(S1),CLZERO(S0,2),
+CLPOLE(S0,2),NCLZ,NCLP,CLF,CLNK,CLDK
CALL OVERLAY(TNDISCRET,22,0)
END

xxxxx DESCRIPTION OF DISCRET PROGRAM xxxxx

THE DISCRET PROGRAM CONVERTS A GENERAL ANALOG OR CONTINUOUS
TRANSFER FUNCTION IN THE S-PLANE, G(S), INTO A VALID
DISCRETE DOMAIN - THE Z, U, OR U' PLANE. DISCRETE CAN
CALCULATE THE STANDARD DELAYED, OR ADVANCED DISCRETE
TRANSFORM. DATA HOLDS INCLUDING THE ZERO ORDER, FIRST
ORDER, SECOND ORDER, AND SLEWER CAN BE INSERTED INTO
THE TRANSFORMATION.

THE PROGRAM TAKES THE PARTIAL FRACTION EXPANSION OF
THE G(S) TRANSFER FUNCTION, INCLUDING THOSE ZEROS AND
POLES INTRODUCED BY THE PARTICULAR DATA HOLD SELECTED
(EG. THE ZERO ORDER HOLD INTRODUCES A POLE AT S=0 AND
THE SLEWER TWO POLES AT S=0). THE U PLANE TRANSFORM OF
THE INDIVIDUAL PARTIAL FRACTION EXPANSION TERMS ARE
OBTAINED AND RECORDED TO GIVE THE NUMERATOR AND
DENOMINATOR POLYNOMIALS IN THE DISCRETE U DOMAIN. THE
ROOTS ARE COMPUTED AND PRINTED ALONG WITH THE
NUMERATOR AND DENOMINATOR POLYNOMIALS. DEPENDING ON THE
OPTION SELECTED, THE CORRESPONDING Z AND U' TRANSFORMS
CAN BE CALCULATED USING THE BILINEAR TRANSFORMATION

U = Z-1/2+1

AND THE RELATIONSHIP BETWEEN THE U AND U' PLANES
EXPRESSED AS

U = (T/2)ZU' , T= SAMPLING TIME (SEC)

THE POSITIVE TIME ADVANCE AND NEGATIVE TIME DELAY
ASSOCIATED WITH THE ADVANCED AND DELAYED DISCRETE


```

CINT *****
C INPUT READ SECTION FOR BATCH MODE OPERATION
C
      READ(5,1)NRTN,N,T,AG,AM
      READ(5,2070) CPLR
      2070 FORMAT(A3)
      NH=1
      IF(CPLR.EQ.3)NON) NH=0
      READ(5,2080) TUDORM
      2080 FORMAT(A2)
C
      IF(NRTN.EQ.0) GO TO 2081
      N2 = 2 * NRTN
      READ(5,3)(RM(I),I=1,N2)
      2081 N2=2*N
      READ(5,3)(RD(I),I=1,N2)
C
      END INPUT READ SECTION FOR BATCH MODE
C
      ADD EQUAL ZERO AND POLE IF NRTN=0.
      PARTR CANCELS THIS ZERO/POLE
      COMBINATION.
      IF(NRTN.GT.0) GO TO 2082
      NRTN=1
      N2=2*NRTN
      RM(1)=1.123456789
      RM(2)=0.0
      DO 2083 I=1,N2
      I1=N2-I+1
      2083 RD(I1+2)=RD(I1)
      RD(1)=1.123456789
      RD(2)=0.0
      N=H+1
      N2=2*N
      2082 CONTINUE
C
      END EQUAL ZERO AND POLE SECTION
C
      INSERT COMPLEX CONJUGATE ROOT IF
      NOT INPUTTED BY USER.
      J=2
      19 IF (RM(J)) 20,21,20
      20 RM(J+1) = RM(J-1)
      RM(J+2) = -RM(J)
      J = J+4
      GO TO 28
      21 J=J+2
      22 IF(J=N2) 19,19,30
      30 CONTINUE

```

```

000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275

```

```

000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330

J=2
24 IF(RD(J)) 22,23,22
22 RD(J+1) = RD(J-1)
20 RD(J+2) = -RD(J)
J = J+4
GO TO 29
23 J=J+2
29 IF(J-M2) 24,24,31
31 CONTINUE
END COMPLEX CONJUGATE ROOT SECTION
IF ZOH SELECTED, ADD ADDITIONAL POLE
TO THE INPUT G(S) TRANSFER FUNCTION.
IF (NM -1 ) 500, 4, 500
500 CONTINUE
NRTD=M
GO TO 5
NRTD=M+1
L=M+M+1
RD(L)=0.0
RD(L+1)=0.0
CONTINUE
END ZOH SECTION
INSERT ADDITIONAL ZEROS AND POLES IN THE
S-DOMAIN FOR THE REMAINING DATA HOLDS
IF(CPLR .EQ. 3)M1ST) GO TO 5000
IF(CPLR .EQ. 3)M5LE) GO TO 5100
IF(CPLR .EQ. 3)M2ND) GO TO 5200
GO TO 5300
5000 NRTD=M+2
L=M+M+3
RD(L)=0.0
RD(L+1)=0.0
L1=NRTN+NRTN+1
NRTN=NRTN+1
RN(L1)= -1.0/T
RN(L1+1)=0.0
GO TO 5300
5100 NRTD=M+2
L=M+M+3
RD(L)=0.0
RD(L+1)=0.0
GO TO 5300
5200 NRTD=M+3
L=M+M+3
RD(L)=0.0
RD(L+1)=0.0
RD(L+2)=0.0

```

```

000331 RD(L+3)=-0.0
000332 L11=NRTH+NRTH+1
000333 NRTH=NRTH+2
000334 RN(L11)=-0.75/T
000335 RN(L11+1)=- (2.645751311)/(4.08T)
000336 RN(L11+2)=-0.75/T
000337 RN(L11+3)=- (2.645751311)/(4.08T)
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365
000366
000367
000368
000369
000370
000371
000372
000373
000374
000375
000376
000377
000378
000379
000380
000381
000382
000383
000384
000385

RD(L+3)=-0.0
L11=NRTH+NRTH+1
NRTH=NRTH+2
RN(L11)=-0.75/T
RN(L11+1)=- (2.645751311)/(4.08T)
RN(L11+2)=-0.75/T
RN(L11+3)=- (2.645751311)/(4.08T)
5300 CONTINUE

END ADDITIONAL ZEROS AND POLES SECTION

OUTPUT WRITE SECTION FOR BASIC DEFINING
PARAMETERS.
WRITE(7,3052)
3052 FORMAT(1H1,////)
PRINT*,TSAMP (SEC)=-T, DELTA (SEC)=-,AM
WRITE(7,3050) CPLE TRANSFORM
3050 FORMAT(1H1,THOLD OPTION = 2,A3,3X,STRANSFORM OPTION = 2,A2)
WRITE(7,3051) A0
3051 FORMAT(1H1,IGAIN = 2,D17.10)
PRINT*,.
PRINT*,.

END OUTPUT WRITE SECTION

CALL OVERLAY(7NDISCRET,22,1)
CALL OVERLAY(7NDISCRET,22,2)

CALL PARTFR
CALL UPLN

END

CHECK PARTFR
OVERLAY(26,1)
PROGRAM PARTFR

```

```

SUBROUTINE PARTER
C
C
C
CONTINUE/ADJCL/AN,ND,NRT(N,NRTD,MN,UR,U1,2R,B1,T,MH,AM,AG,
+1X Urd,CLR
COPRAN/10112/CLPOLY(51),CLDPOLY(51),CLZERO(50,2),
+CLPOLY(50,2),NCLZ,MCLP,CLK,CLWK,CLDK
C
C
DOUBLE UCOO(2),RN(102),RD(102)
DIMENSION NEGCT(51),RUL(51),MH(51)
DOUBLE D1,D2,ONE,ZFN,SCALEF,D,ZFD,COEFF
DOUBLE UN(2),UP(2),RF(6),A(262),RN(612),PD(306),P2(102)
DOUBLE UK(51),U1(51),R(51),B1(51),T,MH,AG
5 FORMAT(31X R3,D16.8,X INDI6.8))
7 FORMAT(31X R3,D16.8,X INDI6.8))
20 FORMAT(///180PARTIAL FRACTIONS //16X,17M U M E R A T O R,31X,
1 21H D E N O M I N A T O R,10X,SAREAL PART,11X,
3 51X,EXONENT )
21 FUKAT (1P,D2,8,D0,8,13X,D17.8,D20.8,11X,12 )
24 FORMAT (1P)
COEFF = AG * 1.D-4
NMN=NRT(NRTN
NRD=NRTD+NRD
ONE=1.
WRITE(7,24)
UNITE(7,6)
6 FORMAT(3 INPUT ROOTS--NUMERATORS//)
WRITE(7,5)(RN(I),I=1,NMN)
UNITE(7,1)
1 FORMAT(///3 INPUT ROOTS--DENOMINATORS//)
WRITE(7,5)(RD(I),I=1,NRD)
L=1
300 IU=500 I=L,NMN,2
11-I
DO 300 J=1,NRD,2
JJ-J
U1 =JMS(RN(I),0001)
D2 =JMS(RN(I+1),0001)
IF (RN(I))510,511,510
IF (RD(J)) 512, 511,512
510 U1 =.00001
512 IF (RN(I+1))513,514,513
514 IF (RD(J+1))515,514,515
514 U2=.00001
515 IF (DMS(RN(I)-RD(J))-01)516,516,300
516 IF (DMS(RN(I+1)-RD(J+1))-02)517,517,300
517 CONTINUE
NEXT=NRD-2
NRD=NRD-2
DO 370 K=11,NMN

```

5

```

KKK=1
NO 500 I=1,NRD,2
N=I/2+1
IFAC=0
IF(NEGLCT(N))470,527,470
527 CONTINUE
IF(RD(I+1)) 528,460,528
528 CONTINUE
NEGLCT(N+1)=1
460 NEGLCT(N)=1
CALL FORM(RD,NRD,MULT,NEGLCT,P2,NP2)
470 CALL CONVAT(A,NP,NM,NM)
CALL CONVAT(P2,NP2,PB,NPD)
N=MULT(N)
IF(RD(I+1)) 529,475,529
529 CONTINUE
NRF=6
RF(1)=1.
RF(2)=0.
RF(3)=1.
RF(4)=RD(I)
RF(5)=RD(I+1)
RF(6)=0.
DO 474 J=1,N
CALL MULTIP(PD,NPD,RF,NRF,A(103),L,3)
CALL MLTPL(A(103),L,ONE,PB,NPD)
474 CONTINUE
475 DO 490 J=1,N
CROOT(1)=RD(I)
CROOT(2)=RD(I+1)
IF(J-1)530,480,530
530 CONTINUE
IFAC=IFAC+1
CALL DIRIU(PM,NPM,PB,NPD)
CALL EVALUB(PM,NPM,CROOT,UN,ZFN)
UN(1)=UN(1)/FACT(IFAC)
UN(2)=UN(2)/FACT(IFAC)
CALL EVALUB(PD,NPD,CROOT,UD,ZFD)
SCALEF=ZFN/ZFD
D=UD(1)
IF(DABS(UD(1))-DABS(UN(1)))484,482,486
482 IF(UD(1)) 486,531,486
531 CONTINUE
D=UD(2)
IF(DABS(UD(2))-DABS(UN(2)))532,533,533
532 D=UN(2)
533 CONTINUE
GO TO 486
484 D=UN(1)
486 UD(1)=UD(1)/D
UN(1)=(UN(1)/D)*SCALEF
UD(2)=UD(2)/D
UN(2)=(UN(2)/D)*SCALEF
D=UD(1)*UD(2)+UD(2)*UD(2)
UN(KKK)=(UN(1)*UD(1)+UN(2)*UD(2))/D SCOFF

```

```

400      VI(KKK)=(VD(1)XUN(2)-UN(1)XUD(2))/DBCOEFF
      K=K+2
      NEGLCT(N)=0
      CONTINUE
      WRITE(7,24)
      WRITE(7,26)
      J=1
      IJJ=1
      DO 502 L=1,NBD,2
      N=L/2+1
      M=MULT(N)
      DO 501 I=1,M
      IX=M-I+1
      WRITE(7,21) UR(IJJ),VI(IJJ),RD(L),RD(L+1),IX
      II=(J+1)/2
      IR(IJJ)=RD(L)
      RI(IJJ)=RD(L+1)
      RI(II)=IX
      IJJ=JJJ+1
      JJJ=JJJ+1
      J=J+2
      CONTINUE
      WRITE(7,24)
      MTD=11
      RETURN
      END
501
502
C-----
CHECK MULTIP
C-----
SUBROUTINE MULTIP(C1,NT1,C2,NT2,C3,NT3,N)
      DOUBLE C1(1),C2(1),C3(1)
      K=1-N
      DO 110 I=1,NT1,N
      DO 100 J=1,NT2,N
      K=K+N
      IF (N-3) 529,90,529
      CONTINUE
      C3(K)=C1(I)XC2(J)
      C3(K+1)=C1(I+1)XC2(J+1)
      GO TO 100
90      C3(K) =C1(I)XC2(J)-C1(I+1)XC2(J+1)
      C3(K+1)=C1(I)XC2(J+1)+C1(I+1)XC2(J)
      C3(K+2)=C1(I+2)XC2(J+2)
      CONTINUE
      NT3=K+N-1
      IF (I-2) 110,530,530
      CONTINUE
      CALL SIMPL(C3,NT3,N)
      K=NT3-N+1
      CONTINUE
      RETURN
      END
C-----
CHECK MULTIP
SUBROUTINE MULTPL(C,N,D,E,M)

```

```

000551
000552
000553
000554
000555
000556
000557
000558
000559
000560
000561
000562
000563
000564
000565
000566
000567
000568
000569
000570
000571
000572
000573
000574
000575
000576
000577
000578
000579
000580
000581
000582
000583
000584
000585
000586
000587
000588
000589
000590
000591
000592
000593
000594
000595
000596
000597
000598
000599
000600
000601
000602
000603
000604
000605
000606

```



```

70 WRITE(7,1)
80 CALL DUMP
90 RETURN
END
C-----
CHECK FORM
SUBROUTINE FORM(R,NR,M,JUMP,P,MP)
C-----
DOUBLE P(1)
DOUBLE R(1),TS(180)
DIMENSION N(1),JUMP(1)
KONJ=0
K=1
DO 3 I=1,NR,2
IF (KONJ)2,531,2
531 CONTINUE
II=(I+1)/2
IF (JUMP(II))2,500,2
500 CONTINUE
N=N(II)
IF (DABS(R(I))-R(I+2))-0.0001)532,532,534
532 IF (DABS(R(I+1)-R(I+3))-0.0001)533,533,534
533 KONJ=1
534 CONTINUE
IF (I+2)-NR) 535,535,536
536 KONJ=0
535 CONTINUE
DO 1 J=1,N
TS(K)=R(I)
TS(K+1)=R(I+1)
K=K+2
IF (KONJ) 537,1,537
537 CONTINUE
TS(K)=R(I+2)
TS(K+1)=R(I+3)
K=K+2
1 CONTINUE
GO TO 3
2 KONJ=0
3 CONTINUE
TS(K)=1.D-8
K=K/2
CALL GETPOL(TS,K,P,MP)
RETURN
END
C-----
CHECK CONVRT
SUBROUTINE CONVRT (P,MP,Q,MQ)
C-----
DOUBLE P(1),Q(1)
MQ=3*MP/2
K=1
DO 1 I=1,MP,2
Q(K)=P(I)
Q(K+1)=0

```

```

000716      Q(K+2)-P(I+1)
000717      K=K+3
000718      RETURN
000719      END
000720
000721      CHECK DIRIU
000722      SUBROUTINE DIRIU(PH,NPH,PD,NPD)
000723      C-----
000724      DOUBLE      PH(1),PD(1),A(306),B(306),C(306),ONE
000725      ONE = 1.0
000726      DO 1 I=1,NPH,3
000727      A(I) =PH(I+2)*PH(I)
000728      A(I+1)=PH(I+2)*PH(I+1)
000729      A(I+2)=PH(I+2)-1.
000730      CONTINUE
000731      N=NPH
000732      CALL MULTIP(PD,NPD,A,N,B,N,3)
000733      DO 2 I=1,NPD,3
000734      A(I) =-PD(I+2)*PD(I)
000735      A(I+1)=-PD(I+2)*PD(I+1)
000736      A(I+2)=PD(I+2)-1.
000737      CONTINUE
000738      N=NPD
000739      CALL MULTIP(PH,NPH,A,N,C,K,3)
000740      CALL ADD(B,M,C,K,PH,NPH,3)
000741      CALL MULTIP(PD,NPD,PD,NPD,A,N,3)
000742      CALL RLTP(L(A,N,ONE,PD,NPD)
000743      RETURN
000744      END
000745
000746      CHECK EVALUS
000747      SUBROUTINE EVALUS (P,NP,R,U,ZF)
000748      C-----
000749      DOUBLE      R(1),EXP,RP,CP,ZF
000750      DOUBLE      P(1),U(2)
000751      CALL ORDER3(P,NP,3)
000752      ZF=1.
000753      P(NP+1)=0
000754      P(NP+2)=0
000755      P(NP+3)=0
000756      EXP=P(3)
000757      U(1)=P(1)
000758      U(2)=P(2)
000759      IF (EXP) 538,4,538
000760      538 CONTINUE
000761      DO 3 I=1,NP,3
000762      IF (P(I+2)) 539,4, 539
000763      539 CONTINUE
000764      RP=U(1)*R(1)-U(2)*R(2)
000765      CP=U(1)*R(2)+U(2)*R(1)
000766      IF (ABS(RP)-1.000011,540,540
000767      540 CONTINUE
000768      ZF = ZF * 1.E10
000769      RP = RP/1.E10
000770      CP = CP / 1.E10
000771

```

```

11 CONTINUE
EXP = EXP - 1.0
IF (EXP - P(I+5)) 2,2,541
541 CONTINUE
U(1) = RP
U(2) = CP
GO TO 1
2 U(1) = RP + P(I+3) / ZF
I11 = I + 4
U(2) = CP + P(I11) / ZF
3 CONTINUE
4 RETURN
END
C-----
CHECK FACT
DOUBLE FUNCTION FACT(M)
C-----
FACT = 1
IF (M) 542,2,542
542 CONTINUE
P = M
1 IF (P - 1.) 543,2,543
543 CONTINUE
FACT = FACT * P
P = P - 1
GO TO 1
2 RETURN
END
C-----
CHECK ADD
SUBROUTINE ADD(C1,NT1,C2,NT2,C3,NT3,M)
C-----
DOUBLE C1(1),C2(1),C3(1)
DO 100 I = 1,NT1
C3(I) = C1(I)
DO 110 J = 1,NT2
J = J + NT1
110 C3(J) = C2(J)
NT3 = NT1 + NT2
CALL SIMPLE(C3,NT3,M)
RETURN
END
C-----
CHECK SIMPLE
SUBROUTINE SIMPLE(C,NT,M)
C-----
DOUBLE C(2)
M = NT - M
DO 110 I = 1,M,M
JJ = I + M
DO 110 J = JJ,NT,M
K = M - I - 1
L = M - J - 1
IF (C(I) - C(L)) 110,100,110
100 C(I) = C(I) * C(J)

```

000771
000772
000773
000774
000775
000776
000777
000778
000779
000780
000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825

```

C(J)=0
IF(N-2)534,110,534
534 CONTINUE
C(I+1)=C(I+1)+C(J+1)
C(J+1)=0
110 CONTINUE
111 I=1
115 DO 140 I=1,NT,M
IF(C(I))140,535,140
535 CONTINUE
536 IF(N-2)536,120,536
536 CONTINUE
IF(C(I+1))140,537,140
537 CONTINUE
120 JJ=I+M
DO 130 J=JJ,NT
K=J-M
130 C(K)=C(J)
NT=NT-M
111
IF(I=M-1-NT)115,115,145
140 CONTINUE
145 IF(NT)155,150,155
150 NT=M
C(1)=0
C(2)=0
C(3)=0
155 RETURN
END

C-----
CHECK ORDER3
SUBROUTINE ORDER3(P,N,K)
C-----
DOUBLE T3489,T3481
DOUBLE P(1),TEMP
N=N-K
IF(N)/5,5,1
1 REVERS=1.
DO 4 I=K,M,K
J=I-K
IF(P(1)-P(J))2,4,4
2 REVERS=0
DO 3 L=1,K
LI=I-L+1
LI=I-L+1
TEMP=P(LJ)
P(LJ)=P(LI)
P(LI)=TEMP
3 CONTINUE
4 CONTINUE
IF(REVERS)/5,1,5
5 RETURN
END

```

6

```

00026
00027
00028
00029
00030
00031
00032
00033
00034
00035
00036
00037
00038
00039
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049
00050
00051
00052
00053
00054
00055
00056
00057
00058
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077
00078
00079
00080

```


22

```

001211 RR(J)-RR(J+1)
001212 RI(J)-RI(J+1)
001213 NN=NN-1
001214 C THIS SUBSECTION WANTS FOR (U=0.0) DENOMINATOR ROOTS AND
001215 C CANCELS THEM INSTEAD OF ADDING (U=0.0) TO NUMERATOR ROOTS.
001216 C
001217 2400 CONTINUE
001218 A=0.0
001219 DO 2410 I=1,L1
001220 ISH1=I
001221 A1=DR(I)
001222 A2=DI(I)
001223 IF (ABS(A1-A).LE.1.E-12.AND.ABS(A2).LE.1.E-12.AND.KK(I).EQ.1)
001224 * GO TO 2420
001225 2410 CONTINUE
001226 NN=NN+1
001227 RR(NN)=0.0
001228 RI(NN)=0.0
001229 GO TO 2500
001230 2420 I=ISH1
001231 NN(I)=NN(I)-1
001232 IF (NN(I).EQ.0) KK(I)=0
001233 2500 CONTINUE
001234 IF (LOOP.GT.1) GO TO 2600
001235 IF (CPLR.EQ.3.AND) GO TO 2900
001236 2600 CONTINUE
001237 C
001238 I=0
001239 491 I=I+1
001240 IF (I.GT.L1) GO TO 496
001241 IF (KK(I).EQ.1) GO TO 495
001242 L1=L1-1
001243 DO 483 J=1,L1
001244 DR(J)=DR(J+1)
001245 DI(J)=DI(J+1)
001246 KK(J)=KK(J+1)
001247 NN(J)=NN(J+1)
001248 483 I=I-1
001249 495 GO TO 491
001250 496 CONTINUE
001251 RR(I)=RR(I)
001252 RI(I)=RI(I)
001253 A1(I)=A1(I)
001254 A2(I)=A2(I)
001255 IF (2)=0.
001256 IF (2)=0.
001257 IF (NN.EQ.1) GO TO 615
001258 DO 610 I=2,NN
001259 DO 605 J=1,IE
001260 IIE=IE-J+1
001261 X=AR(IIE+1)
001262 Y=AR(IIE)-AR(IIE+1)+RI(I)
001263 X=AR(IIE+1)+AR(IIE)-RI(I)
001264 Y=AR(IIE)-AR(IIE+1)+RI(I)
001265 Y=AR(IIE)-AR(IIE+1)+RI(I)

