

เอกสารอ้างอิง

1. K.S. Chao, D.K. Liu, C.T. Pan, "A symmetrical search method for obtaining multiple solutions of simultaneous nonlinear equations," IEEE Trans. Circuits and Systems, Vol. CAS-22, No.9, pp. 748-753, Sept. 1975
2. M.J.D. Powel, "An efficient method for finding the minimum of a function for several variables without calculating derivatives," Computer J., Vol. 7, pp. 155-162, July 1964
3. J.W. Bandler, "Optimization methods for computer aided design," IEEE Trans. Microwave Theory And Techniques, Vol. MTT-17, No. 8, pp.533-551, August 1969
4. A.K. Seth and K. Singhal, "Time-domain sensitivity using the adjoint network," Electronics Letters, pp.563-656, Sept. 1971
5. F.H. Branin, Jr., "Network sensitivity and noise analysis simplified," IEEE Trans Circuit Theory, Vol, CT-20, No. 3, pp. 285-288, May 1973
6. S.W. Director, "LU factorization in network sensitivity computations," IEEE Trans. Circuit Theory, Vol. CT-18, pp. 184-185, Jan 1971
7. S.R.K. Dutta and M. Vidyasagar, "Optimal design of nonlinear DC transistor circuits without solving network equations," IEEE Trans. Circuits and Systems, Vol. CAS-22, No. 8, Aug 1975
8. W.L. Demars and A.R. Thorbjornsen, "Circuit design in the time domain using frequency domain analysis data," IEEE Proc. pp. 33-35, 1975 ISCAS
9. S.W. Director and R.A. Rohrer, "Automated network design-the frequency domain case," IEEE Trans. Circuit Theory, Vol. GT-16, pp. 330-337, Aug. 1969
10. K.U. Wang and F.H. Branin, Jr., "A new hybrid formulation of the network equations," IEEE Conference, Mexico 1971

11. M. Nakhla, K. singhal and J. Vlach, "An optimal pivoting order for the solution of sparse system of equations, "IEEE Trans. Vol. CAS-21, No. 2, pp. 222-225, March 1974
12. F.H. Branin, Jr., G.H. Hogsett, R.L. Lunde, and L.E. Kugel, " ECAP II-A New electronics Circuit Analysis Program," IEEE Trans, Solid state Circuits, Vol. SC-6, pp.146-166, Aug 1971
13. W.M.G. Van Bokhoven, "Linear implicit differentiation formulas of variable step and order," IEEE Trans, Circuits and Systems, Vol. CAS-22, No. 2, Feb 1975
14. D.A. Calahan, Computer-Aided Network Design, New York, McGraw-Hill, 1972
15. L. Cooper and D. Steinberg, Introduction to Methods of optimization. Philadelphia, W.B. Saunders, 1970
16. S.D. Conte and C. de Boor, Elementary numerical analysis, Second Edition, New York, McGraw-Hill, 1972
17. สุขุมวิทย์ ภูมิวิศิสาร, การใช้คอมพิวเตอร์วิเคราะห์วงจรนอนลิเนียร์และค่าพารามิเตอร์แปรตามเวลา, รายงานฉบับสมบูรณ์ของโครงการวิจัย ทุนอุดหนุนการวิจัยของสำนักงานคณะกรรมการวิจัยแห่งชาติ, มิถุนายน 2526

ภาคผนวก ก.

การหาอนุพันธ์ของ Implicit Function

การประมาณค่าสมการโพลีโนเมียลอันดับ 2 (2^{nd} order polynomial) อาจทำ

ได้ดังนี้

$$x(t) = at^2 + bt + c \quad (\text{ก.1})$$

โดยที่ $x(t)$ คือสมการโพลีโนเมียลอันดับ 2

และให้ $x_k = x(t_k) = at_k^2 + bt_k + c \quad (\text{ก.2})$

$$x_{k-1} = x(t_{k-1}) = at_{k-1}^2 + bt_{k-1} + c \quad (\text{ก.3})$$

$$x_{k-2} = x(t_{k-2}) = at_{k-2}^2 + bt_{k-2} + c \quad (\text{ก.4})$$

$$\dot{x}_k = dx(t_k)/dt = 2at_k + b \quad (\text{ก.5})$$

โดยการพิจารณาสมการ (ก.2), (ก.3) และ (ก.4) และแก้สมการหาค่าพาดเตอร์ a, b

$$a = \frac{[(t_{k-1} - t_{k-2})(x_k - x_{k-1}) - (t_k - t_{k-1})(x_{k-1} - x_{k-2})]}{[(t_k - t_{k-1})(t_{k-1} - t_{k-2})(t_k - t_{k-2})]}$$

$$b = -\frac{[(t_{k-1}^2 - t_{k-2}^2)(x_k - x_{k-1}) - (t_k^2 - t_{k-1}^2)(x_{k-1} - x_{k-2})]}{[(t_k - t_{k-1})(t_{k-1} - t_{k-2})(t_k - t_{k-2})]}$$

โดยให้ $h_k = t_k - t_{k-1}$ และ $h_{k-1} = t_{k-1} - t_{k-2}$

โดยการแทนค่าพาดเตอร์ a และ b กลับไปในสมการ (ก.5) เราจะได้สมการต่อไปนี้

$$x_k = \frac{[(2h_k + h_{k-1})h_{k-1}x_k - (h_k + h_{k-1})^2 x_{k-1} + h_k^2 x_{k-2}]}{[h_k h_{k-1} (h_k + h_{k-2})]} \quad (\text{ก.6})$$

ภาคผนวก ข.

การคำนวณค่าต่ำสุดสำหรับอนุพันธ์ของ Quadratic Interpolation

ให้ x_0, x_1, \dots, x_n คือเลขจำนวนจริงบวกบนแกนของเลขจำนวนจริง (real axis) และให้ $f(x)$ คือฟังก์ชันของ x ซึ่งอยู่ในช่วงระหว่าง a และ b Lagrange โพลีโนเมียลของ $p(x)$ ซึ่งมีองศา (order) น้อยกว่าหรือเท่ากับ n อาจเขียนรูปสมการได้ดังนี้

$$p(x) = \sum_{k=0}^n f(x_k) L_k(x) \quad (\text{ข.1})$$

$$L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^n (x-x_i)/(x_k-x_i) \quad (\text{ข.2})$$

L_k คือโพลีโนเมียลของ Lagrange สำหรับ x_0, x_1, \dots, x_n สำหรับวิธี Quadratic Interpolation n จะเท่ากับ 2 และโดยการรวมสมการ (ข.1) และ (ข.2) เราจะได้

$$\begin{aligned} p(x) &= \sum_{k=0}^2 \prod_{\substack{i=0 \\ i \neq k}}^2 (x-x_i)/(x_k-x_i) \\ &= f(x_0)(x-x_1)(x-x_2)/[(x_0-x_1)(x_0-x_2)] + f(x_1)(x-x_0) \\ &\quad (x-x_2)/[(x_1-x_0)(x_1-x_2)] + f(x_2)(x-x_0)(x-x_1)/ \\ &\quad [(x_2-x_0)(x_2-x_1)] \end{aligned} \quad (\text{ข.3})$$

เราต้องการสมการของ $p(x)$ สำหรับค่าต่ำสุดนั้นคือ

$$dp(x)/dx = 0 \quad d^2 p(x)/dx^2 > 0$$

ดังนั้นโดยการดิฟเฟอเรนเชียลสมการ (ข.3) เทียบกับฟังก์ชัน x และให้เท่ากับศูนย์ เราจะได้สมการดังนี้

$$\begin{aligned} p'(x) &= \left[f(x_0)(2x-(x_1+x_2))(x_2-x_1) + f(x_1)(2x-(x_0+x_2)) \right. \\ &\quad \left. (x_0-x_2) + f(x_2)(2x-(x_0+x_1))(x_1-x_0) \right] / (x_0-x_1) \\ &\quad (x_1-x_2)(x_2-x_0) = 0 \\ f(x_0)(2x-(x_1+x_2))(x_2-x_1) + f(x_1)(2x-(x_0+x_2))(x_0-x_2) + f(x_2) \\ &\quad (2x-(x_0+x_1))(x_1-x_0) = 0 \end{aligned} \quad (\text{ข.4})$$

โดยการแก้สมการ (ข.4) อาจหาค่า x ได้ดังนี้คือ

$$x = (1/2) \left[f(x_0)(x_2^2-x_1^2) + f(x_1)(x_0^2-x_2^2) + f(x_2)(x_1^2-x_0^2) \right] / \left[f(x_0)(x_2-x_1) + f(x_1)(x_0-x_2) + f(x_2)(x_1-x_0) \right]$$

ภาคผนวก ค.

การแก้สมการ simultaneous โดยวิธี LU Decomposition

สมการ simultaneous โดยทั่วไปอาจอยู่ในรูปสมการเมทริกซ์ต่อไปนี้

$$Ax = b \quad (\text{ค.1})$$

เมทริกซ์ A จะมีมิติ $n \times b$ โดยที่ n คือจำนวนโหนดและ b คือจำนวนบรรทัด ด้วยวิธี LU Decomposition อาจแบ่งเมทริกซ์ A ออกเป็นเมทริกซ์ย่อย 2 ส่วน ดังนี้คือ

$$A = LU \quad (\text{ค.2})$$

โดยการแทนค่า A จากสมการ (ค.2) กลับไปในสมการ (ค.1) และจัดเรียงสมมติค่าบางส่วนเราจะได้สมการดังนี้ คือ

$$LUX = b \quad (\text{ค.3})$$

$$\text{ให้ } Y = UX \quad (\text{ค.4})$$

$$\text{ดังนั้น } LY = b \quad (\text{ค.5})$$

ด้วยวิธีเปลี่ยนแปลงสมการ (ค.1) และจัดเรียงสมการเสียใหม่ให้เป็นตามสมการ (ค.3), (ค.4) และ (ค.5) และด้วยวิธีแทนค่าไปและกลับ (forward and backward substitution) อาจแก้สมการ simultaneous โดยใช้ขั้นตอนต่อไปนี้

- 1) หาค่า y โดยการแก้สมการ (ค.5) ด้วยวิธีแทนค่าไปข้างหน้า (forward substitution)
- 2) หลังจากหาค่า y แล้ว อาจหาค่า x โดยการแก้สมการ (ค.4) ด้วยวิธีแทนค่ากลับ (backward substitution)

เมทริกซ์ A ซึ่งถูกแบ่งออกเป็นเมทริกซ์ L และ เมทริกซ์ U ดังสมการ (ค.2) อาจทำได้โดยใช้วิธีการดังต่อไปนี้ คือ สมมติต้องการแบ่งเมทริกซ์ A ที่มีมิติ 4×4 ออกเป็นเมทริกซ์ L และ U ดังวิธีดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1_{11} & 0 & 0 & 0 \\ 1_{21} & 1_{22} & 0 & 0 \\ 1_{31} & 1_{32} & 1_{33} & 0 \\ 1_{41} & 1_{42} & 1_{43} & 1_{44} \end{bmatrix} \begin{bmatrix} 1 & U_{12} & U_{13} & U_{14} \\ 0 & 1 & U_{23} & U_{24} \\ 0 & 0 & 1 & U_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} l_{11} & l_{11}U_{12} & l_{11}U_{13} & l_{11}U_{14} \\ l_{21} & l_{21}U_{12}+l_{22} & l_{21}U_{13}+l_{22}U_{23} & l_{21}U_{14}+l_{32}U_{24}+l_{33}U_{34} \\ l_{31} & l_{31}U_{12}+l_{32} & l_{31}U_{13}+l_{32}U_{23}+l_{33} & l_{31}U_{14}+l_{32}U_{24}+l_{33}U_{34} \\ l_{41} & l_{41}U_{12}+l_{42} & l_{41}U_{13}+l_{42}U_{23}+l_{43} & l_{41}U_{14}+l_{42}U_{24}+l_{43}U_{34}+l_{44} \end{bmatrix} \quad (\text{ค.6})$$

ดังนั้นเราอาจสรุปได้ว่า

$$\begin{aligned} l_{i1} &= a_{i1} & i &= 1, \dots, 4 \\ U_{12} &= a_{12}/l_{11} \\ l_{22} &= a_{22}-l_{21}U_{12} \\ l_{32} &= a_{32}-l_{31}U_{12} \\ l_{42} &= a_{42}-l_{41}U_{12} \\ U_{13} &= a_{13}/l_{11} \\ U_{23} &= (a_{23}-l_{21}U_{13})/l_{22} \\ l_{33} &= a_{33}-l_{31}U_{13}-l_{32}U_{23} \\ l_{43} &= a_{43}-l_{41}U_{13}-l_{42}U_{23} \end{aligned} \quad (\text{ค.7})$$

โดยการพิจารณาสมการ (ค.6) และ (ค.7) อาจเขียนคอสมันซ์ของเมตริกซ์ L และเมตริกซ์ U

ให้อยู่ในรูปสมการ ซึ่งอาจนำไปใช้กับเมตริกซ์ที่มีมิติ $n \times n$ ได้ดังนี้

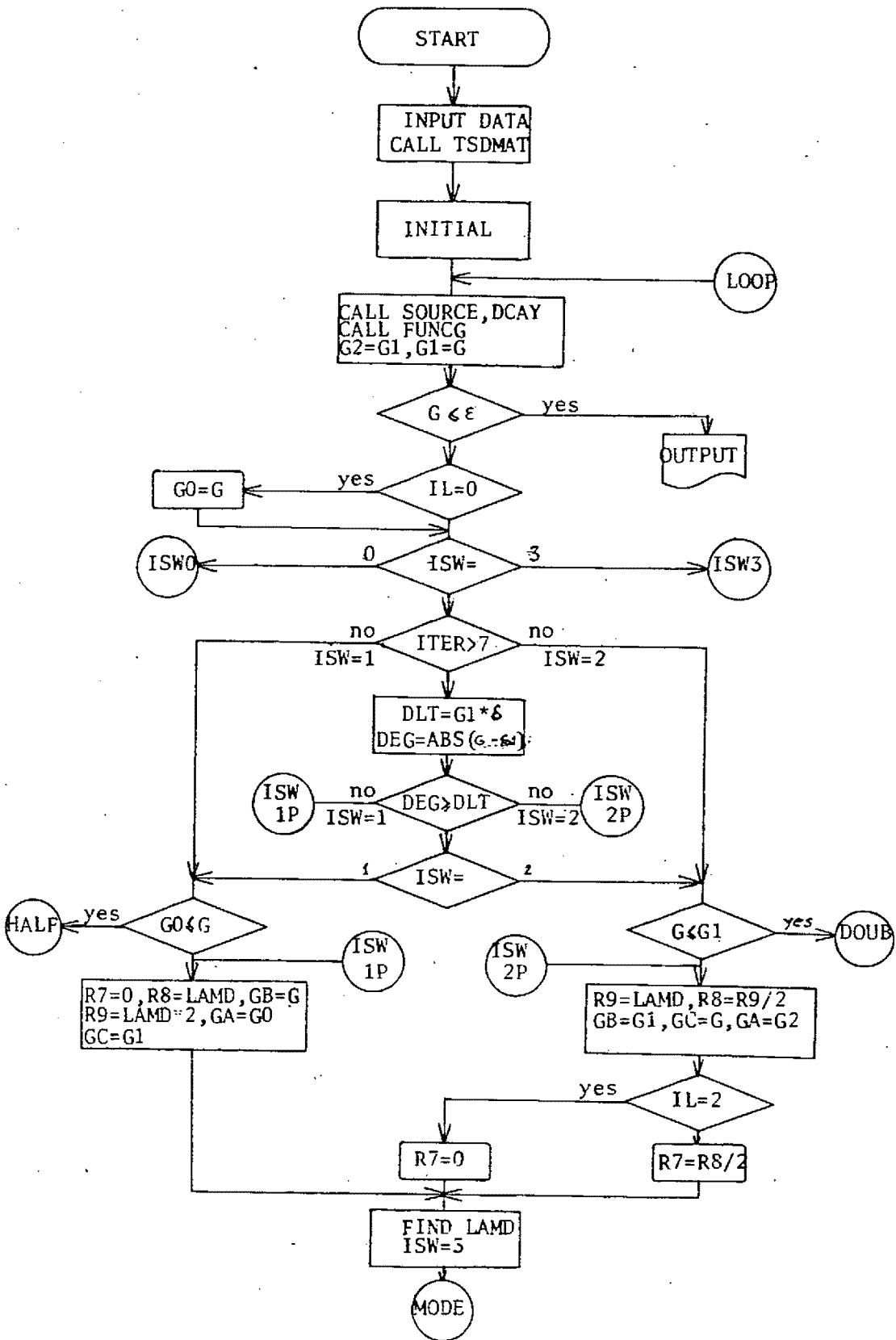
$$\begin{aligned} U_{ij} &= (a_{ij} - \sum_{k=1}^{i-1} l_{ik}U_{kj})/l_{ii} & i < j \\ l_{ij} &= a_{ij} - \sum_{k=1}^{j-1} l_{ik}U_{kj} & i \geq j \end{aligned}$$

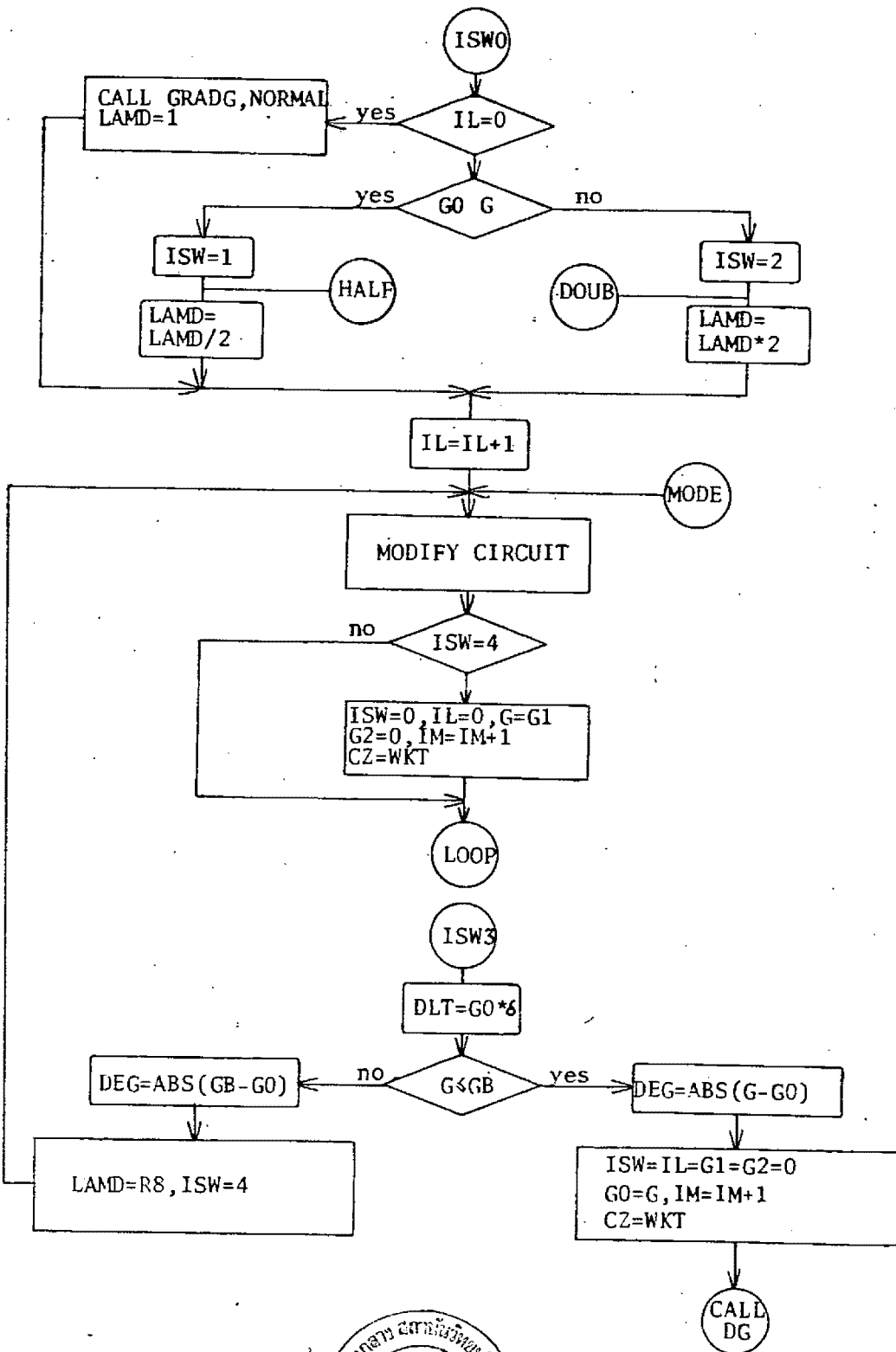
ภาคผนวก ง.

