

OK

BREAK: HHH

; OS \_\_\_\_\_

LTST

100 REM

110 REM

120 REM

130 REM

140 REM

150 REM

160 REM

170 REM

180 REM

\*\*\*\*\* CNAP \*\*\*\*\*

MATRIX SOLUTION BY H-P FOR THE 9830. RE-WRITTEN  
BY DEANE JENSEN. MODIFIED FOR MICROSOFT BY GORDON  
MASSENBURG AND ROB ROBINET. FALL 1978

```

190 REM
200 REM REQUIRES 9 DIGIT FLOATING POINT AND >16K MEMORY
210 REM
220 REM FOR CLARITY ALL CONSTANTS ARE SHOWN IN RAW NUMERIC
230 REM FORM. HOWEVER, A SIGNIFICANT SPEED ADVANTAGE IS
240 REM GAINED WITH MICROSOFT WHEN CONSTANTS ARE REPRESENTED BY
250 REM VARIABLES. THIS IS AN ADVANTAGE IN RUNNING THE MATRIX.
260 REM FOR INSTANCE, IN LINES 2590 & 2600 REPLACE THE '2' WITH
270 REM A VARIABLE 'TU' WHICH HAS BEEN SET IN INITIALIZATION 'TU=2'.
280 REM
290 REM NOTE USE OF (.00001) FACTOR TO CORRECT MICROSOFT
300 REM BINARY ROUND-OFF ERROR!!
310 REM
320 REM TO SAVE SPACE ELIMINATE REM'S TO LINE 99. DO NOT
330 REM DROP REM'S IN PROGRAM AS THEY ARE OFTEN RECIEVERS
340 REM FOR GOTO STATEMENTS
350 REM
360 REM UPDATED & RENUMBERED 02/11/81 - GYM
370 REM
380 REM INITIALIZATION
390 LN=LOG(10)
400 R1=0:N1=0:C1=1:Y1=1:N2=2:S1=3:N3=4
410 PI=3.14159266
420 INPUT "# OF NODES"; N4
430 C2=5*N4
440 DIMY(N4,N4),Z(N4,N4),C(2*C2)
450 O1=N4
460 IF N4=2 GOTO 510
470 PRINT"OUTPUT NODE 2 OR"; O1;
480 INPUT O1
490 IF O1=2 THEN O2=Y1
500 IF O1<>2 AND O1<>N4 THEN O1=N4
510 PRINT"INPUT IS NODE 1 OUTPUT IS NODE"; O1; " GROUND IS NODE 0"
520 REM * FIRST COMPONENT
530 C3=0
540 REM * NEXT COMPONENT
550 ON C3+1 GOTO 560, 590, 620, 650, 1310
560 C#="RESISTOR (OHMS) "
570 E1=0
580 GOTO 680
590 C#="CAPACITOR (MFD) "
600 E1=6

610 GOTO 680
620 C#="INDUCTOR (HY) "
630 E1=0
640 GOTO 680
650 C#="SOURCE (MHO) "
660 E1=0
670 REM 'J' IS USED TO SAVE OLD VALUE. USE ANY CONVENIENT SYMBOL
680 PRINTC#;:INPUT"ENTER NEW VALUE [OR 'J' TO SAVE OLD VALUE]"; V1#
690 IF V1#="J" GOTO 750
700 V1=VAL(V1#)
710 IF V1<0 THEN : PRINT"ERROR! ";: GOTO680
720 IF V1>0 GOTO 750
730 IF V1=0 THEN C3=C3+1
740 GOTO 540
750 REM * STORE VALUE
760 N(C3+1)=N(C3+1)+1
770 C(2*N(C3+1)+1)=V1*(10^LN)
780 N(C3+1)=N(C3+1)+1