```

```

001266 X=AR(1)
001267 AR(1)=-RR(1)ZAR(1)+RI(1)SAI(1)
001268 AI(1)=-RR(1)RX-RR(1)SAI(1)
001269 610 CONTINUE
001270
001271 NUMERATOR POLY GAIN FACTOR FOR ZERO ORDER HOLD (ZOH)
001272 IF(IFLG.EQ.2)XG=2.XYG
001273 IF(IFLG.EQ.2)YG=2.YYG
001274
001275 NUMERATOR POLY GAIN FACTOR FOR FIRST ORDER HOLD (1ST)
001276 IF(CPLR.EQ.3H1ST)XG=2.0XYG
001277 IF(CPLR.EQ.3H1ST)YG=2.0XYG
001278
001279 NUMERATOR POLY GAIN FACTOR FOR SECOND ORDER HOLD (2ND)
001280 IF(CPLR.EQ.3H2ND)XG=4.0XYG
001281 IF(CPLR.EQ.3H2ND)YG=4.0XYG
001282
001283 NUMERATOR GAIN FACTOR FOR SLEWER DATA HOLD (SLE)
001284 IF(CPLR.EQ.3HSLE)XG=(2.0/T)XYG
001285 IF(CPLR.EQ.3HSLE)YG=(2.0/T)YG
001286
001287 DO 617 I=1,IE
001288 CALL MULT(AR(I),AI(I),XG,YG,X,Y)
001289 AR(I)=X
001290 AI(I)=Y
001291
001292 DO 617 I=1,IE
001293 CALL MULT(AR(I),AI(I),XG,YG,X,Y)
001294 AR(I)=X
001295 AI(I)=Y
001296
001297 TRANSFER LOGIC FOR Z OR U TRANSFORM
001298 IF(TXFORM.EQ.2HZ)GO TO 3000
001299 IF(TXFORM.EQ.2HU)GO TO 5000
001300
001301
001302
001303
001304
001305
001306
001307
001308
001309 THE U' TRANSFORM IS OBTAINED DIRECTLY FROM THE
001310 U TRANSFORM BY USING THE FOLLOWING:
001311
001312 U'-(ZEROS & POLES) = U-(ZEROS & POLES)x(2/T)
001313 U'x(2 (T/2)x(2) x(a2) + U'x(1 (T/2) x (a1) + a0 -
001314 Ux(2) x (a2) + Ux(1) x (a1) + a0
001315
001316
001317
001318
001319
001320
001321
001322
001323
001324
001325
001326
001327
001328
001329
001330
001331
001332
001333
001334
001335
001336
001337
001338
001339
001340
001341
001342
001343
001344
001345
001346
001347
001348
001349
001350
001351
001352
001353
001354
001355
001356
001357
001358
001359
001360
001361
001362
001363
001364
001365
001366
001367
001368
001369
001370
001371
001372
001373
001374
001375
001376
001377
001378
001379
001380
001381
001382
001383
001384
001385
001386
001387
001388
001389
001390
001391
001392
001393
001394
001395
001396
001397
001398
001399
001400
001401
001402
001403
001404
001405
001406
001407
001408
001409
001410
001411
001412
001413
001414
001415
001416
001417
001418
001419
001420
001421
001422
001423
001424
001425
001426
001427
001428
001429
001430
001431
001432
001433
001434
001435
001436
001437
001438
001439
001440
001441
001442
001443
001444
001445
001446
001447
001448
001449
001450
001451
001452
001453
001454
001455
001456
001457
001458
001459
001460
001461
001462
001463
001464
001465
001466
001467
001468
001469
001470
001471
001472
001473
001474
001475
001476
001477
001478
001479
001480
001481
001482
001483
001484
001485
001486
001487
001488
001489
001490
001491
001492
001493
001494
001495
001496
001497
001498
001499
001500
001501
001502
001503
001504
001505
001506
001507
001508
001509
001510
001511
001512
001513
001514
001515
001516
001517
001518
001519
001520
001521
001522
001523
001524
001525
001526
001527
001528
001529
001530
001531
001532
001533
001534
001535
001536
001537
001538
001539
001540
001541
001542
001543
001544
001545
001546
001547
001548
001549
001550
001551
001552
001553
001554
001555
001556
001557
001558
001559
001560
001561
001562
001563
001564
001565
001566
001567
001568
001569
001570
001571
001572
001573
001574
001575
001576
001577
001578
001579
001580
001581
001582
001583
001584
001585
001586
001587
001588
001589
001590
001591
001592
001593
001594
001595
001596
001597
001598
001599
001600
001601
001602
001603
001604
001605
001606
001607
001608
001609
001610
001611
001612
001613
001614
001615
001616
001617
001618
001619
001620
001621
001622
001623
001624
001625
001626
001627
001628
001629
001630
001631
001632
001633
001634
001635
001636
001637
001638
001639
001640
001641
001642
001643
001644
001645
001646
001647
001648
001649
001650
001651
001652
001653
001654
001655
001656
001657
001658
001659
001660
001661
001662
001663
001664
001665
001666
001667
001668
001669
001670
001671
001672
001673
001674
001675
001676
001677
001678
001679
001680
001681
001682
001683
001684
001685
001686
001687
001688
001689
001690
001691
001692
001693
001694
001695
001696
001697
001698
001699
001700
001701
001702
001703
001704
001705
001706
001707
001708
001709
001710
001711
001712
001713
001714
001715
001716
001717
001718
001719
001720
001721
001722
001723
001724
001725
001726
001727
001728
001729
001730
001731
001732
001733
001734
001735
001736
001737
001738
001739
001740
001741
001742
001743
001744
001745
001746
001747
001748
001749
001750
001751
001752
001753
001754
001755
001756
001757
001758
001759
001760
001761
001762
001763
001764
001765
001766
001767
001768
001769
001770
001771
001772
001773
001774
001775
001776
001777
001778
001779
001780
```

```

5300 RI(I)=(2/T)RI(I)
C
5000 CONTINUE
C
WRITE(7,619)
619 FORMAT(//,10X,COEFFICIENTS OF NUMERATOR IN UZ/)
C
CLAK=AR(IE)
X=AR(IE)
DO 618 I=1,IE
IEK=IE-I+1
CLMPOLY(I)=AR(IEK)/X
IJ=I-1
WRITE(7,621)CLMPOLY(I), IEK-I
621 FORMAT(1H,10X,I(1,G18.10,5H)UZ,13)
C
620 WRITE(7,620)AR(I),AI(I),IJ
620 FORMAT(5X,I(1,G18.30,1,1D38.30,5H)UZ,13)
618 CONTINUE
C
NCLZ=NN
DO 3100 I=1,NN
CLZERO(I,1)=RR(I)
3100 CLZERO(I,2)=RI(I)
C
C
505 WRITE(7,505)
505 FORMAT(1H1)
506 WRITE(7,506)
506 FORMAT(//,10X,NUMERATOR ROOTS IN UZ/)
516 WRITE(7,516)(CLZERO(I,1),CLZERO(I,2),I=1,NN)
516 FORMAT(5X,I(1,G18.10,1,1D38.30,5H)UZ,13)
C
515 WRITE(7,515)(AR(I),RI(I),I=1,NN)
515 FORMAT(5X,I(1,G18.30,1,1D38.30,5H)UZ,13)
C
C
TRANSFER STATEMENT FOR Z TRANSFORM
C
3000 CONTINUE
C
DO 650 I=1,51
BR(I)=0.
650 RI(I)=0.
BR(I)=DR(I)
RI(I)=DI(I)
BR(2)=1.
RI(2)=0.
IIE=2
DO 660 I=1,11
IF(I.EQ.1.AND.NN(I).EQ.1)GO TO 660
K=NN(I)
IF(I.EQ.1)K=K-1
DO 640 J=1,K
640 IJ=I+J,IIE
IJ=IIE-IJ+1
XX=BR(IIJ+1)
BR(IIJ+1)=BR(IIJ)+DI(I)XX+AR(I)RI(IIJ+1)
630 RI(IIJ+1)=RI(IIJ)+DI(I)XX+AR(I)RI(IIJ+1)

```

```

001321
001322
001323
001324
001325
001326
001327
001328
001329
001330
001331
001332
001333
001334
001335
001336
001337
001338
001339
001340
001341
001342
001343
001344
001345
001346
001347
001348
001349
001350
001351
001352
001353
001354
001355
001356
001357
001358
001359
001360
001361
001362
001363
001364
001365
001366
001367
001368
001369
001370
001371
001372
001373
001374
001375

```



```

669 DO 668 I=1, IIE
668 XSUM=XSUM+BR(I)
      B=XSUM
      IF (ABS(B).GT.1.E-7) GO TO 662
      XSUM=0.
      DO 663 K=1, IIE
        J=K-1
        I=IIE-K
        IF (J.EQ.0) XSUM=XSUM+BR(K)*I
        IF (J.EQ.0) GO TO 663
        IF (I.EQ.0) XSUM=XSUM-BR(K)*J
        IF (I.EQ.0) GO TO 663
        XSUM=XSUM+BR(K)*I*(-J+1)
      XSUM=XSUM+BR(K)*I*(-J+1)
663 CONTINUE
662 IF (XSUM.EQ.0) WRITE(7,661)
661 XG=XG/XSUM
661 FORMAT(' ERROR IN GAIN CALCULATION')
      C
      WRITE(7,595)
      DO 700 I=1, IIE
        CR(1,I)=AR(I)
        CI(1,I)=AI(I)
      700 DO 705 I=1, IIE
        CR(2,I)=BR(I)
        CI(2,I)=BI(I)
      705 NK=NN
      DO 710 I=1, NN
      709 CONTINUE
      X=(1.-RR(1))*X2+RI(1)*X2
      IF (DABS(X).GT.1.D-12) GO TO 707
      NK=NK-1
      IF (I.GT.NK) GO TO 710
      DO 708 J2=1, NK
      708 RI(J2)=AR(J2+1)
      GO TO 709
      707 RI(1)=(1.-RR(1))*X2-RI(1)*X2)/X
      RI(1)=2.*RI(1)/X
      710 CONTINUE
      805 NN=NK
      LK=L1
      DO 720 I=1, L1
      719 CONTINUE
      X=(1.-DR(1))*X2+DI(1)*X2
      IF (DABS(X).GT.1.D-12) GO TO 718
      LK=LK-1
      IF (I.GT.LK) GO TO 720
      DO 717 J2=1, LK
      717 DI(J2)=AR(J2+1)
      DR(J2)=AK(J2+1)
      GO TO 719
      718 DI(1)=(1.-DR(1))*X2-DI(1)*X2)/X
      DI(1)=2.*DI(1)/X

```

```

001486
001487
001488
001489
001490
001491
001492
001493
001494
001495
001496
001497
001498
001499
001500
001501
001502
001503
001504
001505
001506
001507
001508
001509
001510
001511
001512
001513
001514
001515
001516
001517
001518
001519
001520
001521
001522
001523
001524
001525
001526
001527
001528
001529
001530
001531
001532
001533
001534
001535
001536
001537
001538
001539
001540

```



```

J=NM1-I+1
RR(I)=FR(J,1)
853 RI(I)=FI(J,1)
WRITE(7,857)
857 FORMAT(///,10X,SCOEFFICIENTS OF NUMERATOR IN ZL/)
DO 566 I=1,NM1
RR(I)=RR(I)XQ
566 RI(I)=RI(I)XQ
C
CLNK=RR(NM1)
CLK=RR(NM1)
X=RR(NM1)
C
DO 854 I=1,NM1
IEK=NM1-I+1
CLNPOLY(I)=RR(IEK)/X
C
NI=I-1
WRITE(7,856)CLNPOLY(I),IEK-1
856 FORMAT(1H 10X,1(18.10,1)X,5H Z1X,15)
WRITE(7,157) CLNK,CLK
C
157 FORMAT(1H 10X,1(18.10,1)X,5H Z1X,15)
WRITE(7,855)RR(I),RI(I),NI
855 FORMAT(1(18.10,1)X,1(18.10,1)X,4H Z1X,15)
854 CONTINUE
K2=L1
DO 581 I=1,K2
IF(NM1-I.EQ.1)GO TO 581
J2=NM1(I)-1
DO 580 J=1,J2
L1=L1+1
DR(L1)=DR(I)
580 DI(L1)=DI(I)
581 CONTINUE
CALL POLYCO(FR,FI,DR,DI,L1)
LP1=L1+1
DO 843 I=1,LP1
J=LP1-I+1
DR(I)=FR(J,1)
843 DI(I)=FI(J,1)
WRITE(7,873)
873 FORMAT(///,10X,SCOEFFICIENTS OF DENOMINATOR IN ZL/)
C
NCLP=LP1-1
CLDK=1.0
C
DO 848 I=1,LP1
C
C
IEK=LP1-I+1
CLDPOLY(I)=DR(IEK)
C
I1=I-1

```

```

001596
001597
001598
001599
001600
001601
001602
001603
001604
001605
001606
001607
001608
001609
001610
001611
001612
001613
001614
001615
001616
001617
001618
001619
001620
001621
001622
001623
001624
001625
001626
001627
001628
001629
001630
001631
001632
001633
001634
001635
001636
001637
001638
001639
001640
001641
001642
001643
001644
001645
001646
001647
001648
001649
001650

```

```

001651 WRITE(7,856)CLDPOLY(I),IEK-1
001652 WRITE(7,856)DR(I),DI(I),I1
001653 CONTINUE
001654
001655 TRANSFER STATEMENT FOR U OR V' TRANSFORM
001656
001657
001658
001659 CONTINUE
001660
001661 WRITE(7,9977) CLK,CLKX,CLKX
001662 FORMAT(IH,/,/,/,3 CLK-3,G18.10,/,3 CLKX-3,G18.10)
001663 WRITE(7,5599) NCLP,NCLZ
001664 FORMAT(IH,ENCLP-3,15,5X,ENCLZ-3,15)
001665
001666
001667 END
001668
001669 CDECK ROOTS
001670
001671 SUBROUTINE ROOTS(A,B,MN,RR,RI)
001672
001673 DIMENSION A(1),B(1),RR(1),RI(1)
001674 DOUBLE PRECISION A,B,RR,RI,C,D,C1,C2,D1,D2,X,Y,F,G,F1,F2,G1,G2,FH,
001675 IFX,FH
001676 L=1
001677 N=MN
001678 FM=1.
001679 NPLUS=N+1
001680 DO 265 I=1,NPLUS
001681 A1=A(I)
001682 B1=B(I)
001683 IF(ABS(A1).GT..00000001.OR.ABS(B1).GT..00000001)GO TO 266
001684 A(J)=A(J+1)
001685 B(J)=B(J+1)
001686 J=N-1
001687 IF(N.EQ.0)RR(1)=0.
001688 IF(N.EQ.0)RI(1)=0.
001689 IF(N.EQ.0)RETURN
001690 CONTINUE
001691 CONTINUE
001692 MN=N
001693 DO 303 I=1,MN
001694 RR(I)=0.
001695 RI(I)=0.
001696 I12=M1-N+1
001697 IF(N.EQ.1)GO TO 103
001698 D1=DPMX1(DABS(A(N+1)),DABS(B(N+1)))
001699 IF(D1.EQ.0.)GO TO 111
001700 D2=DPMX1(DABS(A),DABS(B))
001701 FX=N
001702 FM=D2*D1. /FX)/D1*D2(1. /FX)
001703 DO 3 I=1,M
001704 A(I)=A(I)/(D1*D2*FX)
001705

```

```

B(I)=B(I)/(D12FTHK)
3 K-K-1
A(M)=A(M)/D1
B(M)=B(M)/D1
7 X=.9876532
Y=.9864312
LL=1
IL=1
8 C-A
D-B
C1-A
C2-B
DO 5 I=1,N
F=XG-YD+A(I+1)
G=XD+YC+B(I+1)
IF(I.EQ.N)GO TO 5
F1=XG1-YD2+
F2=XD2+YC1+G
C-F
D-G
C1-F1
C2-F2
D-F1X2+F2X2
FM=-(F1F1+G1G2)/D
FK=-(G1F1-F1F2)/D
12 X=X-FH
Y=Y-FK
IF((X22+Y22).EQ.(X+FH)X2+(Y+FK)Y2))GO TO 22
IF(X.EQ.0..OR.Y.EQ.0.)GO TO 22
A1-FH/X
A2-FK/Y
IF(ABS(A1).GT.1.E-4..OR.ABS(A2).GT.1.E-4)GO TO 21
GO TO(21,21).IL
211 LL=194
IL=2
21 LL=LL+1
IF(LL.GT.200)GO TO 22
GO TO 8
22 RI(L)=X/FH
RI(L)=Y/FH
DO 100 I=1,N
A(I+1)=X2A(I)-Y2B(I)+A(I+1)
B(I+1)=X2B(I)+Y2A(I)+B(I+1)
100 N=N+1
L=L+1
IF(N.GT.1)GO TO 7
103 D=AX2+222
RI(L)=(-A2A(2)-22B(2))/(D2FH)
RI(L)=(-A2B(2)+22A(2))/(D2FH)
DO 104 I=1,NH
X=RI(I)X2
Y=RI(I)Y2
D=X+Y
IF(D.EQ.X)RI(I)=0.
IF(D.EQ.Y)RI(I)=0.

```

```

001706
001707
001708
001709
001710
001711
001712
001713
001714
001715
001716
001717
001718
001719
001720
001721
001722
001723
001724
001725
001726
001727
001728
001729
001730
001731
001732
001733
001734
001735
001736
001737
001738
001739
001740
001741
001742
001743
001744
001745
001746
001747
001748
001749
001750
001751
001752
001753
001754
001755
001756
001757
001758
001759
001760

```

```

104 CONTINUE
RETURN
111 RR(L)=0.
RI(L)=0.
N=N+1
L=L+1
GO TO 112
END
001761
001762
001763
001764
001765
001766
001767
001768
001769
001770
001771
001772
001773
001774
001775
001776
001777
001778
001779
001780
001781
001782
001783
001784
001785
001786
001787
001788
001789
001790
001791
001792
001793
001794
001795
001796
001797
001798
001799
001800
001801
001802
001803
001804
001805
001806
001807
001808
001809
001810
001811
001812
001813
001814
001815

C-----
CDECK CDEXP
SUBROUTINE CDEXP(A,B,X,Y)
C-----
DOUBLE A,B,X,Y
X=DEXP(A)*DCOS(B)
Y=DEXP(A)*DSIN(B)
RETURN
END
C-----
CDECK MULT
SUBROUTINE MULT(A,B,C,D,X,Y)
C-----
DOUBLE A,B,X,Y,C,D
X=ARC-BXD
Y=BXC+AYD
RETURN
END
C-----
CDECK DIVI
SUBROUTINE DIVI(A,B,C,D,X,Y)
C-----
DOUBLE A,B,C,D,X,Y
X=(ARC-BXD)/(CXS2+DSX2)
Y=(BXC+AYD)/(CXS2+DSX2)
RETURN
END
C-----
CDECK POLYCO
SUBROUTINE POLYCO(A,B,RR,RI,N)
C-----
DIMENSION A(51,4),B(51,4),RR(1),RI(1)
DOUBLE A,B,RR,RI
NP1=NP
DO 1 I=2,NP1
  RI(1)=0.
  1 RI(1)=0.
  RI(1)=1.
  RI(1)=0.
  DO 3 I=1,N
    K=I+1
    DO 3 J=1,I
      RI(K,1)=A(K-1,1)*RR(I,1)+A(K,1)
      RI(K,1)=B(K-1,1)*RR(I,1)+B(K,1)
    3 K=K+1
  RETURN
END

```

```

000001 000001
000002 000002
000003 000003
000004 000004
000005 000005
000006 000006
000007 000007
000008 000008
000009 000009
000010 000010
000011 000011
000012 000012
000013 000013
000014 000014
000015 000015
000016 000016
000017 000017
000018 000018
000019 000019
000020 000020
000021 000021
000022 000022
000023 000023
000024 000024
000025 000025
000026 000026
000027 000027
000028 000028
000029 000029
000030 000030
000031 000031
000032 000032
000033 000033
000034 000034
000035 000035
000036 000036
000037 000037
000038 000038
000039 000039
000040 000040
000041 000041
000042 000042
000043 000043
000044 000044
000045 000045
000046 000046
000047 000047
000048 000048
000049 000049
000050 000050
000051 000051
000052 000052
000053 000053
000054 000054
000055 000055

*****
THE "TXCONU" COMPUTER PROGRAM CALCULATES A LOW-RATE
DISCRET TRANSFORM FROM A GIVEN HIGH-RATE DISCRETE TRANSFORM.
THE INPUT TO "TXCONU" IS A HIGH-RATE TRANSFER FUNCTION IN THE
Z-, U-, OR W'-PLANE AND THE OUTPUT A LOW-RATE TRANSFER
FUNCTION IN THE Z-, U-, OR W'-PLANE.

TXCONU IS WRITTEN IN FORTRAN FOR THE CONTROL DATA
CORPORATION CDC1 CYBER 175 SERIES COMPUTER. THE PROGRAM CAN
HANDLE POLE MULTIPLICITY UP TO THREE. THE PROGRAM IS
PRESENTLY SET-UP TO HANDLE SYSTEM ORDER UP TO 50TH.
THE SYSTEM ORDER IS VARIABLE AND CAN BE EASILY CHANGED. THE
REQUIRED CHANGES TO THE DIMENSIONING STATEMENTS ARE OUTLINED
BELOW.

THERE IS NO RESTRICTION ON THE NUMBER OF SETS OF MULTIPLE
POLES. DOUBLE PRECISION ARITHMETIC IS USED THROUGHOUT THE
PROGRAM.

THE FIVE AVAILABLE INPUT OPTIONS ARE OUTLINED BELOW:

"2" OPTION - INPUT HIGH-RATE ZEROS AND POLES IN Z-PLANE.
OUTPUT LOW-RATE ZEROS AND POLES IN Z-PLANE.
ALL CALCULATIONS IN THE W'-PLANE VIA THE
POLYNOMIAL TRANSFORMATION METHOD, I.E.,
HIGH-RATE Z-PLANE NUMERATOR AND DENOMINATOR
POLYNOMIALS ARE FIRST TRANSFORMED TO
W'-PLANE USING BILINEAR TRANSFORMATION.
FOLLOWING MATHEMATICAL CALCULATIONS, LOW-RATE
W'-PLANE TRANSFER FUNCTION IS TRANSFORMED
BACK TO THE Z-PLANE.

"U" OPTION - INPUT HIGH-RATE ZEROS AND POLES IN U-PLANE.
OUTPUT LOW-RATE ZEROS AND POLES IN U-PLANE.
ALL CALCULATIONS IN THE U-PLANE.

*****
OVERLAY(TXCONU,0,0)
PROGRAM TXCONU(INPUT=1008,OUTPUT=1008,TAPES=INPUT,TAPE7=OUTPUT)
CALL OVERLAY(8HTXCONU,23,0)
END

```


KFORM - INTERGER SPECIFYING FORMAT OF POLYNOMIALS.
 2 - REAL COEFF, PUR
 3 - REAL COEFF, IMAG COEFF, PUR
 ALPHA, SIGMA1, SIGMA2 - VARIABLES IN GENERAL BILINEAR TRANSFORMATION.
 Z, U, U' - ALPHASIGMA1+Z, U, U' /
 ISIGMA2+Z, U, U'
 KRES12 - NUMBER OF ZEROS IN LOW-RATE TRANSFORM ASSOCIATED WITH POLES WITH MULTIPLICITY-1.
 KRES22 - NUMBER OF ZEROS IN LOW-RATE TRANSFORM ASSOCIATED WITH POLES WITH MULTIPLICITY-2.
 KRES32 - NUMBER OF ZEROS IN LOW-RATE TRANSFORM ASSOCIATED WITH POLES WITH MULTIPLICITY-3.
 KJRES2 - INTEGER FLAG FOR POLES WITH MULT-2.
 THIS FLAG INVOKED WHEN TXFORM=2 OR
 OR TXFORM=ZR (I.E., WHEN INPUT Z-PLANE
 POLYS AND ROOTS TRANSFORMED TO U'-PLANE).
 KJRES2=0 - POLES OBTAINED THRU STRAIGHT
 TRANSFORMING NUMERATOR AND ZEROS THRU
 FACTORING NUMERATOR POLY.
 KJRES2=10 - POLES AND ZEROS OBTAINED THRU
 FACTORING NUM AND DEN POLYS.
 KJRES3 - INTERGER FLAG FOR POLES WITH MULT-3.
 SAME FUNCTION AS KJRES2.
 NPOLES - NUMBER OF DISTINCT POLES
 JRES1 - RESIDUE COUNTER FOR 1ST-ORDER POLES
 (UPDATED IN SUBROUTINE ZMULT1 & WMULT1)
 JRES2 - RESIDUE COUNTER FOR 2ND-ORDER POLES.
 (UPDATED IN SUBROUTINE WMULT2)
 JRES3 - RESIDUE COUNTER FOR 3RD-ORDER POLES.
 (UPDATED IN SUBROUTINE WMULT3)
 IPOLE - MASTER COUNTER FOR ALL RESIDUE CALCULATIONS.
 (ORIGINATES IN MAIN PROGRAM)
 NM - NUMBER OF ELEMENTS IN MULTPOLE
 ND - 2 * (ORDER DBNPOLY + 1)
 ND - 2 * (ORDER DBDPOLY + 1)
 DBNPOLY - ARRAY CONTAINING HIGH-RATE NUMERATOR POLY:
 L (REAL COEFF), (Z, U, OR U', PUR) J
 DBDPOLY - ARRAY CONTAINING HIGH-RATE DENOMINATOR POLY:
 L (REAL COEFF), (Z, U, OR U', PUR) J
 DBZERO - ARRAY CONTAINING HIGH-RATE ZEROS (REAL, IMAG)
 DBPOLE - ARRAY CONTAINING HIGH-RATE POLES (REAL, IMAG)
 MULTPOLE - ARRAY CONTAINING MULTIPLICITY OF EACH POLE
 A, B, C3, C6 - WORKING ARRAYS FOR INTERNAL CALCULATIONS
 THAT IS USED BY ALL ROUTINES.
 RES1POL - ARRAY CONTAINING LOW-RATE TRANSFORM 1ST ORDER
 RES2POL - ARRAY CONTAINING LOW-RATE TRANSFORM 2ND ORDER
 RES3POL - ARRAY CONTAINING LOW-RATE TRANSFORM 3RD ORDER
 RES1K - ARRAY CONTAINING NUMERATOR TERM FOR 1ST ORDER
 RES2K1, RES2K2 - ARRAYS CONTAINING NUMERATOR TERMS FOR
 2ND ORDER RESIDUES (SUBROUTINE WMULT2)

x 000166
 x 000167
 x 000168
 x 000169
 x 000170
 x 000171
 x 000172
 x 000173
 x 000174
 x 000175
 x 000176
 x 000177
 x 000178
 x 000179
 x 000180
 x 000181
 x 000182
 x 000183
 x 000184
 x 000185
 x 000186
 x 000187
 x 000188
 x 000189
 x 000190
 x 000191
 x 000192
 x 000193
 x 000194
 x 000195
 x 000196
 x 000197
 x 000198
 x 000199
 x 000200
 x 000201
 x 000202
 x 000203
 x 000204
 x 000205
 x 000206
 x 000207
 x 000208
 x 000209
 x 000210
 x 000211
 x 000212
 x 000213
 x 000214
 x 000215
 x 000216
 x 000217
 x 000218
 x 000219
 x 000220

[illegible]


```

000387 IF(NCLZ .EQ. 0) GO TO 27
000388 DO 25 I=1,NCLZ
000389 A(I)=CLZERO(1,1)
000390 25 B(I)=CLZERO(1,2)
000391 CALL COMPOLY(CLNPOLY,C3,A,B,NCLZ)
000392 GO TO 28
000393 27 CLNPOLY(1)=1.0
000394
000395 OBTAIN DENOMINATOR POLY FROM POLES
000396
000397 DO 26 I=1,NCLP
000398 A(I)=CLPOLE(1,1)
000399 26 B(I)=CLPOLE(1,2)
000400 CALL COMPOLY(CLPOLY,C3,A,B,NCLP)
000401
000402 ***** WRITE SECTION FOR INPUT *****
000403 *****
000404 *****
000405
000406 WRITE(7,8) XNT,TINT,TEXT
000407 8 FORMAT(1H1,RATIO =X,G15.8,4X,TINT =X,G15.8,4X,XTOUT =X,G15.8)
000408 WRITE(7,9) DBCLK,NCLZDB,NCLPDB
000409 9 FORMAT(1H ,GAIN =X,G15.8,5X,XZEROS =X,I3,I3X,XNPOLES =X,I3)
000410 WRITE(7,701) TXFORM
000411 701 FORMAT(1H ,,,,11X,A2,X-PLANE HIGH-RATE NUMERATOR%,/)
000412 NZ=NCLZ+1
000413 DO 203 I=1,NZ
000414 K=NCLZ-I+1
000415 WRITE(7,702) CLNPOLY(1),TXFORM,K
000416 702 FORMAT(1H ,10X,1X,G18.10,1,X,A2,3HX ,I3)
000417 WRITE(7,703) TXFORM
000418 703 FORMAT(1H ,,,,10X,A2,X-PLANE HIGH-RATE DENOMINATOR%,/)
000419 NP=NCLP+1
000420 DO 204 I=1,NP
000421 K=NCLP-I+1
000422 WRITE(7,702) CLNPOLY(1),TXFORM,K
000423 704 FORMAT(1H ,,,,14X,A2,X-PLANE HIGH-RATE ZEROS%,/)
000424 IF(NCLZ.EQ.0) GO TO 603
000425 WRITE(7,705) (CLZERO(1,1),CLZERO(1,2),I=1,NCLZ)
000426 705 FORMAT(1H ,5X,(X,G18.10,1,X,G18.10,1,X) )
000427 WRITE(7,706) TXFORM
000428 706 FORMAT(1H ,,,,14X,A2,X-PLANE HIGH-RATE POLES%,/)
000429 WRITE(7,705) (CLPOLE(1,1),CLPOLE(1,2),I=1,NCLP)
000430
000431 *****
000432 *****
000433 ***** Z-PLANE TO U'-PLANE TRANSFER *****
000434 *****
000435 *****
000436 *****
000437
000438 IF(TXFORM.NE.2HZ.AND.TXFORM.NE.2HZR)GO TO 8000
000439
000440 TRANSFORM INPUT Z-PLANE POLYS AND ROOTS TO U'-PLANE
000441 USING BILINEAR TRANSFORMATION:

```

```

000441      Z = [(2/T)+UP]/[(2/T)-UP]
000442      Z = ALPHA*(S1MA1+UP)/(S1MA2+UP)
000443      NDPOLE=NCLP
000444      ZFORM=2HZ
000445      IF(TXFORM.EQ.2HZR) ZFORM=2HZR
000446      KEY=2
000447      T=TINT
000448      ALPHA=-1.0
000449      S1MA1=2.0/TINT
000450      S1MA2=-2.0/TINT
000451      CALL BILIN(ALPHA,S1MA1,S1MA2,T,KEY,ZFORM,NDPOLE)
000452      NCLZDB=NCLZ
000453      NCLPDB=NCLP
000454      TXFORM=2HUP
000455      KTXFORM=10
000456
000457      8000 CONTINUE
000458
000459      ***** NUMERATOR POLY TRANSFER *****
000460      *****
000461      TRANSFER SINGLE PRECISION ARRAY CLNPOLY IFORMAT: REAL COEFFS
000462      INTO DOUBLE PRECISION ARRAY DBNPOLY I REAL COEF,PUR]
000463      WITH ALL ZERO COEFFICIENTS IN CLNPOLY DELETED.
000464      NN = 2 * (NCLZ+1) ---- NUMBER DBNPOLY ELEMENTS
000465      NCLZ = NUMBER ZEROS IN HIGH-RATE DISCRETE TRANSFORM
000466
000467      J=-1
000468      K=NCLZ+1
000469      KK=NCLZ+1
000470      DO 50 I=1,K
000471      C COUNTER FOR DESCENDING Z-PUR, U-PUR, OR U'-PUR
000472      KK=KK-1
000473      C CHECK FOR ZERO COEFFICIENT IN CLNPOLY
000474      IF(CLNPOLY(I)) 62,50,62
000475      C TRANSFER COEFFS & ADD Z-PUR, U-PUR, OR U'-PUR TO DBNPOLY
000476      62 J=J+2
000477      DBNPOLY(J)=CLNPOLY(I)
000478      DBNPOLY(J+1)=KK
000479      50 CONTINUE
000480      C SET DBNPOLY ELEMENTS, I.E., DBNPOLY(NN)
000481      NN=J+1
000482      C ORDER DBNPOLY IN DESCENDING POWERS OF Z, U, OR U'
000483      CALL ORDER3(DBNPOLY,NN,2)
000484
000485      ***** DENOMINATOR POLY TRANSFER *****
000486      *****
000487
000488
000489
000490
000491
000492
000493
000494
000495

```


144

```

000586
000587
000588
000589
000590
000591
000592
000593
000594
000595
000596
000597
000598
000599
000600
000601
000602
000603
000604
000605
000606
000607
000608
000609
000610
000611
000612
000613
000614
000615
000616
000617
000618
000619
000620
000621
000622
000623
000624
000625
000626
000627
000628
000629
000630
000631
000632
000633
000634
000635
000636
000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
000648
000649
000650
000651
000652
000653
000654
000655
000656
000657
000658
000659
000660

      THAT THE (AP + UP) ROOT APPEARS IN THE NUMERATOR
      AND SHOULD BE INCLUDED AS AN CANCELLING FACTOR IN
      THE RESIDUE CALCULATIONS.

      THE (AP +UP) FACTOR IS MULTIPLIED BY THE ORIGINALLY
      INPUTTED DENOMINATOR POLYNOMIAL D(UP) AND ADDED AS
      A DENOMINATOR POLE IF THIS FACTOR IS NOT A ROOT IN
      THE NUMERATOR. THE RESIDUE OF THE (AP + UP) POLE
      IS THEN CALCULATED ALONG WITH THE OTHER POLES OF D(UP).

      THE OVERALL RESIDUE EXPRESSION FOR THE U OR U' PLANE
      IS GIVEN BY:

      NUMERATOR = 2*PI*I*(UP)/D(P)*C((A+U)/(1+Y*XXNT))
      DENOMINATOR = U + A*(1-Y*XXNT)/(1+Y*XXNT))

      WHERE

      D(P) = D(UP) * (AP+UP) * (AP-UP)
      Y = C((AP+UP)/(AP-UP))
      Y*XXNT = C((AP+UP)/(AP-UP))*XXNT

      U = LOW-RATE U OR U' VARIABLE IZ = EXP(ST/N)
      UP = HIGH-RATE U OR U' VARIABLE IZ = EXP(ST/N)

      BILINEAR TRANSFORMATION BETWEEN Z, U, AND U' PLANES:
      U = (Z-1)/(Z+1)
      U' = (Z-1)/(Z+1)
      Z = (1+U)/(1-U)
      Z = (2/T+U)/(2-T-U)

      U TRANSFORM:
      A=1.0
      AP=1.0
      U = U
      UP = UP

      U' TRANSFORM:
      A = 2/TEXT
      AP = 2/TINT
      U = U'
      UP = UP'

      TEXT = LOW-RATE SAMPLING PERIOD (SEC)
      TINT = HIGH-RATE SAMPLING PERIOD (SEC)
      XNT = TEXT/TINT

      KZERO=0
      C SET XAP VARIABLE TO U OR U' ZERO AT UP = -AP
      XAP=-1.0
      DXAP=-1.0
      IF(TXFORM.EQ. 2HUP) XAP=-2.0/TINT
      IF(TXFORM.EQ. 2HUP) DXAP=-2.0/TINT
      C DO LOOP TO HUNT FOR (AP+UP) ZERO IN NUMERATOR

```

```

000661 IF(NCLZDB.EQ.0) GO TO 4001
000662 DO 4000 I=1,NCLZDB
000663 A1=DBZERO(I,1)
000664 A2=DBZERO(I,2)
000665 C CHECK FOR ZERO AT (AP+UP). IF FOUND, SET THE KZERO
000666 C VARIABLE TO "10" TO INVOKE THE CANCELLATION OF THIS
000667 C ZERO DURING THE RESIDUE CALCULATIONS.
000668 IF(ABS(A1-XAP).LE.1.E-11.AND.ABS(A2).LE.1.E-11) GO TO 5010
000669 4000 CONTINUE
000670 4001 CONTINUE
000671 C NO ZERO AT (AP+UP) WAS FOUND. ADD DOUBLE PRECISION POLE AT
000672 C (UP = -AP)
000673 DBPOLE(NCLPDB,1) = DXAP
000674 DBPOLE(NCLPDB,2) = 0.0
000675 C MULTIPLY THE G(UP) DENOMINATOR TIMES (AP+UP).
000676 C THAT IS, D(UP) * (AP+UP)
000677 AP(1) = 1.0
000678 AP(2) = 1.0
000679 AP(3) = -DXAP
000680 AP(4) = 0.0
000681 CALL MULTIP(AP,4,DBDPOLY,ND,A,NT3,2)
000682 C TRANSFER NEW DENOMINATOR POLY TO DBDPOLY
000683 DO 5020 I=1,NT3
000684 5020 DBDPOLY(I)=A(I)
000685 C SET DBDPOLY ELEMENTS, I.E., DBDPOLY(ND)
000686 ND=NT3
000687 C ORDER DBDPOLY IN DESCENDING POWERS OF U OR U'
000688 CALL ORDER3(DBDPOLY,ND,2)
000689 GO TO 5030
000690 C
000691 C SET KZERO VARIABLE TO INDICATE (AP+UP) ZERO PRESENT
000692 5010 KZERO = 10
000693 5030 CONTINUE
000694 C
000695 C
000696 C
000697 C
000698 C
000699 C
000700 C
000701 C THIS SECTION MULTIPLIES THE DENOMINATOR POLYNOMIAL
000702 C D(UP) BY THE FACTOR (AP-UP) PRIOR TO TAKING THE
000703 C DERIVATIVE WITH RESPECT TO THE HIGH-RATE U OR U'.
000704 C TRANSFORM VARIABLE UP. THE INCLUSION OF THE
000705 C (AP-UP) FACTOR OR POLE IN THE DENOMINATOR POLYNOMIAL
000706 C IS NOT ACTUALLY NECESSARY SINCE THE (AP-UP) POLE
000707 C IS NOT INCLUDED IN THE INTEGRATION CONTOUR THAT IS
000708 C BEING USED. INCLUDING THE (AP-UP) FACTOR IN THE
000709 C DENOMINATOR D(UP) OR CARRYING IT ALONG AS A SEPARATE
000710 C FACTOR HAS THE SAME EFFECT ON THE RESIDUES.
000711 AP(1) = -1.0
000712 AP(2) = 1.0
000713 AP(3) = -DXAP
000714 AP(4) = 0.0
000715

```

```

000711 CALL MULTIP(AP,4,DBDPOLY,ND,A,NT3,2)
000712 C TRANSFER NEW DENOMINATOR POLYNOMIAL TO DBDPOLY
000713 DO 5040 I=1,NT3
00719 5040 DBDPOLY(I)=A(I)
000720 C SET DBDPOLY ELEMENTS .I.E., DBDPOLY(ND)
000721 ND=NT3
000722 C ORDER DBDPOLY IN DESCENDING POWERS OF W OR W'
000723 (CALL ORDER3(DBDPOLY,ND,2))
000724 C
000725 5100 CONTINUE
000726 C
000727 C
000728 C OBTAIN MULTIPLICITY OF EACH POLE AND DELETE EXTRA POLES
000729 C
000730 IROW=NONZER+1
000731 ICOL=2
000732 CALL ORDPOLE(DBPOLE,IMOW,ICOL,NCLPDB,NPOLES,MULPOLE,MM)
000733 C
000734 C
000735 C
000736 C
000737 C
000738 C
000739 C
000740 C
000741 C
000742 C
000743 C
000744 C
000745 C
000746 C
000747 C
000748 C
000749 C
000750 C
000751 C
000752 C
000753 C
000754 C
000755 C
000756 C
000757 C
000758 C
000759 C
000760 C
000761 C
000762 C
000763 C
000764 C
000765 C
000766 C
000767 C
000768 C
000769 C
000770 C

WRITE(7,7) TXFORM
7 FORMAT(1H, '//,14X,A2,X-PLANE HIGH-RATE POLES#,,)
WRITE(7,15) (DBPOLE(I,1),DBPOLE(I,2),I=1,MM)
15 FORMAT(1H,5X,1(X,G18.10),X,G18.10,X)
WRITE(7,10)
10 FORMAT(1H, X X)
WRITE(7,17) (MULPOLE(I),I=1,MM)
17 FORMAT(1H,16X,X MULTIPLICITY -X,13,)

*****
***** MASTER DO LOOP FOR RESIDUES *****
*****
*****
*****

THE LOW-RATE Z-PLANE TRANSFORM IS OBTAINED FROM THE
HIGH-RATE Z-PLANE TRANSFORM USING THE FOLLOWING
GENERAL EXPRESSION:

G(Z) = SUM RESIDUES [ G(P)/P * (Z)/(Z-P*XXNT) ]
      [ P-POLES G(P)/P ]

G(Z) = SUM RESIDUES [ (N(P)*Z) / (D1(P)*Z - D1(P)*P*XXNT) ]
      [ P-POLES G(P)/P ]

G(Z) = N(Z)/D(Z) --- LOW-RATE TXFORM Z=EXP(S*TX)
G(P) = N(P)/D(P) --- HIGH-RATE TXFORM P=Z*EXP(S*TX)
D1(P) = D(P) * P
T = LOW-RATE SAMPLING INTERVAL
T/M = HIGH-RATE SAMPLING INTERVAL
XNT = (T) / (T/M) = N

```



```

C INITIALIZE RESIDUE COUNTERS TO ZERO
JRES1=0
JRES2=0
JRES3=0
C START OF MASTER DO LOOP FOR RESIDUES
C NM = NUMBER OF DISTINCT POLES
DO 400 I=1,NM
C SET COUNTER VARIABLE FOR RESIDUE SUBROUTINES
IPOLE=I
C CHECK FOR MULTIPLICITY .GT. 3
IF(MULPOLE(IPOLE) .GT. 3) GO TO 999
C BRANCH ACCORDING TO MULTIPLICITY OF POLE
GO TO (1000,2000,3000), MULPOLE(IPOLE)
999 WRITE(7,1)
1 FORMAT(1H ,*MULTIPLICITY GREATER THAN 3 - EXIT PROGRAM *)
STOP
1000 IF(TXFORM .EQ. 2HUP .OR. TXFORM .EQ. 2HU ) GO TO 1001
C-----
C CALL OVERLAY(IGHTXCONU,23,1)
C-----
C CALL ZMULT1
C-----
GO TO 400
C-----
1001 CALL OVERLAY(IGHTXCONU,23,2)
C-----
C1001 CALL UMULT1
C-----
GO TO 400
2000 IF(TXFORM .EQ. 2HUP .OR. TXFORM .EQ. 2HU ) GO TO 2001
WRITE(7,2010)
2010 FORMAT(1H ,*MULTIPLICITY .EQ. 2 IN ZT-PLANE - EXIT PROGRAM*)
STOP
C-----
2001 CALL OVERLAY(IGHTXCONU,23,3)
C-----
C2001 CALL UMULT2
C-----
GO TO 400
3000 IF(TXFORM .EQ. 2HUP .OR. TXFORM .EQ. 2HU ) GO TO 3001
WRITE(7,2011)
2011 FORMAT(1H ,*MULTIPLICITY .EQ. 3 IN ZT-PLANE - EXIT PROGRAM*)

```

```

000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
000846
000847
000848
000849
000850
000851
000852
000853
000854
000855
000856
000857
000858
000859
000860
000861
000862
000863
000864
000865
000866
000867
000868
000869
000870
000871
000872
000873
000874
000875
000876
000877
000878
000879
000880

```

```

000881
000882
000883
000884
000885
000886
000887
000888
000889
000890
000891
000892
000893
000894
000895
000896
000897
000898
000899
000900
000901
000902
000903
000904
000905
000906
000907
000908
000909
000910
000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935

STOP
C-----
C 3001 CALL OVERLAY(6HTXCONU,23,4)
C-----
C
C 3001 CALL UMULT3
C-----
C
C 400 CONTINUE
C-----
C
C IF(JRES1 .EQ. 0) GO TO 401
C-----
C CALL OVERLAY(6HTXCONU,23,5)
C-----
C
C CALL RES1
C-----
C
C IF(KTXFORM .LT. 5) GO TO 401
C IF(JRES1 .EQ. 0) GO TO 401
C-----
C CALL WZBILIN TO TRANSFER U' POLY AND ROOTS BACK INTO Z-PLANE
C NCD=2
C ALPHA=2.0/TEXT
C SIGNAL=-1.0
C SIGMA2=1.0
C KEY=4
C T=TEXT
C CALL WZBILIN(RESIN,KRESIN,RESID,KRESID,
C 1 RESIPOL,JRES1,NRD,NCD,ALPHA,SIGMA1,SIGMA2,3,T,KEY,ZFORM)
C 401 CONTINUE
C-----
C
C IF(JRES2 .EQ. 0) GO TO 402
C-----
C CALL OVERLAY(6HTXCONU,23,6)
C-----
C
C CALL RES2
C-----
C
C DO 412 I=1,JRES2
C DBPOLE(I,1)=RES2POL(I,1)
C 412 DBPOLE(I,2)=RES2POL(I,2)
C IF(KTXFORM .LT. 5) GO TO 402

```

```

C C CALL I      JM TO TRANSFER U' POLY AND ROOTS BACK INTO Z-PLANE
C NP      RDER+1
C N1
C A-2.0/TEXT
C .MA1=-1.0
C SIGMA2=1.0
C KEY=4
C T-TEXT
C CALL UZBILIN(RES2N,KRES2N,RES2D,KRES2D,
1 DBPOLE,JRES2,NRD,NCD,ALPHA,SIGMA1,SIGMA2,3,T,KEY,ZFORM)
IF(ZFORM.EQ. 2H2R) GO TO 402
KJRES2=10
402 CONTINUE
C
C IF(JRES3.EQ. 0) GO TO 403
C
C CALL OVERLAY(6HTXCONU,23,7)
C
C CALL RES3
C
C DO 413 I=1,JRES3
DBZERO(I,1)=RES3POL(I,1)
413 DBZERO(I,2)=RES3POL(I,2)
IF(KTXFORM.EQ. 5) GO TO 403
C CALL UZBILIN TO TRANSFER U' POLYS AND ROOTS BACK INTO Z-PLANE
NRD=NORDER+1
NCD=2
ALPHA=2.0/TEXT
SIGMA1=-1.0
SIGMA2=1.0
KEY=4
T-TEXT
CALL UZBILIN(RES3N,KRES3N,RES3D,KRES3D,
1 DBZERO,JRES3,NRD,NCD,ALPHA,SIGMA1,SIGMA2,3,T,KEY,ZFORM)
IF(ZFORM.EQ. 2H2R) GO TO 403
KJRES3=10
403 CONTINUE
C
C *****
C ***** LOW-RATE TRANSFER FUNCTION ADDITION *****
C *****
C ADD THE INDIVIDUAL LOW-RATE TRANSFER FUNCTIONS FOR THE 1ST,
C 2ND AND 3RD ORDER RESIDUES TO OBTAIN THE OUTPUT LOW-RATE
C TRANSFER FUNCTION.

```

```

000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975
000976
000977
000978
000979
000980
000981
000982
000983
000984
000985
000986
000987
000988
000989
000990

```

```

00993      FORM THE COMBINED NUMERATOR POLYNOMIALS
C
C
C      RESN=RESN#RES2D#RES3D
C      RES2N=RESN#RES1D#RES3D
C      RES3N=RESN#RES1D#RES2D
C
C
C      IF(KRES1N .EQ. 0) GO TO 800
C      IF(JRES2 .EQ. 0) GO TO 805
C      CALL MULTIP(RESIN,KRES1N,RES2D,KRES2D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,PES1N,KRES1N)
C
C      805 IF(JRES3 .EQ. 0) GO TO 800
C      IF(JRES3 .EQ. 0) GO TO 800
C      CALL MULTIP(RESIN,KRES1N,RES2D,KRES3D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,PES1N,KRES1N)
C
C      800 CONTINUE
C      IF(JRES2 .EQ. 0) GO TO 810
C      IF(KRES1N .EQ. 0) GO TO 815
C      CALL MULTIP(RES2N,KRES2N,RES1D,KRES1D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,PES2N,KRES2N)
C
C      815 IF(JRES3 .EQ. 0) GO TO 810
C      CALL MULTIP(RES2N,KRES2N,RES2D,KRES3D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,PES2N,KRES2N)
C
C      810 CONTINUE
C
C      IF(JRES3 .EQ. 0) GO TO 820
C      IF(KRES1N .EQ. 0) GO TO 825
C      CALL MULTIP(RES3N,KRES3N,RES1D,KRES1D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,PES3N,KRES3N)
C
C      825 IF(JRES2 .EQ. 0) GO TO 820
C      CALL MULTIP(RES3N,KRES3N,RES2D,KRES2D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,PES3N,KRES3N)
C
C      820 CONTINUE
C
C
C      ADD THE COMBINED NUMERATOR POLYS [RESIN=RES1N#RES2N#RES3N]
C      AND FORM THE COMMON DENOMINATOR POLY [IRESID=RES1D#RES2D#RES3D]
C
C
C      IF(KRES1N .EQ. 0) GO TO 830
C      IF(JRES2 .EQ. 0) GO TO 840
C      IF(JRES3 .EQ. 0) GO TO 850
C
C
C      CALL ADD(RES1N,KRES1N,RES2N,KRES2N,C3,NT3,3)
C      CALL ADD(C3,NT3,KRES3N,KRES3N,C6,NT6,3)
C      CALL DOLOOP(C6,NT6,RES1N,KRES1N)
C      CALL MULTIP(RES1D,KRES1D,RES2D,KRES2D,C3,NT3,3)
C      CALL MULTIP(C3,NT3,RES3D,KRES3D,C6,NT6,3)
C      CALL DOLOOP(C6,NT6,RES1D,KRES1D)
C      GO TO 855
C
C
C      830 IF(JRES2 .EQ. 0) GO TO 832
C      IF(JRES3 .EQ. 0) GO TO 834
C      CALL ADD(RES2N,KRES2N,RES1N,KRES1N,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,RES1N,KRES1N)
C      CALL MULTIP(RES2D,KRES2D,RES3D,KRES3D,C3,NT3,3)
C      CALL DOLOOP(C3,NT3,RES1D,KRES1D)

```

```

GO TO 855
832 CALL DOLLOOP(RES3N,KRES3N,RES1N,KRES1N)
CALL DOLLOOP(RES3D,KRES3D,RES1D,KRES1D)
GO TO 855
834 CALL DOLLOOP(RES2N,KRES2N,RES1N,KRES1N)
CALL DOLLOOP(RES2D,KRES2D,RES1D,KRES1D)
GO TO 855
C
840 IF(JRES3.EQ.0) GO TO 855
CALL ADD(RES1N,KRES1N,RES3N,KRES3N,C3,NT3,3)
CALL DOLLOOP(C3,NT3,RES1N,KRES1N)
CALL MULTIPLY(RES1D,KRES1D,RES3D,KRES3D,C3,NT3,3)
CALL DOLLOOP(C3,NT3,RES1D,KRES1D)
GO TO 855
C
850 CALL ADD(RES1N,KRES1N,RES2N,KRES2N,C3,NT3,3)
CALL DOLLOOP(C3,NT3,RES1N,KRES1N)
CALL MULTIPLY(RES1D,KRES1D,RES2D,KRES2D,C3,NT3,3)
CALL DOLLOOP(C3,NT3,RES1D,KRES1D)
C
855 CONTINUE
C
C
C ELIMINATE ZERO LEADING COEFFICIENT IN NUMERATOR POLY
II-1
607 DO 609 I=1,KRESIN,3
IF(DABS(RESIN(I)).GE.1.D-11) GO TO 610
KRESIN=KRESIN-3
IF(KRESIN.EQ.0) GO TO 610
DO 608 J=1,KRESIN
608 RESIN(J)=RESIN(J)+3
II-1
GO TO 607
609 CONTINUE
610 CONTINUE
C
C ELIMINATE ZERO LEADING COEFFICIENTS IN DENOMINATOR POLY
II-1
611 DO 613 I=1,KRESID,3
IF(RESID(I).NE.0.0) GO TO 614
KRESID=KRESID-3
IF(KRESID.EQ.0) GO TO 614
DO 612 J=1,KRESID
612 RESID(J)=RESID(J)+3
II-1
GO TO 611
613 CONTINUE
614 CONTINUE
C
C
C ORDER THE TOTAL NUMERATOR AND DENOMINATOR POLYNOMIALS IN
C DESCENDING POWERS AND ADD MISSING COEFFICIENTS
C

```

```

001046
001047
001048
001049
001050
001051
001052
001053
001054
001055
001056
001057
001058
001059
001060
001061
001062
001063
001064
001065
001066
001067
001068
001069
001070
001071
001072
001073
001074
001075
001076
001077
001078
001079
001080
001081
001082
001083
001084
001085
001086
001087
001088
001089
001090
001091
001092
001093
001094
001095
001096
001097
001098
001099
001100

```

```

      IF(KRESIN.EQ.3) GO TO 605
      CALL ORDER3(RESIN,KRESIN,3)
605  IF(KRESID.EQ.3) GO TO 606
      CALL ORDER3(RESID,KRESID,3)
606  NCLZ=RESIN(3)
      NCLP=RESID(3)
      C NCLZ1=NCLZ
      C NCLP1=NCLP
      IF(NCLZ.EQ.0) GO TO 604
      CALL COEFF(RESIN,KRESIN,NCLZ,3)
604  CALL COEFF(RESID,KRESID,NCLP,3)
      C
      C NORMALIZE NUMERATOR AND DENOMINATOR POLYS AND TRANSFER
      C REAL COEFFICIENTS (ONLY) INTO ARRAYS CLNPOL1 AND CLDPOL1.
      C
      K=1
      CLNK=RESIN(1)
      CLDK=RESID(1)
      DBCLK=DBCLK*(CLNK/CLDK)
      DO 870 I=1,KRESIN,3
      IF(DABS(RESIN(I)/CLNK).LE.1.D-20) RESIN(I)=0.0
      IF(DABS(RESID(I)/CLDK).LE.1.D-20) RESID(I)=0.0
      CLNPOL1(K)=RESIN(I)/CLDK
      CLDPOL1(K)=RESID(I)/CLDK
      RESIN(I)=RESIN(I)/CLNK
      RESID(I)=RESID(I)/CLDK
870  K=K+1
      C
      K=1
      DO 871 I=1,KRESID,3
      IF(DABS(RESID(I)/CLDK).LE.1.D-20) RESID(I)=0.0
      IF(DABS(RESIN(I+1)/CLDK).LE.1.D-20) RESID(I+1)=0.0
      CLDPOL1(K)=RESID(I)/CLDK
      RESID(I)=RESID(I)/CLDK
      RESID(I+1)=RESID(I+1)/CLDK
871  K=K+1
      C
      CLNK1=CLNPOL1(1)
      CLK1=CLNPOL1(1)
      CLDK1=CLDPOL1(1)
      C
      C OBTAIN THE LOW-RATE NUMERATOR ROOTS (ZEROS)
      C
      IF(NCLZ.EQ.0) GO TO 876
      K=1
      DO 874 I=1,KRESIN,3
      A(K)=RESIN(I)
      B(K)=RESIN(I+1)
874  K=K+1
      CALL ROOTS(A,B,NCLZ,C3,C6)
      DO 875 I=1,NCLZ
      IF(DABS(C3(I)).LE.1.D-20) C3(I)=0.0
      IF(DABS(C6(I)).LE.1.D-20) C6(I)=0.0
      CLZERO(I,1)=C3(I)
      CLZERO(I,2)=C6(I)
      CLZERO(I,1)=C3(I)
      C