สัญลักษณ์ที่ใช้ในโปรแกรมและไฟล์วาร์ท

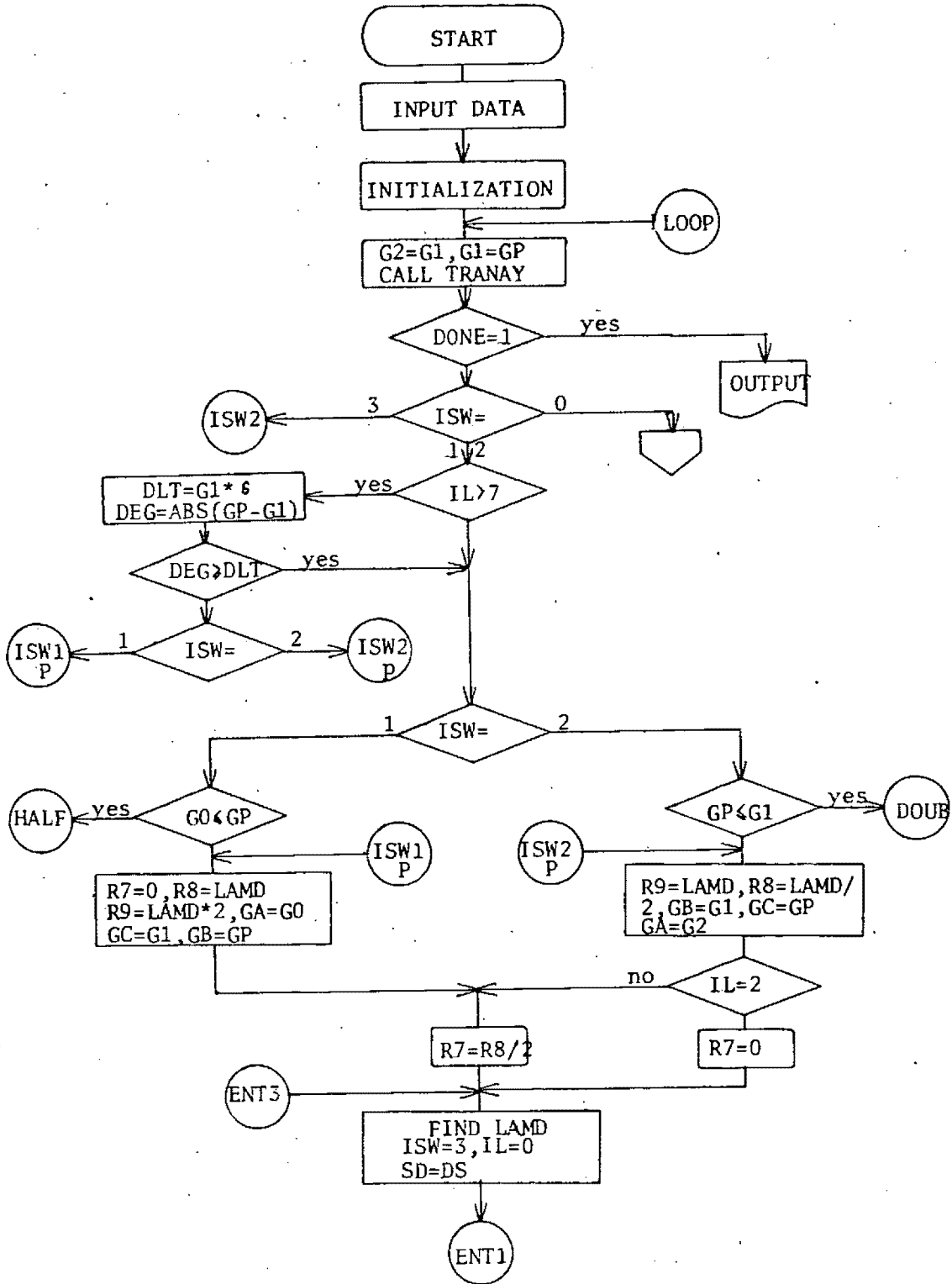
สัญลักษณ์ที่ใช้ในโปรแกรม

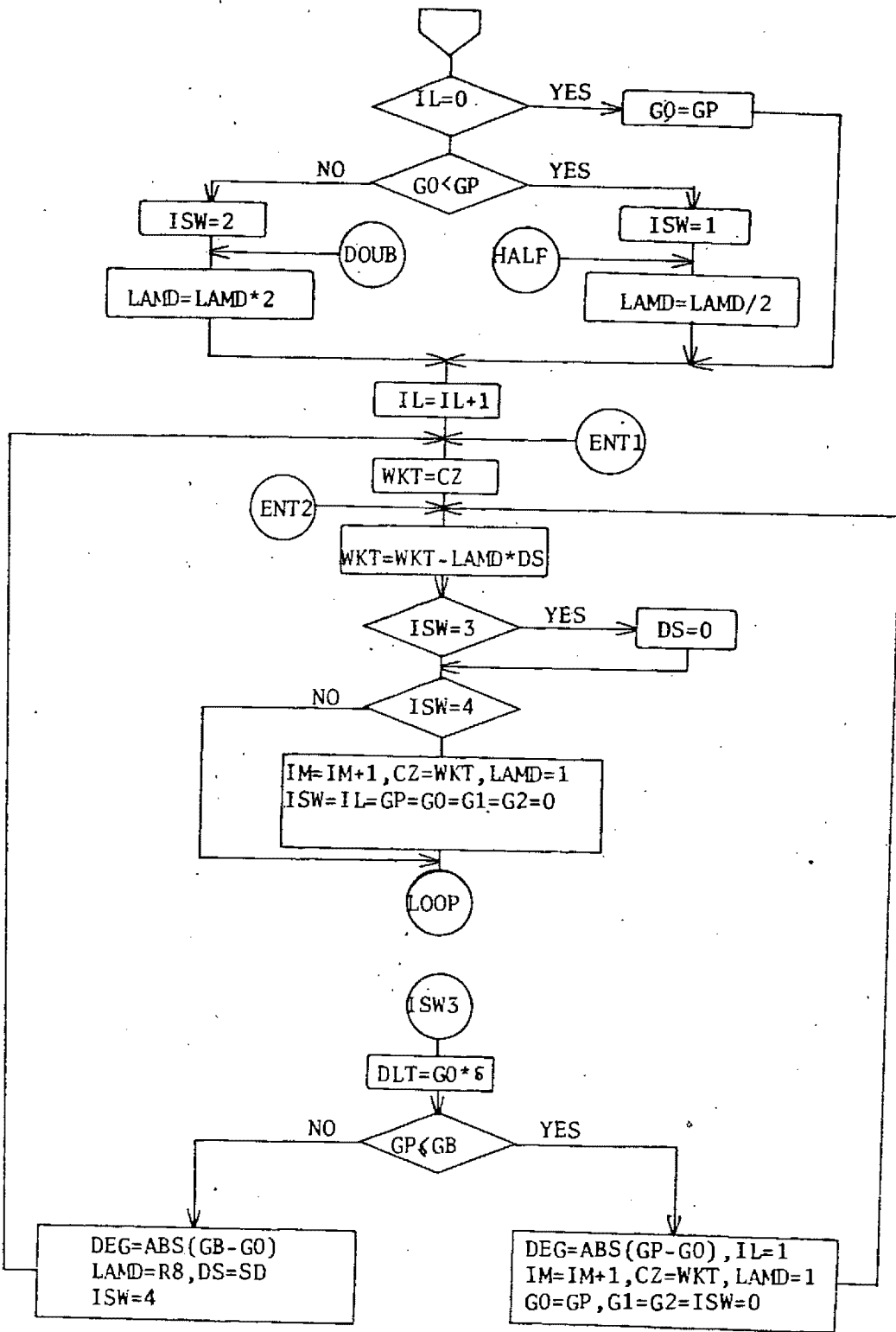
AID	คือแฟลคซึ่งแสดงว่า เซอร์กิตประกอบด้วยนอนลิเนียร์ เอลเมนทหรือไม่
CPIV	คือพีโวกเวคเตอร์ทางแนวคออัมน์
MT,MS,MD	คือเมตริกซ์โครงสร้าง T,S และ D ตามลำดับ
XM,WM,MM	คือเวคเตอร์ของ x,W และ m ตามลำดับ
KB	คือเคาน์เตอร์ในโดเมนของเวลา
RTEP	คือแฟคเตอร์ r ในการออกแบบวงจรไฟฟ้าโดยอัตโนมัติ
LAMD	คือแฟคเตอร์ r ในการวิเคราะห์ท่วงจรไฟฟ้า
DXM,DWM,DMM	คือ $dx/dt, dW/dt$ และ dm/dt ตามลำดับ
DV,DC	คือ dv/dt และ di/dt ตามลำดับ
D8,D9	คือ Δv และ Δi ตามลำดับ
DGDP	คือ ∇g
EM	คือเวคเตอร์ของความผิดพลาด
ESD	คือเวคเตอร์ของ e_i^t s หรือ $e_i^t D$
ETSI	คือเวคเตอร์ของ $e_i^t, T^t e_i$ หรือ se_i
MODE	คือสตริง (string) ประกอบด้วย 'AU' หรือ 'AY' สำหรับการ ออกแบบวงจรโดยอัตโนมัติ หรือการวิเคราะห์ท่วงจรตามลำดับ
HM	คือเมตริกซ์ของ H_x, H_w, H_x และ H_w
H1,H2	คือ $(\partial H_n / \partial W) S$ และ $(\partial H_n / \partial W) S$ ตามลำดับ
HDP	คือ $-(\partial H / \partial P + H_w S \partial f / \partial p + H_w S \partial f / \partial p)$ หรือ $-(\partial H / \partial p + H_w S \partial f / \partial p)$
SI	คือ $(dx/dp)_i, (dW/dp)_i$ หรือ $(dm/dp)_i$

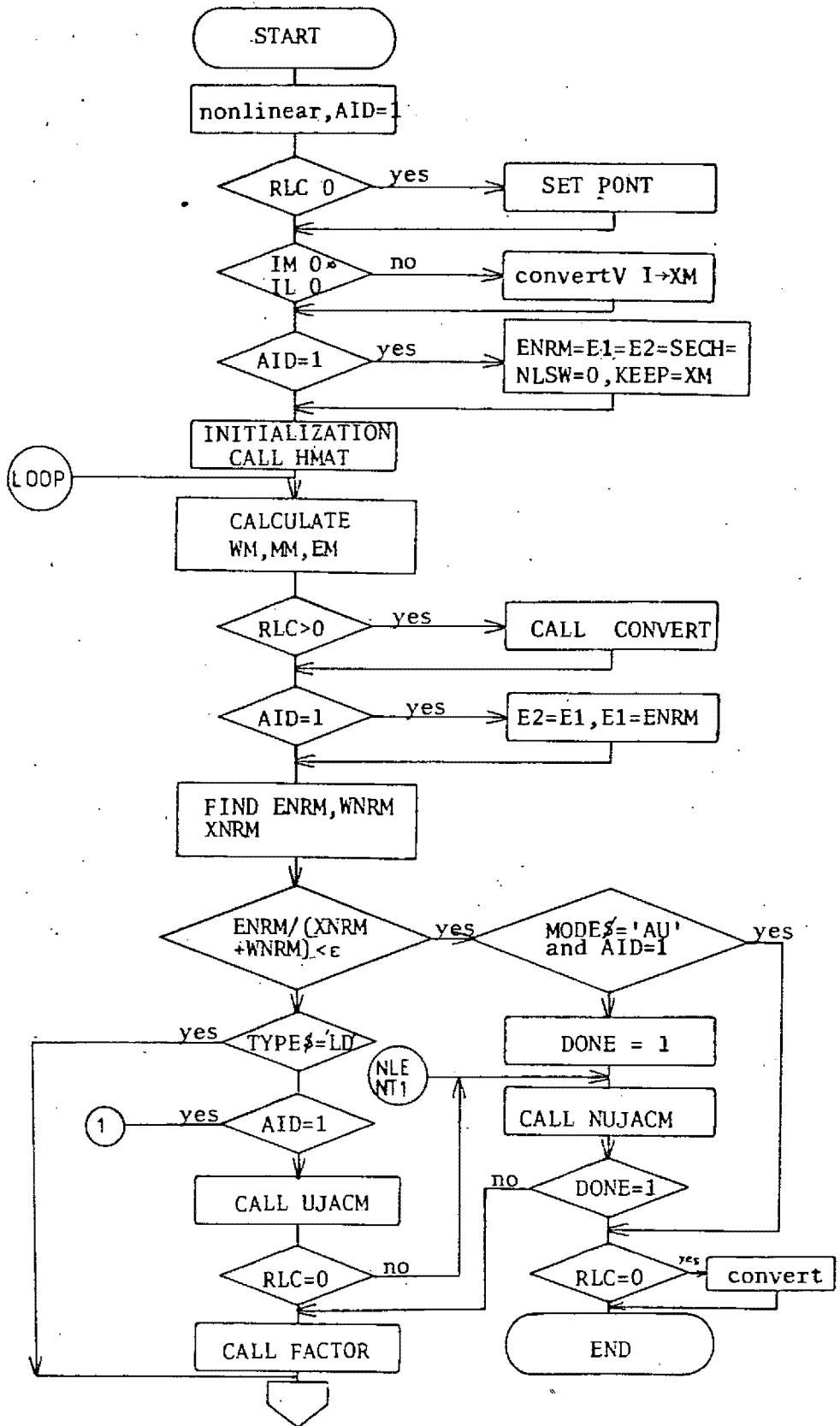


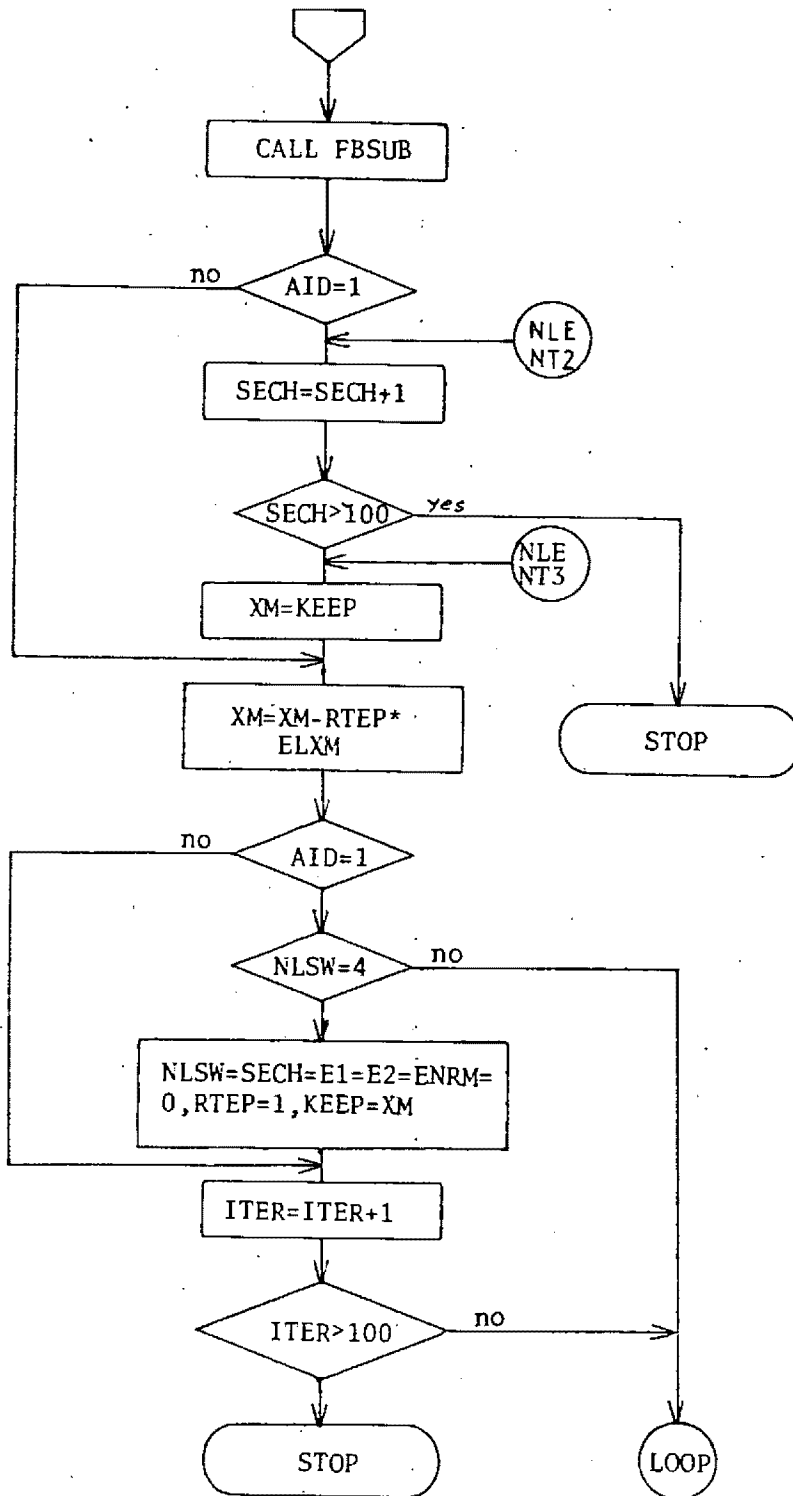


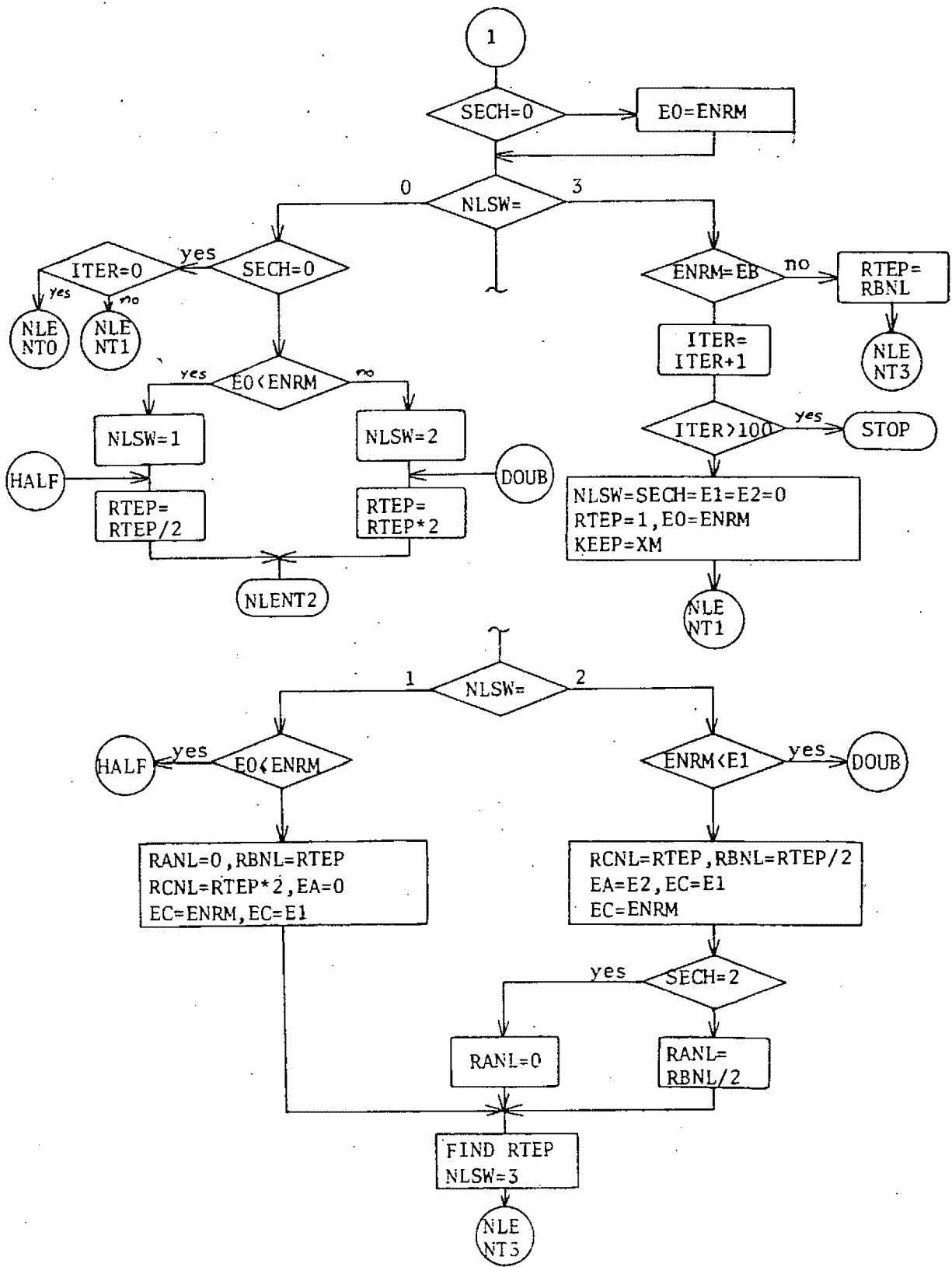
โปรแกรม AIRTRAN

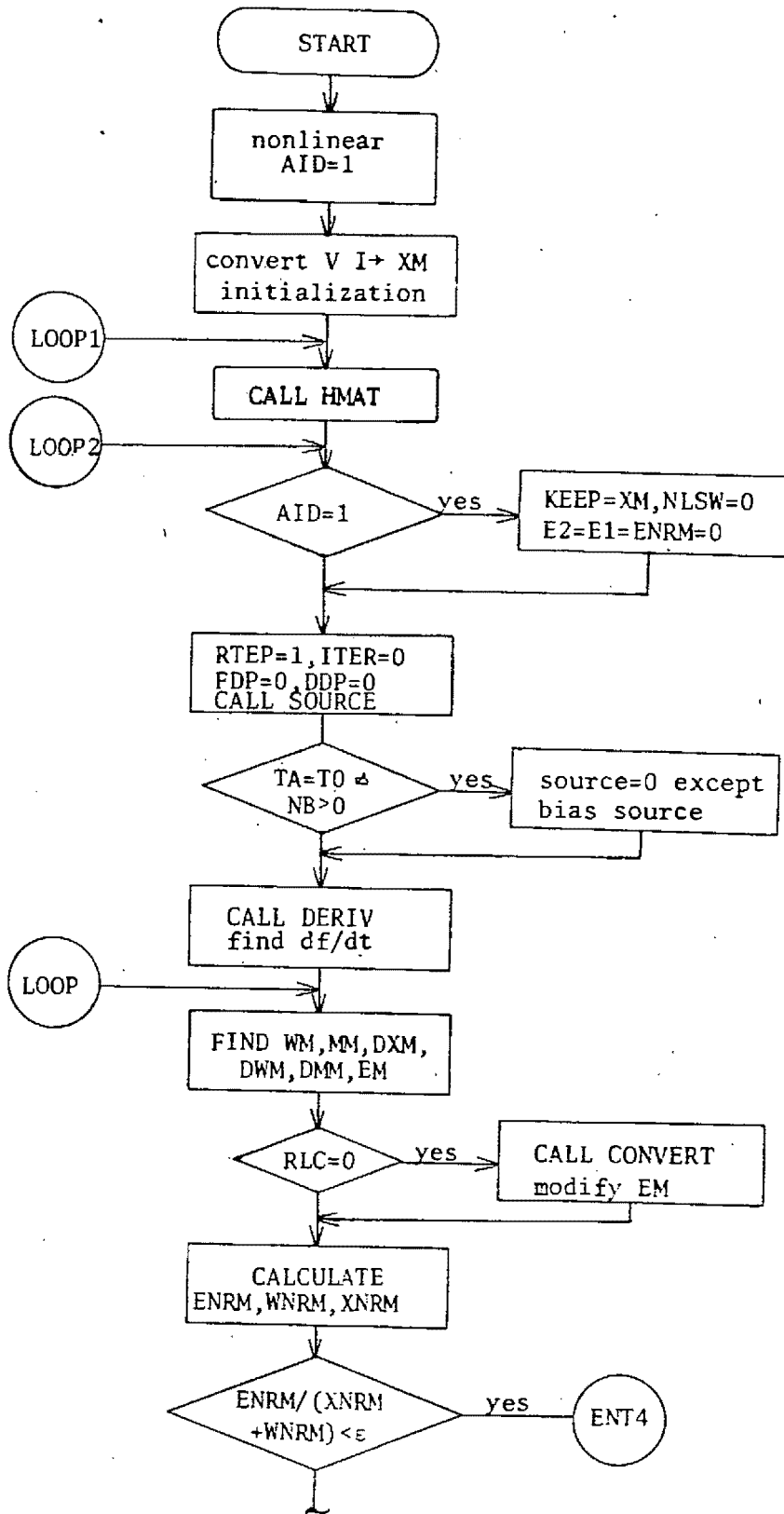


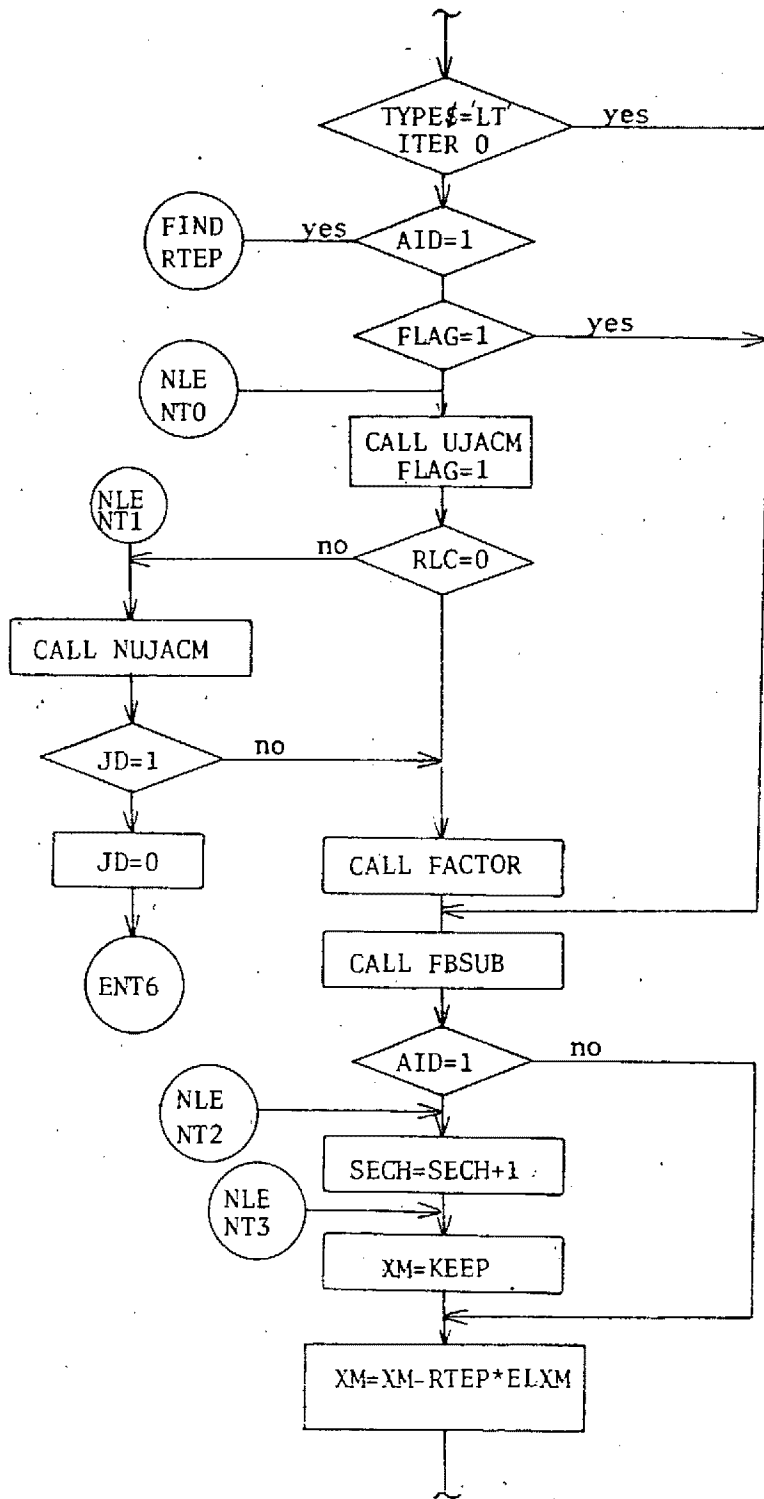


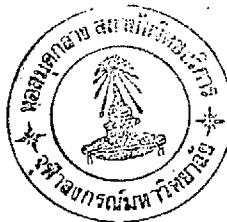
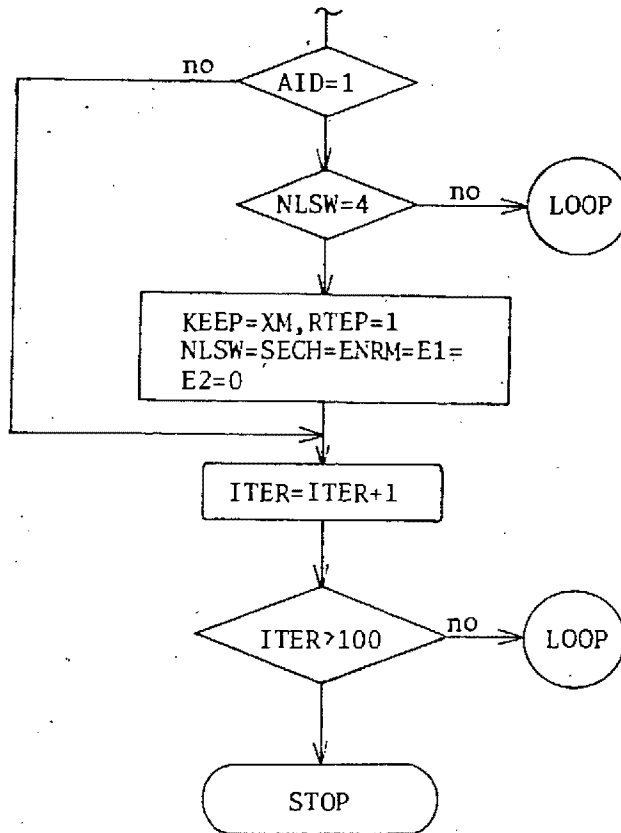