```

```

800 IF C3=51 THEN D1=1
810 IF C3=51 THEN D1=1
820 IF Y<0 OR Y>N4 THEN PRINT "ERROR! "; GOTO 808
830 Y=INT(Y)
840 Z=0
850 IF C3=51 GOTO 930
860 REM *
870 PRINT "TO NODE";
880 INPUT Z
890 IF Z<0 OR Z>N4 THEN PRINT "ERROR! "; GOTO 808
900 Z=INT(Z)
910 PRINT C#; V1;
920 PRINT TAB(36); "STORED IN C#(2*N(5)+1)"; FROM NODE"; Y; "TO NODE"; Z
930 GOSUB 1160
940 C=Y
950 Q1=Y
960 N(7)=2
970 IF C<>51 GOTO 1090
980 REM *
990 INPUT "NON-INVERTING INPUT NODE"; Y
1000 IF Y<0 OR Y>N4 THEN PRINT "ERROR! "; GOTO 980
1010 Y=INT(Y)
1020 Q2=Y
1030 REM *
1040 INPUT "INVERTING INPUT NODE"; Z
1050 IF Z<0 OR Z>N4 THEN PRINT "ERROR! "; GOTO 1030
1060 Z=INT(Z)
1070 Q3=Z
1080 GOSUB 1160
1090 REM * STORE NODES
1100 IF C3=51 THEN D1=1
1110 IF C3<>51 THEN D1=0; GOTO 1130
1120 PRINT "OF-RMP NODES: OUT -"; Q1; " NON-INV INPUT -"; Q2; " INV INPUT -"; Q
1130 C(2*N(5))=C+N(7)/1E2+(Y+1E2+Z/1E2)+D1
1140 REM *
1150 GOTO 540
1160 REM * ENCODE NODES
1170 IF Q1=2 THEN RETURN
1180 IF Y=2 GOTO 1230
1190 IF Y=N4 GOTO 1250
1200 IF Z=2 GOTO 1270

1210 IF Z=N4 GOTO 1290
1220 RETURN
1230 Y=N4
1240 GOTO 1200
1250 Y=2
1260 GOTO 1200
1270 Z=N4
1280 RETURN
1290 Z=2
1300 RETURN
1310 REM * FIRST SWEEP
1320 L1=Y1
1330 REM * NEXT SWEEP (REENTRY POINT)
1340 IFL1=1 THEN PRINT "LOG OR LIN SWEEP (ENTER 1 TO RETAIN LOG)";
1350 IFL1=0 THEN PRINT "LOG OR LIN SWEEP (ENTER 0 TO RETAIN LIN)";
1360 INPUT R#
1370 IF R#="0" GOTO 1420
1380 IF R#="LOG" THEN L1=1; GOTO 1440

```

```

1390 IF A$="LIN" THEN L1=0: GOTO 1440
1400 IF VAL(A$)=1 THEN A$="LOG": GOTO 1440
1410 GOTO 1360
1420 IF L1=1 THEN A$="LOG"
1430 IF L1=0 THEN A$="LIN"
1440 REM * ONWARDS
1450 PRINT A$; " SWEEP"
1460 PRINT "START FREQ="; F1; "HZ ";
1470 INPUT "INPUT NEW START FREQ [OR ^] TO SAVE OLD VALUE)"; F1#
1480 IF F1#="]" GOTO 1510
1490 F1=VAL(F1#)
1500 IF F1<0 GOTO 1470
1510 REM * END FREQUENCY
1520 PRINT "END FREQ="; F2; "HZ ";
1530 INPUT "INPUT NEW END FREQ [OR ^] TO SAVE OLD VALUE)"; F2#
1540 IF F2#="]" GOTO 1570
1550 F2=VAL(F2#)
1560 IF F2<F1 GOTO 1530
1570 REM * INTERVALS
1580 PRINT "INTERVALS="; I2;
1590 INPUT "INPUT NEW INTERVALS [OR ^] TO SAVE OLD VALUE)"; I2#
1600 IF I2#="]" GOTO 1630
1610 I2=VAL(I2#)
1620 IF I2<1 GOTO 1570
1630 I3=L1*EXP(LOG(F2/F1)/I2)
1640 IF L1<>0 THEN L2=0
1650 IF L1=0 THEN L2=1
1660 T3=I3*(L2+I2/F1)*2*PI/I2
1670 REM CLEAR THE SCREEN HERE (RADIO SHACK - CLS)
1680 REM (RS-232C TERM - CHR$(12)+CHR$(13))
1690 PRINT TAB(126); CHR$(28);
1700 FOR WA=1 TO 10: NEXT WA: REM WAIT LOOP IF NECESSARY
1710 PRINT "FREQ IF NOT CHANGED TO "; TAB(39); "PHASE (DEG)";
1720 PRINT TAB(56); "TIME (MICROSEC'S)"
1730 REM START PLOT
1740 PRINT "CALCULATING"
1750 O3=2*PI*F1
1760 REM CLEAR MATRIX
1770 FOR X=1 TO 4
1780 FOR B=1 TO 4
1790 Y(X,B)=0
1800 Z(X,B)=0