```

```

001101
001102
001103
001104
001105
001106
001107
001108
001109
001110
001111
001112
001113
001114
001115
001116
001117
001118
001119
001120
001121
001122
001123
001124
001125
001126
001127
001128
001129
001130
001131
001132
001133
001134
001135
001136
001137
001138
001139
001140
001141
001142
001143
001144
001145
001146
001147
001148
001149
001150
001151
001152
001153
001154
001155

```

```

C      CLZERO(I,2)=CS(I)
      875 CONTINUE
      876 CONTINUE
C
C      TRANSFER LOW-RATE POLES INTO ARRAY CLPOLE
C
      IF(JRES1.EQ.0) GO TO 881
      DO 880 I=1,JRES1
      IF(DABS(RES1POL(I,1)) .LE. 1.D-20) RES1POL(I,1)=0.0
      IF(DABS(RES1POL(I,2)) .LE. 1.D-20) RES1POL(I,2)=0.0
      CLPOLE(I,1)=RES1POL(I,1)
      CLPOLE(I,2)=RES1POL(I,2)
      CLPOLE1(I,1)=RES1POL(I,1)
      CLPOLE1(I,2)=RES1POL(I,2)
      880 CONTINUE
C
      881 IF(JRES2.EQ.0) GO TO 887
      KP=2
      IF(JRES2.GT.5) KP=1
      DO 886 K=1,JRES2
      IF(DABS(DBPOLE(K,1)) .LE. 1.D-20) DBPOLE(K,1)=0.0
      IF(DABS(DBPOLE(K,2)) .LE. 1.D-20) DBPOLE(K,2)=0.0
      DO 885 I=1,KP
      CLPOLE(KK,1)=DBPOLE(K,1)
      CLPOLE(KK,2)=DBPOLE(K,2)
      CLPOLE1(KK,1)=DBPOLE(K,1)
      CLPOLE1(KK,2)=DBPOLE(K,2)
      885 KK=KK+1
      886 CONTINUE
C
      887 IF(JRES3.EQ.0) GO TO 892
      KP=3
      IF(JRES3.GT.5) KP=1
      DO 891 K=1,JRES3
      IF(DABS(DBZERO(K,1)) .LE. 1.D-20) DBZERO(K,1)=0.0
      IF(DABS(DBZERO(K,2)) .LE. 1.D-20) DBZERO(K,2)=0.0
      DO 890 I=1,KP
      CLPOLE(KK,1)=DBZERO(K,1)
      CLPOLE(KK,2)=DBZERO(K,2)
      CLPOLE1(KK,1)=DBZERO(K,1)
      CLPOLE1(KK,2)=DBZERO(K,2)
      890 KK=KK+1
      891 CONTINUE
      892 CONTINUE
C
      CANCEL EQUAL ZEROS AND POLES VIA THE SUBROUTINE 'CANROOT'
C
      TOLR=1.D-11
      TOLI=1.D-11
      NZ=NCLZ
      NP=NCLP
      CALL CANROOT(TOLR,TOLI,CLZERO,NZ,CLPOLE,NP)

```

```

001156
001157
001158
001159
001160
001161
001162
001163
001164
001165
001166
001167
001168
001169
001170
001171
001172
001173
001174
001175
001176
001177
001178
001179
001180
001181
001182
001183
001184
001185
001186
001187
001188
001189
001190
001191
001192
001193
001194
001195
001196
001197
001198
001199
001200
001201
001202
001203
001204
001205
001206
001207
001208
001209
001210

```

```

001211 IF(NZ .EQ. NCLZ) GO TO 950
001212
001213 C OBTAIN NEW NUMERATOR POLYNOMIAL VIA SUBROUTINE "COMPOLY"
001214 C
001215
001216 NCLZ=NZ
001217 NCLP=NP
001218 NCLP1=NP
001219 IF(NCLZ .EQ. 0) GO TO 910
001220 DO 905 I=1,NCLZ
001221 A(I)=CLZERO(I,1)
001222 B(I)=CLZERO(I,2)
001223 CLZERO(I,1)=CLZERO(I,1)
001224 CLZERO(I,2)=CLZERO(I,2)
001225
001226 C 905 CONTINUE
001227 CALL COMPOLY(C6,C3,A,B,NCLZ)
001228
001229 C TRANSFER NEW NUMERATOR POLY INTO ARRAY "RESIN"
001230 C
001231 PUR=NCLZ
001232 K=1
001233 NZ=NCLZ+1
001234 KRESIN=3*NZ
001235 DO 906 I=1,NZ
001236 IF(DABS(C6(I)) .LE. 1.D-20) C6(I)=0.0
001237 IF(DABS(C3(I)) .LE. 1.D-20) C3(I)=0.0
001238 RESIN(K)=C6(I)
001239 RESIN(K+1)=C3(I)
001240 RESIN(K+2)=PUR
001241 CLNPOL(I)=C6(I)*(CLNK/CLDK)
001242 PUR=PUR-1.0
001243 K=K+3
001244 GO TO 911
001245
001246 C 910 RESIN(I)=1.0
001247 RESIN(2)=0.0
001248 RESIN(3)=0.0
001249 CLNPOL(1)=CLNK/CLDK
001250
001251 C OBTAIN NEW DENOMINATOR POLYNOMIAL VIA SUBROUTINE "COMPOLY"
001252 C
001253
001254 911 DO 915 I=1,NCLP
001255 A(I)=CLPOLE(I,1)
001256 B(I)=CLPOLE(I,2)
001257 CLPOLE(I,1)=CLPOLE(I,1)
001258 CLPOLE(I,2)=CLPOLE(I,2)
001259
001260 C 915 CONTINUE
001261 CALL COMPOLY(C6,C3,A,B,NCLP)
001262
001263 C TRANSFER NEW DENOMINATOR POLY INTO ARRAY "RESID"
001264 C
001265 PUR=NCLP
001266 K=1
001267 NP=NCLP+1
001268 KRESID=3*NP
001269 DO 920 I=1,NP

```



```

C      NDPOLES = NUMBER OF DISTINCT POLES SENT TO
C      SUBROUTINE 'SZUR00T'.
C      ZFORM = ALPHANUMERIC (A2) FLAG THAT INDICATES
C      EITHER THE 'Z' OR 'ZR' OPTIONS
C      NOTE: COMMON/TXCONU9/ USED TO TRANSFER POLYS
C      AND ROOTS TO AND FROM 'BILIN'.
C
C      DOUBLE ALPHA,SIGMA1,SIGMA2,CLK,CLNK,CLDX,T
C      DOUBLE A(106),B(106),C3(318),C6(318)
C      DOUBLE CLNPOLY(51),CLDPOLY(51),CLZERO(50,2),CLPOLE(50,2)
C      COMMON/TXCONU9/CLNPOLY,CLDPOLY,CLZERO,CLPOLE,
C      * NCLZ,NCLP,CLK,CLNK,CLDX
C      COMMON/TXCONU2/A,B,C3,C6
C      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NM,NORDER
C
C      IORD=2*(NORDER+3)
C      DO 100 I=1,IORD
C      A(I)=0.0
C      B(I)=0.0
C      IORD=6*(NORDER+3)
C      DO 150 I=1,IORD
C      C3(I)=0.0
C
C      M=NCLZ+1
C      N=NCLP+1
C      DO 1 I=1,M
C      1 A(N+1-I)=CLNPOLY(M+1-I)
C      DO 2 I=1,N
C      2 B(N+1-I)=CLDPOLY(N+1-I)
C      IORD=NORDER+1
C      DO 175 I=1,IORD
C      CLNPOLY(I)=0.0
C      CLDPOLY(I)=0.0
C      175 CLDPOLY(I)=0.0
C
C      DO 50 I=1,N
C      CALL TERM(N-I,N-1,ALPHA,SIGMA1,SIGMA2)
C      DO 50 J=1,N
C      CLNPOLY(J)=CLNPOLY(J) + A(I)*C3(J)
C      CLDPOLY(J)=CLDPOLY(J) + B(I)*C3(J)
C      50 CONTINUE
C
C      ELIMINATE ZERO LEADING COEFFICIENT IN NUMERATOR POLY
C      AND SET NUMBER OF ZEROS (NCLZ)
C
C      NZ=N
C      II=1
C      61 DO 63 I=II,NZ
C      IF(CLNPOLY(I).NE.0.0) GO TO 64
C      NZ=NZ-1
C      IF(NZ.EQ.0) GO TO 64
C      DO 62 J=I,NZ
C      62 CLNPOLY(J)=CLNPOLY(J+1)
C      II=I

```

```

001376
001377
001378
001379
001380
001381
001382
001383
001384
001385
001386
001387
001388
001389
001390
001391
001392
001393
001394
001395
001396
001397
001398
001399
001400
001401
001402
001403
001404
001405
001406
001407
001408
001409
001410
001411
001412
001413
001414
001415
001416
001417
001418
001419
001420
001421
001422
001423
001424
001425
001426
001427
001428
001429
001430

```

```

001431 GO TO 61
001432 63 CONTINUE
001433 64 NCLZ=NZ-1
001434
001435 C
001436 C ELIMINATE ZERO LEADING COEFFICIENT IN DENOMINATOR POLY
001437 C AND SET NUMBER OF POLES (NCLP)
001438
001439 NP=N
001440 II=1
001441 65 DO 67 J=II,MP
001442 IF (CLDPOLY(I) .NE. 0.0) GO TO 68
001443 NP=NP-1
001444 IF (NP .EQ. 0) GO TO 68
001445 DO 66 J=1,MP
001446 66 CLDPOLY(J)=CLDPOLY(J+1)
001447 II=I
001448 GO TO 65
001449 67 CONTINUE
001450 68 NCLP=NP-1
001451
001452 C
001453 C DEVIDE NUMERATOR AND DENOMINATOR POLYS BY HIGHEST
001454 C COEFFICIENT OF DENOMINATOR AND SET POLY GAIN
001455 C VARIABLES - CLK, CLNK, AND CLDK
001456
001457 CLDK=CLDPOLY(1)
001458 DO 60 I=1,M
001459 CLNPOLY(I)=CLNPOLY(I)/CLDK
001460 60 CLDPOLY(I)=CLDPOLY(I)/CLDK
001461 CLNK=CLNPOLY(1)
001462 CLK=CLNPOLY(1)
001463 CLDK=CLDPOLY(1)
001464
001465 C
001466 C OBTAIN NUMERATOR AND DENOMINATOR ROOTS
001467 C
001468 IF (NCLZ .EQ. 0) GO TO 275
001469 DO 200 I=1,NZ
001470 A(I)=CLNPOLY(1)
001471 200 B(I)=0.0
001472 NZ=NCLZ
001473 CALL ROOTS(A,B,NZ,C3,C6)
001474 C TRANSFER NUMERATOR ROOTS INTO ARRAY CLZERO(50,2)
001475 DO 250 I=1,NZ
001476 CLZERO(I,1)=C3(I)
001477 CLZERO(I,2)=C6(I)
001478 250 CONTINUE
001479
001480 C
001481 IF (NCLP .EQ. 0) GO TO 375
001482 IF (ZFORM .EQ. 2HZ) GO TO 310
001483
001484 C THIS SECTION TRANSFERS POLES WITH STRAIGHT TRANSFORMATIONS
001485 C AND DOES NOT FACTOR DENOMINATOR POLY.
001486 C THIS SECTION INVOKED WHEN ZFORM=2HZR.

```

```

C
DO 305 I=1,NMPOLE
A(I)=CLPOLE(I,1)
305 B(I)=CLPOLE(I,2)
CALL SZROOT(A,B,NMPOLE,T,REV)
DO 325 I=1,NMPOLE
CLPOLE(I,1)=A(I)
325 CLPOLE(I,2)=B(I)
GO TO 375
310 CONTINUE
C
DO 340 I=1,NP
A(I)=CLDPOLY(I)
340 B(I)=0.0
NP=NCLP
CALL ROOTS(A,B,NP,C3,C6)
C TRANSFER DENOMINATOR ROOTS INTO ARRAY CLPOLE(50,2)
DO 350 I=1,NP
CLPOLE(I,1)=C3(I)
350 CLPOLE(I,2)=C6(I)
375 CONTINUE
C
RETURN
END
C-----
CDECK TERM
SUBROUTINE TERM(I,NP,ALPHA,SIGMA1,SIGMA2)
C-----
THIS SUBROUTINE CALCULATES THE INDIVIDUAL TERMS USED IN THE
BILINEAR TRANSFORMATION BETWEEN THE Z, W, AND U PLANES.

      [(A+W)*X] X [(A-U)*X]
      J = NP-I

THE INDIVIDUAL TERMS CALCULATED IN THIS SUBROUTINE ARE
PASSED TO THE SUBROUTINE 'BILIN' VIA ARRAY 'C3' IN
COMMON/TXCONU2/A,B,C3,C6

DOUBLE ALPHA,SIGMA1,SIGMA2,PUR
DOUBLE RP(50),RI(50),POLYA(102),POLYB(102)
DOUBLE A(106),B(106),C3(318),C6(318)
COMMON/TXCONU2/A,B,C3,C6

J=NP-I
JJ=J+1
II=I+1
IF(I.EQ.0) GO TO 20
DO 10 K=1,I
RP(K)=-SIGMA1
10 RI(K)=0.0
CALL COMPOLY(C6,POLYA,RR,RI,I)
PUR=FLOAT(I)
C

```

```

001486
001487
001488
001489
001490
001491
001492
001493
001494
001495
001496
001497
001498
001499
001500
001501
001502
001503
001504
001505
001506
001507
001508
001509
001510
001511
001512
001513
001514
001515
001516
001517
001518
001519
001520
001521
001522
001523
001524
001525
001526
001527
001528
001529
001530
001531
001532
001533
001534
001535
001536
001537
001538
001539
001540

```

```

001541
001542
001543
001544
001545
001546
001547
001548
001549
001550
001551
001552
001553
001554
001555
001556
001557
001558
001559
001560
001561
001562
001563
001564
001565
001566
001567
001568
001569
001570
001571
001572
001573
001574
001575
001576
001577
001578
001579
001580
001581
001582
001583
001584
001585
001586
001587
001588
001589
001590
001591
001592
001593
001594
001595

      KL=1
      DO 11 JL=1,17
      POLYA(KL)-C6(JL)
      POLYA(KL+1)-PUR
      PUR-PUR-1.0
      11 KL=KL+2
      IPOLYA-2*II
      GO TO 25
      20 POLYA(1)-1.0
      POLYA(2)-0.0
      IPOLYA-2
      25 CONTINUE
      C
      IF(J.EQ.0) GO TO 40
      DO 30 K=1,J
      RR(K)= -SIGMA2
      30 RI(K)= 0.0
      CALL COMPOLY(C6,POLYB,RR,RI,J)
      PUR=FLOAT(J)
      KL=1
      DO 31 JL=1,JJ
      POLYB(KL)-C6(JL)
      POLYB(KL+1)-PUR
      PUR-PUR-1.0
      31 KL=KL+2
      IPOLYB-2*JJ
      GO TO 45
      40 POLYB(1)-1.0
      POLYB(2)-0.0
      IPOLYB-2
      45 CONTINUE
      C
      CALL MULTIP(POLYA,IPOLYA,POLYB,IPOLYB,C6,N3,2)
      C
      ADD MISSING COEFFICIENTS TO C6 POLYNOMIAL
      C
      CALL COEFF(C6,N3,NP,2)
      N=2*(NP+1)
      JKK=1
      DO 50 JK=1,N,2
      C3(JKK)=C6(JK)*Z(ALPHAZZ1)
      50 JKK=JKK+1
      C
      RETURN
      END
      C
      CDECK COEFF
      SUBROUTINE COEFF(A,NA,NO,KFORM)
      C
      C THIS SUBROUTINE ADDS THE MISSING POWER TERMS IN AN
      C POLYNOMIAL BY INSERTING A ZERO COEFFICIENT WITH THE
      C APPROPRIATE POWER AND MOVING THE ORIGINAL TERMS TO
      C MAKE ROOM FOR THIS MISSING TERM.
      C A = DOUBLE PRECISION ARRAY CONTAINING INPUT POLY IN

```

```

001596
001597
001598
001599
001600
001601
001602
001603
001604
001605
001606
001607
001608
001609
001610
001611
001612
001613
001614
001615
001616
001617
001618
001619
001620
001621
001622
001623
001624
001625
001626
001627
001628
001629
001630
001631
001632
001633
001634
001635
001636
001637
001638
001639
001640
001641
001642
001643
001644
001645
001646
001647
001648
001649
001650

FORMAT SPECIFIED BY KFORM
NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'
NC = ORDER OF INPUT POLY IN ARRAY 'A'
KFORM = INTEGER SPECIFYING POLY FORMAT
KFORM = 2 ---- (REAL COEFF, PUR)
KFORM = 3 ---- (REAL COEFF, IMAG COEFF, PUR)
OUTPUT:
A = POLY WITH ZERO INSERTED FOR MISSING COEFFS
NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'

DOUBLE A(1),DPUR
IF (NA .EQ. KFORM) GO TO 50
CALL SIMPLX(A,NA,KFORM)
CALL ORDER3(A,NA,KFORM)
SET NUMBER OF ELEMENTS ACCORDING TO POLY ORDER
50 N=2*(NC+1)
IF (KFORM .EQ. 3) N=3*(NC+1)
IF (NA .EQ. N) RETURN
C INITIALIZE ADDITIONAL ELEMENTS THAT ARE NEEDED IN THE
C POLY ARRAY 'A' TO MAKE ROOM FOR MISSING TERMS.
N1=NA+1
DO 100 I=N1,N
DO 100 J=N1,N
100 A(I)=0.0
C INITIALIZE THE POWER COUNTER 'PUR'
PUR=FLOAT(N0)
DPUR=FLOAT(N0)
DO 200 I=1,N,KFORM
JKEY=I+KFORM-1
R=A(JKEY)
C CHECK FOR MISSING POWERS
IF (PUR-R) 300,210,300
C SHIFT COEFFICIENTS AND POWERS UP TWO ELEMENTS IN
C ARRAY 'A' FROM THE POINT OF THE MISSING POWER AS
C SPECIFIED BY JKEY
300 NX=NA+KFORM-JKEY
DO 350 J=1,NX
NN=NA+(KFORM+1)-J
A(NN)=A(NN-KFORM)
NA=NA+KFORM
A(JKEY)=0.0
A(JKEY)=DPUR
IF (KFORM .EQ. 3) A(JKEY-2)=0.0
C REDUCE POWER COUNTER BY ONE (1)
210 DPUR=DPUR-1.0
200 PUR=PUR-1.0
NA=N
RETURN
END

C-----CDECK WZBILIN
SUBROUTINE WZBILIN(A,NA,B,NB,D,NDFOLE,NRD,NCD,
* ALPHA,SIGMA1,SIGMA2,KFORM,T,KEY,ZFORM)
C-----
001596
001597
001598
001599
001600
001601
001602
001603
001604
001605
001606
001607
001608
001609
001610
001611
001612
001613
001614
001615
001616
001617
001618
001619
001620
001621
001622
001623
001624
001625
001626
001627
001628
001629
001630
001631
001632
001633
001634
001635
001636
001637
001638
001639
001640
001641
001642
001643
001644
001645
001646
001647
001648
001649
001650

```

```

001652 THIS SUBROUTINE CALCULATES THE SPECIFIC BILINEAR
001653 TRANSFORMATION FROM THE U'-PLANE TO THE Z-PLANE.
001654
001655 A = DOUBLE PRECISION ARRAY CONTAINING INPUT UP-PLANE
001656 NUMERATOR POLY AND OUTPUT Z-PLANE NUMERATOR POLY
001657 IN FORMAT SPECIFIED BY KEFORM
001658 NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'
001659 R = SAME AS ARRAY 'A' FOR DENOMINATOR POLY
001660 NR = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'B'
001661 D = DOUBLE PRECISION ARRAY CONTAINING POLES (REAL, IMAG)
001662 NPOLES = NUMBER OF DISTINCT POLES
001663 NRD = NUMBER OF ROOTS IN ARRAY 'D'
001664 NCD = NUMBER OF COLUMNS IN ARRAY 'D'
001665 NOTE1 FOR DEFINITION OF REMAINING ARGUMENTS SEE
001666 SUBROUTINE 'BILIN' AND 'COEFF'
001667
001668 DOUBLE A(1),B(1),D(NRD,NCD),PUR,T
001669 DOUBLE CLK,CINK,CLKX,CLKX,ALPHA,SIGMA1,SIGMA2
001670 DOUBLE CLNPOLY(151),CLDPOLY(51),CZERO(50,2),C1POLE(50,2)
001671 COMMON/TXGNOUO/CLNPOLY,CLDPOLY,CZERO,CLNPOLY,
001672 * NCLZ,NCLP,CLK,CINK,CLKX
001673
001674 ORDER POLYS IN DECREASING POWERS AND ADD MISSING COEFFICIENTS
001675 CALL ORDER3(A,NA,KEFORM)
001676 NOZ=A(KEFORM)
001677 CALL COEFF(A,NA,NOZ,KEFORM)
001678 CALL ORDER3(B,NR,KEFORM)
001679 NOD=B(KEFORM)
001680 CALL COEFF(B,NR,NOD,KEFORM)
001681
001682 TRANSFER REAL POLY COEFFICIENTS TO ARRAYS CLNPOLY AND CLDPOLY
001683 K=1
001684 DO 100 I=1,NA,KEFORM
001685 CLNPOLY(K)=A(I)
001686 K=K+1
001687 DO 110 I=1,NR,KEFORM
001688 CLDPOLY(K)=B(I)
001689 K=K+1
001690 NCLZ=NOZ
001691 NCLP=NOD
001692
001693 TRANSFER POLES INTO ARRAY C1POLE
001694 DO 130 I=1,NPOLES
001695 C1POLE(I,1)=D(I,1)
001696 C1POLE(I,2)=D(I,2)
001697
001698 CALL SUBROUTINE BILIN TO PERFORM THE GENERAL BILINEAR
001699 TRANSFORMATION OF THE POLYNOMIALS AND ROOTS
001700 CALL BILIN(ALPHA,SIGMA1,SIGMA2,T,KEY,ZFORM,NBPOLE)
001701 N=NCLZ+1
001702 M=NCLP+1
001703
001704
001705

```

```

001706 C
001707 C TRANSFER NEW NUMINATOR POLY COEFFICIENTS INTO ARRAY 'A'
001708 C IN THE PROPER FORMAT AS SPECIFIED BY 'KFORM'
001709 PUR=FLOAT(NCLZ)
001710 K=1
001711 DO 200 I=1,N
001712 A(K)=CLNPOLY(I)
001713 A(K+1)=PUR
001714 IF(KFORM.EQ. 3) A(K+1)=0.0
001715 IF(KFORM.EQ. 3) A(K+2)=PUR
001716 PUR=PUR-1.0
001717 200 K=K+KFORM
001718 NA=NXKFORM
001719 C
001720 C TRANSFER NEW DENOMINATOR POLY COEFFICIENTS INTO ARRAY 'B'
001721 C IN THE PROPER FORMAT AS SPECIFIED BY 'KFORM'
001722 PUR=FLOAT(NCLP)
001723 K=1
001724 DO 300 I=1,M
001725 B(K)=CLDPOLY(I)
001726 B(K+1)=PUR
001727 IF(KFORM.EQ. 3) B(K+1)=0.0
001728 IF(KFORM.EQ. 3) B(K+2)=PUR
001729 PUR=PUR-1.0
001730 300 K=K+KFORM
001731 NB=NBKFORM
001732 C
001733 C TRANSFER NEW POLES INTO ARRAY 'D'
001734 KP=NCLP
001735 IF(ZFORM.EQ. 2HZR) KP=NDPOLE
001736 DO 500 I=1,KP
001737 D(I,1)=CLPOLE(I,1)
001738 D(I,2)=CLPOLE(I,2)
001739 IF(ZFORM.EQ. 2HZR) RETURN
001740 NDPOLE=NCLP
001741 C
001742 RETURN
001743 END
001744 C
001745 CDECK SZURROOT
001746 SUBROUTINE SZURROOT(RR,RI,NR,T,KEY)
001747 C
001748 C THIS SUBROUTINE PERFORMS THE ROOT CONVERSION BETWEEN
001749 C THE S, Z, U, AND U' PLANES.
001750 C THE DESIRED CONVERSION IS SELECTED BY THE INPUT
001751 C VARIABLE 'KEY':
001752 C 1 - CONVERSION FROM Z TO U PLANE
001753 C 2 - CONVERSION FROM U TO U' PLANE
001754 C 3 - CONVERSION FROM U' TO Z PLANE
001755 C 4 - CONVERSION FROM U' TO Z PLANE
001756 C 5 - CONVERSION FROM S TO Z PLANE
001757 C
001758 C
001759 C
001760 C

```

```

001765 6 - CONVERSION FROM Z TO S PLANE
001766 7 - CONVERSION FROM S TO U PLANE
001767 8 - CONVERSION FROM S TO U' PLANE
001768 9 - CONVERSION FROM U TO S PLANE
001769 10 - CONVERSION FROM U' TO S PLANE
001770
001771 RR - DOUBLE PRECISION ARRAY CONTAINING REAL PART OF ROOT
001772 RI - DOUBLE PRECISION ARRAY CONTAINING IMAG PART OF ROOT
001773 NR - NUMBER OF ROOTS
001774 T - SAMPLING PERIOD (SEC)
001775
001776 DOUBLE RR(1),RI(1),T,AP,DEN
001777 GO TO(1,2,2,3,4,5,5,6,6),KEY
001778
001779 1 CONTINUE
001780
001781 TRANSFORMATION EQUATIONS FOR Z TO U' AND Z TO U:
001782
001783 
$$Z = 1 - U/v + U \quad Z = (2/T)U'/(2/T) - U'$$

001784
001785 
$$XU = AP * [(XZ * XZ) + (YZ * YZ) - 1] * C(XZ + 1) * XZ * YZ * XZ$$

001786
001787 
$$YU = AP * [2 * YZ] / [C(XZ + 1) * XZ * 2 + YZ * XZ]$$

001788
001789 
$$AP = 1.0 \quad Z \text{ TO } U \quad AP = 2/T \quad Z \text{ TO } U'$$

001790
001791 AP=1.0
001792 IF(KEY .EQ. 2) AP=2.0/T
001793 NRR=NRR-1
001794 DO 10 I=1,NR
001795 10 CONTINUE
001796 DEN=(RR(I)+1.)*X(RR(I)+1.)*RI(I)*RI(I)
001797
001798 CHECK FOR Z = -1.0 WHICH TRANSFORMS TO U-U'-INFINITY
001799 IF(DABS(DEN) .GT. 1.D-12) GO TO 11
001800 NRR=NRR-1
001801 IF(I .GT. NRR) GO TO 10
001802 DO 12 J=1,NRR
001803 RR(J)=RR(J+1)
001804 RI(J)=RI(J+1)
001805 WRITE(7,15)
001806 15 FORMAT(1H ,*ROOT AT Z = -1.0 TRANSFORMS TO U-U'-INFINITY*,/)
001807 1 * THIS ROOT HAS BEEN DELETEDS)
001808 GO TO 13
001809
001810 11 RRI)=AP*(RR(I)*RR(I)+RI(I)*RI(I)-1.)))/DEN
001811 RI(I)=AP*(2.*RI(I))/DEN
001812 10 CONTINUE
001813 NRR=NRR
001814 RETURN
001815
001816 2 CONTINUE
001817
001818 TRANSFORMATION EQUATIONS FOR U TO Z AND U' TO Z:
001819
001820