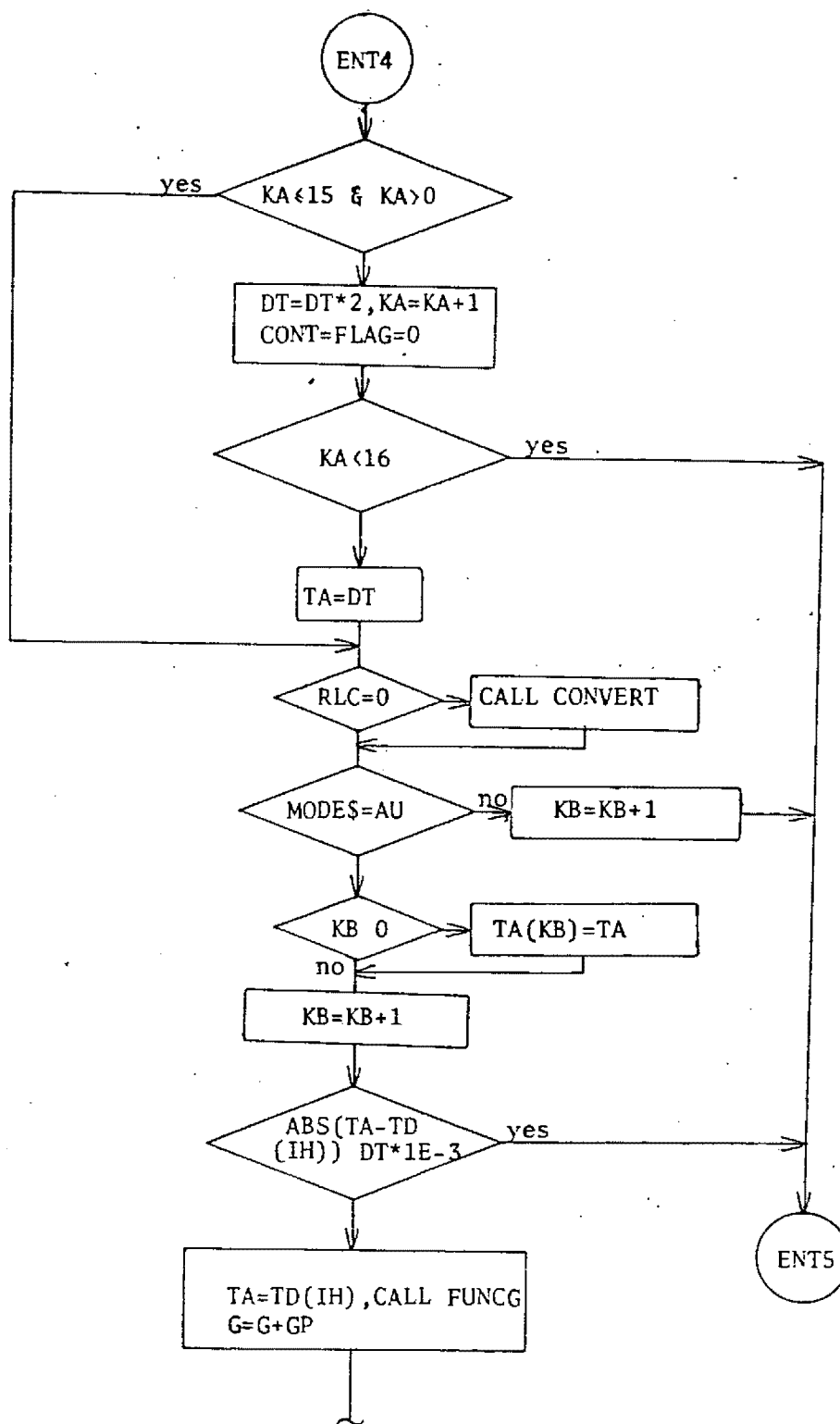


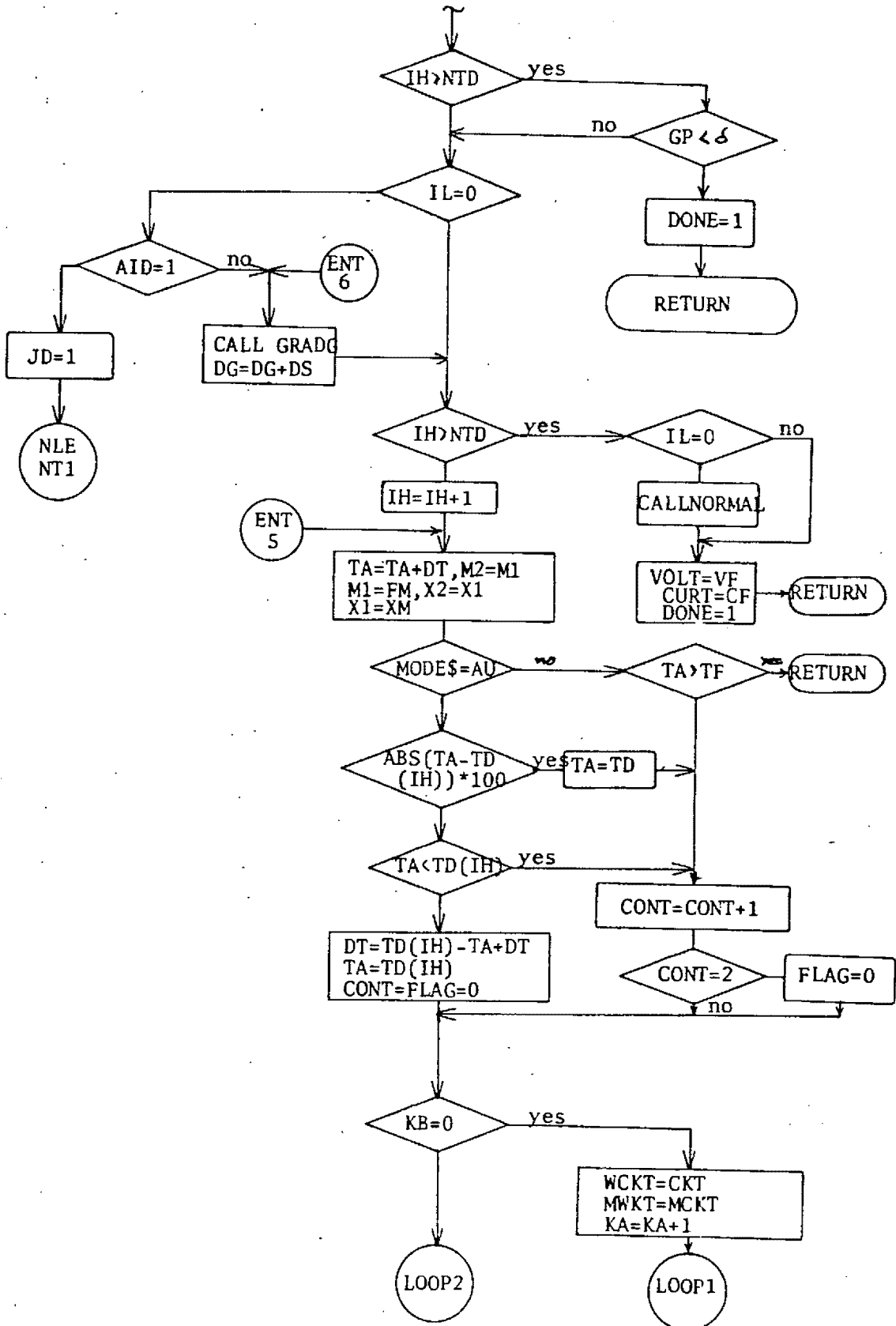


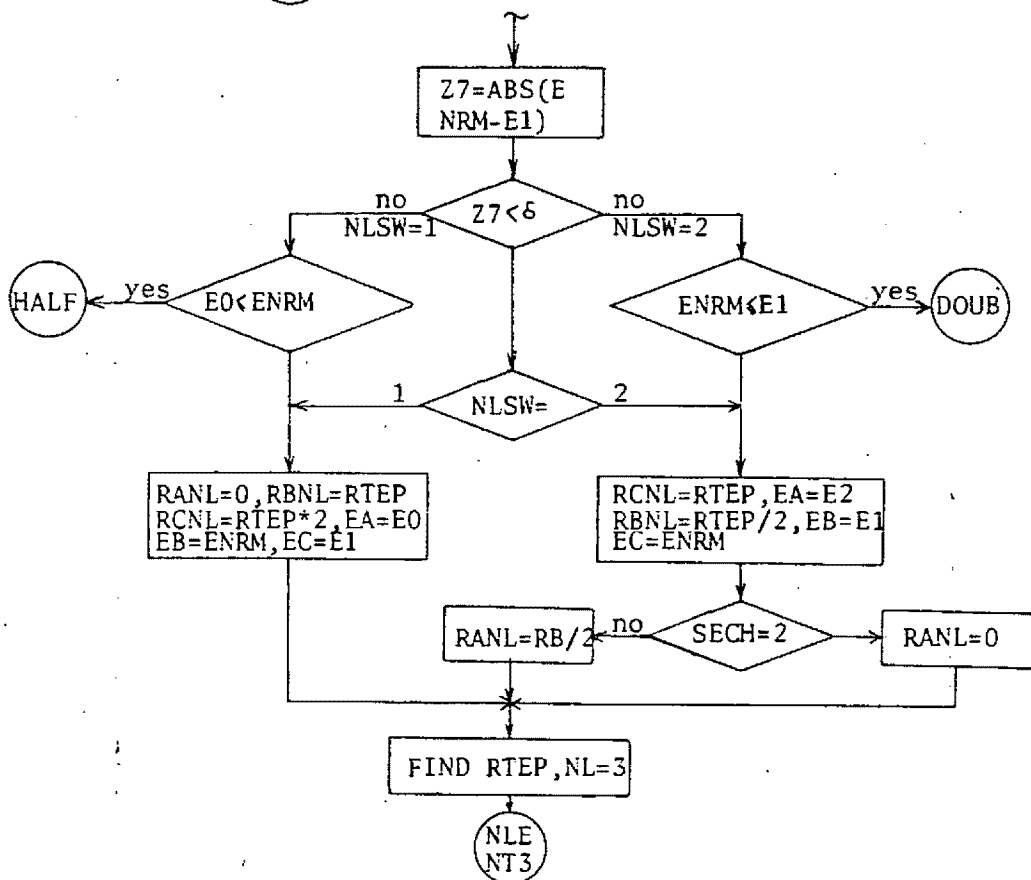
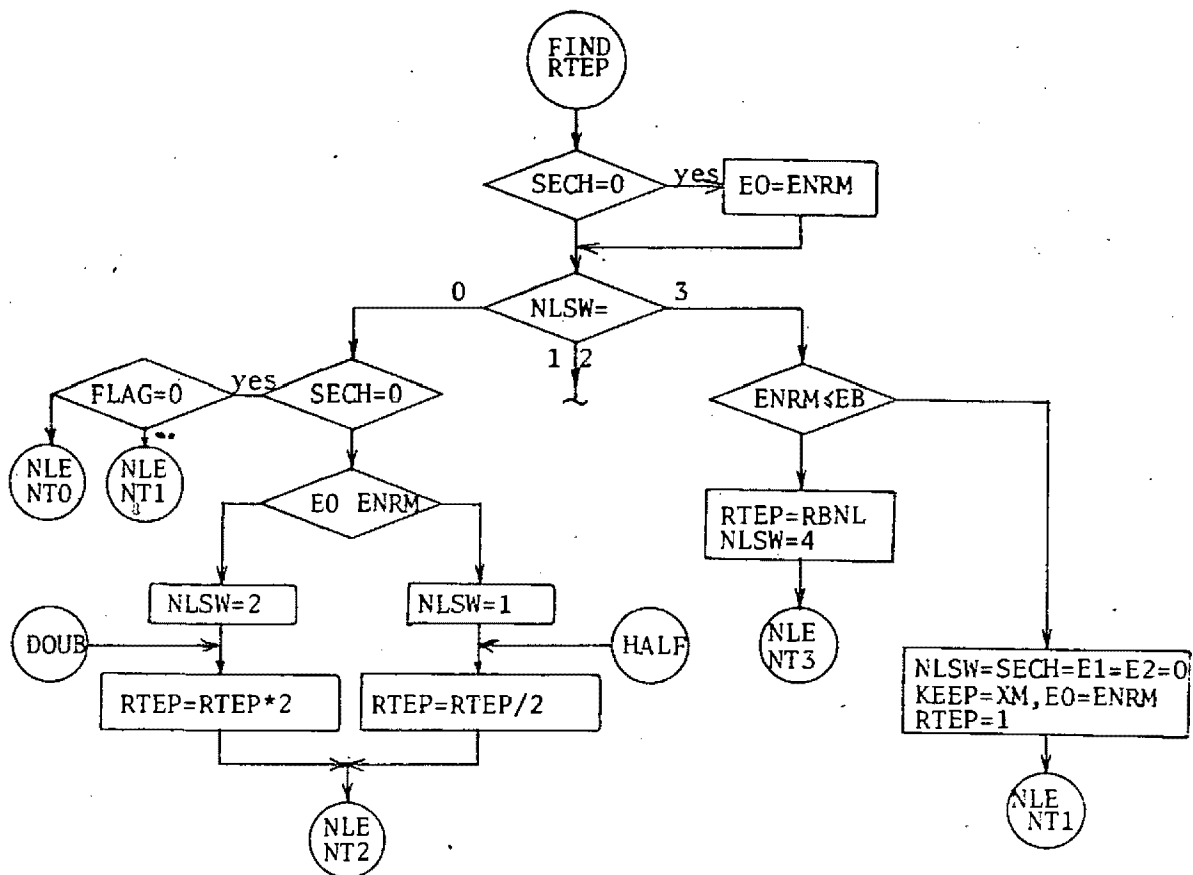