```

```

1810 NEXT B
1820 NEXT X
1830 X=0
1840 IF N(1)<1 GOTO 1890
1850 FOR B=1 TO N(1)
1860 N(9)=1/C(2*B-1)
1870 GOSUB 1900
1880 NEXT B
1890 GOTO 2110
1900 REM * CALCULATE MATRIX
1910 Z=C(2*B)
1920 C=INT(Z)
1930 REM THE .00001 FACTOR CORRECTS ROUNDING ERROR
1940 REM IN MICROSOFT AND ALL 9 DIGIT BINARY BASICS
1950 REM THIS STEP IS MANDATORY, BUT MAY BE HANDLED WITH A VARIABLE.
1960 Z=INT(1E2*(Z-C+.00001))
1970 IF X<>0 GOTO 2000
1980 Y(C,C)=Y(C,C)+N(9)
1990 GOTO 2010
2000 Z(C,C)=Z(C,C)+N(9)
2010 IF Z=0 THEN X=X+1
2020 IF X<>0 GOTO 2070
2030 Y(C,Z)=Y(C,Z)-N(9)
2040 Y(Z,C)=Y(Z,C)-N(9)
2050 Y(Z,Z)=Y(Z,Z)+N(9)
2060 RETURN
2070 Z(C,Z)=Z(C,Z)-N(9)
2080 Z(Z,C)=Z(Z,C)-N(9)
2090 Z(Z,Z)=Z(Z,Z)+N(9)
2100 RETURN
2110 X=1
2120 IF N(1)+1>N(2)+N(1) GOTO 2170
2130 FOR B=N(1)+1 TO N(2)+N(1)
2140 N(9)=03/C(2*B-1)
2150 GOSUB 1900
2160 NEXT B
2170 IF N(5)-N(4)<N(2)+N(1)+1 GOTO 2240
2180 FOR B=N(2)+N(1)+1 TO N(5)-N(4)
2190 N(9)=-1/C(2*B-1)*03)
2200 GOSUB 1900
2210 NEXT B
2220 REM
2230 IF N(5)<N(5)-N(4)+1 GOTO 2460
2240 FOR B=N(5)-N(4)+1 TO N(5)
2250 Z=C(2*B)
2260 X=INT(Z)
2270 Z=Z-X+.00001
2280 N(6)=INT(X/100)
2290 N(8)=INT(X-N(6)*100)
2300 N(9)=INT(Z*100)
2310 N(7)=INT(Z*10000-N(9)*100)
2320 C=-3
2330 X=-3
2340 C=C+2
2350 IF C>1 GOTO 2440
2360 X=X+2
2370 IF X>1 GOTO 2330
2380 Y=N((C+13)/2)
2390 IF Y=0 GOTO 2360
2400 Z=N((X+17)/2)

```



```

2410 IF Z=0 GOTO 2360
2420 Y(Z,Y)=Y(Z,Y)+C(2*B-1)*D*X
2430 GOTO 2360
2440 NEXT Y
2450
2460 PRINT
2470
2480
2490
2500 D=N(8)*N(8)+N(9)*N(9)
2510 C=0
2520 IF B<>2 GOTO 2570
2530 F3=1
2540 X=1
2550 C=2
2560
2570
2580
2590 IF C<>B GOTO 2620
2600
2610 GOTO 2460
2620
2630 IF C=1 GOTO 2670
2640 N(6)=Z(C,B)
2650 N(7)=Z(B,X)
2660 Y=(1-2*F3)*Y(C,B)*Y(B,X)-N(6)*N(7)
2670 Z=(1-2*F3)*Y(C,B)*N(7)+N(6)*Y(B,X)
2680 IF D=0 THEN D=1E-37 :REM UNDERFLOW
2690 IF F3=1 THEN
2700 Y(C,X)=Y(C,X)-(Y*N(8)+Z*N(9))/D
2710 Z(C,X)=Z(C,X)-(Z*N(8)-Y*N(9))/D
2720 GOTO 2620
2730 Y=Y/D
2740 Z=Z/D
2750 X=SQR(Y*Y+Z*Z)
2760 IF X<>0 GOTO 2800
2770 REM NO CONTINUITY FLAG
2780 PRINT "NO CONTINUITY FOR F=";O3/(2*PI); "HZ"
2790 GOTO 2930
2800 IF Z>0 THEN Z4=1
2810 IF Z<0 THEN Z4=0
2820 C=(2*Z4-1)*PI/2
2830 IF Y=0 GOTO 2890
2840 C=ATN(Z/Y)
2850 IF Y<=0 GOTO 2890
2860 C=C+PI
2870 IF Z<=0 GOTO 2890
2880 C=C-2*PI
2890 REM OUT DATA
2900 DB=(INT(2E3*LOG(X)/LN+.5))/100
2910 REM ADJUST TABS TO SUIT VIDEO DISPLAY
2920 PRINT O3/(2*PI);TAB(19);DB;TAB(39);180*C/PI;TAB(56);1E6*C/O3
2930 REM INCREMENT OMEGA
2940 IF O3>=2*PI*(F2+.004) GOTO 3030
2950 IF L1<>0 THEN L5=0
2960 IF L1=0 THEN L5=1
2970 O3=L1*O3*I3+L5*(O3+I3)
2980 IF O3>2*PI*F2 GOTO 3000
2990 GOTO 1760
3000 REM * LAST POINT

```

3010 03=2\*PI\*F2

3020 GOTO 1760

3030 INPUT "ENTER 'J' FOR NEW SWEEP"; R#

3040 IF R#="J" GOTO 1760