```

```

001816      U = Z-1/2+1      U' = (2/T)*Z-1/2+1
001817
001818      XZ = [(1-XU)*YU]/[E(1-XU)**2+YU**2]
001819      YZ = [2.0*YU]/[E(1-XU)**2+YU**2]
001820
001821      FOR U' PLANE - XU = T/2*YU, YU = T/2*YU
001822
001823      NRR=NR
001824      DO 20 I=1,NR
001825      23 CONTINUE
001826      IF(KEY.EQ.4) RR(I)=(T/2.)*RR(I)
001827      IF(KEY.EQ.4) RI(I)=(T/2.)*RI(I)
001828      DEN=(1.-RR(I))*(1.-RR(I))+RI(I)*RI(I)
001829
001830      CHECK FOR U = 1.0 OR U' = 2/T WHICH TRANSFORM TO Z-INFINITY
001831      IF(DABS(DEN).GT.1.D-12) GO TO 21
001832      NRR=NRR-1
001833      IF(1.GT.NRR) GO TO 20
001834      DO 22 J=1,NRR
001835      22 RI(J)=RI(J+1)
001836      RR(J)=RR(J+1)
001837      WRITE(7,25)
001838      25 FORMAT(1H,ROOT AT U=1.0 OR U'=2/T TRANSFORMS,
001839      1 X TO Z-INFINITY,/,* THIS ROOT HAS BEEN DELETED*)
001840      GO TO 23
001841      21 RI(I)=(1.-RR(I))*RR(I)-RI(I)*RI(I)/DEN
001842      RI(I)=(2.0*RI(I))/DEN
001843      20 CONTINUE
001844      NR=NRR
001845      RETURN
001846
001847      3 CONTINUE
001848
001849      TRANSFORMATION EQUATIONS FOR S TO Z:
001850
001851      XZ = EXP(XST)*COS(YST)
001852      YZ = EXP(XST)*SIN(YST)
001853      Z = EXP(SST)
001854
001855      DO 30 I=1,NR
001856      DEN=EXP(RR(I)*ST)
001857      RI(I)=DEN*DCOS(RI(I)*ST)
001858      RR(I)=DEN*DSIN(RI(I)*ST)
001859      30 RI(I)=DEN*DSIN(RI(I)*ST)
001860      RETURN
001861
001862      4 CONTINUE
001863
001864      TRANSFORMATION EQUATIONS FOR Z TO S:
001865
001866      XS = (1/2*ST)*LN(XZ**2+YZ**2)
001867      YS = (1/T)*ARCTAN(YZ/XZ)
001868
001869
001870

```

```

001871
001872
001873
001874
001875
001876
001877
001878
001879
001880
001881
001882
001883
001884
001885
001886
001887
001888
001889
001890
001891
001892
001893
001894
001895
001896
001897
001898
001899
001900
001901
001902
001903
001904
001905
001906
001907
001908
001909
001910
001911
001912
001913
001914
001915
001916
001917
001918
001919
001920
001921
001922
001923
001924
001925

      Z = EXP(SST)

      DO 40 I=1,NR
      DEN=((1./T)*XDATA2(RI(I),RR(I))+RI(I)*RI(I))
      RI(I)=(1./T)*XDATA2(RI(I),RR(I))
40  RI(I)=DEN
      RETURN

      5  CONTINUE

      TRANSFORMATION EQUATIONS FOR S TO U AND S TO U':

      DENOMINATOR = EXP(XSST)+EXP(-XSST)+2*COS(VSST)
      XU = AP*EXP(XSST)-EXP(-XSST)/DENOMINATOR
      YU = AP*(2.*SIN(VSST)/DENOMINATOR

      AP=1.0
      IF(KEY.EQ.8) AP=2.0/T
      DO 50 I=1,NR
      DEN=EXP(RR(I)*ST)+DEXP(-RR(I)*ST)+2.*CDS(RI(I)*ST)
      RR(I)=AP*(EXP(RR(I)*ST)-DEXP(-RR(I)*ST))/DEN
50  RI(I)=AP*(2.*SIN(RI(I)*ST))/DEN
      RETURN

      6  CONTINUE

      TRANSFORMATION EQUATIONS FOR U TO S AND U' TO S:

      XS = (1/2*ST)*LNC(1+XU)*X2+YU*X2/[(1-XU)*X2+YU*X2]
      VS = (1/T)*SARCTAN(2*YU/[1-XU*XU-YU*YU])

      FOR U' PLANE - XU=(T/2)*XU, YU=(T/2)*YU

      DO 60 I=1,NR
      IF(KEY.EQ.10) RR(I)=(T/2)*RR(I)
      IF(KEY.EQ.10) RI(I)=(T/2)*RI(I)
      DEN=((1.+RR(I))*X(1.+RR(I))+RI(I)*RI(I))
      DEN=DEN/((1.-RR(I))*X(1.-RR(I))+RI(I)*RI(I))
      DEN=(1./T)*XDATA2(DEN)
      RI(I)=(1./T)*XDATA2(2.*RI(I)),(1.-RR(I)*RR(I)-RI(I)*RI(I))
60  RR(I)=DEN
      RETURN
      END

      CHECK COMPOLY
      SUBROUTINE COMPOLY(A,B,RR,RI,NR)

      THIS SUBROUTINE FORMS THE POLYNOMIAL FROM A SET OF ROOTS.
      BOTH REAL AND IMAGINARY POLY COEFFICIENTS ARE CALCULATED.
      A = DOUBLE PRECISION ARRAY CONTAINING REAL COEFFS OF POLY

```

```

C C C C C
      B = DOUBLE PRECISION ARRAY CONTAINING IMAG COEFFS OF POLY
      RP = DOUBLE PRECISION ARRAY CONTAINING REAL PART OF ROOT
      RI = DOUBLE PRECISION array CONTAINING IMAG PART OF ROOT
      NP = NUMBER OF ROOTS

      DOUBLE A(I),B(I),DR(1),RI(1)
      NPO=NR+1
      DO 1 I=2,NRR
        A(I)=0.0
        B(I)=0.0
      1 B(I)=0.0
        A(I)=1.0
        B(I)=0.0
      DO 2 I=1,NR
        K=I+1
      DO 2 J=1,I
        A(K)=-A(K-1)*RR(I)+B(K-1)*RI(I)+A(K)
        B(K)=-B(K-1)*RR(I)-A(K-1)*RI(I)+B(K)
      2 K=K-1
      RETURN
      END

C C C C C
      CDECK ORDPOL
      SUBROUTINE ORDPOL(DBPOLE,IRPOL,ICOL,NCLPDB,NPOLES,MULPOLE,NM)
C C C C C
      THIS SUBROUTINE CHECKS FOR MULTIPLE POLES AND STORES THE
      MULTIPLICITY IN ARRAY "MULPOLE". THE EXTRA MULTIPLE
      POLES ARE DELETED AND ONLY A SINGLE COPY OF EACH POLE
      IS STORED IN array "DBPOLE".

      DBPOLE(IRPOL,ICOL) = DOUBLE PRECISION array CONTAINING POLES
      (REAL,IMAG)
      MULPOLE = SINGLE PRECISION array CONTAINING POLE
      MULTIPLICITY
      NCLPDB = NUMBER OF TOTAL POLES INPUTTED.
      NPOLES = NUMBER OF DISTINCT POLES OUTPUTTED.
      NM = NUMBER OF OCCUPIED ELEMENTS IN array "MULPOLE".
      ALSO, NUMBER OF DISTINCT POLES OUTPUTTED.
      IRPOL = ROW DIMENSION OF array "DBPOLE".
      ICOL = COLUMN DIMENSION OF array "DBPOLE".

      DOUBLE DBPOLE(IRPOL,ICOL),D1,D2
      DIMENSION MULPOLE(1)
      DO 510 I=1,NCLPDB
        MULPOLE(I)=1
      510 L=1
      NPOLES=NCLPDB
      DO 550 I=L,NPOLES
        M=I+1
      550 DO 590 I=L,NPOLES
      C CHECK FOR LAST POLE
      IF(M=NPOLES) 510,510,550
      510 CONTINUE
      DO 535 J=M,NPOLES

```

```

001926
001927
001928
001929
001930
001931
001932
001933
001934
001935
001936
001937
001938
001939
001940
001941
001942
001943
001944
001945
001946
001947
001948
001949
001950
001951
001952
001953
001954
001955
001956
001957
001958
001959
001960
001961
001962
001963
001964
001965
001966
001967
001968
001969
001970
001971
001972
001973
001974
001975
001976
001977
001978
001979
001980

```

```

C SET PERCENTAGE TOLERANCE USED TO CHECK POLES
  D1=DABS(DBPOLE(I,1))*0.0001
  D2=DABS(DBPOLE(I,2))*0.0001
C CHECK FOR ZERO IMAGINARY PART OF POLE BEING CHECKED
C AGAINST REMAINING POLES.
  IF(DBPOLE(I,2)) 519,520,519
C CHECK FOR ZERO REAL PART OF NEXT POLE TO BE CHECKED
  519 IF(DBPOLE(J,1)) 521,520,521
C SET DECIMAL TOLERANCE FOR REAL PART OF POLE IF IMAG
C PART OF POLE BEING CHECKED AGAINST REMAINING POLES
C IS ZERO, OR IF REAL PART OF NEXT POLE IS ZERO.
  520 D1=.00001
  521 CONTINUE
C CHECK FOR ZERO IMAGINARY PART OF POLE BEING CHECKED
C AGAINST REMAINING POLES
  IF(DBPOLE(I,2)) 522,523,522
C CHECK FOR ZERO IMAG PART OF NEXT POLE TO BE CHECKED
  522 IF(DBPOLE(J,2)) 524,523,524
C SET DECIMAL TOLERANCE FOR IMAG PART OF POLE IF IMAG
C PART OF POLE BEING CHECKED AGAINST REMAINING POLES
C IS ZERO, OR IF IMAG PART OF NEXT POLE IS ZERO.
  523 D2=.00001
  524 CONTINUE
C HUNT FOR EQUAL POLES THAT MEET THE 'D1' AND 'D2' TOLERANCE
  IF(DABS(DBPOLE(I,1))-DBPOLE(J,1))-D1) 525,525,535
  525 IF(DABS(DBPOLE(I,2))-DBPOLE(J,2))-D2) 526,526,560
C HUNT FOR POLES WHOSE REAL PARTS MEET THE 'D1' TOLERANCE
C AND HAVE AN IMAG PART WHICH IS .LE. 1.D-6 THE REAL
C PART.
  560 IF(DBPOLE(I,1)) 561,562,561
  562 IF(DABS(DBPOLE(J,2)) .LE. 1.D-6) GO TO 526
  GO TO 535
  561 IF(DABS(DBPOLE(J,2))/DBPOLE(I,1)) .LE. 1.D-6) GO TO 526
  GO TO 535
  526 CONTINUE
C EQUAL POLES FOUND, OVERWRITE POLE AND INCREMENT MULTIPLICITY
  NPOLES=NPOLES+1
  MULPOLE(I)=MULPOLE(I)+1
  DO 530 K=J,NPOLES
  MULPOLE(K)=MULPOLE(K)+1
  DBPOLE(K,1)=DBPOLE(I,1)
  DBPOLE(K,2)=DBPOLE(I,2)
  530 L=1
  IF(NPOLES-L-1) 550,555,555
  535 CONTINUE
  500 CONTINUE
  550 CONTINUE
  NM=NPOLES
  RETURN
  END
C-----
C CHECK CANROOT
C-----
C SUBROUTINE CANROOT(TOLR,TOLI,CZERO,NZ,CPOLE,MP)
C-----

```

001981
 001982
 001983
 001984
 001985
 001986
 001987
 001988
 001989
 001990
 001991
 001992
 001993
 001994
 001995
 001996
 001997
 001998
 001999
 002000
 002001
 002002
 002003
 002004
 002005
 002006
 002007
 002008
 002009
 002010
 002011
 002012
 002013
 002014
 002015
 002016
 002017
 002018
 002019
 002020
 002021
 002022
 002023
 002024
 002025
 002026
 002027
 002028
 002029
 002030
 002031
 002032
 002033
 002034
 002035

```

002036
002037
002038
002039
002040
002041
002042
002043
002044
002045
002046
002047
002048
002049
002050
002051
002052
002053
002054
002055
002056
002057
002058
002059
002060
002061
002062
002063
002064
002065
002066
002067
002068
002069
002070
002071
002072
002073
002074
002075
002076
002077
002078
002079
002080
002081
002082
002083
002084
002085
002086
002087
002088
002089
002090

C THIS SUBROUTINE CANCELS EQUAL ZEROS AND POLES ACCORDING TO A
C SPECIFIED TOLERANCE. SEPARATE TOLERANCES ARE PROVIDED FOR
C THE REAL PART OF EACH ROOT (TOLR) AND FOR THE IMAGINARY PART
C OF EACH ROOT (TOLI).
C
C TOLR = TOLERANCE PERCENTAGE FOR REAL PART OF ROOTS
C TOLI = TOLERANCE PERCENTAGE FOR IMAG PART OF ROOTS
C ZERO = ARRAY CONTAINING ZEROS (REAL,IMAG)
C POLE = ARRAY CONTAINING POLES (REAL,IMAG)
C NZ = NUMBER OF ZEROS
C NP = NUMBER OF POLES
C
C DOUBLE TOLR,TOLI,CZERO(50,2),CPOLE(50,2),D1,D2
C L=1
150 DO 100 I=L,NZ
110 I=I
JJ=J
D1=DABS(CZERO(I,1)*TOLR)
D2=DABS(CZERO(I,2)*TOLI)
IF(CZERO(I,1)) 510,511,510
510 IF(CPOLE(J,1)) 512,511,512
511 D1=D1-D1
512 IF(CZERO(I,2)) 513,514,513
513 IF(CPOLE(I,2)) 515,514,515
514 D2=D2-D2
515 IF(DABS(CZERO(I,1)-CPOLE(J,1))-D1) 516,516,100
516 IF(DABS(CZERO(I,2)-CPOLE(J,2))-D2) 517,517,100
517 CONTINUE
NZ=NZ-1
NP=NP-1
DO 200 K=1,NZ
CZERO(K,1)=CZERO(K+1,1)
CZERO(K,2)=CZERO(K+1,2)
200 DO 300 K=JJ,NP
CPOLE(K,1)=CPOLE(K+1,1)
CPOLE(K,2)=CPOLE(K+1,2)
300 L=1
IF(NZ-L) 400,150,150
100 CONTINUE
400 CONTINUE
RETURN
END
C-----
CDECK DERIV3
SUBROUTINE DERIV3(P,NP,A,NA,K)
C-----
C THIS SUBROUTINE TAKES THE DERIVATIVE OF A POLYNOMIAL.
C
C P = INPUT POLY
C NP = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'P'
C A = OUTPUT POLY ( DERIVATIVE OF POLY 'P' )
C NA = NUMBER OF OCCUPIED ELEMENTS IN ARRAY 'A'

```

```

C      K = FORMAT OF INPUT POLY
C      K = 2 ---- (REAL COEFF, Z-PUR)
C      K = 3 ---- (REAL COEFF, IMAG COEFF, Z-PUR)
C
C      DOUBLE P(1),A(1)
C      BRANCH ACCORDING TO INPUT POLY FORMAT
C      IF (K-2) 10,20,10
C      K=2 SECTION
C      MULTIPLY COEFFICIENTS BY POWERS OF Z
C      REDUCE POWER OF Z BY ONE
C      STORE DERIVATIVE IN ARRAY 'A'
C      P0 CONTINUE
C      DO 2 I=1,NP,K
C      A(I)=P(I+1)*P(I)
C      A(I+1)=P(I+1)-1.0
C      CHECK FOR NEGATIVE Z PUR AFTER DIFFERENTIATION
C      IF (A(I+1) .LT. 0.0) A(I+1)=0.0
C      2 CONTINUE
C      GO TO 30
C
C      K=3 SECTION
C      MULTIPLE BOTH REAL & IMAG COEFF BY
C      POWER OF Z.
C      REDUCE POWER OF Z BY ONE
C      STORE DERIVATIVE IN ARRAY 'A'
C      10 CONTINUE
C      DO 3 I=1,NP,K
C      A(I)=P(I+2)*P(I)
C      A(I+1)=P(I+2)*P(I+1)
C      A(I+2)=P(I+2)-1.0
C      CHECK FOR NEGATIVE Z PUR AFTER DIFFERENTIATION
C      IF (A(I+2) .LT. 0.0) A(I+2)=0.0
C      3 CONTINUE
C      30 CONTINUE
C      SET NUMBER OF 'A' ELEMENTS
C      NA=NP
C      IF (P(NP) .EQ. 0.0) NA=NP-K
C      RETURN
C      END
C
C      CHECK EVALU3
C      SUBROUTINE EVALU3(P,NP,R,U,ZF,K)
C
C      THIS SUBROUTINE EVALUATES A POLYNOMIAL FOR A COMPLEX
C      VALUE OF (Z = X + JY). IT CAN EVALUATE BOTH REAL
C      COEFFICIENT POLYS AND COMPLEX COEFFICIENT POLYS.
C
C      P = INPUT POLY TO BE EVALUATED
C      NP = NUMBER OF WORKING ELEMENTS IN P(NP)
C      R = POLE OR ROOT TO BE EVALUATED (R=X+JY)
C      U = VALUE OF POLY 'P' AT POLE 'R' (U=A+JB)
C      ZF = SCALE FACTOR IMPOSED ON LARGE VALUES OF 'U'
C           TO MAINTAIN NUMERICAL ACCURACY.
C      K = FORMAT OF INPUT POLY 'P'

```

```

002091
002092
002093
002094
002095
002096
002097
002098
002099
002100
002101
002102
002103
002104
002105
002106
002107
002108
002109
002110
002111
002112
002113
002114
002115
002116
002117
002118
002119
002120
002121
002122
002123
002124
002125
002126
002127
002128
002129
002130
002131
002132
002133
002134
002135
002136
002137
002138
002139
002140
002141
002142
002143
002144
002145

```

```

002146
002147
002148
002149
002150
002151
002152
002153
002154
002155
002156
002157
002158
002159
002160
002161
002162
002163
002164
002165
002166
002167
002168
002169
002170
002171
002172
002173
002174
002175
002176
002177
002178
002179
002180
002181
002182
002183
002184
002185
002186
002187
002188
002189
002190
002191
002192
002193
002194
002195
002196
002197
002198
002199
002200

      K = 2      ---- (REAL COEFF, Z-PUR)
      K = 3      ---- (REAL COEFF, IMAG COEFF, Z-PUR)

      THE FOLLOWING STEPS REPRESENT AN EXAMPLE OF THE
      ALGORITHM USED TO EVALUATE THE INPUT POLY *P*
      AT ROOT *R*:

      A3XS3 + A2XS2 + A1XS + A0      [ (NP-8) & (K-2) ]
      R = S - [R(1) + J R(2)]

      I=1
      [A3R(1) + J A3R(2)]XS2 + A2XS2 + A1XS + A0
      [(RP) + J (CP)]XS2 + A2XS2 + A1XS + A0
      [(RP+A2) + J (CP)]XS2 + A1XS + A0
      [U(1) + J U(2)]XS2 + A1XS + A0
      I=3
      [U + R]XS + A1XS + A0
      [(RP) + J (CP)]XS + A1XS + A0
      [(RP+A1) + J (CP)]XS + A0
      [U(1) + J U(2)]XS + A0
      I=5
      [U + R]XS + A1XS + A0
      [(RP) + J (CP)]XS + A1XS + A0
      [(RP+A1) + J (CP)]XS + A0
      [U(1) + J U(2)]XS + A0
      I=7
      RETURN AS RESULT OF IF(P(N)) 539,4,539

      A3XS3 + A2XS2 + A1XS      [ (NP-6) & (K-2) ]
      R = S - [R(1) + J R(2)]

      I=1
      [A3R(1) + J A3R(2)]XS2 + A2XS2 + A1XS
      [(RP) + J (CP)]XS2 + A2XS2 + A1XS
      [(RP+A2) + J (CP)]XS2 + A1XS
      [U(1) + J U(2)]XS2 + A1XS
      I=3
      [U + R]XS + A1XS
      [(RP) + J (CP)]XS + A1XS
      [(RP+A1) + J (CP)]XS
      [U(1) + J U(2)]XS
      I=5
      [U + R]XS + A1XS
      [(RP) + J (CP)]XS
      [(RP+A1) + J (CP)]XS
      [U(1) + J U(2)]XS
      I=7
      RETURN (NP-6)

      DOUBLE R(1),EXP,RP,CP,ZF,P(1),U(1)
      ZF=1.0
      ORDER POLY *P* IN DESCENDING POWERS OF Z
      CALL ORDER3(P,NP,K)
      BRANCH ACCORDING TO POLY *P* FORMAT
      IF (K-2) 10,20,10
      K=2 SECTION
      SET FIRST 2 UNUSED ELEMENTS IN *P* TO ZERO

```

```

002201 C INITIALIZE EXPONENT VARIABLE 'EXP' TO FIRST
002202 C Z POWER IN POLY 'P' AND SET EVALUATION
002203 C VARIABLE 'U' TO FIRST COEFFICIENT OF 'P'
002204 20 CONTINUE
002205 P(NP+1)=0.0
002206 P(NP+2)=0.0
002207 EXP=P(2)
002208 U(1)=P(1)
002209 U(2)=0.0
002210 GO TO 25
002211 K=3 SECTION
002212 C SET FIRST 3 UNUSED ELEMENTS IN 'P' TO ZERO
002213 C INITIALIZE EXPONENT VARIABLE 'EXP' TO FIRST
002214 C Z POWER IN POLY 'P' AND SET EVALUATION
002215 C VARIABLE 'U' TO FIRST COEFFICIENT OF 'P'
002216 10 CONTINUE
002217 P(NP+1)=0.0
002218 P(NP+2)=0.0
002219 P(NP+3)=0.0
002220 EXP=P(3)
002221 U(1)=P(1)
002222 U(2)=P(2)
002223 25 CONTINUE
002224 C CHECK FOR 0TH-ORDER POLY 'P' --- RETURN
002225 IF(EXP) 538,4,538
002226 538 CONTINUE
002227 DO 3 I=1,NP,K
002228 C MASTER DO LOOP FOR EVALUATING POLY 'P'
002229 C SET 'N' TO LOCATION OF Z PUR IN POLY 'P'
002230 N=I*K-1
002231 C CHECK FOR ZERO Z PUR IN POLY 'P' -- RETURN
002232 IF LAST TERM IN POLY 'P' HAS ZERO Z-PUR,
002233 LAST COEFFICIENT ALREADY ADDED IN LAST LOOP
002234 THROUGH ROUTINE BY U(1) & U(2) CODE BELOW
002235 C IF LAST TERM NOT ZERO Z-PUR, ONE MORE
002236 MULTIPLICATION BY ROOT 'R' REQUIRED AND THUS
002237 ONE MORE LOOP REQUIRED THROUGH ROUTINE
002238 IF(P(N)) 539,4,539
002239 539 CONTINUE
002240 C MULTIPLY COMPLEX COEFF STORED IN U(1),U(2)
002241 C BY COMPLEX POLE IN R(1),R(2)
002242 C STORE RESULTS IN VARIABLES 'RP' AND 'CP'
002243 1 RP=U(1)*R(1) - U(2)*R(2)
002244 CP=U(1)*R(2) + U(2)*R(1)
002245 C CHECK FOR EXCESSIVELY LARGE VALUES OF 'RP'
002246 IF(DABS(RP)-1.D20) 11,540,540
002247 540 CONTINUE
002248 C SCALE DOWN LARGE VALUES OF EVALUATION RESULTS
002249 C IN VARIABLES 'RP' AND 'CP' BY 1.E10 AND STORE
002250 C TOTAL SCALE FACTOR IN 'ZF'
002251 ZF=ZF*1.D10
002252 RP=RP/1.D10
002253 CP=CP/1.D10
002254 11 CONTINUE
002255 C REDUCE 'Z' POWER VARIABLE BY ONE

```

```

EXP-EXP-1.0
C SET 'NN' TO NEXT POWER OF 'Z' IN POLY 'P'
  NN-I+(2*K)-1
C CHECK FOR MISSING POWER OF 'Z' IN POLY 'P'
  IF (EXP-P(NN)) 2,2,541
C BRANCH HERE FOR MISSING Z-POWER
C NOTE: NEXT COEFF NOT ADDED TO RUNNING SUM
C VARIABLE U(1) & J U(2)
  541 CONTINUE
    U(1)-RP
    U(2)-CP
    GO TO 1
C BRANCH HERE FOR CONSECUTIVE 2 POWERS
C NEXT COEFFICIENT ADDED TO RUNNING
C SUM VARIABLE U(1) & J U(2)
  2 U(1)-RP+P(I+K)/ZF
C BRANCH ACCORDING TO POLY 'P' FORMAT
  IF (K-2) 700,5,700
C ADD IMAG PART OF NEXT COEFF TO RUNNING
C SUM VARIABLE U(2) FOR K=3
  700 U(2)-CP+P(I+K)/ZF
  GO TO 3
C SINCE NO IMAGINARY PART OF COEFF FOR K=2
C SIMPLE TRANSFER PRIOR RUNNING SUM TO U(2)
  5 U(2)-CP
  3 CONTINUE
  4 RETURN
  END
C-----
CDECK ORDER3
SUBROUTINE ORDER3(P,N,K)
C-----
C THIS SUBROUTINE ORDERS A POLYNOMIAL IN DESCENDING POWERS
C
C P = INPUT POLY
C N = NUMBER OCCUPIED ELEMENTS IN 'P'
C K = FORMAT OF INPUT POLY
C K = 2 ---- (REAL COEFF, PUR)
C K = 3 ---- (REAL COEFF, IMAG COEFF, PUR)
C
C DOUBLE P(1),TEMP
M=N-K
IF (M)5,5,1
1 REVERS=1
DO 4 I=K,M,K
  J=I+K
  IF (P(I)-P(J))2,4,4
2 REVERS=0
DO 3 L=1,K
  LJ=J-L+1
  LI=I-L+1
  TEMP=P(LJ)
  P(LJ)=P(LI)
  P(LI)=TEMP