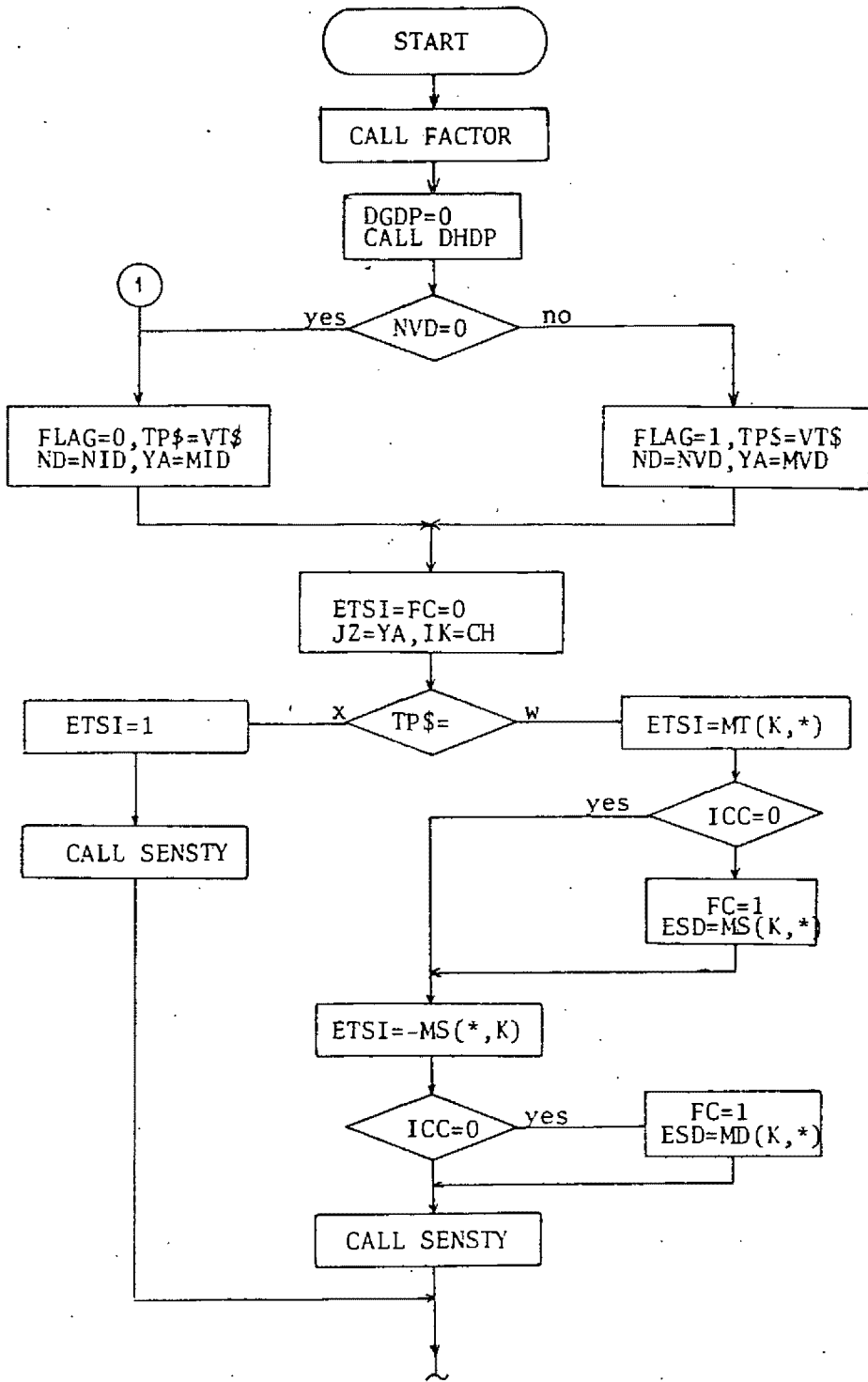


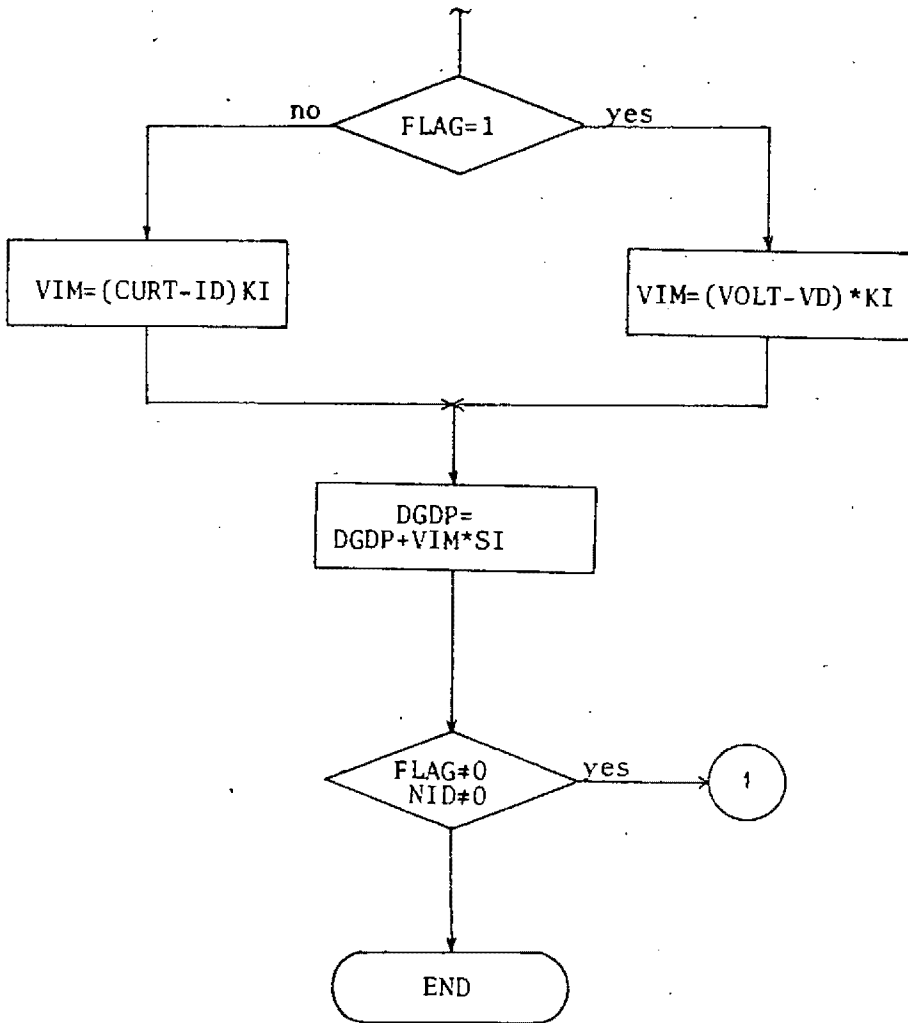


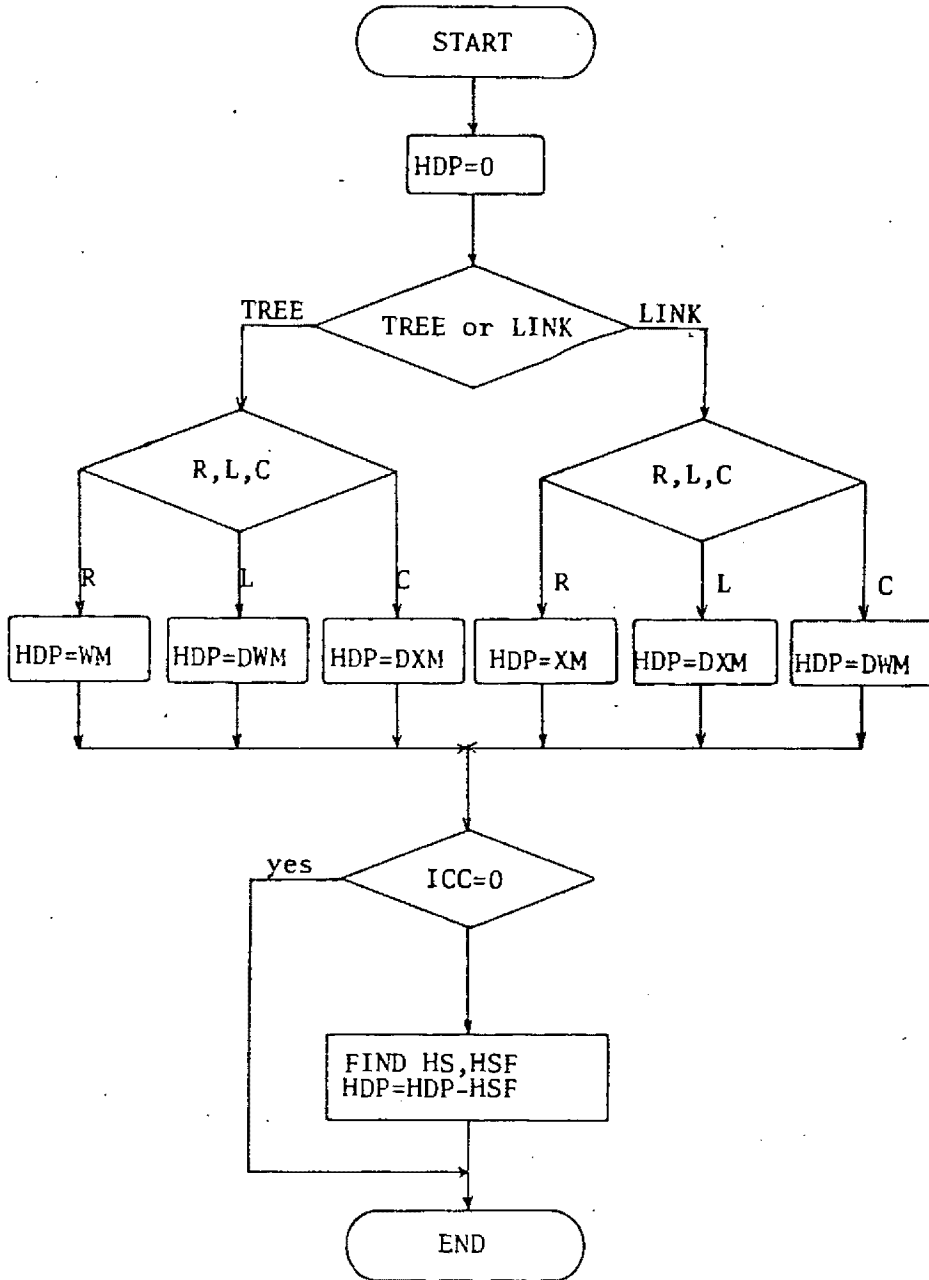


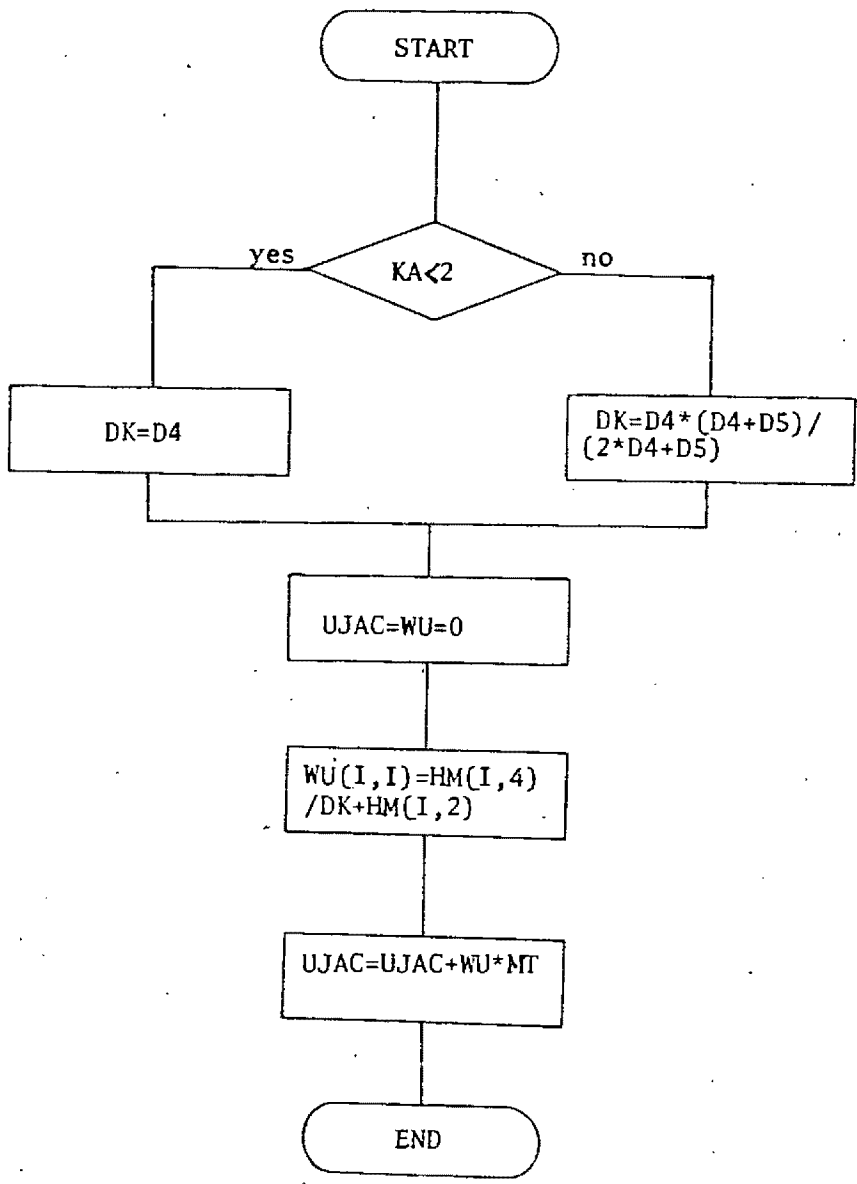


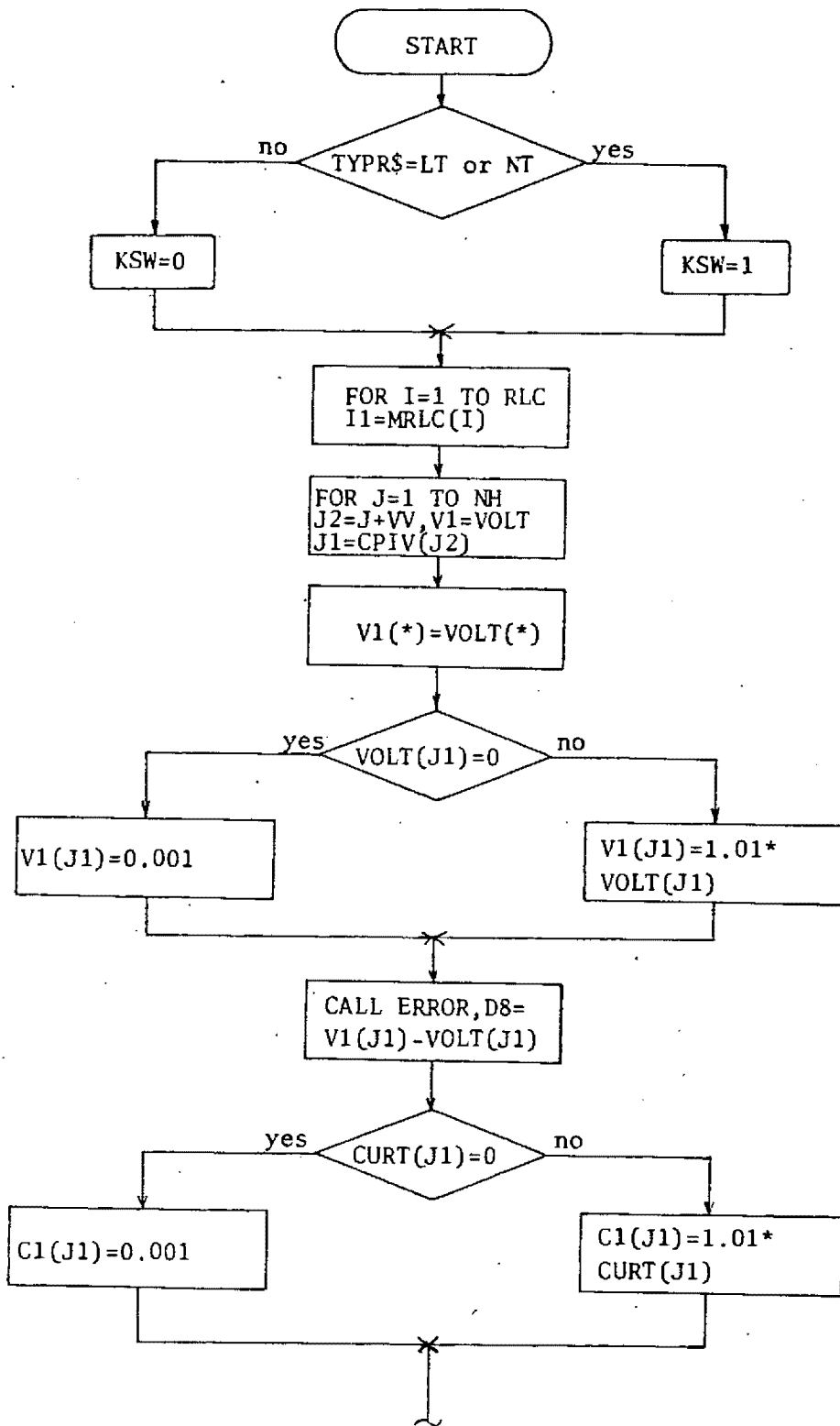


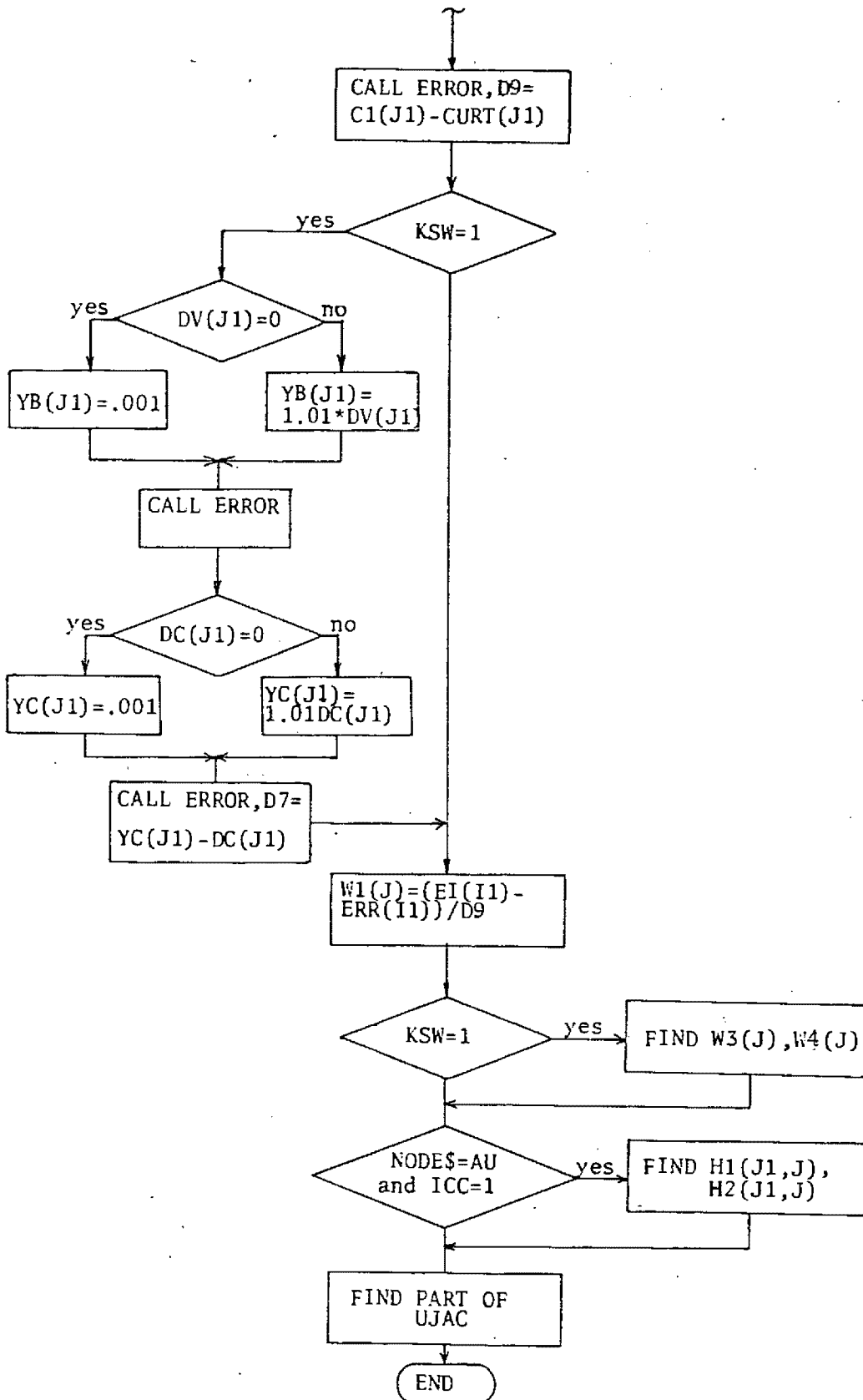


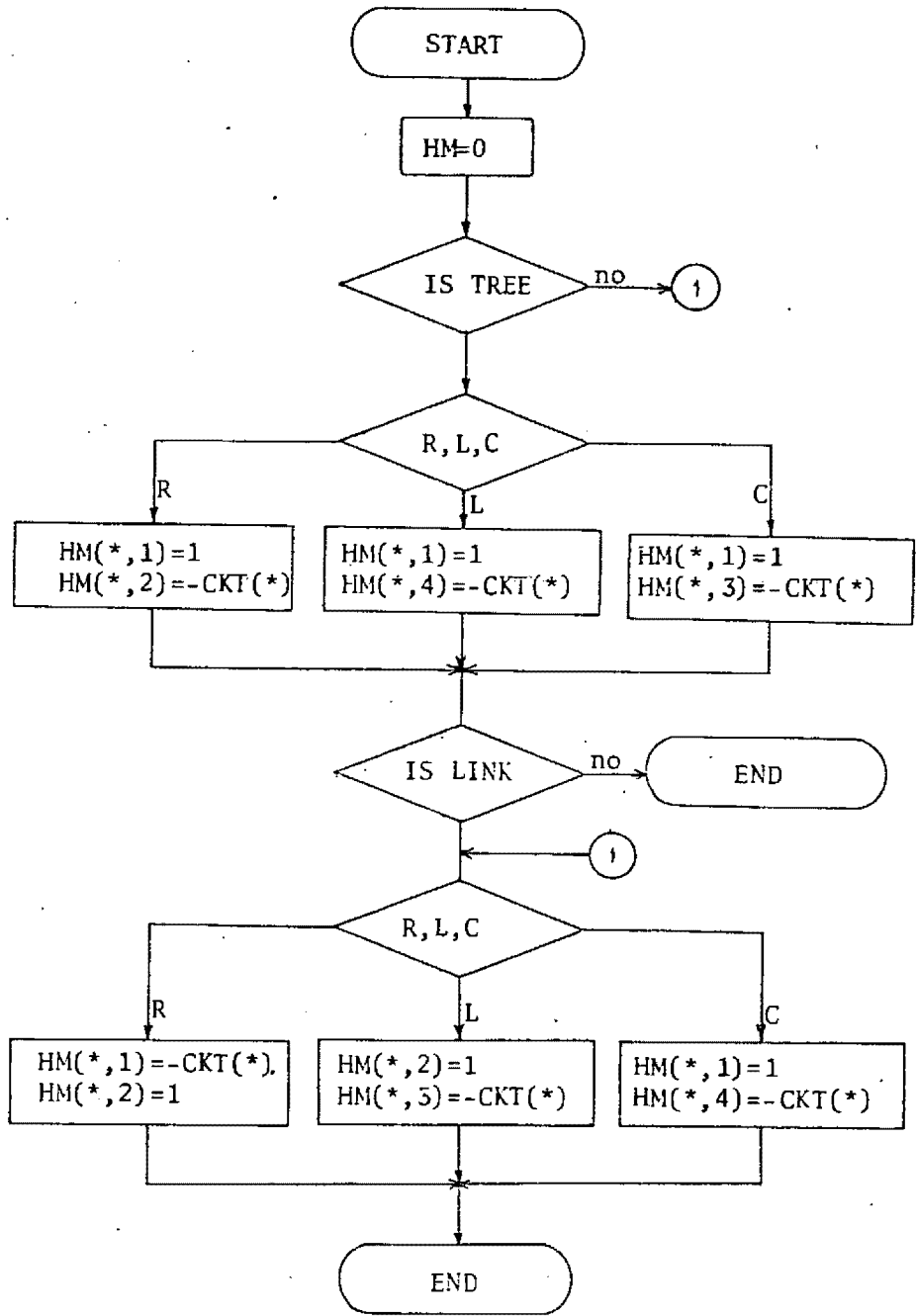


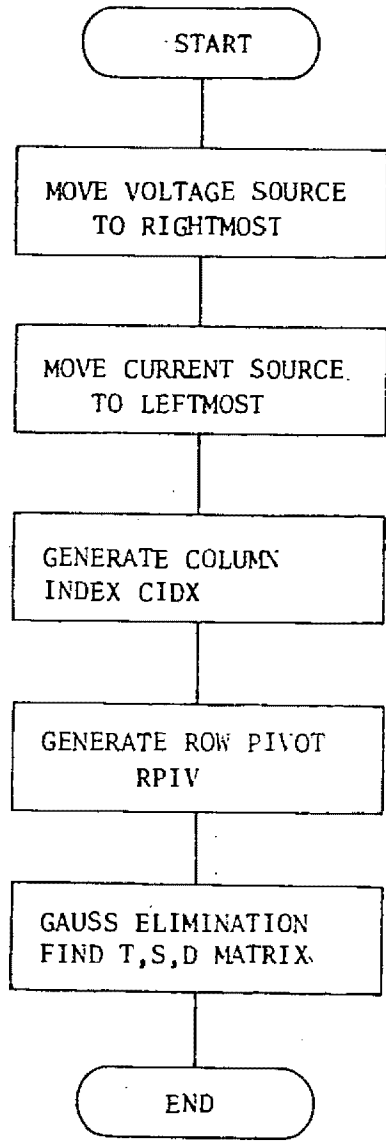


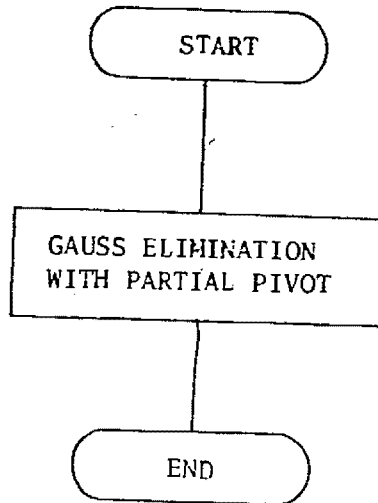




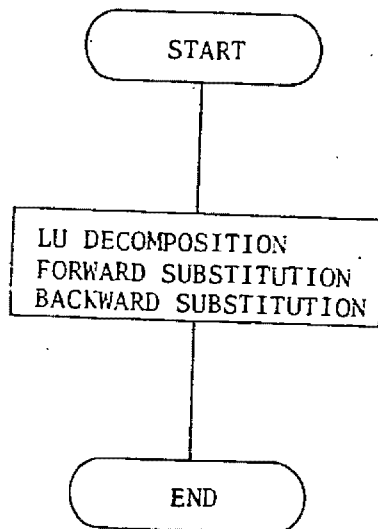


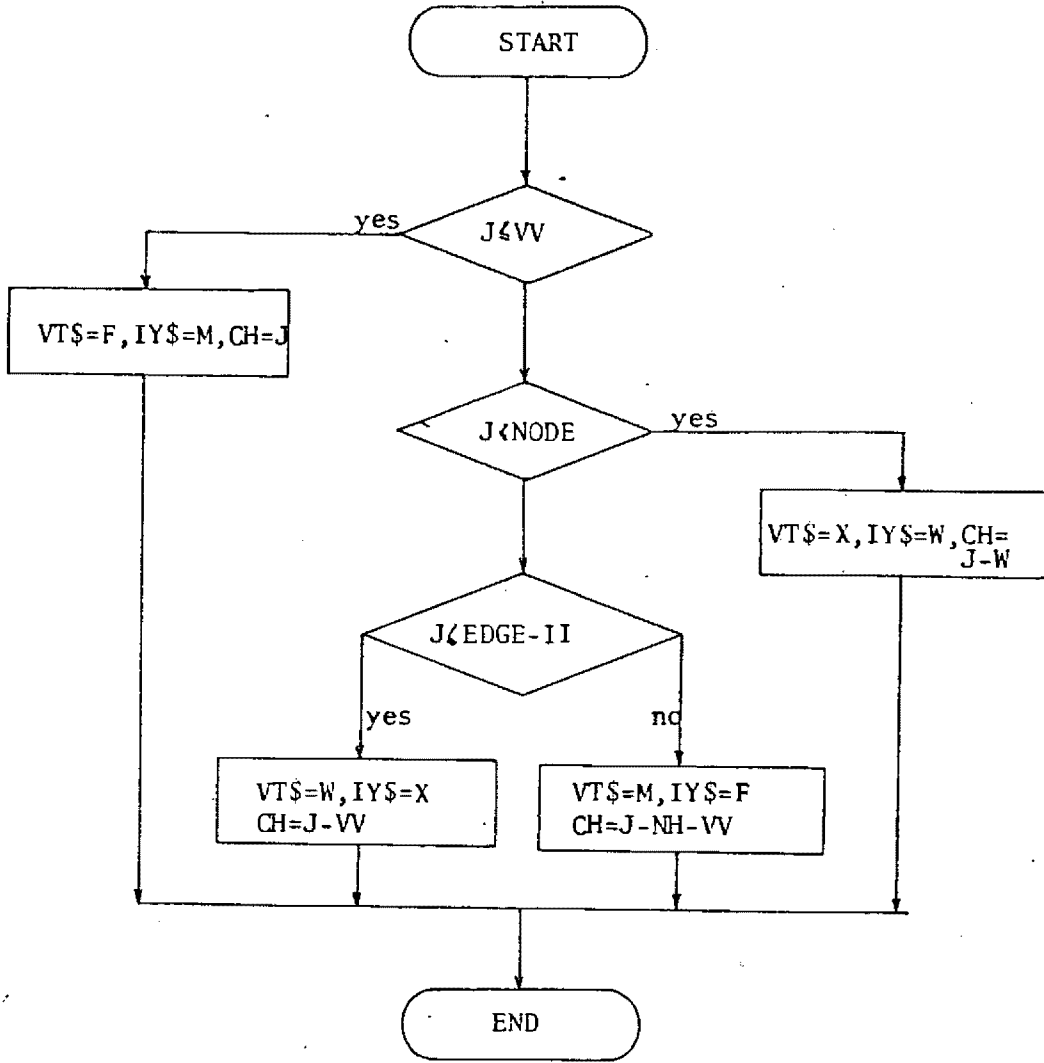


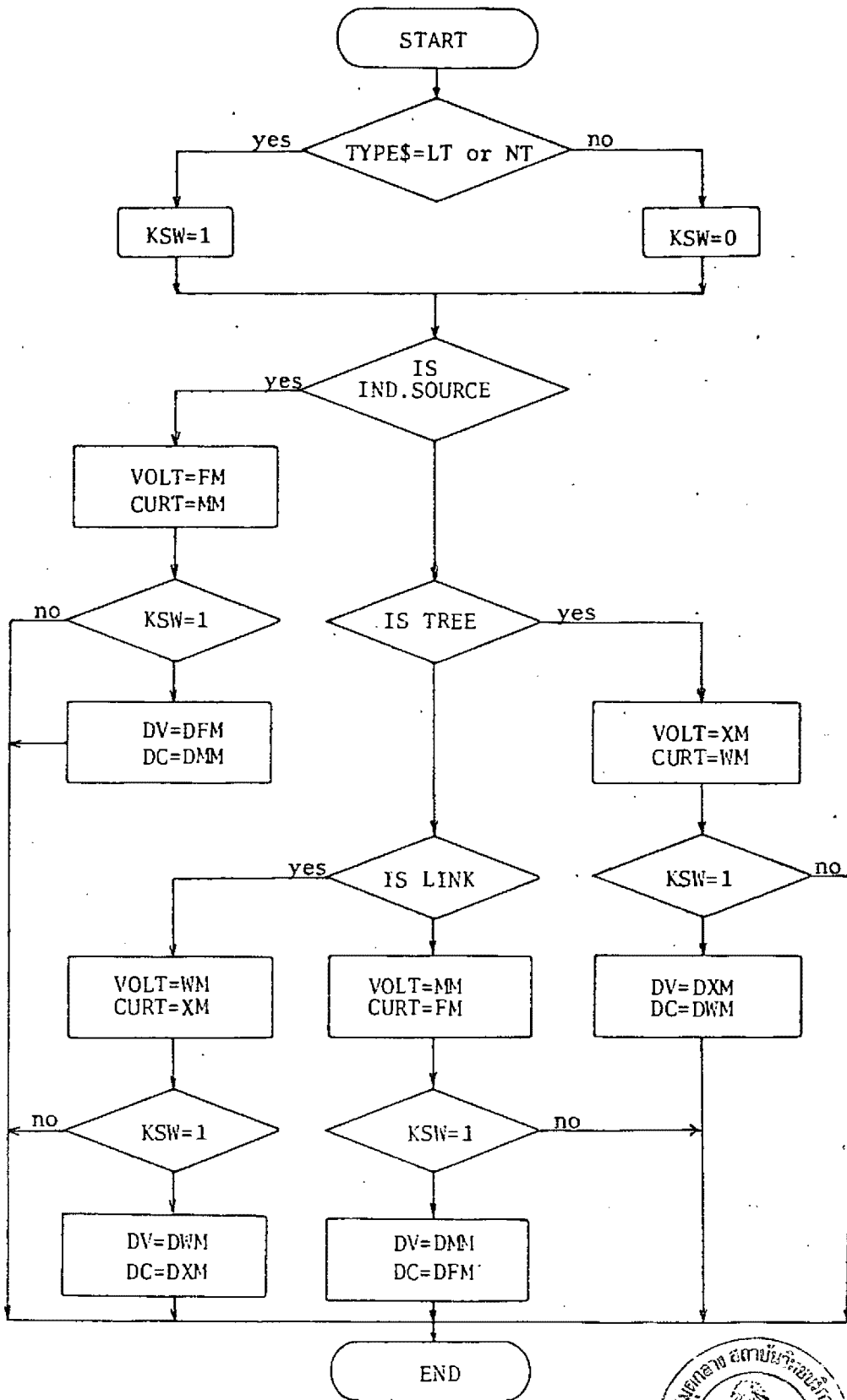




ສົມຮູບ FBSUB







ภาคผนวก จ.