```

```

002256
002257
002258
002259
002260
002261
002262
002263
002264
002265
002266
002267
002268
002269
002270
002271
002272
002273
002274
002275
002276
002277
002278
002279
002280
002281
002282
002283
002284
002285
002286
002287
002288
002289
002290
002291
002292
002293
002294
002295
002296
002297
002298
002299
002300
002301
002302
002303
002304
002305
002306
002307
002308
002309
002310

```

```

002311
002312
002313
002314
002315
002316
002317
002318
002319
002320
002321
002322
002323
002324
002325
002326
002327
002328
002329
002330
002331
002332
002333
002334
002335
002336
002337
002338
002339
002340
002341
002342
002343
002344
002345
002346
002347
002348
002349
002350
002351
002352
002353
002354
002355
002356
002357
002358
002359
002360
002361
002362
002363
002364
002365

3 CONTINUE
4 CONTINUE
  IF (REVERS) 5,1,5
5 RETURN
  END

C-----
CDECK DIVI
  SUBROUTINE DIVI(A,B,C,D,X,Y)
C-----
C  THIS ROUTINE DIVIDES TWO COMPLEX NUMBERS
C  X+JY = (A+JB)/(C+JD)
C
  DOUBLE A,B,C,D,X,Y
  X=(A*C+B*D)/(C*C+D*D)
  Y=(B*C-A*D)/(C*C+D*D)
  RETURN
  END

C-----
CDECK MULT
  SUBROUTINE MULT(A,B,C,D,X,Y)
C-----
C  THIS ROUTINE MULTIPLIES TWO COMPLEX NUMBERS
C  X+JY = (A+JB)*(C+JD)
C
  DOUBLE A,B,C,D,X,Y
  X=A*C-B*D
  Y=B*C+A*D
  RETURN
  END

C-----
CDECK MULTIP
  SUBROUTINE MULTIP(C1,NT1,C2,NT2,C3,NT3,N)
C-----
C  THIS ROUTINE MULTIPLIES TWO POLYNOMIALS
C  C1 = INPUT POLY
C  NT1 = NUMBER OCCUPIED ELEMENTS IN ARRAY "C1"
C  C2 = INPUT POLY
C  NT2 = NUMBER OCCUPIED ELEMENTS IN ARRAY "C2"
C  C3 = C1*C2
C  NT3 = NUMBER OCCUPIED ELEMENTS IN ARRAY "C3"
C  N = FORMAT OF POLYNOMIALS
C  N = 2 ---- (REAL COEFF, PUR)
C  N = 3 ---- (REAL COEFF, IMAG COEFF, PUR)
C
  DOUBLE C1(1),C2(1),C3(1)
  K=1-N
  DO 110 I=1,NT1,N
  DO 100 J=1,NT2,N
  K=K+N
  IF (N=3) 529,90,529
529 CONTINUE
  C3(K)=C1(I)*C2(J)
  C3(K+1)=C1(I+1)+C2(J+1)
  GO TO 100

```



```

JJ-I+M
DO 110 J=JJ,NT,M
K=M-I-1
L=M-J-1
IF(C(K)-C(L))110,100,110
100 C(I)=C(I)+C(J)
C(J)=0
IF(M-2)534,110,534
534 CONTINUE
C(I+1)=C(I+1)+C(J+1)
C(J+1)=0
110 CONTINUE
111
115 DO 140 I=I-1,NT,M
IF(C(I))140,535,140
535 CONTINUE
536 IF(M-2)536,120,536
536 CONTINUE
IF(C(I+1))140,537,140
537 CONTINUE
120 JJ-I+M
DO 130 J=JJ,NT
K=J-M
130 C(K)=C(J)
NT=NT-M
111
IF(I+M-I-NT)115,115,145
140 CONTINUE
145 IF(NT)155,150,155
150 NT=M
C(1)=0
C(2)=0
C(3)=0
155 RETURN
END
C-----
CDECK ROOTS
C-----
SUBROUTINE ROOTS(A,B,NN,RR,RI)
C-----
C THIS ROUTINE FINDS THE ROOTS OF POLYNOMIAL
C A - REAL COEFF OF POLY IN DESCENDING ORDER
C B - IMAG COEFF OF POLY IN DESCENDING ORDER
C RR - REAL PART OF ROOTS
C RI - IMAG PART OF ROOTS
C NN - ORDER OF POLYNOMIAL
C LA(1)+JB(1)JSxx(NN) + LA(2)+JB(2)JSxx(NN-1) + ...
C
C DOUBLE A(1),B(1),RR(1),RI(1)
C DOUBLE PRECISION C,D,C1,C2,D1,D2,X,Y,F,G,F1,F2,G1,G2,FH,
C 1FK,FH
C L=1
C N=NN
C FH=1.
C NPLUS=N+1

```

```

002421
002422
002423
002424
002425
002426
002427
002428
002429
002430
002431
002432
002433
002434
002435
002436
002437
002438
002439
002440
002441
002442
002443
002444
002445
002446
002447
002448
002449
002450
002451
002452
002453
002454
002455
002456
002457
002458
002459
002460
002461
002462
002463
002464
002465
002466
002467
002468
002469
002470
002471
002472
002473
002474
002475

```

```

DO 265 I=1,NPLUS
  A1=A(I)
  B1=B(I)
  IF (ABS(A1).GT..00000001.OR.ABS(B1).GT..00000001)GO TO 266
DO 264 J=1,NPLUS
  A(J)=A(J+1)
  B(J)=B(J+1)
  N=N-1
  IF (N.EQ.0)RR(1)=0.
  IF (N.EQ.0)RI(1)=0.
  IF (N.EQ.0)RETURN
265 CONTINUE
266 CONTINUE
NN=N
DO 303 I=1,NN
  RR(I)=0.
  RI(I)=0.
112 N1=N+1
  IF (N.EQ.1)GO TO 103
  D1=DMAX1(DABS(A(N+1)),DABS(B(N+1)))
  IF (D1.EQ.0.)GO TO 111
  D2=DMAX1(DABS(A),DABS(B))
  FK=N
  FM=D2*(1./FK)/D1*(1./FK)
  K=N
DO 3 I=1,N
  A(I)=A(I)/(D1*FM*(K))
  B(I)=B(I)/(D1*FM*(K))
3 K=K-1
  A(N1)=A(N1)/D1
  B(N1)=B(N1)/D1
7 X=.9876532
  Y=.9654312
  LL=1
  IL=1
8 C=A
  D=B
  C1=A
  C2=B
DO 5 I=1,M
  F=X*C-V*D+A(I+1)
  G=X*D+V*C+B(I+1)
  IF (I.EQ.M)GO TO 5
  F1=X*C1-V*C2+F
  F2=X*C2+V*C1+G
  C=F
  D=G
  C1=F1
  C2=F2
5 D=F1*X2+F2*X2
  FM=-(F*F1+G*F2)/D
  FK=-(G*F1-F*F2)/D
12 X=X+FK
  Y=Y+FK
  IF ((X*X2+Y*X2).EQ.((X+FK)*X2+(Y+FK)*X2))GO TO 22

```

```

002476
002477
002478
002479
002480
002481
002482
002483
002484
002485
002486
002487
002488
002489
002490
002491
002492
002493
002494
002495
002496
002497
002498
002499
002500
002501
002502
002503
002504
002505
002506
002507
002508
002509
002510
002511
002512
002513
002514
002515
002516
002517
002518
002519
002520
002521
002522
002523
002524
002525
002526
002527
002528
002529
002530

```

```

002531 IF(X.EQ.0..OR.Y.EQ.0.)GO TO 22
002532 A1=FM/X
002533 A2=FK/Y
002534 IF(ABS(A1).GT.1.E-4.OR. ABS(A2).GT.1.E-4)GO TO 21
002535 GO TO(211,21),IL
002536
002537 LL=194
002538 IL=2
002539
002540 21 LL=LL+1
002541 IF(LL.GT.200)GO TO 22
002542 GO TO 8
002543
002544 22 RR(L)=X/FM
002545 RI(L)=Y/FM
002546 DO 100 I=1,M
002547 A(I+1)=XA(I)-YXB(I)+A(I+1)
002548 B(I+1)=XB(I)+YXA(I)+B(I+1)
002549
002550 100 M=M+1
002551 L=L+1
002552 IF(N.GT.1)GO TO 7
002553 D=A222+B222
002554 RR(L)=(-XA(2)-BXB(2))/(D*FM)
002555 RI(L)=(-XB(2)+BXA(2))/(D*FM)
002556 DO 104 I=1,N
002557 X=RR(I)*2
002558 Y=RI(I)*2
002559 D=X*Y
002560 IF(D.EQ.X)RI(I)=0.
002561 IF(D.EQ.Y)RR(I)=0.
002562
002563 104 CONTINUE
002564
002565 111 RR(L)=0.
002566 RI(L)=0.
002567 M=M+1
002568 L=L+1
002569 GO TO 112
002570 END
002571
002572 C-----
002573 CDECK DOLOOP
002574 SUBROUTINE DOLOOP(C,NC,D,ND)
002575 C-----
002576 C
002577 C THIS ROUTINE IMPLEMENTS A STANDARD DO LOOP
002578 C TO TRANSFER ONE ARRAY INTO ANOTHER ARRAY
002579 C - INPUT ARRAY
002580 NC = NUMBER OCCUPIED ELEMENTS IN 'C'
002581 D = OUTPUT ARRAY
002582 ND = NC
002583
002584 DOUBLE C(1),D(1)
002585 ND=NC
002586 DO 1 I=1,NC
002587 1 D(I)=C(I)
002588 RETURN
002589 END
002590 C-----
002591 C

```

```

CDECK POLE
SUBROUTINE POLE(REAL,RIMAG)
C-----
C THIS SUBROUTINE CALCULATES A LOW-RATE POLE [ Z=EXP(S*T) ]
C FROM A GIVEN HIGH-RATE POLE [ Z=EXP(S*T/N) ].
C THE NEW POLE LOCATION IS SIMPLY THE ORIGINAL POLE
C LOCATION IN THE Z-DOMAIN TAKEN TO A POWER EQUAL
C TO THE RATIO OF THE LOW-RATE TO HIGH-RATE SAMPLING
C INTERVALS, XNT = (T) / (T/N).
C
C [ Z=EXP(S*T) ] [ Z=EXP(S*T/N) ]
C [ NEW POLE ] = [ ORIGINAL POLE ]**XNT
C
C REAL = REAL PART OF HIGH-RATE POLE ON INPUT
C RIMAG = REAL PART OF LOW-RATE POLE ON OUTPUT
C XNT = (T) / (T/N)
C
C DOUBLE REAL,RIMAG,REAL1,REAL2,RIMAG1,C,ANG,X,Y
C COMMON/PAR/TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK
C EXIT ROUTINE IF XNT=1
C IF (XNT.EQ.1.0) RETURN
C TRANSFER INTEGER VALUE OF XNT INTO NT
C NT=XNT
C CHECK FOR INTEGER RATIO XNT
C IF (XNT-NT) 50,100,50
C POLE TXFORM SECTION FOR NON-INTEGER XNT
C 50 CONTINUE
C CHECK FOR SPECIAL CASE OF REAL NEGATIVE POLE
C THIS CONDITION EXISTS ONLY WHEN IMAG PART OF
C COMPLEX S-PLANE POLE IS AT EXACTLY ONE-HALF
C THE SAMPLING FREQUENCY  $U=2\pi/T$ .
C FOR THIS CONDITION, THE IMAG PART OF THE
C Z-PLANE POLE IS EXACTLY ZERO AND THUS THE
C COMPLEX S-PLANE POLE BECOMES A NEGATIVE REAL
C POLE IN THE Z-PLANE.
C THE GENERAL TRANSFORMATION FOR THIS SPECIAL
C CASE IS BASED ON THE FOLLOWING:
C  $Z/Z - (\exp(-XT) \pm \exp(-JYT))$ 
C WHERE,  $(S + X + JY) = S$ -PLANE POLE
C  $Y = 1/T \pm \pi$ 
C  $X = -(1/T) \pm \ln(\text{REAL}/\cos(YT))$ 
C WITH,  $\text{REAL} = \exp(-XT) \pm \cos(YT)$ 
C RIMAG =  $-\exp(-XT) \pm \sin(YT)$ 
C  $\cos(YT) = \cos(\pi) = -1$ 
C NOTE:  $\exp(-XT)$  IS ALWAYS NEGATIVE
C
C IF (REAL.LT.0.0 .AND. RIMAG.EQ.0.0) GO TO 51
C GO TO 52
C 51 Y = (1.0/TINT) * (3.141592654)
C X = (-1.0/TINT) * DLOG(-REAL)

```

```

002641 REAL = DEXP(-X*TEXT) * DCOS(Y*TEXT)
002642 RIMAG = -DEXP(-X*TEXT) * DSIN(Y*TEXT)
002643 GO TO 200
002644 52 CONTINUE
002645 C CALCULATE MAGNITUDE OF LOW-RATE POLE
002646 IF (REAL.EQ. 0.0 .AND. RIMAG.EQ. 0.0) GO TO 200
002647 C-(REAL*REAL+RIMAG*RIMAG)**(XNT*0.5)
002648 CALCULATE PHASE ANGLE OF LOW-RATE POLE
002649 ANG=(DATAN2(RIMAG,REAL))/XNT
002650 CALCULATE REAL & IMAG PART OF LOW-RATE POLE
002651 REAL=CDCCOS(ANG)
002652 RIMAG=CDCSIN(ANG)
002653 GO TO 200
002654 C POLE TXFORM SECTION FOR INTEGER XNT
002655 100 CONTINUE
002656 REAL1=REAL
002657 RIMAG1=RIMAG
002658 C MULTIPLY HIGH-RATE POLE Z=EXP(S*TX/N)
002659 C TIMES ITSELF XNT TIMES
002660 NIT=NT-1
002661 DO 300 K=1,NTT
002662 REAL2=REAL
002663 REAL=REAL*REAL1 - RIMAG*RIMAG1
002664 RIMAG=REAL2*RIMAG1 + REAL1*RIMAG
002665 300 CONTINUE
002666 200 CONTINUE
002667 RETURN
002668 END
002669 C-----
002670 CDECK UPPOLE
002671 SUBROUTINE UPPOLE(UREAL, UIMAG, YPLUSR, YPLUSI, YN1R, YN1I,
002672 * YN2R, YN2I, YP1R, YP1I, YP2R, YP2I)
002673 C-----
002674 C THIS SUBROUTINE CALCULATES THE U OR U' LOW-RATE POLE
002675 FROM THE HIGH-RATE POLE USING THE FOLLOWING:
002676 U = -A * [(1-Y*XXNT)/(1+Y*XXNT)]
002677 WHERE
002678 Y*XXNT = [(AP+UP)/(AP-UP)]*XXNT
002679 U = LOW-RATE U OR U' VARIABLE [Z=EXP(S*TX)]
002680 UP = HIGH-RATE U OR U' VARIABLE [Z=EXP(S*TX)]
002681 U TRANSFORM:
002682 A = 1.0
002683 AP = 1.0
002684 U = U
002685 UP = UP
002686 U' TRANSFORM:
002687 A = 2/TEXT
002688
002689
002690
002691
002692
002693
002694
002695

```

```

002696      AP = 2./TINT
002697      U = U'
002698      UP = UP'
002699
002700      TEXT = LOW-RATE SAMPLING PERIOD (SEC)
002701      TEXT = HIGH-RATE SAMPLING PERIOD (SEC)
002702      XNT = TEXT/TINT
002703
002704      UP = NUMERICAL VALUE OF HIGH-RATE M OR M'
002705      POLE TO BE CONVERTED TO LOW-RATE POLE.
002706
002707      THE PROCEDURE IS TO SIMPLY INSERT THE VALUE OF THE
002708      HIGH-RATE POLE UP = UREAL, UIMAG INTO THE ABOVE
002709      EXPRESSION WITH THE APPROPRIATE VALUES OF A AND AP
002710      TO DETERMINE THE VALUE OF THE LOW-RATE POLE.
002711
002712      ALSO, THE FOLLOWING TERMS ARE OUTPUTTED FOR USE IN
002713      THE RESIDUE CALCULATIONS.
002714
002715      VPLUSR, VPLUSI = (1 + VXX(XNT))
002716      VNIR, VNII = VXX(XNT-1)
002717      VNBR, VNBI = VXX(XNT-2)
002718      VPIR, VPII = V'
002719      VPER, VPEI = V''
002720
002721      DOUBLE UREAL, UIMAG, VPLUSR, VPLUSI, REAL, REALI, RIMAG, RIMAGI, REAL2
002722      DOUBLE VNIR, VNII, VNBR, VNBI, VPIR, VPII, VPER, VPEI, DBCLK, RIMAG2
002723      DOUBLE DXA, DXAP, ARD, ARN, TINT, TEXT, XNT, YMINR, YMINI, C, ANG
002724      COMMON/PAR/TINT, TEXT, XNT, TXFORM, KZERO, NCLDDB, NCLPDB, DBCLK
002725
002726      KXNT=0
002727
002728      C SET VALUE OF 'A'-DXA AND 'AP'-DXAP FOR U OR U' TRANSFORM
002729      DXA=1.0
002730      DXAP=1.0
002731      IF(TXFORM.EQ. 2HUP) DXA=2.0/TEXT
002732      IF(TXFORM.EQ. 2HUP) DXAP=2.0/TINT
002733      C TRANSFER INTEGER VALUE OF XNT INTO NT
002734      NT=XNT
002735      C CHECK FOR INTEGER RATIO XNT
002736      IF((XNT-NT).EQ.0.0) KXNT=10
002737      C CALCULATE Y(REAL, RIMAG) = I(AP+UP)/(AP-UP)J
002738      ARD = DXAP + UREAL
002739      ARN = DXAP - UREAL
002740
002741      C CHECK FOR POLE AT U=1.0 OR U'=-2/T
002742      C THESE POLES TRANSFORM BACK INTO
002743      C THE Z-PLANE AT Z-INFINITY
002744
002745      IF(ARN.EQ.0.0.AND. UIMAG.EQ.0.0) GO TO 500
002746
002747      C CALL DIVI(ARD, UIMAG, ARN, -UIMAG, REAL, RIMAG)
002748
002749      C CALCULATE THE FOLLOWING TERMS USED IN THE NUMERATOR
002750      C EXPRESSION FOR THE RESIDUES:

```

```

002696
002697
002698
002699
002700
002701
002702
002703
002704
002705
002706
002707
002708
002709
002710
002711
002712
002713
002714
002715
002716
002717
002718
002719
002720
002721
002722
002723
002724
002725
002726
002727
002728
002729
002730
002731
002732
002733
002734
002735
002736
002737
002738
002739
002740
002741
002742
002743
002744
002745
002746
002747
002748
002749
002750

```

```

002751      Y = ((AP+UP)/(AP-UP))
002752      YX(XNT-1) = YNIR,YNII
002753      YX(XNT-2) = YN2R,YN2I
002754      Y' = YPIR,YP1I
002755      Y'' = YP2R,YP2I
002756      Y' = (2.0*AP)/(AP-UP)X2J
002757      Y'' = (4.0*AP)/(AP-UP)X3J
002758      WHERE
002759      Y' = (2.0*AP)/(AP-UP)X2J
002760      Y'' = (4.0*AP)/(AP-UP)X3J
002761      CALL MULT(ARN,-UIMAG,ARN,-UIMAG,YPIR,YP1I)
002762      YP2R=YP1R
002763      YP2I=YP1I
002764      REAL2= 2.0*DXAP
002765      RIMAG2=0.0
002766      CALL MULT(REAL2,RIMAG2,YP2R,YP2I,YPIR,YP1I)
002767      CALL MULT(ARN,-UIMAG,YP2R,YP2I,REAL1,RIMAG1)
002768      REAL2= 4.0*DXAP
002769      RIMAG2=0.0
002770      CALL MULT(REAL2,RIMAG2,REAL1,RIMAG1,YP2R,YP2I)
002771      YNIR=0.0
002772      YNII=0.0
002773      YN2R=0.0
002774      YN2I=0.0
002775      IF(REAL.EQ.0.0.AND.RIMAG.EQ.0.0.AND.XNT.EQ.1.0) YNIR=1.0
002776      IF(REAL.EQ.0.0.AND.RIMAG.EQ.0.0.AND.XNT.EQ.2.0) YN2R=1.0
002777      IF(REAL.EQ.0.0.AND.RIMAG.EQ.0.0) GO TO 100
002778      C=(REAL*REAL+RIMAG*RIMAG)X((XNT-1.0)*0.5)
002779      ANG=(DATAN2(RIMAG,REAL))X(XNT-1.0)
002780      YNIR=C*DCOS(ANG)
002781      YNII=C*DSIN(ANG)
002782      C=(REAL*REAL+RIMAG*RIMAG)X((XNT-2.0)*0.5)
002783      ANG=(DATAN2(RIMAG,REAL))X(XNT-2.0)
002784      YN2R=C*DCOS(ANG)
002785      YN2I=C*DSIN(ANG)
002786      IF(KXNT.GT. 5) GO TO 200
002787      POLE TRANSFORM SECTION FOR NON-INTEGGER VALUES OF XNT
002788      CALCULATE THE YXNT MAGNITUDE
002789      C=(REAL*REAL+RIMAG*RIMAG)X(XNT*0.5)
002790      CALCULATE THE YXNT PHASE ANGLE
002791      ANG=(DATAN2(RIMAG,REAL))XNT
002792      CALCULATE REAL AND IMAGINARY PART OF YXNT=REAL,RIMAG
002793      REAL=C*DCOS(ANG)
002794      RIMAG=C*DSIN(ANG)
002795      GO TO 300
002796      POLE TRANSFORM SECTION FOR INTEGER VALUES OF XNT.

```

```

002806 C MULTIPLY 'V' TIMES ITSELF XNT TIMES - V*XXNT
002807 200 CONTINUE
002808 IF(XNT.EQ. 1.0) GO TO 300
002809 REALI=REAL
002810 RIMAGI=RIMAG
002811 NIT=NT-1
002812 DO 400 K=1,NTT
002813 REAL2=REAL
002814 REAL=REAL*REALI-RIMAG*RIMAGI
002815 RIMAG=REAL*RIMAGI+REALI*RIMAG
002816 400 CONTINUE
002817 300 CONTINUE
002818 C CALCULATE YMINR,YMINI=1-V*XXNT AND YPLUSR,YPLUSI=1+V*XXNT
002819 100 CONTINUE
002820 YMINR=1-REAL
002821 YMINI=-RIMAG
002822 YPLUSR=1+REAL
002823 YPLUSI=RIMAG
002824 IF(XNT.EQ. 1.0) RETURN
002825 C CALCULATE THE LOW-RATE POLE U = -AXE((1-V*XXNT)/(1+V*XXNT))
002826 YMINR = -DXAYMINI
002827 CALL DIVI(YMINR,YMINI,YPLUSR,YPLUSI,UREAL,URIMAG)
002828 GO TO 50
002829 500 WRITE(7,1)
002830 1 F=FORMAT('H, *POLE AT U=1.0 OR U'=2/T TRANSFORME',
002831 * * BACK INTO Z-PLANE AT Z-INFINITY',/, * EXIT PROGRAM)
002832 CALL DUMP
002833 50 RETURN
002834 END
002835
002836
002837
002838
002839
002840
002841
002842
002843
002844
002845
002846
002847
002848
002849
002850
002851
002852
002853
002854
002855
002856
002857
002858
002859
002860

```

THIS SUBROUTINE CALCULATES THE RESIDUES FOR POLES WITH
MULTIPLICITY EQUAL TO ONE (1). THE GENERAL EXPRESSION
USED IS

$$\text{RESIDUE} = \left[\frac{N(P)Z}{P} \right] / \left[\frac{D(P)Z}{P} \right] - (D(P) \times \text{PXXNT})' \quad]$$

$$\text{RESIDUE} = \frac{Z \times [N(P)/D(P)]'}{[P \times \text{POLES } G(P)/P] \times [Z - (D(P) \times \text{PXXNT})'/D(P)]'}$$

G(P) = N(P)/D(P) --- HIGH-RATE TXFORM P-Z-EXP(SST/N)
Z = EXP(SST) --- DEFINITION LOW-RATE TXFORM
D(P) = D(P) * P


```

002916 C      ARRAY RESIK(51,2).
002917 C
002918 C      STORE POLE TO BE EVALUATED IN R
002919 C      R(1)=DBPOLE(IPOLE,1)
002920 C      R(2)=DBPOLE(IPOLE,2)
002921 C
002922 C      EVALUATE N(P) - NUMERATOR POLY
002923 C      DBNPOLY - POLY TO BE EVALUATED AT *R*
002924 C      NM = 2 * (DBNPOLY ORDER + 1)
002925 C      R - POLE TO BE EVALUATED (REAL,IMAG)
002926 C      R - OR NUMBER OF OCCUPIED ELEMENTS IN DBNPOLY
002927 C
002928 C      V1 - RESULTS OF EVALUATION (REAL,IMAG)
002929 C      ZF - SCALE FACTOR FOR NUMERICAL ACCURACY
002930 C      Z - FORMAT OF DBNPOLY (REAL COEFF, 2-PUR)
002931 C      CALL EVALU3(DBNPOLY,NM,R,V1,ZF,2)
002932 C      V1(1)=V1(1)*ZF
002933 C      V1(2)=V1(2)*ZF
002934 C
002935 C      OBTAIN D(P)' - DENOMINATOR DERIVATIVE
002936 C      DBDPOLY - POLY TO BE DIFFERENTIATED
002937 C      MD = 2 * (DBDPOLY ORDER + 1)
002938 C      R - OR NUMBER OF OCCUPIED ELEMENTS IN DBDPOLY
002939 C      A - ARRAY TO CONTAIN THE DERIVATIVE
002940 C      NA = 2 * (ORDER OF *A* + 1)
002941 C      Z - OR NUMBER OF OCCUPIED ELEMENTS IN ARRAY *A*
002942 C      Z - FORMAT OF DBDPOLY (REAL, 2-PUR)
002943 C      CALL DERIV3(DBDPOLY,MD,A,NA,2)
002944 C
002945 C      EVALUATE D(P)' AT THE POLE *R*
002946 C      CALL EVALU3(A,NA,R,V2,ZF,2)
002947 C      V2(1)=V2(1)*ZF
002948 C      V2(2)=V2(2)*ZF
002949 C
002950 C      CALCULATE EVALUATED EXPRESSION N(P)/D(P)'
002951 C      THE REAL,IMAG VALUES ARE SIMPLY DIVIDED
002952 C      CALL DIVI(V1(1),V1(2),V2(1),V2(2),X,Y)
002953 C
002954 C      STORE NUMERATOR TERM IN ARRAY RESIK(51,2)
002955 C      RESIK(JRESI,1)=X
002956 C      RESIK(JRESI,2)=Y
002957 C      RETURN
002958 C      END
002959 C
002960 C
002961 C
002962 C
002963 C
002964 C
002965 C
002966 C
002967 C
002968 C
002969 C
002970 C