โปรแกรมการวิเคราะห์วงจรไฟฟ้าและการออกแบบวงจรไฟฟ้าโดยอัตโนมัติ

Tshunsw AUTODC

```

50 REM AUTODC
51 REM ! INTEGER EDGE, NODE, VV, II, NS, NH, RLC, NP, NVD, NID, IM, IL
, ITER, ICC, MT, MS, MD, CP1V, MVD, MID, MCKT, CH, MPAR, MRLC, SS
52 REM ! DEFCOMMON EDGE, NODE, VV, II, NS, NH, RLC, NP, NVD, NID, IM, I
L, ITER, ICC, TYPE$, MODE$
53 REM ! DEFCOMMON XM(8), WM(8), MM(2), FM(2), HM(8, 4), UJAC(8, 8
), CKT(9), VOLT(9), CURT(9), H1(8, 2), FAUJ(8, 8), ELXM(8), ERR(9)

54 REM ! DEFCOMMON MVD(2), MID(2), CH(9), MPAR(3), SS(2), MT(8, 8
), MS(8, 2), MD(2, 2), MCKT(9), CP1V(9), MRLC(4), IP(8), FONT(9), K
EEP(8), C(9), WU(8, 8)
55 REM ! DEFCOMMON VT$(9), IY$(9), DFM(2), DXM(8), DWM(8), DMM(2
), WCKT(9), W1(8), W2(8), W3(8), W4(8), V1(9), C1(9), EV(9), EI(9
), D1(8)
56 REM ! DEFCOMMON VD(9), ID(9), DGD(3), H2(8, 2), FDP(2, 3), DDP
(2, 3), FBND(2, 3), W(8, 8), B(8), X(8), A(8, 8), KV(9), KI(9), FJ(8,
8), HDF(8, 3), ETSI(8), SI(3)
57 REM ! DEFCOMMON ESD(2), UT(8, 8), YY(8), TP$(9), YA(10), HF(8,
3), HS(8, 2), YZV(8), EM(8), ZP(8)
58 REM ! DEFCOMMON FU, FV, FW, FX
59 HOME
60 INPUT "CREATE NEW FILE (Y/N) "; Z$
62 IF Z$ = "N" THEN 190
65 PRINT : PRINT "EDGE NODE VV II ICC"
66 INPUT EDGE: VTAB 4: HTAB 6: INPUT NODE: VTAB 4: HTAB 11: INPUT
VV: VTAB 4: HTAB 14: INPUT II: VTAB 4: HTAB 17: INPUT ICC

70 MODE$ = "AU": NS = VV + II: NH = EDGE - NS
90 DIM XM(8), WM(8), MM(2), FM(2), HM(8, 4), UJAC(8, 8), CKT(9), VOLT
(9), CURT(9), H1(8, 2), FAUJ(8, 8), ELXM(8), ERR(9)
100 DIM MVD(2), MID(2), CH(9), MPAR(3), SS(2)
110 DIM MT(8, 8), MS(8, 2), MD(2, 2), MCKT(9), CP1V(9), MRLC(4), IP(8
), FONT(9), KEEP(8), C(9), WU(8, 8)
120 DIM VT$(9), IY$(9), DFM(2), DXM(8), DWM(8), DMM(2), WCKT(9)
130 DIM W1(8), W2(8), W3(8), W4(8), V1(9), C1(9), EV(9), EI(9), D1(8
)
140 DIM VD(9), ID(9), DGD(3), H2(8, 2), FDP(2, 3), DDP(2, 3), FBND(2
, 3)
150 DIM W(8, 8), B(8), X(8), A(8, 8), KV(9), KI(9), FJ(8, 8), HDF(8, 3)
, ETSI(8), SI(3)
160 DIM ESD(2), UT(8, 8), YY(8), TP$(9), YA(10), HF(8, 3), HS(8, 2), Y
ZV(8)
170 DIM MA(5, 9), RONE(5), RPIV(5), CIDX(9), WORK(8), EM(8)
180 DIM ZP(8), Z1$(9)
185 GOSUB 2500: REM INITIAL
190 GOSUB 2960: REM RETRIEVE FILE
192 IF FX = 1 THEN PR$ 1
196 GOSUB 1000: REM TDMAT
270 GOSUB 2000: REM MAP
280 PRINT : PRINT " DESIRED RESPONSE WEIGHTING FACTOR"
290 IF NVD = 0 THEN 345
340 FOR I = 1 TO NVD: J = MVD(I): PRINT : PRINT "V("; J; ") = "
: VD(J); "VOLT KV("; J; ") = "; KV(J): NEXT I
345 IF NID = 0 THEN 420
390 FOR I = 1 TO NID: J = MID(I): PRINT : PRINT " I("; J; ") =

```

```

";ID(J);" AMP          KI(";J;") = ";KI(J): NEXT I
420 PRINT : PRINT "    INITIAL GUESS CKT = ": PRINT
430 FOR I = 1 TO NF: PRINT "  CKT(";MPAR(I);") = ";CKT(MPAR(
I));: NEXT I
435 PRINT
440 REM ! CLEAR CHAIN
450 PRINT CHR$(4)"BRUN AUTOGRADG.OBJ"
910 END
1000 REM TSDMAT
1020 FOR I = 1 TO NODE:ZZ = 0: FOR J = 1 TO EDGE:ZZ = ZZ + ABS
(MA(I,J)): NEXT J:RONE(I) = ZZ: NEXT I
1030 FOR J = 1 TO EDGE
1040 FOR I = 1 TO NODE
1050 IF MA(I,J) = 0 THEN 1070
1060 CPIV(J) = CPIV(J) + RONE(I)
1070 NEXT I
1080 NEXT J
1090 K = 1
1100 FOR I = 1 TO EDGE
1110 FOR J = 1 TO NS
1120 IF I = SS(J) THEN 1150
1130 NEXT J
1140 WORK(K) = I:K = K + 1
1150 NEXT I
1160 IF VV = 0 THEN 1180
1170 FOR I = 1 TO VV:CIDX(I) = SS(I): NEXT I
1180 IF II = 0 THEN 1200
1190 FOR I = 1 TO II:CIDX(EDGE - II + I) = SS(VV + I): NEXT
I
1200 L = NH
1210 FOR J = (VV + 1) TO (EDGE - II - 1):K = 1:TEST = CPIV(W
ORK(1))
1220 FOR I = 2 TO L
1230 IF TEST > = CPIV(WORK(I)) THEN 1250
1240 TEST = CPIV(WORK(I)):K = I
1250 NEXT I
1260 CIDX(J) = WORK(K): IF K = L THEN 1280
1270 FOR I = K TO (L - 1):WORK(I) = WORK(I + 1): NEXT I
1280 L = L - 1
1290 NEXT J
1300 CIDX(J) = WORK(1)
1310 FOR I = 1 TO NODE:RPIV(I) = I: NEXT I
1320 N1 = 0:N2 = NODE:J = 0
1330 J = J + 1:J1 = CIDX(J):MN = 0:TEST = EDGE + 1
1340 FOR I1 = 1 TO NODE
1350 IF MA(I1,J1) = 0 THEN 1430
1360 IF J = 1 THEN 1400
1370 K1 = 1: IF N1 = 0 THEN 1400
1380 IF I1 = RPIV(K1) THEN 1430
1390 K1 = K1 + 1: IF K1 < = N1 THEN 1380
1400 NN = 0: FOR Z = 1 TO EDGE:NN = NN + ABS (MA(I1,Z)): NEXT
Z
1410 IF TEST < = NN THEN 1430
1420 TEST = NN:MN = I1
1430 NEXT I1
1440 IF MN = 0 THEN 1580

```

```

1450 N1 = N1 + 1:CPIV(N1) = J1
1460 I = N1
1470 IF RPIV(I) = MN THEN 1490
1480 I = I + 1: IF I < = NODE THEN 1470
1490 RPIV(I) = RPIV(N1):RPIV(N1) = MN:I3 = RPIV(N1):J3 = CPIX
(N1):ZZ = MA(I3,J3)
1500 FOR Z = 1 TO EDGE:MA(I3,Z) = MA(I3,Z) / ZZ: NEXT Z
1510 FOR K2 = 1 TO NODE
1520 IF RPIV(K2) = MN THEN 1560
1530 K3 = RPIV(K2): IF MA(K3,J1) = 0 THEN 1560
1540 ZZ = MA(K3,J3)
1550 FOR Z = 1 TO EDGE:MA(K3,Z) = MA(K3,Z) - MA(I3,Z) * ZZ: NEXT
Z
1560 NEXT K2
1570 GOTO 1590
1580 CPIX(N2) = J1:N2 = N2 + 1
1590 IF N1 < (NODE - 1) THEN 1330
1600 FOR I = (J + 1) TO EDGE:CPIX(I) = CPIX(I): NEXT I
1610 PRINT : PRINT " MCPIV =": FOR I = 1 TO EDGE: PRINT
CPIX(I);" ";: NEXT I: PRINT
1620 PRINT : PRINT " MA ="
1630 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(RPIV(I),
CPIX(J));" ";: NEXT J: PRINT : NEXT I
1640 FOR I = 1 TO NH: FOR J = 1 TO NH:MT(I,J) = 0: NEXT J: NEXT
I
1650 FOR I = 1 TO NH: FOR J = 1 TO NS:MS(I,J) = 0: NEXT J: NEXT
I
1660 FOR I = 1 TO NS: FOR J = 1 TO NS:MD(I,J) = 0: NEXT J: NEXT
I
1670 FOR I = (VV + 1) TO (NODE - 1):I1 = RPIV(I):I2 = I - VV
1680 FOR J = NODE TO (EDGE - I1):J1 = CPIX(J):J2 = J - VV:MT
(I2,J2) = - MA(I1,J1):MT(J2,I2) = MA(I1,J1): NEXT J
1690 IF I1 = 0 THEN 1710
1700 FOR J = (EDGE - I1 + 1) TO EDGE:J1 = CPIX(J):J2 = J - N
H:MS(I2,J2) = - MA(I1,J1): NEXT J
1710 NEXT I
1720 IF VV = 0 THEN 1760
1730 FOR I = 1 TO VV:I1 = RPIV(I): FOR J = NODE TO (EDGE - I
I):J1 = CPIX(J):I2 = J - VV:MS(I2,I) = MA(I1,J1): NEXT J:
NEXT I
1740 IF I1 = 0 THEN 1760
1750 FOR I = 1 TO VV:I1 = RPIV(I):I2 = I: FOR J = (EDGE - I1
+ 1) TO EDGE:J1 = CPIX(J):J2 = J - NH:MD(I2,J2) = - MA(
I1,J1):MD(J2,I2) = MA(I1,J1): NEXT J: NEXT I
1760 PRINT : PRINT : PRINT " MATRIX T ="
1770 FOR I = 1 TO NH: FOR J = 1 TO NH: PRINT MT(I,J);" ";: NEXT
J: PRINT : NEXT I
1780 PRINT : PRINT : PRINT " MATRIX S ="
1790 FOR I = 1 TO NH: FOR J = 1 TO NS: PRINT MS(I,J);" ";: NEXT
J: PRINT : NEXT I
1800 PRINT : PRINT : PRINT " MATRIX D ="
1810 FOR I = 1 TO NS: FOR J = 1 TO NS: PRINT MD(I,J);" ";: NEXT
J: PRINT : NEXT I
1820 RETURN
2000 REM MAP

```

```

2010 FOR I = 1 TO EDGE
2020 FOR J = 1 TO EDGE
2030 IF I < > CPIV(J) THEN 2110
2040 IF J < = VV THEN 2080
2050 IF J < NODE THEN 2090
2060 IF J < = (EDGE - I) THEN 2100
2070 VT$(I) = "M": IY$(I) = "F": CH(I) = J - (NH + VV): GOTO 21
20      20
2080 VT$(I) = "F": IY$(I) = "M": CH(I) = J: GOTO 2120
2090 VT$(I) = "X": IY$(I) = "W": CH(I) = J - VV: GOTO 2120
2100 VT$(I) = "W": IY$(I) = "X": CH(I) = J - VV: GOTO 2120
2110 NEXT J
2120 NEXT I
2130 RETURN
2500 REM INITIAL
2505 HOME: PRINT "E N1 N2 TYPE K.OHM VOLT M.AMP"
2510 VTAB 24: GET Z$: IF Z$ = CHR$(13) THEN 2530
2515 I = VAL (Z$): J = I * 2: IF I > ED THEN 2510
2520 VTAB J: PRINT I: VTAB J: HTAB 3: INPUT KV(I): VTAB J: HTAB
6: INPUT KI(I): VTAB J: HTAB 10: GET Z1$(I): VTAB J: HTAB
10: PRINT Z1$(I)
2525 VTAB J: HTAB 15: INPUT CKT(I): VTAB J: HTAB 23: INPUT V
OLT(I): VTAB J: HTAB 31: INPUT CURT(I): GOTO 2510
2530 RLC = 0: TYPE$ = "LD"
2535 FOR I = 1 TO EDGE: MCKT(I) = 0
2540 IF Z1$(I) = CHR$(82) THEN MCKT(I) = - 1
2545 IF Z1$(I) = CHR$(76) THEN MCKT(I) = 0
2550 IF Z1$(I) = CHR$(67) THEN MCKT(I) = 1
2555 IF Z1$(I) = CHR$(78) THEN 2560
2557 GOTO 2565
2560 RLC = RLC + 1: MRLC(RLC) = I: TYPE$ = "ND": MCKT(I) = 2
2565 NEXT I
2570 HOME
2580 FOR I = 1 TO NS: INPUT "INDEPENDENT SOURCE VECTOR": SS(I
): NEXT I
2585 FOR I = 1 TO NS: INPUT "INDEPENDENT SOURCE VALUE ": FM(I
): NEXT I
2590 HOME
2600 INPUT "NUMBER OF PARAMETER": NP: PRINT
2610 FOR I = 1 TO NP: VTAB 4: INPUT "ELEMENT ": MPAR(I): VTAB
4: HTAB 15: INPUT "MAX ": FBND(1, I): VTAB 4: HTAB 22: INPUT
"MIN ": FBND(2, I): NEXT I
2614 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: MA(I, J) = 0: NEXT
J: NEXT I
2616 FOR I = 1 TO EDGE: MA(KV(I), I) = 1: MA(KI(I), I) = - 1: NEXT
I
2620 HOME
2630 INPUT "NUMBER OF DESIRED O/P VOLTAGE": NVD
2640 IF NVD < = 0 THEN 2660
2645 FOR I = 1 TO EDGE: VD(I) = 0: KV(I) = 0: NEXT I
2650 FOR I = 1 TO NVD: VTAB 4: INPUT "ELEMENT": MVD(I): VTAB
4: HTAB 11: INPUT "DESIRED O/P VOLT": VD(MVD(I)): KV(MVD(I)
) = 1: NEXT I
2660 HOME
2670 INPUT "NUMBER OF DESIRED O/P CURRENT": NID
2680 IF NID < = 0 THEN 2710

```

```

2690 FOR I = 1 TO EDGE:ID(I) = 0:KI(I) = 0: NEXT I
2700 FOR I = 1 TO NID: VTAB 4: INPUT "ELEMENT";MID(I): VTAB
    4: HTAB 11: INPUT "DESIRED O/P CURRENT";ID(MID(I)):KI(MID
    (I)) = 1: NEXT I
2710 HOME
2740 INPUT "ALPHA F ";FV: INPUT " ALPHA R ";FW: INPUT " VT "
    ;FU
2750 INPUT "YOU WANT HARD COPY(1/0) ";FX
2760 GOSUB 2850: REM CREATE FILE
2770 RETURN
2850 REM CREATE FILE
2860 HOME : INPUT "FILE NAME ";Z#
2865 PRINT CHR# (4);"OPEN ";Z#
2870 PRINT CHR# (4);"DELETE ";Z#
2875 PRINT CHR# (4);"OPEN ";Z#
2880 PRINT CHR# (4);"WRITE ";Z#
2885 PRINT EDGE: PRINT NODE: PRINT VV: PRINT II: PRINT NVD: PRINT
    NID: PRINT NF: PRINT RLC: PRINT ICC: PRINT TYPE#: PRINT M
    ODE#: PRINT NS: PRINT NH: PRINT FU: PRINT FV: PRINT FW: PRINT
    FX
2890 FOR I = 1 TO NS: PRINT SS(I): NEXT I
2895 FOR I = 1 TO NF: PRINT MPAR(I): NEXT I
2900 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(I,J): NEXT
    J: NEXT I
2905 FOR I = 1 TO EDGE: PRINT MCKT(I): PRINT CKT(I): PRINT V
    OLT(I): PRINT CURT(I): NEXT I
2910 FOR I = 1 TO 2: FOR J = 1 TO NF: PRINT PBND(I,J): NEXT
    J: NEXT I
2915 FOR I = 1 TO EDGE: PRINT VD(I): PRINT KV(I): NEXT I
2916 IF NVD < = 0 THEN 2925
2920 FOR I = 1 TO NVD: PRINT MVD(I): NEXT I
2925 FOR I = 1 TO EDGE: PRINT ID(I): PRINT KI(I): NEXT I
2927 IF NID < = 0 THEN 2935
2930 FOR I = 1 TO NID: PRINT MID(I): NEXT I
2935 IF RLC < = 0 THEN 2945
2940 FOR I = 1 TO RLC: PRINT MRLC(I): NEXT I
2945 FOR I = 1 TO NS: PRINT FM(I): NEXT I
2948 PRINT CHR# (4);"CLOSE ";Z#
2950 RETURN
2960 REM RETRIEVE FILE
2961 HOME : INPUT "FILE NAME ";Z#
2962 PRINT CHR# (4);"OPEN ";Z#
2963 PRINT CHR# (4);"READ ";Z#
2964 INPUT EDGE: INPUT NODE: INPUT VV: INPUT II: INPUT NVD: INPUT
    NID: INPUT NF: INPUT RLC: INPUT ICC: INPUT TYPE#: INPUT M
    ODE#: INPUT NS: INPUT NH: INPUT FU: INPUT FV: INPUT FW: INPUT
    FX
2965 FOR I = 1 TO NS: INPUT SS(I): NEXT I
2966 FOR I = 1 TO NF: INPUT MPAR(I): NEXT I
2967 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: INPUT MA(I,J): NEXT
    J: NEXT I
2968 FOR I = 1 TO EDGE: INPUT MCKT(I): INPUT CKT(I): INPUT V
    OLT(I): INPUT CURT(I): NEXT I
2970 FOR I = 1 TO 2: FOR J = 1 TO NF: INPUT PBND(I,J): NEXT
    J: NEXT I
2971 FOR I = 1 TO EDGE: INPUT VD(I): INPUT KV(I): NEXT I

```

```
2972 IF NVD < = 0 THEN 2974
2973 FOR I = 1 TO NVD: INPUT MVD(I): NEXT I
2974 FOR I = 1 TO EDGE: INPUT ID(I): INPUT KI(I): NEXT I
2976 IF NID < = 0 THEN 2979
2977 FOR I = 1 TO NID: INPUT MID(I): NEXT I
2979 IF RLC < = 0 THEN 2982
2980 FOR I = 1 TO RLC: INPUT MRLC(I): NEXT I
2982 FOR I = 1 TO NS: INPUT FM(I): NEXT I
2983 PRINT CHR$(4);"CLOSE ";Z$
2984 RETURN
```

โปรแกรมคอมพิวเตอร์ AUTOGRADG

```
300 REM AUTOGRADG
320 REM ! INTEGER EDGE,NODE,VV,II, NS,NH,RLC,NF,NVD,NID,IM,I
L,ITER,ICC,MT,MS,MD,CPIV,MVD,MID,MCKT,CH,MPAR,MRLC,SS
330 REM ! USECOMMON EDGE,NODE,VV,II,NS,NH,RLC,NF,NVD,NID,IM,
IL,ITER,ICC,TYPE$,MODE$
340 REM ! USECOMMON XM(8),WM(8),MM(2),FM(2),HM(8,4),UJAC(8,
8),CKT(9),VOLT(9),CURT(9),H1(8,2),FAUJ(8,2),ELXM(8),ERR(9
)
350 REM ! USECOMMON MVD(2),MID(2),CH(9),MPAR(3),SS(2),MT(8,
8),MS(8,2),MD(2,2),MCKT(9),CPIV(9),MRLC(4),IP(8),FONT(9),
KEEP(8),C(9),WU(8,8)
360 REM ! USECOMMON VT$(9),IY$(9),DFM(2),DXM(8),DWM(8),DMM(
2),WCKT(9),W1(8),W2(8),W3(8),W4(8),V1(9),C1(9),EV(9),E1(9
),D1(8)
370 REM ! USECOMMON VD(9),ID(9),DGDP(3),H2(8,2),FDP(2,3),DD
P(2,3),PBND(2,3),W(8,8),B(8),X(8),A(8,8),KV(9),KI(9),FJ(8
,8),HDP(8,3),ETS1(8),SI(3)
380 REM ! USECOMMON ESD(2),UT(8,2),YY(8),TF$(9),YA(10),HF(8
,3),HS(8,2),YZV(8),EM(8),ZF(8)
390 REM ! USECOMMON FU,FV,FW,FX
400 IF PX = 1 THEN PR# 1
440 IV = 0: IW = 0
450 FOR I = 1 TO EDGE: WCKT(I) = CKT(I): NEXT I
460 G = 0: G1 = 0: G2 = 0: IL = 0: ISW = 0: IM = 0
490 GOSUB 2100: REM DCAY
510 GOSUB 6200: REM FUNC6
520 IF G < = 1.0E - 6 THEN 990
560 GOSUB 6500: REM GRADG
570 GOSUB 8000: REM NORMAL
620 IL = IL + 1
630 FOR I = 1 TO EDGE: WCKT(I) = CKT(I): NEXT I
640 FOR I = 1 TO NF
650 J = MPAR(I): WCKT(J) = WCKT(J) - DGDP(I)
660 IF WCKT(J) < PBND(2,I) THEN WCKT(J) = PBND(2,I)
670 IF WCKT(J) > PBND(1,I) THEN WCKT(J) = PBND(1,I)
680 NEXT I
710 GOTO 490
980 PRINT : PRINT " RANGEN:STOP": END
990 IV = IV + IM + 1: IW = IW + IL + 1
1000 PRINT : PRINT " THE TOTAL NUMBER OF MAJOR ITERATIONS
";IV
1010 PRINT : PRINT " TOTAL NUMBER OF ITERATIONS FOR SEARCH
ING LAMD ";IW
1020 PRINT : PRINT " THE SQUARE ERROR ";G
1030 PRINT : PRINT " THE FINAL CIRCUIT CKT ": PRINT
1040 FOR I = 1 TO NF: PRINT "CKT(";MPAR(I);") = ";WCKT(MPAR(
I)): PRINT : NEXT I
1050 PRINT : PRINT " THE FINAL-RESPONSES "
1060 FOR I = 1 TO EDGE: PRINT "V(";I;") = ";VOLT(I);" VOLT
I(";I;") = ";CURT(I);"MILLIAMP": PRINT : NEXT I
1070 END
2100 REM DCAY
2102 FOR I = 1 TO EDGE: CKT(I) = WCKT(I): NEXT I
2104 AID = 0: IF TYPE$ = "ND" THEN AID = 1
2108 IF RLC = 0 THEN 2130
```

```

2110 FOR I = 1 TO EDGE:ERR(I) = 0:FONT(I) = 0: NEXT I
2120 FOR I = 1 TO NH:K = CPIV(VV + I):FONT(K) = I: NEXT I
2130 IF IM > 0 OR IL > 0 THEN 2170
2140 FOR I = 1 TO NH:K = I + VV:J = CPIV(K)
2150 XM(I) = CURT(J): IF K < NODE THEN XM(I) = VOLT(J)
2160 NEXT I
2170 IF AID < > 1 THEN 2190
2180 ENRM = 0:E1 = 0:E2 = 0:SECH = 0:NLSW = 0: FOR I = 1 TO N
H:KEEP(I) = XM(I): NEXT I
2190 DONE = 0:ITER = 0:RTEP = 1: FOR I = 1 TO NH: FOR J = 1 TO
NS:H1(I,J) = 0: NEXT J: NEXT I
2200 GOSUB 3000: REM HMAT
2210 FOR I = 1 TO NH
2220 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 + MT(I,J) * XM(J): NEXT
J
2230 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MS(I,J) * FM(J): NEXT
J
2240 WM(I) = Z1 + Z2
2250 NEXT I
2260 FOR I = 1 TO NS
2270 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 - MS(J,I) * XM(J): NEXT
J
2280 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MD(I,J) * FM(J): NEXT
J
2290 MM(I) = Z1 + Z2
2300 NEXT I
2310 FOR I = 1 TO NH:EM(I) = HM(I,1) * XM(I) + HM(I,2) * WM(
I): NEXT I
2320 IF RLC = 0 THEN 2360
2330 GOSUB 3500: REM CONVERT
2340 GOSUB 4000: REM ERROR
2350 FOR I = 1 TO RLC:J = MRLC(I):K = PONT(J):EM(K) = ERR(J)
: NEXT I
2360 Z1 = 0:Z2 = 0:Z3 = 0: FOR I = 1 TO NH:Z1 = Z1 + EM(I) *
EM(I):Z2 = Z2 + XM(I) * XM(I):Z3 = Z3 + WM(I) * WM(I): NEXT
I:ENRM = SQR(Z1):XNRM = SQR(Z2):WNRM = SQR(Z3)
2370 IF AID < > 1 THEN 2400
2380 E2 = E1:E1 = ENRM:ENRM = 0:XNRM = 0:WNRM = 0
2390 FOR I = 1 TO NH:ENRM = ENRM + ABS(EM(I)):XNRM = XNRM +
ABS(XM(I)):WNRM = WNRM + ABS(WM(I)): NEXT I
2400 IF ENRM > (XNRM + WNRM) * 1.0E - 6 THEN 2430
2410 IF MODE$ < > "AU" OR AID = 0 THEN 2860
2420 DONE = 1: GOTO 2570
2430 IF TYPE$ = "LD" AND ITER > 0 THEN 2600
2440 IF AID < > 1 THEN 2550
2450 IF SECH = 0 THEN EQ = ENRM
2460 ON NLSW GOTO 2720,2740,2800
2470 IF SECH < > 0 THEN 2500
2480 IF ITER = 0 THEN 2550
2490 IF ITER < > 0 THEN 2570
2500 IF EQ < ENRM THEN 2530
2510 NLSW = 2
2520 RTEP = RTEP * 2: GOTO 2620
2530 NLSW = 1
2540 RTEP = RTEP / 2: GOTO 2620
2550 DUMY = 1: GOSUB 4500: REM UJADN

```



```

2560 IF RLC = 0 THEN 2590
2570 GOSUB 5000: REM NUJADM
2580 IF DONE = 1 THEN 2860
2590 GOSUB 5500: REM FACTOR
2600 GOSUB 6000: REM FBSUB
2610 IF AID < > 1 THEN 2650
2620 SECH = SECH + 1: IF SECH < = 100 THEN 2640
2630 PRINT : PRINT " SECH>100:STOP": END
2640 FOR I = 1 TO NH:XM(I) = KEEP(I): NEXT I
2650 FOR I = 1 TO NH:XM(I) = XM(I) - RTEP * ELXM(I): NEXT I
2660 IF AID < > 1 THEN 2690
2670 IF NLSW < > 4 THEN 2210
2680 NLSW = 0:SECH = 0:ENRM = 0:E1 = 0:E2 = 0:RTEP = 1: FOR I
= 1 TO NH:KEEP(I) = XM(I): NEXT I
2690 ITER = ITER + 1
2700 IF ITER < = 100 THEN 2210
2710 PRINT : PRINT : PRINT " ITER>100:STOP": END
2720 IF EO < = ENRM THEN 2540
2730 RANL = 0:RBNL = RTEP:RCNL = RTEP * 2:EA = EO:EB = ENRM:EC
= E1: GOTO 2770
2740 IF ENRM < E1 THEN 2520
2750 RCNL = RTEP:RBNL = RTEP / 2:EA = E2:EB = E1:EC = ENRM
2760 RANL = RBNL / 2: IF SECH = 2 THEN RANL = 0
2770 EA = EA * (RCNL - RBNL):ENRM = EB * (RANL - RCNL):EC = E
C * (RBNL - RANL)
2780 RTEP = (EA * (RCNL + RBNL) + ENRM * (RANL + RCNL) + EC *
(RBNL + RANL)) / (2 * (EA + ENRM + EC))
2790 NLSW = 3: GOTO 2640
2800 IF ENRM < = EB THEN 2820
2810 RTEP = RBNL:NLSW = 4: GOTO 2640
2820 ITER = ITER + 1: IF ITER < = 100 THEN 2840
2830 PRINT : PRINT " ITER > 100 :STOP": END
2840 NLSW = 0:SECH = 0:EO = ENRM:E1 = 0:E2 = 0:RTEP = 1: FOR
I = 1 TO NH:KEEP(I) = XM(I): NEXT I
2850 GOTO 2570
2860 IF RLC < > 0 THEN 2880
2870 GOSUB 3500: REM CONVERT
2880 PRINT : PRINT : PRINT " NO.OF ITERATIONS = ";ITER
2890 FOR I = 1 TO EDGE: PRINT "VOLT(";I;") = ";VOLT(I);" VOL
T
CURT(";I;") = ";CURT(I);" MILLIAMPS": PRINT : NEXT I
2895 FOR I = 1 TO EDGE:WCKT(I) = CKT(I): NEXT I
2900 RETURN
3000 REM HMAT
3010 FOR I = 1 TO NH: FOR J = 1 TO 4:HM(I,J) = 0: NEXT J: NEXT
I
3020 FOR I = 1 TO (NODE - 1 - VV)
3030 K = I + VV:I1 = OP1V(K): IF MCKT(I1) > 1 THEN 3100
3040 IF MCKT(I1) = - 1 THEN 3070
3050 IF MCKT(I1) = 0 THEN 3080
3060 IF MCKT(I1) = 1 THEN 3090
3045 PRINT " CHECK CKT:STOP": END
3070 HM(I,1) = 1:HM(I,2) = - CKT(I1): GOTO 3100
3080 HM(I,1) = 1:HM(I,4) = - CKT(I1): GOTO 3100
3090 HM(I,2) = 1:HM(I,3) = - CKT(I1)
3100 NEXT I

```

```

3110 FOR I = (NODE - VV) TO NH
3120 K = I + VV: I1 = CPIV(K): IF MCKT(I1) > 1 THEN 3200
3130 IF MCKT(I1) = - 1 THEN 3170
3140 IF MCKT(I1) = 0 THEN 3180
3150 IF MCKT(I1) = 1 THEN 3190
3160 PRINT " CHECK CKT:STDF": END
3170 HM(I,1) = - CKT(I1):HM(I,2) = 1: GOTO 3200
3180 HM(I,2) = 1:HM(I,3) = - CKT(I1): GOTO 3200
3190 HM(I,1) = 1:HM(I,4) = - CKT(I1): GOTO 3200
3200 NEXT I
3210 RETURN
3500 REM CONVERT
3510 KSW = 0: IF TYPE$ = "LT" OR TYPE$ = "NT" THEN KSW = 1
3520 K = VV
3530 FOR I = 1 TO EDGE:J = CPIV(I)
3540 IF I < = VV THEN 3600
3550 IF I < NODE THEN 3630
3560 IF I < = (EDGE - I1) THEN 3660
3570 K = K + 1:VOLT(J) = MM(K):CURT(J) = FM(K)
3580 IF KSW < > 1 THEN 3700
3590 DV(J) = DMM(K):DC(J) = DFM(K): GOTO 3700
3600 VOLT(J) = FM(I):CURT(J) = MM(I)
3610 IF KSW < > 1 THEN 3700
3620 DV(J) = DFM(I):DC(J) = DMM(I): GOTO 3700
3630 L = I - VV:VOLT(J) = XM(L):CURT(J) = WM(L)
3640 IF KSW < > 1 THEN 3700
3650 DV(J) = DXM(L):DC(J) = DWM(L): GOTO 3700
3660 L = I - VV:VOLT(J) = WM(L):CURT(J) = XM(L)
3670 IF KSW < > 1 THEN 3700
3680 DV(J) = DWM(L):DC(J) = DXM(L): GOTO 3700
3700 NEXT I
3710 RETURN
4000 REM ERROR
4010 FOR I = 1 TO RLC:C(MRLC(I)) = 1.0E - 6 * ( EXP (VOLT(MR
LC(I)) * FU) - 1): NEXT I
4030 ERR(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(
2))
4040 ERR(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(
1))
4050 RETURN
4500 REM UJADM
4510 FOR I = 1 TO NH: FOR J = 1 TO NH:UJAC(I,J) = 0:WU(I,J) =
0: NEXT J: NEXT I
4520 FOR I = 1 TO NH:WU(I,I) = HM(I,4) + HM(I,2): NEXT I
4530 FOR I = 1 TO NH: FOR J = 1 TO NH:Z1 = 0: FOR K = 1 TO N
H:Z1 = Z1 + WU(I,K) * MT(K,J): NEXT K:UJAC(I,J) = Z1: NEXT
J: NEXT I
4540 FOR I = 1 TO NH:UJAC(I,I) = UJAC(I,I) + HM(I,1) + HM(I,
3): NEXT I
4550 RETURN
5000 REM NUJADM
5010 KSW = 0: IF TYPE$ = "LT" OR TYPE$ = "NT" THEN KSW = 1
5020 FOR I = 1 TO NH: FOR J = 1 TO NS:H1(I,J) = 0:H2(I,J) =
0: NEXT J: NEXT I
5030 FOR I = 1 TO EDGE:EV(I) = 0:EI(I) = 0: NEXT I
5040 FOR I = 1 TO RLC

```

```

5050 I1 = MRLC(I)
5060 FOR J = 1 TO NH:J2 = J + VV:J1 = CP1V(J2)
5070 FOR K = 1 TO EDGE:V1(K) = VOLT(K): NEXT K
5080 V1(J1) = 1.01 * VOLT(J1): IF VOLT(J1) = 0 THEN V1(J1) =
    0.001
5090 C(MRLC(1)) = 1.0E - 6 * ( EXP (V1(MRLC(1)) * FU) - 1 )
5100 C(MRLC(2)) = 1.0E - 6 * ( EXP (V1(MRLC(2)) * FU) - 1 )
5110 EV(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2)
    )
5120 EV(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1)
    )
5130 D8 = V1(J1) - VOLT(J1)
5140 FOR K = 1 TO EDGE:C1(K) = CURT(K): NEXT K
5150 C1(J1) = 1.01 * CURT(J1): IF CURT(J1) = 0 THEN C1(J1) =
    0.001
5160 C(MRLC(1)) = 1.0E - 6 * ( EXP (VOLT(MRLC(1)) * FU) - 1 )
5170 C(MRLC(2)) = 1.0E - 6 * ( EXP (VOLT(MRLC(2)) * FU) - 1 )
5180 EI(MRLC(1)) = C1(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
5190 EI(MRLC(2)) = C1(MRLC(2)) - C(MRLC(2)) + FV * C(MRLC(1))

5200 D9 = C1(J1) - CURT(J1)
5210 IF J2 < NODE THEN 5230
5220 W1(J) = (EI(I1) - ERR(I1)) / D9:W2(J) = (EV(I1) - ERR(I1)
    ) / D8: GOTO 5240
5230 W1(J) = (EV(I1) - ERR(I1)) / D8:W2(J) = (EI(I1) - ERR(I1)
    ) / D9: GOTO 5240
5240 NEXT J
5250 IF MODE# = "AU" AND ICC = 1 THEN 5270
5260 GOTO 5280
5270 J1 = PONT(I1): FOR J = 1 TO NS:Z1 = 0: FOR K = 1 TO NH:Z
    1 = Z1 + W2(K) * MS(K,J): NEXT K:H1(J1,J) = Z1: NEXT J
5280 FOR K = 1 TO NH:K1 = PONT(I1):Z1 = 0: FOR Z = 1 TO NH:Z
    1 = Z1 + W2(Z) * MT(Z,K): NEXT Z:UJAC(K1,K) = W1(K) + Z1:
    NEXT K
5300 NEXT I
5310 RETURN
5500 REM FACTOR
5510 FOR I = 1 TO NH: FOR J = 1 TO NH:A(I,J) = UJAC(I,J):W(I
    ,J) = FAUJ(I,J): NEXT J:ZF(I) = IF(I): NEXT I
5515 IF CODE < > 1 THEN 5540
5520 FOR I = 1 TO NH: FOR J = 1 TO NH:A(I,J) = UT(I,J):W(I,J
    ) = FJ(I,J): NEXT J:ZF(I) = YY(I): NEXT I
5540 FOR I = 1 TO NH: FOR J = 1 TO NH:W(I,J) = A(I,J): NEXT
    J: NEXT I
5550 FOR I = 1 TO NH:ZF(I) = I:MX = 0
5560 FOR J = 1 TO NH
5570 IF MX < ABS (W(I,J)) THEN MX = ABS (W(I,J))
5580 NEXT J
5590 IF MX = 0 THEN END
5600 D1(I) = MX
5610 NEXT I
5620 N1 = NH - 1
5630 IF N1 = 0 THEN RETURN
5640 FOR K = 1 TO N1:J = K:K1 = K + 1:I1 = ZF(K):CX = ABS
    (W(I1,K)) / D1(I1)

```

```

5650 FOR I = KP1 TO NH: I1 = ZP(I): MX = ABS (W(I1,K)) / D1(I
1)
5660 IF MX < = CX THEN 5680
5670 CX = MX: J = I
5680 NEXT I
5690 IF CX = 0 THEN END
5700 ZZ = ZP(J): ZP(J) = ZP(K): ZP(K) = ZZ
5710 FOR I = KP1 TO NH: I1 = ZP(I): W(I1,K) = W(I1,K) / W(ZZ,K
): R1 = - W(I1,K)
5720 FOR J = KP1 TO NH
5730 W(I1,J) = R1 * W(ZZ,J) + W(I1,J)
5740 NEXT J
5750 NEXT I
5760 NEXT K
5770 IF W(I1,NH) < > 0 THEN 5785
5780 PRINT " CAN NOT BE FACTORIZED"
5785 IF CODE = 1 THEN 5795
5790 FOR I = 1 TO NH: FOR J = 1 TO NH: UJAC(I,J) = A(I,J): FAU
J(I,J) = W(I,J): NEXT J: IP(I) = ZP(I): NEXT I
5792 GOTO 5800
5795 FOR I = 1 TO NH: FOR J = 1 TO NH: UT(I,J) = A(I,J): FJ(I,
J) = W(I,J): NEXT J: YY(I) = ZP(I): NEXT I
5800 RETURN
6000 REM FBSUB
6010 FOR I = 1 TO NH: FOR J = 1 TO NH: W(I,J) = FAUJ(I,J): NEXT
J: B(I) = EM(I): X(I) = ELXM(I): ZP(I) = IP(I): NEXT I
6012 IF CODE < > 1 THEN 6020
6015 FOR I = 1 TO NH: FOR J = 1 TO NH: W(I,J) = FJ(I,J): NEXT
J: B(I) = ETSI(I): X(I) = YZV(I): ZP(I) = YY(I): NEXT I
6020 IF NH > 1 THEN 6040
6030 X(1) = B(1) / W(1,1): RETURN
6040 I1 = ZP(1): X(1) = B(I1)
6050 FOR K = 2 TO NH: I1 = ZP(K): KP1 = K - 1
6060 ZZ = 0: FOR J = 1 TO KP1: ZZ = ZZ + W(I1,J) * X(J): NEXT
J
6070 X(K) = B(I1) - ZZ
6080 NEXT K
6090 X(NH) = X(NH) / W(I1,NH): N1 = NH - 1
6100 FOR K = N1 TO 1 STEP - 1: I1 = ZP(K): KP1 = K + 1
6110 ZZ = 0: FOR J = KP1 TO NH: ZZ = ZZ + W(I1,J) * X(J): NEXT
J
6120 X(K) = (X(K) - ZZ) / W(I1,K)
6130 NEXT K
6135 IF CODE = 1 THEN 6145
6140 FOR I = 1 TO NH: FOR J = 1 TO NH: FAUJ(I,J) = W(I,J): NEXT
J: EM(I) = B(I): ELXM(I) = X(I): IP(I) = ZP(I): NEXT I
6142 GOTO 6150
6145 FOR I = 1 TO NH: FOR J = 1 TO NH: FJ(I,J) = W(I,J): NEXT
J: ETSI(I) = B(I): YZV(I) = X(I): YY(I) = ZP(I): NEXT I
6150 RETURN
6200 REM FUNCB
6210 IF NVD = 0 THEN 6250
6220 IF NVD = 0 THEN 6270
6230 G = 0: FOR I = 1 TO EDGE: G = G + ((VOLT(I) - VD(I)) ^ 2)
* KV(I) + ((CURT(I) - ID(I)) ^ 2) * KI(I): NEXT I
6240 GOTO 6300

```

```

6250 G = 0: FOR I = 1 TO EDGE:G = G + ((VOLT(I) - VD(I)) ^ 2)
      * KV(I): NEXT I
6260 GOTO 6300
6270 G = 0: FOR I = 1 TO EDGE:G = G + ((CURT(I) - ID(I)) ^ 2)
      * KI(I): NEXT I
6300 RETURN
6500 REM GRADG
6510 FOR I = 1 TO NH: FOR J = 1 TO NH:UT(I,J) = UJAC(J,I): NEXT
      J: NEXT I
6520 FOR I = 1 TO NP:DGDP(I) = 0: NEXT I
6530 GOSUB 7000: REM DHDF
6540 CODE = 1: GOSUB 5500: REM FACTOR
6542 CODE = 0
6545 FOR Z = 1 TO 10:YA(Z) = 0: NEXT Z
6550 IF NVD = 0 THEN 6520
6560 FLAG = 1:ND = NVD: FOR I = 1 TO NVD:YA(I) = MVD(I): NEXT
      I
6570 FOR I = 1 TO EDGE:TP$(I) = VT$(I): NEXT I: GOTO 6600
6580 FLAG = 0:ND = NID: FOR I = 1 TO NID:YA(I) = MID(I): NEXT
      I
6590 FOR I = 1 TO EDGE:TP$(I) = IY$(I): NEXT I
6600 FOR IB = 1 TO ND
6610 FOR J = 1 TO NH:ETSI(J) = 0: NEXT J
6620 FC = 0:JZ = YA(IB):IK = CH(JZ)
6630 IF TP$(JZ) = "X" THEN 6690
6640 IF TP$(JZ) = "W" THEN 6700
6650 FOR Z = 1 TO NH:ETSI(Z) = - MS(Z,IK): NEXT Z
6660 IF ICC = 0 THEN 6730
6670 FC = 1: FOR Z = 1 TO NS:ESD(Z) = MD(IK,Z): NEXT Z
6680 GOTO 6730
6690 ETSI(IK) = 1: GOTO 6730
6700 FOR Z = 1 TO NH:ETSI(Z) = MT(IK,Z): NEXT Z
6710 IF ICC = 0 THEN 6730
6720 FC = 1: FOR Z = 1 TO NS:ESD(Z) = MS(IK,Z): NEXT Z
6725 GOTO 6730
6730 GOSUB 7500: REM SENSTY
6740 VIM = (CURT(JZ) - ID(JZ)) * KI(JZ): IF FLAG = 1 THEN VIM
      = (VOLT(JZ) - VD(JZ)) * KV(JZ)
6750 FOR Z = 1 TO NP:DGDP(Z) = DGDP(Z) + VIM * SI(Z): NEXT Z
6760 NEXT IB
6770 IF FLAG < > 0 AND NID < > 0 THEN 6580
6800 RETURN
7000 REM DHDF
7010 KW = 0: IF TYPE$ = "NT" OR TYPE$ = "LT" THEN KW = 1
7020 FOR I = 1 TO NH: FOR J = 1 TO NP:HDP(I,J) = 0: NEXT J: NEXT
      I
7030 FOR J = 1 TO NP
7040 FOR I = 1 TO NH:K = CPIV(I + VV)
7050 IF K < > MPAR(J) THEN 7150
7060 IF I < (NODE - VV) THEN 7110
7070 IF MCKT(K) = - 1 THEN HDP(I,J) = - XM(I)
7080 IF MCKT(K) = 0 THEN HDP(I,J) = - DXM(I)
7090 IF MCKT(K) = 1 THEN HDP(I,J) = DWM(I)
7100 GOTO 7160
7110 IF MCKT(K) = - 1 THEN HDP(I,J) = - WM(I)

```

```

7120 IF MCKT(K) = 0 THEN HDF(I,J) = - DWM(I)
7130 IF MCKT(K) = 1 THEN HDF(I,J) = - DXM(I)
7140 GOTO 7160
7150 NEXT I
7160 NEXT J
7170 FOR I = 1 TO NH: FOR J = 1 TO NP:HDF(I,J) = - HDF(I,J)
: NEXT J: NEXT I
7180 IF ICC = 0 THEN 7400
7190 FOR I = 1 TO NH: FOR J = 1 TO NS:HS(I,J) = HM(I,2) * MS
(I,J): NEXT J: NEXT I
7200 IF RLC < = 0 THEN 7220
7210 FOR I = 1 TO NH: FOR J = 1 TO NS:HS(I,J) = HS(I,J) + H1
(I,J): NEXT J: NEXT I
7220 FOR I = 1 TO NH
7230 FOR J = 1 TO NP:ZZ = 0: FOR Z = 1 TO NS:ZZ = ZZ + HS(I,
Z) * FDP(Z,J): NEXT Z:HF(I,J) = ZZ: NEXT J
7240 NEXT I
7250 FOR I = 1 TO NH: FOR J = 1 TO NP:HDF(I,J) = HDF(I,J) -
HF(I,J): NEXT J: NEXT I
7260 IF KW = 0 THEN 7400
7270 FOR I = 1 TO NH: FOR J = 1 TO NS:HS(I,J) = HM(I,4) * MS
(I,J): NEXT J: NEXT I
7280 IF RLC < = 0 THEN 7300
7290 FOR I = 1 TO NH: FOR J = 1 TO NS:HS(I,J) = HS(I,J) + H2
(I,J): NEXT J: NEXT I
7300 FOR I = 1 TO NH
7310 FOR J = 1 TO NP
7320 ZZ = 0: FOR Z = 1 TO NS:ZZ = ZZ + HS(I,Z) * DDP(Z,J): NEXT
Z:HF(I,J) = ZZ
7330 NEXT J
7340 NEXT I
7350 FOR I = 1 TO NH: FOR J = 1 TO NP:HDF(I,J) = HDF(I,J) -
HF(I,J): NEXT J: NEXT I
7400 RETURN
7500 REM SENSTY
7510 CODE = 1: GOSUB 6000: REM FBSUB
7515 CODE = 0
7520 FOR Z = 1 TO NP:ZZ = 0: FOR ZK = 1 TO NH:ZZ = ZZ + YZV(
ZK) * HDP(ZK,Z): NEXT ZK:SI(Z) = ZZ: NEXT Z
7530 IF FC = 0 THEN 7550
7540 FOR Z = 1 TO NP:ZZ = 0: FOR ZK = 1 TO NS:ZZ = ZZ + ESD(
ZK) * FDP(ZK,Z): NEXT ZK:SI(Z) = SI(Z) + ZZ: NEXT Z
7550 RETURN
8000 REM NORMAL
8010 K = 1:CST = 0.1:RM = ABS(DGDF(1))
8012 IF G < = 1.0E - 3 THEN CST = 0.05
8014 IF G < = 1.0E - 4 THEN CST = 0.025
8020 IF NP < 2 THEN 8070
8030 FOR I = 2 TO NP:RN = ABS(DGDF(I))
8040 IF RN > = RM THEN 8060
8050 RM = RN:K = I
8060 NEXT I
8070 FCKT = WCKT(MFAR(K))
8080 FOR I = 1 TO NP:DGDF(I) = DGDF(I) * FCKT * CST / RM: NEXT
I
8100 RETURN

```

โปรแกรม DCMAIN

```
1500 REM DCMAIN
1501 HOME
1502 INPUT "CREATE NEW FILE(1/0)";Z
1504 IF Z = 0 THEN 1600
1505 PRINT : PRINT "EDGE NODE,VV II RLC"
1506 INPUT EDGE: VTAB 4: HTAB 6: INPUT NODE: VTAB 4: HTAB 11
: INPUT VV: VTAB 4: HTAB 14: INPUT II: VTAB 4: HTAB 17: INPUT
RLC
1520 NVD = 1:NID = 1:NP = 2:NS = VV + II:NH = EDGE - NS:IM =
0:IL = 0:ICC = 0:MODE$ = "AY"
1530 DIM MVD(NVD),MID(NID),CH(EDGE),MPAR(NP),SS(NS),VT$(EDGE
),IY$(EDGE),H2(NH,NS)
1540 DIM WCKT(EDGE),MA(NODE,EDGE),RONE(NODE),RPIV(NODE),CIDX
(EDGE),WORK(NH)
1550 DIM XM(NH),WM(NH),MM(NS),FM(NS),HM(NH,4),UJAC(NH,NH),CK
T(EDGE),VOLT(EDGE),CURT(EDGE),H1(NH,NS),FAUJ(NH,NH),ELXM(
NH),ERR(EDGE)
1560 DIM MT(NH,NH),MS(NH,NS),MD(NS,NS),MCKT(EDGE),CPIV(EDGE)
,MRLC(RLC),IP(NH),PONT(EDGE),KEEP(NH),C(EDGE),WU(NH,NH)
1570 DIM W1(NH),W2(NH),W3(NH),W4(NH),V1(EDGE),C1(EDGE),EV(ED
GE),EI(EDGE),D1(NH)
1580 DIM W(NH,NH),B(NH),X(NH),A(NH,NH),Z1$(EDGE)
1590 GOSUB 4500: REM INITIAL
1600 GOSUB 4800: REM RETRIEVE FILE
1610 IF PT = 1 THEN PR# 1
2070 GOSUB 2500: REM TSDMAT
2080 PRINT
2142 PRINT CHR$(4);"BLOAD CHAIN,A520"
2144 CALL 520"DCAY"
2450 END
2500 REM TSDMAT
2520 FOR I = 1 TO NODE:ZZ = 0: FOR J = 1 TO EDGE:ZZ = ZZ + ABS
(MA(I,J)): NEXT J:RONE(I) = ZZ: NEXT I
2530 FOR J = 1 TO EDGE
2540 FOR I = 1 TO NODE
2550 IF MA(I,J) = 0 THEN 2570
2560 CPIV(J) = CPIV(J) + RONE(I)
2570 NEXT I
2580 NEXT J
2590 K = 1
2600 FOR I = 1 TO EDGE
2610 J = 1
2620 IF I = SS(J) THEN 2650
2630 J = J + 1: IF J < = NS THEN 2620
2640 WORK(K) = I:K = K + 1
2650 NEXT I
2660 IF VV = 0 THEN 2680
2670 FOR I = 1 TO VV:CIDX(I) = SS(I): NEXT I
2680 IF II = 0 THEN 2700
2690 FOR I = 1 TO II:CIDX(EDGE - II + I) = SS(VV + I): NEXT
I
2700 L = NH
2710 FOR J = (VV + 1) TO (EDGE - II - 1):K = 1:TEST = CPIV(W
ORK(1))
2720 FOR I = 2 TO L
```

```

2730 IF TEST > = CPIV(WORK(I)) THEN 2750
2740 TEST = CPIV(WORK(I)):K = I
2750 NEXT I
2760 CIDX(J) = WORK(K): IF K = L THEN 2780
2770 FOR I = K TO (L - 1):WORK(I) = WORK(I + 1): NEXT I
2780 L = L - 1
2790 NEXT J
2800 CIDX(J) = WORK(1)
2810 FOR I = 1 TO NODE:RPIV(I) = 1: NEXT I
2820 N1 = 0:N2 = NODE:J = 0
2830 J = J + 1:J1 = CIDX(J):MN = 0:TEST = EDGE + 1
2840 FOR I1 = 1 TO NODE
2850 IF MA(I1,J1) = 0 THEN 2930
2860 IF J = 1 THEN 2900
2870 K1 = 1: IF N1 = 0 THEN 2900
2880 IF I1 = RPIV(K1) THEN 2930
2890 K1 = K1 + 1: IF K1 < = N1 THEN 2880
2900 NN = 0: FOR Z = 1 TO EDGE:NN = NN + ABS (MA(I1,Z)): NEXT
Z
2910 IF TEST < = NN THEN 2930
2920 TEST = NN:MN = I1
2930 NEXT I1
2940 IF MN = 0 THEN 3080
2950 N1 = N1 + 1:CPIV(N1) = J1
2960 I = N1
2970 IF RPIV(I) = MN THEN 2990
2980 I = I + 1: IF I < = NODE THEN 2970
2990 RPIV(I) = RPIV(N1):RPIV(N1) = MN:I3 = RPIV(N1):J3 = CPIV
(N1):ZZ = MA(I3,J3)
3000 FOR Z = 1 TO EDGE:MA(I3,Z) = MA(I3,Z) / ZZ: NEXT Z
3010 FOR K2 = 1 TO NODE
3020 IF RPIV(K2) = MN THEN 3060
3030 K3 = RPIV(K2): IF MA(K3,J1) = 0 THEN 3060
3040 ZZ = MA(K3,J3)
3050 FOR Z = 1 TO EDGE:MA(K3,Z) = MA(K3,Z) - MA(I3,Z) * ZZ: NEXT
Z
3060 NEXT K2
3070 GOTO 3090
3080 CPIV(N2) = J1:N2 = N2 + 1
3090 IF N1 < (NODE - 1) THEN 2830
3100 FOR I = (J + 1) TO EDGE:CPIV(I) = CIDX(I): NEXT I
3110 PRINT : PRINT : PRINT " MCPIV =": FOR I = 1 TO EDGE: PRINT
CPIV(I):" "": NEXT I: PRINT
3120 PRINT : PRINT " MA ="
3130 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(RPIV(I),
CPIV(J)):" "": NEXT J: PRINT : NEXT I
3140 FOR I = 1 TO NH: FOR J = 1 TO NH:MT(I,J) = 0: NEXT J: NEXT
I
3150 FOR I = 1 TO NH: FOR J = 1 TO NS:MS(I,J) = 0: NEXT J: NEXT
I
3160 FOR I = 1 TO NS: FOR J = 1 TO NS:MD(I,J) = 0: NEXT J: NEXT
I
3170 FOR I = (VV + 1) TO (NODE - 1):I1 = RPIV(I):I2 = I - VV
3180 FOR J = NODE TO (EDGE - I1):J1 = CPIV(J):J2 = J - VV:MT
(I2,J2) = - MA(I1,J1):MT(J2,I2) = MA(I1,J1): NEXT J