```

THIS ROUTINE CALCULATES THE RESIDUES FOR POLES WITH
MULTIPLICITY EQUAL TO ONE (1). THE EXPRESSION USED
IS:

```

002971
002972
002973
002974
002975
002976
002977
002978
002979
002980
002981
002982
002983
002984
002985
002986
002987
002988
002989
002990
002991
002992
002993
002994
002995
002996
002997
002998
002999
003000
003001
003002
003003
003004
003005
003006
003007
003008
003009
003010
003011
003012
003013
003014
003015
003016
003017
003018
003019
003020
003021
003022
003023
003024
003025

NUMERATOR = 2APZCN(UP)/DX(UP)'JXC(A+U)/(1+YXXNT)J
DENOMINATOR = U + AII(1-YXXNT)/(1+YXXNT)J
WHERE
  DX(UP)' = CD(UP)X(AP+UP)X(AP-UP)J'
  DX(UP)' = DERIVATIVE WITH RESPECT TO UP
  Y = [(AP+UP)/(AP-UP)]J
  YXXNT = [I(AP+UP)/(AP-UP)]JXXNT
  U = LOW-RATE W OR U' VARIABLE [Z = EXP(ST)J]
  UP = HIGH-RATE W OR U' VARIABLE [Z = EXPST/N]J
BILINEAR TRANSFORMATION BETWEEN Z, W, AND W' PLANES:
  W = (Z-1)/(Z+1)
  W' = (2/T)X(Z-1)/(Z+1)
  Z = (1+W)/(1-W)
  U TRANSFORM:
    A = 1.0
    AP = 1.0
    W = U
    UP = UP
  U' TRANSFORM:
    A = 2/TEXT
    AP = 2/TINT
    W = W'
    UP = UP'
TEXT = LOW-RATE SAMPLING PERIOD (SEC)
TINT = HIGH-RATE SAMPLING PERIOD (SEC)
XNT = TEXT/TINT
IN THE SUBROUTINE, THE NUMERATOR TERM FOR THE
RESIDUE IS MECHANIZED VIA:
  RESIK(S1,2) = 2APZCN(UP)/DX(UP)'JXC(1+YXXNT)J
  DBNPOLY(102) = N(UP)
  CDBPOLY(106)J' = DX(UP)'
  YPLUSA,YPLUSI = [1+YXXNT]J
CALCULATION OF THE LOW-RATE POLE FROM THE HIGH-RATE
POLE IS ACCOMPLISHED IN THE SUBROUTINE "UPOLE".
THESE SIMPLE LOW-RATE POLES ARE STORED IN
RESIPOL(S1,2). THAT IS,
  RESIPOL(S1,2) = -AII(1-YXXNT)/(1+YXXNT)J
  Y:AXNT = [(AP+UP)/(AP-UP)]JXXNT
  UP = NUMERICAL VALUE OF HIGH-RATE POLE
DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(S1,2),DBZERO(S1,2)

```



```

003081 IF(KZERO .LT. S) GO TO 100
003082 X=R(1)+DXAP
003083 Y=R(2)
003084 CALL DVI(V1(1),V1(2),X,Y,V2(1),V2(2))
003085 RES1(JRES1,1)=V2(1)
003086 RES1(JRES1,2)=V2(2)
003087
003088 100 CONTINUE
003089 RETURN
003090 END
003091
003092 -----
003093 CDECK WMULT2
003094 OVERLAY(27,3)
003095 PROGRAM WMULT2
003096
003097 -----
003098 SUBROUTINE WMULT2
003099
003100 -----
003101
003102 THIS SUBROUTINE CALCULATES THE RESIDUES FOR POLES WITH
003103 MULTIPLICITY EQUAL TO TWO (2). THE EXPRESSION USED IS:
003104
003105
003106
003107
003108
003109
003110
003111
003112
003113
003114
003115
003116
003117
003118
003119
003120
003121
003122
003123
003124
003125
003126
003127
003128
003129
003130
003131
003132
003133
003134
003135

```

IF(SUM OF RESIDUES OF JJJ)
 ZAPR(A+U) X (NUM1+NUM2+NUM3)U + (NUM4+NUM5+NUM6)J - NUMERATOR
 C U + AX(1-YXXNT)/(1+YXXNT) JJJ2 - DENOMINATOR
 IF EVALUATED AT UP-HIGH-RATE POLES JJJJ

Y = [(AP+UP)/(AP-UP)]
 XNT = TEXT/TINT
 TEXT = LOW-RATE SAMPLING PERIOD (SEC)
 TINT = HIGH-RATE SAMPLING PERIOD (SEC)

NUM1 = [(2IN')/(DX'')X(1+YXXNT)]
 NUM2 = [(-2IN'DX'')/(X(3)X(DX'')X(DX'')X(1+YXXNT))]
 NUM3 = [(-2INXXNT)X(VXXNT-1)X(Y')/(DX'')X(1+YXXNT)]
 NUM4 = [(2IN')X(POLE)/(DX'')X(1+YXXNT)]
 NUM5 = [(-2IN'DX'')X(POLE)/(X(3)X(DX'')X(DX'')X(1+YXXNT))]
 NUM6 = [(2INXXNT)X(VXXNT-1)X(Y')/(DX'')X(1+YXXNT)]

G(UP) = N(UP)/D(UP) -- HIGH-RATE TRANSFER FUNCTION
 G(UP) = N/D
 DX = DX(UP) = D X (AP-UP) X (AP+UP)
 LOW-RATE POLE = -AX[(1-YXXNT)/(1+YXXNT)]
 POLE = AX[(1-YXXNT)/(1+YXXNT)]

YXXNT = [(AP+UP)/(AP-UP)]JXXNT
 YXXNT-1 = [(AP+UP)/(AP-UP)]JXXNT-1
 Y' = DERIVATIVE OF Y WITH RESPECT TO 'UP'
 = (2.0XAP)/[(AP-UP)]JXX2

```

003133 N = NUMERA OR POLYNOMIAL (HIGH-RATE)
003137 N' = FIRST DERIVATIVE OF N WITH RESPECT TO *UP*
003138 D = DENOMINATOR POLYNOMIAL (HIGH-RATE)
003139 D* = D * (AP-LP) * (AP*UP)
003140 D*'' = SECOND DERIVATIVE
003141 D*''' = THIRD DERIVATIVE
003142
003143 U TRANSFORM:
003144 A=1.0
003145 AP=1.0
003146 UP=UP = HIGH-RATE TRANSFORM VARIABLE
003147 U-U = LOW-RATE TRANSFORM VARIABLE
003148
003149 U' TRANSFORM:
003150 A=2/TEXT
003151 AP=2/TEXT
003152 UP=UP' = HIGH-RATE TRANSFORM VARIABLE
003153 U-U' = LOW-RATE TRANSFORM VARIABLE
003154
003155 DOUBLE DBNPOLY(102),DBDPOLY(106),DBPOLE(51,2),DBZERO(51,2)
003156
003157 DOUBLE A(106),C(106),C3(18),C6(318)
003158 DOUBLC R(2),DXAP,DXA,V1(2),V2(2),V3(2),ZF,X,Y
003159 DOUBLC U1(2),U2(2),U3(2)
003160 DOUBLC UNUM(2),UNUM1(2),UDEN2(2),UDEN3(2)
003161 DOUBLC UREAL,UIMAG,YPLUSR,YPLUSI,VN1R,VN1I,VN2R,VN2I,
003162 1 YP1R,YP1I,YP2R,YP2I
003163 DOUBLC TINT,TEXT,XNT,DBCLK
003164 DOUBLC RES2K1(25,2),RES2K2(25,2),RES2POL(25,2)
003165 COMMON/RES,JRES1,JRES2,JRES3,IPOLE,NN,ND,NN,NORDER
003166 COMMON/PAR,TINT,TEXT,XNT,TXFORM,KZERO,NCLZDB,NCLPDB,DBCLK
003167 COMMON/TXCONU1,DBNPOLY,DBDPOLY,DBPOLY,DBPOLE,DBZERO
003168 COMMON/TXCONU2/A,B,C3,C6
003169 COMMON/TXCONU4/RES2K1,RES2K2,RES2POL
003170
003171 ***** POLE TRANSFORM TO LOW-RATE *****
003172 CALL SUBROUTINE UPOLE TO CALCULATE LOW-RATE POLE IN THE
003173 U OR U' PLANE. THESE LOW-RATE POLES ARE STORED IN ARRAY
003174 RES2POLE(25,2) IN THE FORMAT: REAL,IMAG
003175
003176 TRANSFER POLE ACCORDING TO THE *IPOLE* VARIABLE FROM MASTER
003177 DO LOOP 400 I=1,NN IN THE MAIN PROGRAM.
003178 UREAL=DBPOLE(IPOLE,1)
003179 UIMAG=DBPOLE(IPOLE,2)
003180 CALL UPOLE(UREAL,UIMAG,YPLUSR,YPLUSI,VN1R,VN1I,
003181 1 VN2R,VN2I,YP1R,YP1I,YP2R,YP2I)
003182
003183 INCREMENT RESIDUE COUNTER FOR MULTIPLICITY = 2
003184 JRES2=JRES2+1
003185 STORE LOW-RATE POLE IN ARRAY RES2POL(25,2)
003186 RES2POL(JRES2,1)=UREAL
003187 RES2POL(JRES2,2)=UIMAG
003188
003189 ***** CALCULATE NUMERATOR TERMS IN RESIDUE *****
003190

```

AD-A097 864

SYSTEMS TECHNOLOGY INC HAWTHORNE CA F/G 9/2
MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS--ETC(U)
SEP 80 D 6 DIDALEUSKY F33615-79-C-3601
STI-TR-1142-1-3 AFWAL-TR-80-3101-VOL-3 NL

UNCLASSIFIED

2-12
2-12-86

END
DATE
FILMED
8
DTIC

```

C      CALCULATE THE INDIVIDUAL NUMERATOR TERMS IN THE
C      RESIDUE AND STORE THEM AS:
C      CRESK1(25,2)XU + CRESK2(25,2) / U - CRESPOL(25,2)J
C
C      STORE HIGH-RATE POLE TO BE EVALUATED IN ARRAY R(2)
C      R(1)=DBPOLE(IPOLE,1)
C      R(2)=DBPOLE(IPOLE,2)
C      DXA=1.0
C      DXAP=1.0
C      IF(TXFORM.EQ.2HUP) DXA=2.0/TEXT
C      IF(TXFORM.EQ.2HUP) DXAP=2.0/TINT
C
C      OBTAIN: N(UP)=UNUM(2); N(UP)=-UNUM1(2); DX(UP)=-UDEN2(2);
C      AND DX(UP)=-UDEN3(2)
C      CALL EVALU3(DBNPOLY,MN,R,UNUM,ZF,2)
C      UNUM(1)=UNUM(1)*ZF
C      UNUM(2)=UNUM(2)*ZF
C      CALL DERIV3(DBNPOLY,MN,A,NA,2)
C      CALL EVALU3(A,NA,R,UNUM1,ZF,2)
C      UNUM1(1)=UNUM1(1)*ZF
C      UNUM1(2)=UNUM1(2)*ZF
C      CALL DERIV3(DBDPOLY,ND,A,NA,2)
C      CALL DERIV3(A,NA,B,NB,2)
C      CALL EVALU3(B,NB,R,UDEN2,ZF,2)
C      UDEN2(1)=UDEN2(1)*ZF
C      UDEN2(2)=UDEN2(2)*ZF
C      CALL DERIV3(B,NB,A,NA,2)
C      CALL EVALU3(A,NA,R,UDEN3,ZF,2)
C      UDEN3(1)=UDEN3(1)*ZF
C      UDEN3(2)=UDEN3(2)*ZF
C
C      ADD (AP+UP) CANCELLING FACTOR TO NUMERATOR IF
C      KZERO.GT.0. THE VALUE OF KZERO IS SET IN
C      THE (AP+UP) SECTION OF THE MAIN PROGRAM.
C      THIS CANCELLATION OCCURS WHEN THE NUMERATOR
C      CONTAINS THE (AP+UP) ROOT.
C
C      N = N/(AP+UP)
C      N' = [(AP+UP)N' - N]/[(AP+UP)*x2]
C      = [N'/(AP+UP)] - [N/(AP+UP)*x2]
C
C      IF(KZERO.LT.5) GO TO 100
C      U1(1) = DXAP+R(1)
C      U1(2) = R(2)
C      U2(1) = UNUM(1)
C      U2(2) = UNUM(2)
C      CALL DIV1(U2(1),U2(2),U1(1),U1(2),UNUM(1),UNUM(2))
C      CALL DIV1(UNUM1(1),UNUM1(2),U1(1),U1(2),X,Y)
C      CALL DIV1(UNUM(1),UNUM(2),U1(1),U1(2),U3(1),U3(2))
C      UNUM1(1)=X-U3(1)
C      UNUM1(2)=Y-U3(2)
C      100 CONTINUE
C

```

93


```

003411      ***** CALCULATE NUMERATOR TERMS IN RESIDUES *****
003412      CALCULATE THE INDIVIDUAL NUMERATOR TERMS IN THE RESIDUES
003413      AND STORE THEM AS:
003414
003415      [RES3K1(17,2)]*(U**2) + [RES3K2(17,2)]*(U) + [RES3K3(17,2)]
003416
003417
003418      STORE HIGH-RATE POLE TO BE EVALUATED IN ARRAY R(2)
003419      R(1)=DBPOLE(IPOLE,1)
003420      R(2)=DBPOLE(IPOLE,2)
003421      DXA=1.0
003422      DXAP=1.0
003423      IF(TXFORM.EQ.2HUP) DXA=2.0/TEXT
003424      IF(TXFORM.EQ.2HUP) DXAP=2.0/TINT
003425
003426      OBTAIN N=UNUM(2), N'=UNUM1(2), N''=UNUM2(2); DX''=-UDENS(2)
003427      DX''=-UDEN4(2); AND DX''=-UDENS(2)
003428
003429      CALL EVALU3(DBNPOLY,NN,R,UNUM,ZF,2)
003430      UNUM(1)=UNUM(1)*ZF
003431      UNUM(2)=UNUM(2)*ZF
003432      CALL DERIV3(DBNPOLY,NN,A,NA,2)
003433      CALL EVALU3(A,NA,R,UNUM1,ZF,2)
003434      UNUM1(1)=UNUM1(1)*ZF
003435      UNUM1(2)=UNUM1(2)*ZF
003436      CALL DERIV3(A,NA,B,NB,2)
003437      CALL EVALU3(B,NB,R,UNUM2,ZF,2)
003438      UNUM2(1)=UNUM2(1)*ZF
003439      UNUM2(2)=UNUM2(2)*ZF
003440
003441      CALL DERIV3(DBDPOLY,ND,A,NA,2)
003442      CALL DERIV3(A,NA,B,NB,2)
003443      CALL DERIV3(B,NB,A,NA,2)
003444      CALL EVALU3(A,NA,R,UDEN3,ZF,2)
003445      UDEN3(1)=UDEN3(1)*ZF
003446      UDEN3(2)=UDEN3(2)*ZF
003447      CALL DERIV3(A,NA,B,NB,2)
003448      CALL EVALU3(B,NB,R,UDEN4,ZF,2)
003449      UDEN4(1)=UDEN4(1)*ZF
003450      UDEN4(2)=UDEN4(2)*ZF
003451      CALL DERIV3(B,NB,A,NA,2)
003452      CALL EVALU3(A,NA,R,UDEN5,ZF,2)
003453      UDEN5(1)=UDEN5(1)*ZF
003454      UDEN5(2)=UDEN5(2)*ZF
003455
003456      ADD (AP+UP) CANCELLING FACTOR TO NUMERATOR IF
003457      KZERO.GT.0. THE VALUE OF KZERO IS SET IN
003458      THE (AP+UP) SECTION OF THE MAIN PROGRAM.
003459      THIS CANCELLATION OCCURS WHEN THE NUMERATOR
003460      CONTAINS THE (AP+UP) ROOT. THE NUMERATOR
003461      TERMS THEN BECOME:
003462
003463      N = N/(AP+UP)
003464      N' = [N/(AP+UP)] * [N'/(AP+UP)] - [N/(AP+UP)]**2]
003465

```

```

003466      N'' = EN/(AP+UP)J'' = EN'''/(AP+UP)J - [2N'/(AP+UP)]x2J
003467      * [2xN/(AP+UP)]x3J
003468
003469      IF(KZERO .LT. 5) GO TO 100
003470      U1(1)=DXAP+R(1)
003471      U1(2)=R(2)
003472      U2(1)=UNUM(1)
003473      U2(2)=UNUM(2)
003474
003475      CALL DIU(U2(1),U2(2),V1(1),V1(2),UNUM(1),UNUM(2))
003476      U2(1)=UNUM1(1)
003477      U2(2)=UNUM1(2)
003478      CALL DIU(U2(1),U2(2),V1(1),V1(2),UNUM1(1),UNUM1(2))
003479      U2(1)=UNUM2(1)
003480      U2(2)=UNUM2(2)
003481      CALL DIU(U2(1),U2(2),V1(1),V1(2),UNUM2(1),UNUM2(2))
003482
003483      CALL DIU(UNUM(1),UNUM(2),V1(1),V1(2),V3(1),V3(2))
003484      CALL DIU(V3(1),V3(2),V1(1),V1(2),V4(1),V4(2))
003485      CALL DIU(UNUM1(1),UNUM1(2),V1(1),V1(2),V5(1),V5(2))
003486      UNUM2(1)=UNUM2(1)+2.*V5(1)+2.*V4(1)
003487      UNUM2(2)=UNUM2(2)+2.*V5(2)+2.*V4(2)
003488      UNUM1(1)=UNUM1(1)-V3(1)
003489      UNUM1(2)=UNUM1(2)-V3(2)
003490
003491      C 100 CONTINUE
003492
003493      OBTAIN COMMON FACTORS IN THE NUMERATOR TERMS.
003494
003495      UD3V-E(I+YxxXNT)x3-1Jx(Dx''')x3-1
003496      UD3V-E(I+YxxXNT)x3-1Jx(Dx''')x3-2Jx(Dx''')
003497      UD3V-E(I+YxxXNT)x3-1Jx(Dx''')x3-2Jx(Dx''')
003498      UD3V-I(E(I+YxxXNT)x3-1Jx(Dx''')x3-3Jx(Dx'''))x32
003499
003500      V1(1)=1.#
003501      V1(2)=0.#
003502      CALL DIU(V1(1),V1(2),VPLUS,VPLUS,U2(1),U2(2))
003503      CALL DIU(U2(1),U2(2),UDEN3(1),UDEN3(2),UD3V(1),UD3V(2))
003504      CALL DIU(UD3V(1),UD3V(2),UDEN3(1),UDEN3(2),V1(1),V1(2))
003505      CALL MULT(V1(1),V1(2),UDEN5(1),UDEN5(2),UD3V(1),UD3V(2))
003506      CALL MULT(V1(1),V1(2),UDEN4(1),UDEN4(2),UD3V(1),UD3V(2))
003507      CALL MULT(UD3V(1),UD3V(2),UDEN4(1),UDEN4(2),V1(1),V1(2))
003508      CALL DIU(V1(1),V1(2),UDEN3(1),UDEN3(2),UD3V(1),UD3V(2))
003509
003510      OBTAIN THE "UNI" NUMERATOR TERM
003511      UNI=3.*DN'E(I+YxxXNT)x3-1Jx(Dx''')x3-1
003512
003513      V1(1)=3.*UNUM2(1)
003514      V1(2)=3.*UNUM2(2)
003515      CALL MULT(V1(1),V1(2),UD3V(1),UD3V(2),UNI(1),UNI(2))
003516
003517      OBTAIN THE "UN2" NUMERATOR TERM
003518      UN2=-1.5*N'E(C(I+YxxXNT)x3-1Jx(Dx''')x3-2Jx(Dx'''))
003519
003520

```

```

003521
003522
003523
003524
003525
003526
003527
003528
003529
003530
003531
003532
003533
003534
003535
003536
003537
003538
003539
003540
003541
003542
003543
003544
003545
003546
003547
003548
003549
003550
003551
003552
003553
003554
003555
003556
003557
003558
003559
003560
003561
003562
003563
003564
003565
003566
003567
003568
003569
003570
003571
003572
003573
003574
003575

V1(1)=-1.5XNUM(1)
V1(2)=-1.5XNUM(2)
CALL MULT(V1(1),V1(2),UD34Y(1),UD34Y(2),UN2(1),UN2(2))
C
C
C
OBTAIN THE 'UN3' NUMERATOR TERM
UN3=-.375XN(1+VXXNT)X-13X(DX''')X-33X(DX''')X23
C
C
V1(1)=-.375XNUM(1)
V1(2)=-.375XNUM(2)
CALL MULT(V1(1),V1(2),UD34Y(1),UD34Y(2),UN3(1),UN3(2))
C
C
OBTAIN THE 'UN4' NUMERATOR TERM
UN4=-0.32XN(1+VXXNT)X-13X(DX''')X-23X(DX''')
C
C
V1(1)=-0.32XNUM(1)
V1(2)=-0.32XNUM(2)
CALL MULT(V1(1),V1(2),UD35Y(1),UD35Y(2),UN4(1),UN4(2))
C
C
OBTAIN THE 'UN5' NUMERATOR TERM
UN5=-6.0XN(X(1+VXXNT)X-23X(DX''')X-13XNTX(VXXNT-1)XV'
X(XNT)XVXX(XNT-1)XV'
C
C
V1(1)=-6.0XNUM(1)XNT
V1(2)=-6.0XNUM(2)XNT
CALL MULT(V1(1),V1(2),VNIR,VNIR,VN11,VN11,V2(1),V2(2))
CALL MULT(V2(1),V2(2),VP1R,VP1R,VP11,VP11,V1(1),V1(2))
CALL MULT(V1(1),V1(2),UD3V(1),UD3V(2),V2(1),V2(2))
CALL DIV(V2(1),V2(2),VPLUSR,VPLUSR,VN5(1),VN5(2))
C
C
OBTAIN THE 'UN6' NUMERATOR TERM
UN6=-3.0XN(1+VXXNT)X-23X(DX''')X-23X(DX''')
X(XNT)XVXX(XNT-1)XV'
C
C
V1(1)=-3.0XNUM(1)XNT
V1(2)=-3.0XNUM(2)XNT
CALL MULT(V1(1),V1(2),VNIR,VNIR,VN11,VN11,V2(1),V2(2))
CALL MULT(V2(1),V2(2),VP1R,VP1R,VP11,VP11,V1(1),V1(2))
CALL MULT(V1(1),V1(2),UD34Y(1),UD34Y(2),V2(1),V2(2))
CALL DIV(V2(1),V2(2),VPLUSR,VPLUSR,VN6(1),VN6(2))
C
C
OBTAIN THE 'UN7' NUMERATOR TERM
UN7=-1.5XN(1+VXXNT)X-23X(DX''')X-23X(DX''')
X(XNT)XVXX(XNT-1)XV'
C
C
V1(1)=-1.5XNUM(1)XNT
V1(2)=-1.5XNUM(2)XNT
CALL MULT(V1(1),V1(2),VNIR,VNIR,VN11,VN11,V2(1),V2(2))
CALL MULT(V2(1),V2(2),VP1R,VP1R,VP11,VP11,V1(1),V1(2))
CALL MULT(V1(1),V1(2),UD34Y(1),UD34Y(2),V2(1),V2(2))
CALL DIV(V2(1),V2(2),VPLUSR,VPLUSR,VN7(1),VN7(2))
C
C
OBTAIN THE 'UN8' NUMERATOR TERM
UN8=-3.0XN(1+VXXNT)X-23X(DX''')X-13X(XNT)X(XNT-1)
XVXX(XNT-2)X(XNT-1)X23
C
C
V1(1)=-3.0XNUM(1)XNTX(XNT-1.0)