```



```

3190 IF II = 0 THEN 3210
3200 FOR J = (EDGE - II + 1) TO EDGE:J1 = CPIV(J):J2 = J - N
      H:MS(I2,J2) = - MA(I1,J1): NEXT J
3210 NEXT I
3220 IF VV = 0 THEN 3260
3230 FOR I = 1 TO VV:I1 = RPIV(I): FOR J = NODE TO (EDGE - I
      I):J1 = CPIV(J):I2 = J - VV:MS(I2,I) = MA(I1,J1): NEXT J:
      NEXT I
3240 IF II = 0 THEN 3260
3250 FOR I = 1 TO VV:I1 = RPIV(I):I2 = I: FOR J = (EDGE - II
      + 1) TO EDGE:J1 = CPIV(J):J2 = J - NH:MD(I2,J2) = - MA(
      I1,J1):MD(J2,I2) = MA(I1,J1): NEXT J: NEXT I
3260 PRINT : PRINT : PRINT "  MATRIX T ="
3270 FOR I = 1 TO NH: FOR J = 1 TO NH: PRINT MT(I,J):"  ": NEXT
      J: PRINT : NEXT I
3280 PRINT : PRINT : PRINT "  MATRIX S ="
3290 FOR I = 1 TO NH: FOR J = 1 TO NS: PRINT MS(I,J):"  ": NEXT
      J: PRINT : NEXT I
3300 PRINT : PRINT : PRINT "  MATRIX D ="
3310 FOR I = 1 TO NS: FOR J = 1 TO NS: PRINT MD(I,J):"  ": NEXT
      J: PRINT : NEXT I
3320 RETURN
4500 REM INITIAL
4502 HOME
4505 PRINT "E N1 N1 TYPE K.OHM VOLT M.AMF"
4510 VTAB 24: GET Z$: IF Z$ = CHR$(13) THEN 4530
4515 I = VAL(Z$):J = I * 2: IF I > ED THEN 4510
4520 VTAB J: PRINT I: VTAB J: HTAB 3: INPUT EV(I): VTAB J: HTAB
      6: INPUT EI(I): VTAB J: HTAB 10: GET Z1$(I): VTAB J: HTAB
      10: PRINT Z1$(I)
4525 VTAB J: HTAB 15: INPUT CKT(I): VTAB J: HTAB 23: INPUT V
      OLT(I): VTAB J: HTAB 31: INPUT CURT(I): GOTO 4510
4530 J = 0: TYPE$ = "LD"
4535 FOR I = 1 TO EDGE:MCKT(I) = 0
4540 IF Z1$(I) = CHR$(82) THEN MCKT(I) = - 1
4545 IF Z1$(I) = CHR$(76) THEN MCKT(I) = 0
4550 IF Z1$(I) = CHR$(67) THEN MCKT(I) = 1
4555 IF Z1$(I) = CHR$(78) THEN 4560
4557 GOTO 4565
4560 J = J + 1:MRLOC(J) = I:TYPE$ = "ND":MCKT(I) = 2
4565 NEXT I
4570 HOME
4580 FOR I = 1 TO NS: INPUT "VECTOR OF INDEPENDENT SOURCE ":
      SS(I): INPUT "VALUE OF INDEPENDENT SOURCE ":FM(I): NEXT I
4590 FOR I = 1 TO NODE: FOR J = 1 TO EDGE:MA(I,J) = 0: NEXT
      J: NEXT I
4620 FOR I = 1 TO EDGE:MA(EV(I),I) = 1:MA(EI(I),I) = - 1: NEXT
      I
4640 INPUT "ALPHA F":FV: INPUT "ALPHA R":FW: INPUT "VT":FU
4650 INPUT "YOU WANT HARD COPY(1/0)":PT
4660 GOSUB 4700: REM CREATE FILE
4670 RETURN
4700 REM CREATE FILE
4710 HOME : INPUT "FILE NAME":Z$
4720 PRINT CHR$(4):"OPEN ":Z$

```

```

4730 PRINT CHR$(4);"DELETE ";Z$
4740 PRINT CHR$(4);"OPEN ";Z$
4745 PRINT CHR$(4);"WRITE ";Z$
4750 PRINT EDGE: PRINT NODE: PRINT VV: PRINT II: PRINT RLC: PRINT
NS: PRINT NH: PRINT IM: PRINT IL: PRINT ICC: PRINT MODE$:
PRINT NVD: PRINT NID: PRINT NF: PRINT TYPE$
4760 FOR I = 1 TO NS: PRINT SS(I): PRINT FM(I): NEXT I
4770 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(I,J): NEXT
J: NEXT I
4780 FOR I = 1 TO EDGE: PRINT MCKT(I): PRINT CKT(I): PRINT V
OLT(I): PRINT CURT(I): NEXT I
4782 PRINT FU: PRINT FV: PRINT FW
4785 IF RLC < = 0 THEN 4792
4790 FOR I = 1 TO RLC: PRINT MRLC(I): NEXT I
4792 PRINT CHR$(4);"CLOSE ";Z$
4795 RETURN
4800 REM RETRIEVE FILE
4810 HOME : INPUT "FILE NAME";Z$
4820 PRINT CHR$(4);"OPEN ";Z$
4845 PRINT CHR$(4);"READ ";Z$
4850 INPUT EDGE: INPUT NODE: INPUT VV: INPUT II: INPUT RLC: INPUT
NS: INPUT NH: INPUT IM: INPUT IL: INPUT ICC: INPUT MODE$:
INPUT NVD: INPUT NID: INPUT NF: INPUT TYPE$
4860 FOR I = 1 TO NS: INPUT SS(I): INPUT FM(I): NEXT I
4870 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: INPUT MA(I,J): NEXT
J: NEXT I
4880 FOR I = 1 TO EDGE: INPUT MCKT(I): INPUT CKT(I): INPUT V
OLT(I): INPUT CURT(I): NEXT I
4882 INPUT FU: INPUT FV: INPUT FW
4885 IF RLC < = 0 THEN 4892
4890 FOR I = 1 TO RLC: INPUT MRLC(I): NEXT I
4892 PRINT CHR$(4);"CLOSE ";Z$
4900 RETURN

```



โปรแกรม DCAY

```
2100 REM DCAY
2104 AID = 0: IF TYPE# = "ND" THEN AID = 1
2108 IF RLC = 0 THEN 2130
2110 FOR I = 1 TO EDGE:ERR(I) = 0:PONT(I) = 0: NEXT I
2120 FOR I = 1 TO NH:K = CPIV(VV + I):PONT(K) = I: NEXT I
2130 IF IM > 0 OR IL > 0 THEN 2170
2140 FOR I = 1 TO NH:K = I + VV:J = CPIV(K)
2150 XM(I) = CURT(J): IF K < NODE THEN XM(I) = VOLT(J)
2160 NEXT I
2165 REM ENTO
2170 IF AID < > 1 THEN 2190
2180 ENRM = 0:E1 = 0:E2 = 0:SECH = 0:NLSW = 0: FOR I = 1 TO NH:K
    EEP(I) = XM(I): NEXT I
2190 DONE = 0:ITER = 0:RTEP = 1: FOR I = 1 TO NH: FOR J = 1 TO N
    S:HI(I,J) = 0: NEXT J: NEXT I
2200 GOSUB 3000: REM HMAT
2205 REM LOOP
2210 FOR I = 1 TO NH
2220 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 + MT(I,J) * XM(J): NEXT J
2230 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MS(I,J) * FM(J): NEXT J
2240 WM(I) = Z1 + Z2
2250 NEXT I
2260 FOR I = 1 TO NS
2270 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 - MS(J,I) * XM(J): NEXT J
2280 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MD(I,J) * FM(J): NEXT J
2290 MM(I) = Z1 + Z2
2300 NEXT I
2310 FOR I = 1 TO NH:EM(I) = HM(I,1) * XM(I) + HM(I,2) * WM(I):
    NEXT I
2320 IF RLC = 0 THEN 2360
2330 GOSUB 3500: REM CONVERT
2340 GOSUB 4000: REM ERROR
2350 FOR I = 1 TO RLC:J = MRLC(I):K = PONT(J):EM(K) = ERR(J): NEXT
    I
2360 Z1 = 0:Z2 = 0:Z3 = 0: FOR I = 1 TO NH:Z1 = Z1 + EM(I) * EM(
    I):Z2 = Z2 + XM(I) * XM(I):Z3 = Z3 + WM(I) * WM(I): NEXT I:E
    NRM = SQR(Z1):XNRM = SQR(Z2):WNRM = SQR(Z3)
2370 IF AID < > 1 THEN 2400
2380 E2 = E1:E1 = ENRM:ENRM = 0:XNRM = 0:WNRM = 0
2390 FOR I = 1 TO NH:ENRM = ENRM + ABS(EM(I)):XNRM = XNRM + ABS
    (XM(I)):WNRM = WNRM + ABS(WM(I)): NEXT I
2400 IF ENRM > (XNRM + WNRM) * 1.0E - 6 THEN 2430
2410 IF MODE# < > "AU" OR AID = 0 THEN 2860
2420 DONE = 1: GOTO 2570
2430 IF TYPE# = "LD" AND ITER > 0 THEN 2600
2440 IF AID < > 1 THEN 2550
2450 IF SECH = 0 THEN EO = ENRM
2460 ON NLSW GOTO 2720,2740,2800
2470 IF SECH < > 0 THEN 2500
2480 IF ITER = 0 THEN 2550
2490 IF ITER < > 0 THEN 2570
2500 IF EO < ENRM THEN 2530
2510 NLSW = 2
2515 REM NLDOUB
2530 RTEP = RTEP * 2: GOTO 2620
```

```

2530 NLSW = 1
2535 REM NLHALF
2540 RTEP = RTEP / 2: GOTO 2620
2545 REM NLENT0
2550 DUMY = 1: GOSUB 4500: REM UJACM
2560 IF RLC = 0 THEN 2590
2565 REM NLENT1
2570 GOSUB 5000: REM NUJACM
2580 IF DONE = 1 THEN 2860
2585 REM ENT2
2590 GOSUB 5500: REM FACTOR
2595 REM ENT3
2600 GOSUB 6000: REM FBSUB
2610 IF AID < > 1 THEN 2650
2615 REM NLENT2
2620 SECH = SECH + 1: IF SECH < = 100 THEN 2640
2630 PRINT : PRINT " SECH>100:STOP": END
2635 REM NLENT3
2640 FOR I = 1 TO NH:XM(I) = KEEP(I): NEXT I
2650 FOR I = 1 TO NH:XM(I) = XM(I) - RTEP * ELXM(I): NEXT I
2660 IF AID < > 1 THEN 2690
2670 IF NLSW < > 4 THEN 2210
2680 NLSW = 0:SECH = 0:ENRM = 0:E1 = 0:E2 = 0:RTEP = 1: FOR I =
1 TO NH:KEEP(I) = XM(I): NEXT I
2690 ITER = ITER + 1
2700 IF ITER < = 100 THEN 2210
2710 PRINT : PRINT : PRINT " ITER>100:STOP": END
2715 REM NLSW1
2720 IF EO < = ENRM THEN 2540
2730 RANL = 0:RBNL = RTEP:RCNL = RTEP * 2:EA = EO:EB = ENRM:EC =
E1: GOTO 2770
2735 REM NLSW2
2740 IF ENRM < E1 THEN 2520
2750 RCNL = RTEP:RBNL = RTEP / 2:EA = E2:EB = E1:EC = ENRM
2760 RANL = RBNL / 2: IF SECH = 2 THEN RANL = 0
2765 REM NLENT4
2770 EA = EA * (RCNL - RBNL):ENRM = EB * (RANL - RCNL):EC = EC *
(RBNL - RANL)
2780 RTEP = (EA * (RCNL + RBNL) + ENRM * (RANL + RCNL) + EC * (R
BNL + RANL)) / (2 * (EA + ENRM + EC))
2790 NLSW = 3: GOTO 2640
2795 REM NLSW3
2800 IF ENRM < = EB THEN 2820
2810 RTEP = RBNL:NLSW = 4: GOTO 2640
2820 ITER = ITER + 1: IF ITER < = 100 THEN 2840
2830 PRINT : PRINT " ITER > 100 :STOP": END
2840 NLSW = 0:SECH = 0:EO = ENRM:E1 = 0:E2 = 0:RTEP = 1: FOR I =
1 TO NH:KEEP(I) = XM(I): NEXT I
2850 GOTO 2570
2855 REM OUT
2860 IF RLC < > 0 THEN 2880
2870 GOSUB 3500: REM CONVERT
2880 PRINT : PRINT : PRINT " NO.OF ITERATIONS = ":ITER
2890 FOR I = 1 TO EDGE: PRINT "VOLT("":I:"") = "":VOLT(I):" VOLT
CURT("":I:"") = "":CURT(I):" MILLIAMPS": PRINT : NEXT I
2900 END

```

```

3000 REM HMAT
3010 FOR I = 1 TO NH: FOR J = 1 TO 4:HM(I,J) = 0: NEXT J: NEXT
  I
3020 FOR I = 1 TO (NODE - 1 - VV)
3030 K = I + VV: I1 = CPIV(K): IF MCKT(I1) > 1 THEN 3100
3040 IF MCKT(I1) = - 1 THEN 3070
3050 IF MCKT(I1) = 0 THEN 3080
3060 IF MCKT(I1) = 1 THEN 3090
3065 PRINT " CHECK CKT:STOP": END
3070 HM(I,1) = 1:HM(I,2) = - CKT(I1): GOTO 3100
3080 HM(I,1) = 1:HM(I,4) = - CKT(I1): GOTO 3100
3090 HM(I,2) = 1:HM(I,3) = - CKT(I1)
3100 NEXT I
3110 FOR I = (NODE - VV) TO NH
3120 K = I + VV: I1 = CPIV(K): IF MCKT(I1) > 1 THEN 3200
3130 IF MCKT(I1) = - 1 THEN 3170
3140 IF MCKT(I1) = 0 THEN 3180
3150 IF MCKT(I1) = 1 THEN 3190
3160 PRINT " CHECK CKT:STOP": END
3170 HM(I,1) = - CKT(I1):HM(I,2) = 1: GOTO 3200
3180 HM(I,2) = 1:HM(I,3) = - CKT(I1): GOTO 3200
3190 HM(I,1) = 1:HM(I,4) = - CKT(I1): GOTO 3200
3200 NEXT I
3210 RETURN
3500 REM CONVERT
3510 KSW = 0: IF TYPE$ = "LT" OR TYPE$ = "NT" THEN KSW = 1
3520 K = VV
3530 FOR I = 1 TO EDGE:J = CPIV(I)
3540 IF I < = VV THEN 3600
3550 IF I < NODE THEN 3630
3560 IF I < = (EDGE - II) THEN 3660
3570 K = K + 1:VOLT(J) = MM(K):CURT(J) = FM(K)
3580 IF KSW < > 1 THEN 3700
3590 DV(J) = DMM(K):DC(J) = DFM(K): GOTO 3700
3600 VOLT(J) = FM(I):CURT(J) = MM(I)
3610 IF KSW < > 1 THEN 3700
3620 DV(J) = DFM(I):DC(J) = DMM(I): GOTO 3700
3630 L = I - VV:VOLT(J) = XM(L):CURT(J) = WM(L)
3640 IF KSW < > 1 THEN 3700
3650 DV(J) = DXM(L):DC(J) = DWM(L): GOTO 3700
3660 L = I - VV:VOLT(J) = WM(L):CURT(J) = XM(L)
3670 IF KSW < > 1 THEN 3700
3680 DV(J) = DWM(L):DC(J) = DXM(L): GOTO 3700
3700 NEXT I
3710 RETURN
4000 REM ERROR
4010 FOR J = 1 TO NRC:C(MRLC(I)) = 1.0E - 6 * ( EXP (VOLT(MRLC(
  I)) * FU) - 1): NEXT I
4030 ERR(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
4040 ERR(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1))
4050 RETURN
4500 REM UJACM
4510 FOR I = 1 TO NH: FOR J = 1 TO NH:UJAC(I,J) = 0:WU(I,J) = 0
  : NEXT J: NEXT I

```

```

4520 FOR I = 1 TO NH:WU(I,I) = HM(I,4) + HM(I,2): NEXT I
4530 FOR I = 1 TO NH: FOR J = 1 TO NH:Z1 = 0: FOR K = 1 TO NH:Z
1 = Z1 + WU(I,K) * MT(K,J): NEXT K:UJAC(I,J) = Z1: NEXT J: NEXT
I
4540 FOR I = 1 TO NH:UJAC(I,I) = UJAC(I,I) + HM(I,1) + HM(I,3):
NEXT I
4550 RETURN
5000 REM NUJACM
5010 KSW = 0: IF TYPE# = "LT" OR TYPE# = "NT" THEN KSW = 1
5020 FOR I = 1 TO NH: FOR J = 1 TO NS:H1(I,J) = 0: NEXT J: NEXT
I
5030 FOR I = 1 TO EDGE:EV(I) = 0:EI(I) = 0: NEXT I
5040 FOR I = 1 TO RLC
5050 I1 = MRLC(I)
5060 FOR J = 1 TO NH:J2 = J + VV:J1 = CPIV(J2)
5070 FOR K = 1 TO EDGE:V1(K) = VOLT(K): NEXT K
5080 V1(J1) = 1.01 * VOLT(J1): IF VOLT(J1) = 0 THEN V1(J1) = 0.0
01
5090 C(MRLC(1)) = 1.0E - 6 * ( EXP (V1(MRLC(1)) * FU) - 1)
5100 C(MRLC(2)) = 1.0E - 6 * ( EXP (V1(MRLC(2)) * FU) - 1)
5110 EV(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
5120 EV(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1))
5130 D8 = V1(J1) - VOLT(J1)
5140 FOR K = 1 TO EDGE:C1(K) = CURT(K): NEXT K
5150 C1(J1) = 1.01 * CURT(J1): IF CURT(J1) = 0 THEN C1(J1) = 0.0
01
5160 C(MRLC(1)) = 1.0E - 6 * ( EXP (VOLT(MRLC(1)) * FU) - 1)
5170 C(MRLC(2)) = 1.0E - 6 * ( EXP (VOLT(MRLC(2)) * FU) - 1)
5180 EI(MRLC(1)) = C1(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
5190 EI(MRLC(2)) = C1(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1))
5200 D9 = C1(J1) - CURT(J1)
5210 IF J2 < NODE THEN 5230
5220 W1(J) = (EI(I1) - ERR(I1)) / D9:W2(J) = (EV(I1) - ERR(I1)) /
D8: GOTO 5240
5230 W1(J) = (EV(I1) - ERR(I1)) / D8:W2(J) = (EI(I1) - ERR(I1)) /
D9: GOTO 5240
5240 NEXT J
5250 IF MODE# = "AU" AND ICC = 1 THEN 5270
5260 GOTO 5280
5270 J1 = PONT(I1): FOR J = 1 TO NS:Z1 = 0: FOR K = 1 TO NH:Z1 =
Z1 + W2(K) * MS(K,J): NEXT K:H1(J1,J) = Z1: NEXT J
5280 FOR K = 1 TO NH:K1 = PONT(I1):Z1 = 0: FOR Z = 1 TO NH:Z1 =
Z1 + W2(Z) * MT(Z,K): NEXT Z:UJAC(K1,K) = W1(K) + Z1: NEXT K
5300 NEXT I
5310 RETURN
5500 REM FACTOR
5510 FOR I = 1 TO NH: FOR J = 1 TO NH:A(I,J) = UJAC(I,J):W(I,J)
= FAUJ(I,J): NEXT J: NEXT I
5540 FOR I = 1 TO NH: FOR J = 1 TO NH:W(I,J) = A(I,J): NEXT J: NEXT
I
5550 FOR I = 1 TO NH:IF(I) = I:MX = 0
5560 FOR J = 1 TO NH
5570 IF MX < ABS (W(I,J)) THEN MX = ABS (W(I,J))
5580 NEXT J
5590 IF MX = 0 THEN END

```

```

5600 D1(I) = MX
5610 NEXT I
5620 N1 = NH - 1
5630 IF N1 = 0 THEN RETURN
5640 FOR K = 1 TO N1: J = K: KP1 = K + 1: I1 = IP(K): CX = ABS (W(
    I1, K)) / D1(I1)
5650 FOR I = KP1 TO NH: I1 = IP(I): MX = ABS (W(I1, K)) / D1(I1)
5660 IF MX < CX THEN 5680
5670 CX = MX: J = I
5680 NEXT I
5690 IF CX = 0 THEN END
5700 ZZ = IP(J): IF(J) = IP(K): IP(K) = ZZ
5710 FOR I = KP1 TO NH: I1 = IP(I): W(I1, K) = W(I1, K) / W(ZZ, K): R
    1 = - W(I1, K)
5720 FOR J = KP1 TO NH
5730 W(I1, J) = R1 * W(ZZ, J) + W(I1, J)
5740 NEXT J
5750 NEXT I
5760 NEXT K
5770 IF W(I1, NH) < > 0 THEN 5790
5780 PRINT " CAN NOT BE FACTORIZED"
5790 FOR I = 1 TO NH: FOR J = 1 TO NH: UJAC(I, J) = A(I, J): FAUJ(I
    , J) = W(I, J): NEXT J: NEXT I
5800 RETURN
6000 REM FBSUB
6010 FOR I = 1 TO NH: FOR J = 1 TO NH: W(I, J) = FAUJ(I, J): NEXT
    J: B(I) = EM(I): X(I) = ELXM(I): NEXT I
6020 IF NH > 1 THEN 6040
6030 X(1) = B(1) / W(1, 1): RETURN
6040 I1 = IP(1): X(1) = B(I1)
6050 FOR K = 2 TO NH: I1 = IP(K): KP1 = K - 1
6060 ZZ = 0: FOR J = 1 TO KP1: ZZ = ZZ + W(I1, J) * X(J): NEXT J
6070 X(K) = B(I1) - ZZ
6080 NEXT K
6090 X(NH) = X(NH) / W(I1, NH): N1 = NH - 1
6100 FOR K = N1 TO 1 STEP - 1: I1 = IP(K): KP1 = K + 1
6110 ZZ = 0: FOR J = KP1 TO NH: ZZ = ZZ + W(I1, J) * X(J): NEXT J
6120 X(K) = (X(K) - ZZ) / W(I1, K)
6130 NEXT K
6140 FOR I = 1 TO NH: FOR J = 1 TO NH: FAUJ(I, J) = W(I, J): NEXT
    J: EM(I) = B(I): ELXM(I) = X(I): NEXT I
6150 RETURN

```

START TRMAIN

```

100 REM START TRMAIN
105 DIM MA(10,10),MCKT(10),CKT(10),VOLT(10),CURT(10),DV(10),
    DC(10),SS(10),MRLC(10),MB(10),MY(10),MZ(10),YB(10),YR(10)

109 HOME
110 INPUT "CREATE NEW FILE ";Z$
120 IF Z$ = "N" THEN 570
130 HOME : PRINT "EDGE NODE VV II"
140 VTAB 2: INPUT EDGE: VTAB 2: HTAB 6: INPUT NODE: VTAB 2: HTAB
    11: INPUT VV: VTAB 2: HTAB 14: INPUT II
150 INPUT "NUMBER VOLTAGE TO BE PLOTTED ";NA
160 INPUT "NUMBER CURRENT TO BE PLOTTED ";NC
170 INPUT "NUMBER DC BIAS SOURCE ";NB
180 MODE$ = "AY":NS = VV + II:NH = EDGE - NS:NF = 0:ICC = 0
190 HOME : INPUT "INITIAL TIME";TO: INPUT "TIME STEP ";DT
220 TF = TO + DT * 80:NZ = INT ((TF - TO) / DT + 1)
230 HOME : PRINT "E N1 N2 TYPE OHM VOLT AMP "
235 VTAB 24: GET Z$: IF Z$ = CHR$(13) THEN 260
240 I = VAL (Z$):J = I * 2: IF I > ED THEN 235
245 VTAB J: PRINT I: VTAB J: HTAB 3: INPUT DV(I): VTAB J: HTAB
    6: INPUT DC(I): VTAB J: HTAB 10: GET Z1$(I): VTAB J: HTAB
    10: PRINT Z1$(I)
250 VTAB J: HTAB 15: INPUT CKT(I): VTAB J: HTAB 23: INPUT VO
    LT(I): VTAB J: HTAB 31: INPUT CURT(I): GOTO 235
260 FOR I = 1 TO NODE: FOR J = 1 TO EDGE:MA(I,J) = 0: NEXT J
    : NEXT I
270 FOR I = 1 TO EDGE:MA(DV(I),I) = 1:MA(DC(I),I) = - 1: NEXT
    I
275 RLC = 0:TYPE$ = "LT"
280 FOR I = 1 TO EDGE:MCKT(I) = 0
285 IF Z1$(I) = CHR$(82) THEN MCKT(I) = - 1
290 IF Z1$(I) = CHR$(67) THEN MCKT(I) = 1
295 IF Z1$(I) = CHR$(78) THEN 305
300 GOTO 310
305 RLC = RLC + 1:MRLC(RLC) = 1:TYPE$ = "NT":MCKT(I) = 2
310 NEXT I
315 IF RLC < = 0 THEN 320
317 HOME : INPUT "ALPHA F ";FV: INPUT "ALPHA R ";FW: INPUT "
    VT ";FU
320 IF NB < = 0 THEN 340
330 FOR I = 1 TO NB: INPUT "VECTOR OF DC BIAS SOURCE ";MB(I)
    : NEXT I
340 IF NA < = 0 THEN 360
350 FOR I = 1 TO NA: INPUT "VECTOR VOLT TO BE PLOTTED ";MY(I)
    ): NEXT I
360 IF NC < = 0 THEN 374
370 FOR I = 1 TO NC: INPUT "VECTOR CURRENT TO BE PLOTTED ";M
    Z(I): NEXT I
374 HOME : FOR I = 1 TO NS: INPUT "INDEPENDENT SOURCE VECTOR
    ";SS(I): NEXT I
376 HOME : PRINT "FM = K1+K2*SIN(314.16*TA)"
378 FOR I = 1 TO NS: INPUT "K1 ";YD(I): INPUT "K2 ";YR(I): NEXT
    I
380 HOME : INPUT "FILE NAME ";Z$
390 PRINT CHR$(4);"OPEN ";Z$