```

```

003576
003577
003578
003579
003580
003581
003582
003583
003584
003585
003586
003587
003588
003589
003590
003591
003592
003593
003594
003595
003596
003597
003598
003599
003600
003601
003602
003603
003604
003605
003606
003607
003608
003609
003610
003611
003612
003613
003614
003615
003616
003617
003618
003619
003620
003621
003622
003623
003624
003625
003626
003627
003628
003629
003630

U1(2)=-3.0XNUM(2)XNTX(XNT-1.0)
CALL MULT(U1(1),U1(2),VN2R,VN2I,U2(1),U2(2))
CALL MULT(U2(1),U2(2),VP1R,VP1I,U1(1),U1(2))
CALL MULT(U1(1),U1(2),VP1R,VP1I,U2(1),U2(2))
CALL MULT(U2(1),U2(2),UD3Y(1),UD3Y(2),U1(1),U1(2))
CALL DIVI(U1(1),U1(2),YPLUSR,YPLUSI,UN8(1),UN8(2))

C
C OBTAIN THE "UN9" NUMERATOR TERM
C UN9=-3.0XNUM(1)+YXNTJXX-2JXC(D8'')JXX-1JX(XNTJEXX(XNT-1)JY''
C
C
C U1(1)=-3.0XNUM(1)XNT
C U1(2)=-3.0XNUM(2)XNT
CALL MULT(U1(1),U1(2),VN1R,VN1I,U2(1),U2(2))
CALL MULT(U2(1),U2(2),VP2R,VP2I,U1(1),U1(2))
CALL MULT(U1(1),U1(2),UD3Y(1),UD3Y(2),U2(1),U2(2))
CALL DIVI(U2(1),U2(2),YPLUSR,YPLUSI,UN9(1),UN9(2))

C
C OBTAIN THE "UN10" NUMERATOR TERM
C UN10=-6.0XNUM(1)+YXNTJXX-3JXC(D8'')JXX-1JX(XNTJX2
C X(YX(XNT-1))X2JXC(Y'')X2J
C
C
C U1(1)=-6.0XNUM(1)XNTXNT
C U1(2)=-6.0XNUM(2)XNTXNT
CALL MULT(U1(1),U1(2),VN1R,VN1I,U2(1),U2(2))
CALL MULT(U2(1),U2(2),VN1R,VN1I,U1(1),U1(2))
CALL MULT(U1(1),U1(2),VP1R,VP1I,U2(1),U2(2))
CALL MULT(U2(1),U2(2),VP1R,VP1I,U1(1),U1(2))
CALL MULT(U1(1),U1(2),UD3Y(1),UD3Y(2),U2(1),U2(2))
CALL DIVI(U2(1),U2(2),YPLUSR,YPLUSI,U1(1),U1(2))
CALL DIVI(U1(1),U1(2),YPLUSR,YPLUSI,UN10(1),UN10(2))

C
C COMBINE THE NUMERATOR TERMS TO OBTAIN THE COEFFICIENTS
C OF THE NUMERATOR OF THE RESIDUE
C [RES3K1(17,2)JX(U2X2) + [RES3K2(17,2)JX(U) + [RES3K3(17,2)J
C
C U2(1)=UN1(1)+UN2(1)+UN3(1)+UN4(1)
C U2(2)=UN1(2)+UN2(2)+UN3(2)+UN4(2)
C U3(1)=UN5(1)+UN6(1)+UN7(1)+UN8(1)+UN9(1)
C U3(2)=UN5(2)+UN6(2)+UN7(2)+UN8(2)+UN9(2)
C
C RES3K1(JRES3,1)=2.0DXAPX(U2(1)+U3(1)+UN10(1))
C RES3K1(JRES3,2)=2.0DXAPX(U2(2)+U3(2)+UN10(2))
C
C U1(1)=-2.0RES3POL(JRES3,1)
C U1(2)=-2.0RES3POL(JRES3,2)
CALL MULT(U1(1),U1(2),U2(1),U2(2),U4(1),U4(2))
U1(1)=-RES3POL(JRES3,1)-DXA
U1(2)=-RES3POL(JRES3,2)
CALL MULT(U1(1),U1(2),U3(1),U3(2),U5(1),U5(2))
RES3K2(JRES3,1)=2.0DXAPX(U4(1)+U5(1)-2.0DXAPXUN10(1))
RES3K2(JRES3,2)=2.0DXAPX(U4(2)+U5(2)-2.0DXAPXUN10(2))
C
C U1(1)=-RES3POL(JRES3,1)
C U1(2)=-RES3POL(JRES3,2)

```



```

C      C3 = C3(318)
C      C6 = C6(318)
C
C      DOUBLE RESID(156),RESIN(156),RESIDS(6),RESINS(3)
C      DOUBLE A(106),B(106),C3(318),C6(318)
C      DOUBLE RESIK(51,2),RESIPOL(51,2)
C      DOUBLE XMAX,TINT,TEXT,XNT,DBCLK,DZERO,PUR
C      COMMON/RES/JRES1,JRES2,JRES3,IPOLE,NN,ND,NI,NORDER
C      COMMON/ZEROS/KRES1Z,KRES2Z,KRES3Z
C      COMMON/PAR/TINT,TEXT,XNT,TFORM,KZERO,NCLZDB,NCLPDB,DBCLK
C      COMMON/TXCONU2/A,B,C3,C6
C      COMMON/TXCONU3/RESIK,RESIPOL
C      COMMON/TXCONU6/RESIN,RESID,KRESIN,KRESID
C
C      INITIALIZE ARRAYS TO ZERO
C      IORD=3*(NORDER+2)
C      DO 100 I=1,IORD
C      RESIN(I)=0.0
C      RESID(I)=0.0
C      KRESIN=3
C      KRESID=6
C      INITIALIZE OVERALL STORAGE ARRAYS WITH
C      FIRST RESIDUE TERM
C      RESID(1)=1.0
C      RESID(2)=0.0
C      RESID(3)=1.0
C      RESID(4)=-RESIPOL(1,1)
C      RESID(5)=-RESIPOL(1,2)
C      RESID(6)=0.0
C      RESIN(1)=RESIK(1,1)
C      RESIN(2)=RESIK(1,2)
C      RESIN(3)=0.0
C
C      IF(JRES1.EQ.1) GO TO 250
C
C      LOOP TO ADD INDIVIDUAL RESIDUE TERMS
C      JRES1 = NUMBER OF SIMPLE POLES
C      DO 200 K=2,JRES1
C      TEMPORARILY STORE NEXT RESIDUE TERM
C      RESIDS(1)=1.0
C      RESIDS(2)=0.0
C      RESIDS(3)=1.0
C      RESIDS(4)=-RESIPOL(K,1)
C      RESIDS(5)=-RESIPOL(K,2)
C      RESIDS(6)=0.0
C      RESINS(1)=RESIK(K,1)
C      RESINS(2)=RESIK(K,2)
C      RESINS(3)=0.0
C      CALCULATE 1ST NUMERATOR TERM C3=NI*DI
C      KRESIN = 2 * (RESIN ORDER + 1)
C      RESIDS = 2ND INPUT POLY
C      6 = 2 * (RESIDS ORDER + 1)

```

```

003686
003687
003688
003689
003690
003691
003692
003693
003694
003695
003696
003697
003698
003699
003700
003701
003702
003703
003704
003705
003706
003707
003708
003709
003710
003711
003712
003713
003714
003715
003716
003717
003718
003719
003720
003721
003722
003723
003724
003725
003726
003727
003728
003729
003730
003731
003732
003733
003734
003735
003736
003737
003738
003739
003740

```

```

C C3 = OUTPUT POLY = RESINRESIDS
C NT3 = 2 * (C3 ORDER + 1)
C 3 = FORMAT FOR INPUT POLYS: REAL,IMAG,Z-PUR
C CALL MULTIPLY(RESIN,KRESIN,RESIDS,6,C3,NT3,3)
C
C STORE 1ST NUMERATOR TERM: N=C3
C DO 300 I=1,NT3
C 300 RESIN(I)=C3(I)
C SET NEW ORDER OF FIRST NUMERATOR TERM
C KRESIN=NT3
C CALCULATE 2ND NUMERATOR TERM: C3-D*NT1
C CALL MULTIPLY(RESID,KRESID,RESINS,3,C3,NT3,3)
C
C ADD 1ST & 2ND NUMERATOR TERMS
C RESIN = INPUT POLY
C KRESIN = 2 * (RESIN ORDER + 1)
C C3 = INPUT POLY
C NT3 = 2 * (C3 ORDER + 1)
C C6 = OUTPUT POLY = RESIN+C3
C NT6 = 2 * (C6 ORDER + 1)
C 3 = FORMAT FOR INPUT POLYS: REAL,IMAG,Z-PUR
C CALL ADD(RESIN,KRESIN,C3,NT3,C6,NT6,3)
C
C STORE TOTAL NUMERATOR POLY: N=C6
C DO 400 I=1,NT6
C 400 RESIN(I)=C6(I)
C SET NEW ORDER OF TOTAL NUMERATOR POLY
C KRESIN=NT6
C CALCULATE TOTAL DENOMINATOR POLY: C3-D*DI
C CALL MULTIPLY(RESID,KRESID,RESIDS,6,C3,NT3,3)
C
C STORE TOTAL DENOMINATOR POLY: D=C3
C DO 500 I=1,NT3
C 500 RESID(I)=C3(I)
C SET NEW ORDER OF TOTAL DENOMINATOR POLY
C KRESID=NT3
C 200 CONTINUE
C
C 250 CONTINUE
C
C ORDER NUMERATOR POLY IN DESCENDING PURS OF Z, U, OR V
C CALL ORDER3(RESIN,KRESIN,3)
C
C FIND LARGEST NUMERATOR COEFFICIENT
C IF(KRESIN.GT. 0) GO TO 700
C XMAX=0.0
C DO 600 I=1,KRESIN,3
C 600 IF(DABS(RESIN(I)).GT. XMAX) XMAX = DABS(RESIN(I))
C
C ELIMINATE EXTREMELY SMALL HIGHEST ORDER NUM COEFF TERM
C IF(DABS(RESIN(1))/XMAX).LT. 1.D-12) GO TO 710
C GO TO 700
C 710 RESIN(1)=0.0
C RESIN(2)=0.0
C RESIN(3)=0.0

```

```

C CALL SIMPLE ROUTINE TO PUSH ZEROED
C OUT TERM TO BOTTOM OF ARRAY KRESIN
C CALL SIMPLE(RESIN,KRESIN,3)
700 CONTINUE
C ***** ADD ZERO OR CANCEL POLE *****
C ***** AT Z=0.0, U=-1.0, OR U=-2/TEXT *****
C THIS SECTION HUNTS AND CANCELS DENOMINATOR POLE AT Z=0.0
C U=-1.0, OR U=-2/TEXT INSTEAD OF ADDING LIKE ZERO TO THE
C NUMERATOR.
C KSKIP=0
C ZERO=0.0
C DZERO=0.0
C IF(TXFORM.EQ. 2H2T) GO TO 715
C ZERO=-1.0
C DZERO=-1.0
C IF(TXFORM.EQ. 2HUP) ZERO=-2.0/TEXT
C IF(TXFORM.EQ. 2HUP) DZERO=-2.0/TEXT
715 CONTINUE
C DO 720 I=1,JRES1
C ISH=1
C P1=RESIPOL(I,1)
C P2=RESIPOL(I,2)
C IF(ABS(P1-ZERO).LE.1.E-11.AND.ABS(P2).LE.1.E-11) GO TO 725
720 CONTINUE
C GO TO 730
C SET FLAG TO INDICATE THAT POLE HAS BEEN CANCELLED
725 KSKIP=0
C CANCEL POLE BY OVERWRITING IT
C DO 735 J=ISH,JRES1
C RESIPOL(J,1)=RESIPOL(J+1,1)
C RESIPOL(J,2)=RESIPOL(J+1,2)
C JRES1=JRES1-1
730 CONTINUE
C MULTIPLY NUMERATOR POLY IN RESIN(156) BY (Z) OR (U+A)
C IF NO LIKE POLE IS FOUND
C IF(KSKIP.GT. 5) GO TO 740
C A(1)=1.0
C A(2)=0.0
C A(3)=1.0
C A(4)=-DZERO
C A(5)=0.0
C A(6)=0.0
C CALL MULTIPLY(A,6,RESIN,KRESIN,C3,NT3,3)
C TRANSFER NEW NUMERATOR POLY INTO RESIN(156)
C DO 745 I=1,NT3
745 RESIN(I)=C3(I)
C KRESIN=NT3
740 CONTINUE
C ORDER NEW NUMERATOR POLY IN DESCENDING PWS OF Z, U, OR U'
C CALL ORDER3(RESIN,KRESIN,3)
C TRANSFER REAL AND IMAGINARY PART OF NUMERATOR COEFFICIENTS
C IN ARRAY RESIN INTO TWO SEPARATE ARRAYS 'A' AND 'B'

```

```

003796
003797
003798
003799
003800
003801
003802
003803
003804
003805
003806
003807
003808
003809
003810
003811
003812
003813
003814
003815
003816
003817
003818
003819
003820
003821
003822
003823
003824
003825
003826
003827
003828
003829
003830
003831
003832
003833
003834
003835
003836
003837
003838
003839
003840
003841
003842
003843
003844
003845
003846
003847
003848
003849
003850

```

```

003851
003852
003853
003854
003855
003856
003857
003858
003859
003860
003861
003862
003863
003864
003865
003866
003867
003868
003869
003870
003871
003872
003873
003874
003875
003876
003877
003878
003879
003880
003881
003882
003883
003884
003885
003886
003887
003888
003889
003890
003891
003892
003893
003894
003895
003896
003897
003898
003899
003900
003901
003902
003903
003904
003905

K=1
DO 755 I=1,KRESIN,3
  A(I)=RESINI(I)
  B(I)=RESINI(I+1)
755 K=K+1
C SET DEGREE OF NUMERATOR POLY AND NUMBER OF ZEROS IN KRESIZ
C KRESIZ=RESIN(3)
C OBTAIN NUMERATOR ROOTS
  IF(KRESIZ.EQ. 0) GO TO 762
  CALL ROOTS(A,B,KRESIZ,C3,C6)
C TRANSFER NUMERATOR ROOTS INTO ARRAY RESIK(51,2)
  DO 760 I=1,KRESIZ
    RESIK(I,1)=C3(I)
    RESIK(I,2)=C6(I)
760 RESIK(I,2)=C6(I)
C FORM NEW DENOMINATOR POLYNOMIAL IF POLE CANCELLED
762 CONTINUE
C IF(KSKIP.LT. 5) GO TO 765
C TRANSFER REAL AND IMAGINARY PART OF POLES INTO
C ARRAY 'A' IN SEQUENTIAL BACK-TO-BACK FORMAT
  RESID(1)=1.0
  RESID(2)=0.0
  RESID(3)=0.0
  KRESID=3
  IF(JRESI.EQ. 0) GO TO 765
  DO 770 I=1,JRESI
    A(I)=RESIPOL(I,1)
    B(I)=RESIPOL(I,2)
770 B(I)=RESIPOL(I,2)
C CALL SUBROUTINE 'CONPOLY' TO OBTAIN NEW DENOMINATOR POLY
  CALL CONPOLY(C3,C6,A,B,JRESI)
C TRANSFER DENOMINATOR POLY COEFFS INTO ARRAY RESID(156)
C AND ADD POLY POWERS.
  K=1
  N=JRESI+1
  DO 771 I=1,N
    PUR=FLOAT(JRESI-I+1)
    RESID(K)=C3(I)
    RESID(K+1)=C6(I)
    RESID(K+2)=PUR
771 K=K+3
771 KRESID=K-1
765 CONTINUE
  RETURN
  END

C-----
C CDECK RES2
C OVERLAY(27.6)
C PROGRAM RES2
C-----
C-----
C SUBROUTINE RES2
C-----
C-----

```

003906
003907
003908
003909
003910
003911
003912
003913
003914
003915
003916
003917
003918
003919
003920
003921
003922
003923
003924
003925
003926
003927
003928
003929
003930
003931
003932
003933
003934
003935
003936
003937
003938
003939
003940
003941
003942
003943
003944
003945
003946
003947
003948
003949
003950
003951
003952
003953
003954
003955
003956
003957
003958
003959
003960

```

RES2D(5)=-2.0*RES2POL(1,2)
RES2D(6)=-1.0
RES2D(7)=X
RES2D(8)=Y
RES2D(9)=0.0
C IF(JRES2.EQ.1) GO TO 250
C
C LOOP TO ADD INDIVIDUAL RESIDUE TERMS
C JRES2 = NUMBER OF SECOND ORDER POLES
DO 200 K=2,JRES2
C TEMPORARILY STORE NEXT RESIDUE TERM IN RES2NS AND RES2DS
RES2NS(1)=RES2K1(K,1)
RES2NS(2)=RES2K1(K,2)
RES2NS(3)=1.0
RES2NS(4)=RES2K2(K,1)
RES2NS(5)=RES2K2(K,2)
RES2NS(6)=0.0
C
R(1)=RES2POL(K,1)
R(2)=RES2POL(K,2)
CALL MULT(R(1),R(2),R(1),R(2),X,Y)
C
RES2DS(1)=1.0
RES2DS(2)=0.0
RES2DS(3)=2.0
RES2DS(4)=-2.0*RES2POL(K,1)
RES2DS(5)=-2.0*RES2POL(K,2)
RES2DS(6)=1.0
RES2DS(7)=X
RES2DS(8)=Y
RES2DS(9)=0.0
C CALCULATE: C3=ND1
CALL MULTIP(RES2N,KRES2N,RES2DS,9,C3,NT3,3)
DO 300 I=1,NT3
300 RES2N(I)=C3(I)
KRES2N=NT3
C CALCULATE: C3=DN1
CALL MULTIP(RES2D,KRES2D,RES2NS,6,C3,NT3,3)
C ADD (ND1) + (DN1)
CALL ADD(RES2N,KRES2N,C3,NT3,C6,NT6,3)
DO 400 I=1,NT6
400 RES2N(I)=C6(I)
KRES2N=NT6
C CALCULATE DENOMINATOR POLY: C3=DD1
CALL MULTIP(RES2D,KRES2D,RES2DS,9,C3,NT3,3)
DO 500 I=1,NT3
500 RES2D(I)=C3(I)
KRES2D=NT3
C 200 CONTINUE
250 CONTINUE

```

```

003961
003962
003963
003964
003965
003966
003967
003968
003969
003970
003971
003972
003973
003974
003975
003976
003977
003978
003979
003980
003981
003982
003983
003984
003985
003986
003987
003988
003989
003990
003991
003992
003993
003994
003995
003996
003997
003998
003999
004000
004001
004002
004003
004004
004005
004006
004007
004008
004009
004010
004011
004012
004013
004014
004015

```

```

C
C      CALL ORDER3(RES2N,KRES2N,3)
C      C ELIMINATE EXTREMELY SMALL HIGHEST-ORDER NUM COEFF TERM
      IF(KRES2N.GT.0) GO TO 700
      XMAX=0.0
      DO 600 I=1,KRES2N,3
        600 IF(DABS(RES2N(I)).GT.XMAX) XMAX=DABS(RES2N(I))
      IF(DABS(RES2N(1))/XMAX).LT.1.D-12) GO TO 710
      GO TO 700
      710 RES2N(1)=0.0
          RES2N(2)=0.0
          RES2N(3)=0.0
      CALL SIMPLE(RES2N,KRES2N,3)
      700 CONTINUE

C      C MULTIPLE NUMERATOR POLY IN RES2N(156) BY (A+U)
      DZERO=1.0
      IFITXFORM.EQ.2MUP) DZERO=2.0/TEXT
      A(1)=1.0
      A(2)=0.0
      A(3)=1.0
      A(4)=DZERO
      A(5)=0.0
      A(6)=0.0
      CALL MULTIP(A,6,RES2N,KRES2N,C3,NT3,3)
      DO 745 I=1,NT3
        745 RES2N(I)=C3(I)
      KRES2N=NT3

C      C OBTAIN NUMERATOR ROOTS
      CALL ORDER3(RES2N,KRES2N,3)
      K=1
      DO 755 I=1,KRES2N,3
        A(K)=RES2N(I)
        B(K)=RES2N(I+1)
        755 K=K+1
      KRES2Z=RES2N(3)
      IF(KRES2Z.EQ.0) GO TO 770
      CALL ROOTS(A,B,KRES2Z,C3,C6)

C      C TRANSFER NUMERATOR ROOTS INTO ARRAYS RES2K1(25,2)
      C AND RES2K2(25,2)
      K2=KRES2Z
      IF(KRES2Z.GT.25) K2=25
      DO 760 I=1,K2
        RES2K1(I,1)=C3(I)
        760 RES2K1(I,2)=C6(I)
      IF(KRES2Z.GT.25) GO TO 765
      GO TO 770
      765 K2=KRES2Z-25
      DO 775 I=1,K2
        RES2K2(I,1)=C3(I+26)
        775 RES2K2(I,2)=C6(I+26)
      770 CONTINUE

```

```

704016
704017
704018
704019
704020
704021
704022
704023
704024
704025
704026
704027
704028
704029
704030
704031
704032
704033
704034
704035
704036
704037
704038
704039
704040
704041
704042
704043
704044
704045
704046
704047
704048
704049
704050
704051
704052
704053
704054
704055
704056
704057
704058
704059
704060
704061
704062
704063
704064
704065
704066
704067
704068
704069
704070

```



```

RESM(1)=RESK(1,1)
RESM(2)=RESK(1,2)
RESM(3)=2.0
RESM(4)=RESK(2,1)
RESM(5)=RESK(2,2)
RESM(6)=1.0
RESM(7)=RESK(3,1)
RESM(8)=RESK(3,2)
RESM(9)=0.0
C
R1(1)=RESPOL(1,1)
R1(2)=RESPOL(1,2)
CALL MULT(R1(1),R1(2),R1(1),R1(2),R2(1),R2(2))
CALL MULT(R2(1),R2(2),R1(1),R1(2),R3(1),R3(2))
C
RES3D(1)=1.0
RES3D(2)=0.0
RES3D(3)=3.0
RES3D(4)=3.0*R1(1)
RES3D(5)=3.0*R1(2)
RES3D(6)=2.0
RES3D(7)=3.0*R2(1)
RES3D(8)=3.0*R2(2)
RES3D(9)=1.0
RES3D(10)=R3(1)
RES3D(11)=R3(2)
RES3D(12)=0.0
C
IF(JRES3.EQ.1) GO TO 250
C
C LOOP TO ADD INDIVIDUAL RESIDUE TERMS
C JRES3 = NUMBER OF SECOND ORDER POLES
C DO 200 K=2,JRES3
C TEMPORARILY STORE NEXT RESIDUE TERM IN RESJMS AND RES3DS
RESJMS(1)=RESK1(K,1)
RESJMS(2)=RESK1(K,2)
RESJMS(3)=2.0
RESJMS(4)=RESK2(K,1)
RESJMS(5)=RESK2(K,2)
RESJMS(6)=1.0
RESJMS(7)=RESK3(K,1)
RESJMS(8)=RESK3(K,2)
RESJMS(9)=0.0
C
R1(1)=RESPOL(K,1)
R1(2)=RESPOL(K,2)
CALL MULT(R1(1),R1(2),R1(1),R1(2),R2(1),R2(2))
CALL MULT(R2(1),R2(2),R1(1),R1(2),R3(1),R3(2))
C
RES3DS(1)=1.0
RES3DS(2)=0.0
RES3DS(3)=3.0
RES3DS(4)=3.0*R1(1)
RES3DS(5)=3.0*R1(2)
RES3DS(6)=2.0

```

```

004126
004127
004128
004129
004130
004131
004132
004133
004134
004135
004136
004137
004138
004139
004140
004141
004142
004143
004144
004145
004146
004147
004148
004149
004150
004151
004152
004153
004154
004155
004156
004157
004158
004159
004160
004161
004162
004163
004164
004165
004166
004167
004168
004169
004170
004171
004172
004173
004174
004175
004176
004177
004178
004179
004180

```

```

RES3DS(7)=3.04R2(1)
RES3DS(8)=3.04R2(2)
RES3DS(9)=1.0
RES3DS(10)=R3(1)
RES3DS(11)=R3(2)
RES3DS(12)=0.0

C CALCULTE1 C3=ND1
CALL MULTIP(RES3N,KRES3N,RES3DS,12,C3,NT3,3)
DO 300 I=1,NT3
300 RES3(I)=C3(I)
KRES3N=NT3

C CALCULTE1 C3=DN1
CALL MULTIP(RES3D,KRES3D,RES3NS,9,C3,NT3,3)
C ADD (ND1) + (DN1)
CALL ADD(RES3N,KRES3N,C3,NT3,C6,NT6,3)
DO 400 I=1,NT6
400 RES3(I)=C6(I)
KRES3N=NT6

C CALCULATE DENOMINATOR POLY: C3=DN1
CALL MULTIP(RES3D,KRES3D,RES3DS,12,C3,NT3,3)
DO 500 I=1,NT3
500 RES3(I)=C3(I)
KRES3D=NT3

C 200 CONTINUE
250 CONTINUE

C CALL ORDER3(RES3N,KRES3N,3)

C ELIMINATE EXTREMELY SMALL HIGHEST-ORDER NUM COEFF TERM
IF(KRES3N.GT.0) GO TO 700
XMAX=0.0
DO 600 I=1,KRES3N,3
600 IF(DABS(RES3N(I)).GT.XMAX) XMAX=DABS(RES3N(I))
IF(DABS(RES3N(1))>XMAX) .LT. 1.D-12) GO TO 710
GO TO 700
710 RES3N(1)=0.0
RES3N(2)=0.0
RES3N(3)=0.0
CALL SIMPLE(RES3N,KRES3N,3)
700 CONTINUE

C MULTIPLE NUMERATOR POLY IN RES3N(156) BY (A+U)
DZERO=1.0
IF(TXFORM.EQ.2HUP) DZERO=2.0/TEXT
A(1)=1.0
A(2)=0.0
A(3)=1.0
A(4)=DZERO
A(5)=0.0
A(6)=0.0
CALL MULTIP(A,6,RES3N,KRES3N,C3,NT3,3)

```

```

004181
004182
004183
004184
004185
004186
004187
004188
004189
004190
004191
004192
004193
004194
004195
004196
004197
004198
004199
004200
004201
004202
004203
004204
004205
004206
004207
004208
004209
004210
004211
004212
004213
004214
004215
004216
004217
004218
004219
004220
004221
004222
004223
004224
004225
004226
004227
004228
004229
004230
004231
004232
004233
004234
004235

```

004236
004237
004238
004239
004240
004241
004242
004243
004244
004245
004246
004247
004248
004249
004250
004251
004252
004253
004254
004255
004256
004257
004258
004259
004260
004261
004262
004263
004264
004265
004266
004267
004268
004269
004270
004271
004272
004273
004274

```

DO 745 I=1,NT3
745 RES3K(I)=C3(I)
KRES3N=NT3
C
C OBTAIN NUMERATOR ROOTS
CALL ORDER3(RES3N,KRES3N,3)
K=1
DO 755 I=1,KRES3N,3
A(K)=RES3N(I)
B(K)=RES3N(I+1)
755 K=K+1
KRES3Z=RES3N(3)
IF(KRES3Z.EQ.0) GO TO 770
CALL ROOTS(A,B,KRES3Z,C3,C6)
C
C TRANSFER NUMERATOR ROOTS INTO ARRAYS RES3K1(17,2)
C RES3K2(17,2), AND RES3K3(17,2)
KZ=KRES3Z
IF(KRES3Z.GT.17) KZ=17
DO 760 I=1,KZ
RES3K1(I,1)=C3(I)
760 RES3K1(I,2)=C6(I)
IF(KRES3Z.GT.17) GO TO 765
GO TO 770
765 KZ=KRES3Z-17
IF(KZ.GT.17) KZ=17
DO 775 I=1,KZ
RES3K2(I,1)=C3(I+17)
775 RES3K2(I,2)=C6(I+17)
IF(KRES3Z.GT.34) GO TO 766
GO TO 770
766 KZ=KRES3Z-34
DO 777 I=1,KZ
RES3K3(I,1)=C3(I+34)
777 RES3K3(I,2)=C6(I+34)
770 CONTINUE
C
RETURN
END

```

DATE
FILMED
- 8