```



```
400 PRINT CHR$(4);"DELETE ";Z$
410 PRINT CHR$(4);"OPEN ";Z$
420 PRINT CHR$(4);"WRITE ";Z$
430 PRINT EDGE: PRINT NODE: PRINT VV: PRINT II: PRINT RLC: PRINT
NB: PRINT NA: PRINT NC: PRINT MODE$: PRINT NS: PRINT NH: PRINT
NF: PRINT ICC: PRINT TO: PRINT DT: PRINT TYPE$: PRINT TF:
PRINT NZ
440 FOR I = 1 TO NS: PRINT SS(I): NEXT I
450 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(I,J): NEXT
J: NEXT I
460 FOR I = 1 TO EDGE: PRINT MCKT(I): PRINT CKT(I): PRINT VO
LT(I): PRINT CURT(I): NEXT I
470 IF RLC < = 0 THEN 490
480 FOR I = 1 TO RLC: PRINT MRLC(I): NEXT I
490 IF NB < = 0 THEN 510
500 FOR I = 1 TO NB: PRINT MB(I): NEXT I
510 IF NA < = 0 THEN 530
520 FOR I = 1 TO NA: PRINT MY(I): NEXT I
530 IF NC < = 0 THEN 540
535 FOR I = 1 TO NC: PRINT MZ(I): NEXT I
540 PRINT FV: PRINT FW: PRINT FU
550 FOR I = 1 TO NS: PRINT YQ(I): PRINT YR(I): NEXT I
560 PRINT CHR$(4);"CLOSE ";Z$
570 PRINT CHR$(4);"RUN TMAIN"
600 END
```

TUSUNSU AUTRAN

```

100 REM AUTRAN
110 HOME : INPUT "NUMBER EDGE ":EDGE: INPUT "NUMBER NODE ":NODE
120 INPUT "NO.INDEPENDENT VOLTAGE ":VV: INPUT "NO.INDEPENDENT C
CURRENT ":II
130 ICC = 0:NS = VV + II:NH = EDGE - NS
160 DIM XM(8),WM(8),MM(2),FM(2),HM(8,4),UJAC(8,8),DKT(9),VOLT(9
),CURT(9),H1(8,2),FAUJ(8,8),ELXM(8),ERR(9)
165 DIM MVD(2),MID(2),CH(9),MPAR(3),SS(2)
170 DIM MT(8,8),MS(8,2),MD(2,2),MCKT(9),CPIV(9),MRLC(2),IF(8),F
ONT(9),KEEP(8),C(9),WU(8,8)
175 DIM VT$(9),IY$(9),DFM(2),DXM(8),DWM(8),DMM(2),WCKT(9)
180 DIM W1(8),W2(8),W3(8),W4(8),V1(9),C1(9),EV(9),E1(9),D1(8)
185 DIM VD(9),ID(9),DGGP(3),H2(8,2),FDP(2,3),DDP(2,3),PBND(2,2)
190 DIM W(8,8),B(8),X(8),A(8,8),KV(9),KI(9),FJ(8,6),HDP(8,3),ET
SI(8),SI(3)
195 DIM ESD(2),UT(8,8),YY(6),TP$(9),YA(10),HF(8,3),HS(8,2),YZV(
8)
200 DIM MA(5,9),RONE(5),RPIV(5),CIDX(9),WORK(8),EM(E)
205 DIM ZP(8),MWKT(9),DV(9),DC(9),M1(2),M2(2),X1(8),X2(8)
210 DIM YB(9),YC(9),YD(9),YE(9)
215 DIM WKT(9),VR(2,5),IR(2,5),DS(3),KT(2,5),KU(2,5)
220 DIM SD(3),MY(2),MZ(2),MB(2)
225 DIM CZ(9),Z1$(9)
230 HOME : PRINT "E N1 N2 TYPE OHM/H/F VOLT AMP"
235 VTAB 24: GET Z$: IF Z$ = CHR$(13) THEN 255
240 I = VAL(Z$):J = I * 2: IF I > ED THEN 235
245 VTAB J: PRINT I: VTAB J: HTAB 3: INPUT DV(I): VTAB J: HTAB
6: INPUT DC(I): VTAB J: HTAB 10: GET Z1$(I): VTAB J: HTAB 10
: PRINT Z1$(I)
250 VTAB J: HTAB 17: INPUT CZ(I): VTAB J: HTAB 23: INPUT VOLT(I
): VTAB J: HTAB 31: INPUT CURT(I): GOTO 235
255 RLC = 0:TYPE$ = "LT"
260 FOR I = 1 TO EDGE:MCKT(I) = 0
265 IF Z1$(I) = CHR$(82) THEN MCKT(I) = - 1
270 IF Z1$(I) = CHR$(67) THEN MCKT(I) = 1
275 IF Z1$(I) = CHR$(76) THEN 285
280 GOTO 290
285 RLC = RLC + 1:MRLC(RLC) = I:TYPE$ = "NT":MCKT(I) = 2
290 NEXT I
295 HOME : FOR I = 1 TO NS: INPUT "INDEPENDENT SOURCE VECTOR ":
SS(I): NEXT I
300 INPUT "NO.OF DESIRED TIME POINTS "INTD
305 HOME : INPUT "NUMBER PARAMETER ":NP
310 FOR I = 1 TO NP: INPUT "ELEMENT ":MPAR(I):J = I + 1: VTAB J
: HTAB 15: INPUT "MAX ":PEND(1,I): VTAB J: HTAB 23: INPUT "E
IN ":PBND(2,I): NEXT I
315 INPUT "NUMBER BIAS SOURCE ":NB: IF NB < = 0 THEN 390
320 FOR I = 1 TO NB: INPUT "BIAS SOURCE VECTOR ":MB(I): NEXT I
390 FOR I = 1 TO NE
400 FOR J = 1 TO NE
410 IF ME(I) = SS(J) THEN ME(I) = J
420 NEXT J
430 NEXT I

```

```

440 HOME : INPUT "NUMBER DESIRED VOLTAGE ":NVD: IF NVD < = 0 THEN
455
445 FOR I = 1 TO NVD: INPUT "DESIRED VOLTAGE VECTOR ":MVD(I): NEXT
I
447 INPUT "INITIAL TIME ":TO: INPUT "STEP SIZE ":DT
450 FOR I = 1 TO NVD: FOR J = 1 TO NTD: INPUT "TIME ":TD(J): CALL
- 998: HTAB 15: INPUT "VOLT ":VR(I,J):KT(I,J) = 1: NEXT J: NEXT
I
455 HOME : INPUT "NUMBER DESIRED CURRENT ":NID: IF NID < = 0 THEN
470
457 NZ = INT (TD(NTD) / DT + NTD / 2 + 1)
460 FOR I = 1 TO NID: INPUT "DESIRED CURRENT VECTOR ":MID(I): NEXT
I
465 FOR I = 1 TO NID: FOR J = 1 TO NTD: INPUT "TIME ":TD(J): CALL
- 998: HTAB 15: INPUT "AMP ":IR(I,J):KU(I,J) = 1: NEXT J: NEXT
I
467 NZ = INT (TD(NTD) / DT + NTD / 2 + 1)
470 FOR I = 1 TO NODE: FOR J = 1 TO EDGE:MA(I,J) = 0: NEXT J: NEXT
I
480 FOR I = 1 TO EDGE:MA(DV(I),I) = 1:MA(DC(I),I) = - 1: NEXT
I
485 HOME : PRINT "FM = K1+K2*SIN(314.16*T)          ": FOR I = 1 TO
NS: INPUT "K1,K2          ":YR(I),YQ(I): NEXT I
490 GOSUB 1000: REM TSDMAT
500 GOSUB 9000: REM MAP
595 HOME : INPUT "CREATE FILE ":Z$: PRINT CHR$(4);"OPEN ":Z$
600 PRINT CHR$(4);"DELETE ":Z$
605 PRINT CHR$(4);"OPEN ":Z$
610 PRINT CHR$(4);"WRITE ":Z$
620 PRINT EDGE: PRINT NODE: PRINT VV: PRINT II: PRINT RLC: PRINT
NB: PRINT NS: PRINT NH: PRINT NF: PRINT ICC: PRINT TO: PRINT
DT: PRINT TYPE$: PRINT TF: PRINT NZ: PRINT NVD: PRINT NID: PRINT
NTD
630 FOR I = 1 TO NTD: PRINT TD(I): NEXT I
640 FOR I = 1 TO NS: PRINT SS(I): NEXT I
650 FOR I = 1 TO NF: PRINT MPAR(I): NEXT I
660 FOR I = 1 TO NH: FOR J = 1 TO NH: PRINT MT(I,J): NEXT J: NEXT
I
662 FOR I = 1 TO NH: FOR J = 1 TO NS: PRINT MS(I,J): NEXT J: NEXT
I
664 FOR I = 1 TO NS: FOR J = 1 TO NS: PRINT MD(I,J): NEXT J: NEXT
I
666 FOR I = 1 TO EDGE: PRINT CP1V(I): NEXT I
670 FOR I = 1 TO EDGE: PRINT VT$(I): PRINT IY$(I): PRINT CH(I):
NEXT I
680 FOR I = 1 TO EDGE: PRINT MCKT(I): PRINT CZ(I): PRINT VOLT(I
): PRINT CURT(I): NEXT I
690 FOR I = 1 TO 2: FOR J = 1 TO NF: PRINT PBND(I,J): NEXT J: NEXT
I
700 IF RLC < = 0 THEN 730
710 FOR I = 1 TO RLC: PRINT MRLC(I): NEXT I
730 IF NB < = 0 THEN 750
740 FOR I = 1 TO NB: PRINT MB(I): NEXT I
750 IF NVD < = 0 THEN 780
760 FOR I = 1 TO NVD: PRINT MVD(I): NEXT I
770 FOR I = 1 TO NVD: FOR J = 1 TO NTD: PRINT VR(I,J): PRINT KT

```

```

(I,J): NEXT J: NEXT I
780 IF MID < = 0 THEN 800
790 FOR I = 1 TO MID: PRINT MID(I): NEXT I
795 FOR I = 1 TO MID: FOR J = 1 TO NTD: PRINT IR(I,J): PRINT KU
(I,J): NEXT J: NEXT I
800 PRINT FV: PRINT FW: PRINT FU
810 FOR I = 1 TO NS: PRINT YQ(I): PRINT YR(I): NEXT I
820 PRINT CHR$(4):"CLOSE ";Z$
830 PRINT CHR$(4):"BRUN AUT2.GBJ"
999 END
1000 REM TSDMAT
1005 PRINT EDGE,NODE,VV,II,NS,NH: FOR I = 1 TO NS: PRINT SS(I):
NEXT I
1020 FOR I = 1 TO NODE:ZZ = 0: FOR J = 1 TO EDGE:ZZ = ZZ + ABS
(MA(I,J)): NEXT J:RONE(I) = ZZ: NEXT I
1030 FOR J = 1 TO EDGE
1040 FOR I = 1 TO NODE
1050 IF MA(I,J) = 0 THEN 1070
1060 CPIV(J) = CPIV(J) + RONE(I)
1070 NEXT I
1080 NEXT J
1090 K = 1
1100 FOR I = 1 TO EDGE
1110 FOR J = 1 TO NS
1120 IF I = SS(J) THEN 1150
1130 NEXT J
1140 WORK(K) = I:K = K + 1
1150 NEXT I
1160 IF VV = 0 THEN 1190
1170 FOR I = 1 TO VV:CIDX(I) = SS(I): NEXT I
1180 IF II = 0 THEN 1200
1190 FOR I = 1 TO II:CIDX(EDGE - II + 1) = SS(VV + I): NEXT I
1200 L = NH
1210 FOR J = (VV + 1) TO (EDGE - II - 1):K = 1:TEST = CPIV(WORK
(1))
1220 FOR I = 2 TO L
1230 IF TEST > = CPIV(WORK(I)) THEN 1250
1240 TEST = CPIV(WORK(I)):K = I
1250 NEXT I
1260 CIDX(J) = WORK(K): IF K = L THEN 1280
1270 FOR I = K TO (L - 1):WORK(I) = WORK(I + 1): NEXT I
1280 L = L - 1
1290 NEXT J
1300 CIDX(J) = WORK(1)
1310 FOR I = 1 TO NODE:RPIV(I) = I: NEXT I
1320 N1 = 0:N2 = NODE:J = 0
1330 J = J + 1:J1 = CIDX(J):MN = 0:TEST = EDGE + 1
1340 FOR I1 = 1 TO NODE
1350 IF MA(I1,J1) = 0 THEN 1430
1360 IF J = 1 THEN 1400
1370 K1 = 1: IF N1 = 0 THEN 1400
1380 IF I1 = RPIV(K1) THEN 1430
1390 K1 = K1 + 1: IF K1 < = N1 THEN 1380
1400 NN = 0: FOR Z = 1 TO EDGE:NN = NN + ABS (MA(I1,Z)): NEXT Z
1410 IF TEST < = NN THEN 1430

```

```

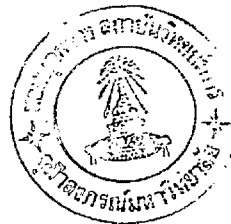
1420 TEST = NN:MN = I1
1430 NEXT I1
1440 IF MN = 0 THEN 1580
1450 N1 = N1 + 1:CP1V(N1) = J1
1460 I = N1
1470 IF RP1V(I) = MN THEN 1490
1480 I = I + 1: IF I < = NODE THEN 1470
1490 RP1V(I) = RP1V(N1):RP1V(N1) = MN:I3 = RP1V(N1):J3 = CP1V(N1)
      ):ZZ = MA(I3,J3)
1500 FOR Z = 1 TO EDGE:MA(I3,Z) = MA(I3,Z) / ZZ: NEXT Z
1510 FOR K2 = 1 TO NODE
1520 IF RP1V(K2) = MN THEN 1560
1530 K3 = RP1V(K2): IF MA(K3,J1) = 0 THEN 1560
1540 ZZ = MA(K3,J3)
1550 FOR Z = 1 TO EDGE:MA(K3,Z) = MA(K3,Z) - MA(I3,Z) * ZZ: NEXT
      Z
1560 NEXT K2
1570 GOTO 1590
1580 CP1V(N2) = J1:N2 = N2 + 1
1590 IF N1 < (NODE - 1) THEN 1330
1600 FOR I = (J + 1) TO EDGE:CP1V(I) = C1DX(I): NEXT I
1610 PRINT : PRINT : PRINT " MCPIV =": FOR I = 1 TO EDGE: PRINT
      CP1V(I);" ";: NEXT I: PRINT
1620 PRINT : PRINT " MA ="
1630 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(RP1V(I),CP1
      V(J));" ";: NEXT J: PRINT : NEXT I
1640 FOR I = 1 TO NH: FOR J = 1 TO NH:MT(I,J) = 0: NEXT J: NEXT
      I
1650 FOR I = 1 TO NH: FOR J = 1 TO NS:MS(I,J) = 0: NEXT J: NEXT
      I
1660 FOR I = 1 TO NS: FOR J = 1 TO NS:MD(I,J) = 0: NEXT J: NEXT
      I
1670 FOR I = (VV + 1) TO (NODE - 1):I1 = RP1V(I):I2 = I - VV
1680 FOR J = NODE TO (EDGE - I1):J1 = CP1V(J):J2 = J - VV:MT(I2
      ,J2) = - MA(I1,J1):MT(J2,I2) = MA(I1,J1): NEXT J
1690 IF I1 = 0 THEN 1710
1700 FOR J = (EDGE - I1 + 1) TO EDGE:J1 = CP1V(J):J2 = J - NH:M
      S(I2,J2) = - MA(I1,J1): NEXT J
1710 NEXT I
1720 IF VV = 0 THEN 1760
1730 FOR I = 1 TO VV:I1 = RP1V(I): FOR J = NODE TO (EDGE - I1):
      J1 = CP1V(J):I2 = J - VV:MS(I2,I) = MA(I1,J1): NEXT J: NEXT
      I
1740 IF I1 = 0 THEN 1760
1750 FOR I = 1 TO VV:I1 = RP1V(I):I2 = I: FOR J = (EDGE - I1 +
      1) TO EDGE:J1 = CP1V(J):J2 = J - NH:MD(I2,J2) = - MA(I1,J1)
      :MD(J2,I2) = MA(I1,J1): NEXT J: NEXT I
1760 PRINT : PRINT : PRINT " MATRIX T ="
1770 FOR I = 1 TO NH: FOR J = 1 TO NH: PRINT MT(I,J);" ";: NEXT
      J: PRINT : NEXT I
1780 PRINT : PRINT : PRINT " MATRIX S ="
1790 FOR I = 1 TO NH: FOR J = 1 TO NS: PRINT MS(I,J);" ";: NEXT
      J: PRINT : NEXT I
1800 PRINT : PRINT : PRINT " MATRIX D ="
1810 FOR I = 1 TO NS: FOR J = 1 TO NS: PRINT MD(I,J);" ";: NEXT
      J: PRINT : NEXT I

```

```
1820 RETURN
2000 REM SOURCE
2010 FOR I = 1 TO NS:FM(I) = YR(I) + Y0(I) * SIN (314.16 * TA)
: NEXT I
2030 RETURN
9000 REM MAP
9005 FOR I = 1 TO EDGE
9010 FOR J = 1 TO EDGE
9015 IF I < > CPIV(J) THEN 9055
9020 IF J < = VV THEN 9040
9025 IF J < NODE THEN 9045
9030 IF J < = (EDGE - 11) THEN 9050
9035 VT$(I) = "M":IY$(I) = "F":CH(I) = J - (NH + VV): GOTO 9060
9040 VT$(I) = "F":IY$(I) = "M":CH(I) = J: GOTO 9060
9045 VT$(I) = "X":IY$(I) = "W":CH(I) = J - VV: GOTO 9060
9050 VT$(I) = "W":IY$(I) = "X":CH(I) = J - VV: GOTO 9060
9055 NEXT J
9060 NEXT I
9065 RETURN
```

โปรแกรมย่อย TRMAIN

```
100 REM TRMAIN
160 DIM XM(8),WM(8),MM(2),FM(2),HM(8,4),UJAC(8,8),CKT(9),VOL
T(9),CURT(9),H1(8,2),FAUJ(8,8),ELXM(8),ERR(9)
165 DIM MVB(2),MID(2),CH(9),MPAR(3),SS(2)
170 DIM MT(8,8),MS(8,2),MD(2,2),MCKT(9),DRIV(9),MRLC(4),IF(8
),PONT(9),KEEP(8),C(9),WU(8,8)
175 DIM VT$(9),IY$(9),DFM(2),DXM(8),DWM(8),DMM(2),WCKT(9)
180 DIM W1(8),W2(8),W3(8),W4(8),V1(9),C1(9),EV(9),EI(9),D1(8
)
185 DIM VD(9),ID(9),DGDF(3),H2(8,2),FDF(2,3),DDF(2,3),FBND(2
,3)
190 DIM W(8,8),B(8),X(8),A(8,8),KV(9),KI(9),FJ(8,8),HDF(8,3)
,ETSI(8),SI(3)
195 DIM ESD(2),UT(8,8),YY(8),TP$(9),YA(10),HF(8,3),HS(8,2),Y
ZV(8)
200 DIM MA(5,9),RONE(5),RFIV(5),CIDX(9),WORK(8),EM(8)
205 DIM ZP(8),MWKT(9),DV(9),DC(9),M1(2),M2(2),X1(8),X2(8)
210 DIM YB(9),YC(9),YD(9),YE(9)
215 DIM WKT(9),VR(2,5),IR(2,5),DS(3),TD(5),KT(2,5),KU(2,5)
220 DIM SD(3),VF(80,9),CF(80,9),TM(80),FT(80),MY(5),MZ(5),MB
(2)
230 HOME : INPUT "FILE NAME ";Z$
235 PRINT CHR$(4);"OPEN ";Z$
240 PRINT CHR$(4);"READ ";Z$
245 INPUT EDGE: INPUT NODE: INPUT VV: INPUT II: INPUT RLC: INPUT
NB: INPUT NA: INPUT NC: INPUT MODE$: INPUT NS: INPUT NH: INPUT
NF: INPUT ICC: INPUT TO: INPUT DT: INPUT TYPE$: INPUT TF:
INPUT NZ
250 FOR I = 1 TO NS: INPUT SS(I): NEXT I
255 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: INPUT MA(I,J): NEXT
J: NEXT I
260 FOR I = 1 TO EDGE: INPUT MCKT(I): INPUT CKT(I): INPUT VD
LT(I): INPUT CURT(I): NEXT I
264 IF RLC < = 0 THEN 270
268 FOR I = 1 TO RLC: INPUT MRLC(I): NEXT I
270 IF NB < = 0 THEN 278
274 FOR I = 1 TO NB: INPUT MB(I): NEXT I
278 IF NA < = 0 THEN 284
280 FOR I = 1 TO NA: INPUT MY(I): NEXT I
284 IF NC < = 0 THEN 294
290 FOR I = 1 TO NC: INPUT MZ(I): NEXT I
294 INPUT FV: INPUT FW: INPUT FU
296 FOR I = 1 TO NS: INPUT YC(I): INPUT YR(I): NEXT I
298 PRINT CHR$(4);"CLOSE ";Z$
300 GOSUB 1000: REM TSDMAT
310 TM(1) = DT
340 FOR I = 1 TO NB
350 FOR J = 1 TO NS
360 IF MB(I) = SS(J) THEN MB(I) = J
370 NEXT J
380 NEXT I
430 GOSUB 2100: REM TRANAY
440 TK = TO: I = 0
450 IF NA = 0 AND NC = 0 THEN END
460 IF NA < = 0 THEN 520
```



```

470 PL# = "V"
480 FOR I = 1 TO NA
490 J = MY(I): FOR K = 0 TO KB: FT(K) = VF(K, J): NEXT K
500 GOSUB 8500: REM PLOT
510 NEXT I
520 IF NC < = 0 THEN END
530 PL# = "I"
540 FOR I = 1 TO NC
550 J = MZ(I): FOR K = 0 TO KB: FT(K) = CF(K, J): NEXT K
560 GOSUB 8500: REM PLOT
570 NEXT I
900 END
1000 REM TSDMAT
1020 FOR I = 1 TO NODE: ZZ = 0: FOR J = 1 TO EDGE: ZZ = ZZ + ABS
      (MA(I, J)): NEXT J: RONE(I) = ZZ: NEXT I
1030 FOR J = 1 TO EDGE
1040 FOR I = 1 TO NODE
1050 IF MA(I, J) = 0 THEN 1070
1060 CPIV(J) = CPIV(J) + RONE(I)
1070 NEXT I
1080 NEXT J
1090 K = 1
1100 FOR I = 1 TO EDGE
1110 FOR J = 1 TO NS
1120 IF I = SS(J) THEN 1150
1130 NEXT J
1140 WORK(K) = I: K = K + 1
1150 NEXT I
1160 IF VV = 0 THEN 1180
1170 FOR I = 1 TO VV: CIDX(I) = SS(I): NEXT I
1180 IF II = 0 THEN 1200
1190 FOR I = 1 TO II: CIDX(EDGE - II + I) = SS(VV + I): NEXT
      I
1200 L = NH
1210 FOR J = (VV + 1) TO (EDGE - II - 1): K = 1: TEST = CPIV(W
      ORK(1))
1220 FOR I = 2 TO L
1230 IF TEST > = CPIV(WORK(I)) THEN 1250
1240 TEST = CPIV(WORK(I)): K = I
1250 NEXT I
1260 CIDX(J) = WORK(K): IF K = L THEN 1280
1270 FOR I = K TO (L - 1): WORK(I) = WORK(I + 1): NEXT I
1280 L = L - 1
1290 NEXT J
1300 CIDX(J) = WORK(1)
1310 FOR I = 1 TO NODE: RPIV(I) = I: NEXT I
1320 N1 = 0: N2 = NODE: J = 0
1330 J = J + 1: J1 = CIDX(J): MN = 0: TEST = EDGE + 1
1340 FOR I1 = 1 TO NODE
1350 IF MA(I1, J1) = 0 THEN 1430
1360 IF J = 1 THEN 1400
1370 K1 = 1: IF N1 = 0 THEN 1400
1380 IF I1 = RPIV(K1) THEN 1430
1390 K1 = K1 + 1: IF K1 < = N1 THEN 1360
1400 NN = 0: FOR Z = 1 TO EDGE: NN = NN + ABS (MA(I1, Z)): NEXT
      Z

```



```

1410 IF TEST < = NN THEN 1430
1420 TEST = NN:MN = I1
1430 NEXT I1
1440 IF MN = 0 THEN 1580
1450 N1 = N1 + 1:CP1V(N1) = J1
1460 I = N1
1470 IF RP1V(I) = MN THEN 1490
1480 I = I + 1: IF I < = NODE THEN 1470
1490 RP1V(I) = RP1V(N1):RP1V(N1) = MN:I3 = RP1V(N1):J3 = CP1V
(N1):ZZ = MA(I3,J3)
1500 FOR Z = 1 TO EDGE:MA(I3,Z) = MA(I3,Z) / ZZ: NEXT Z
1510 FOR K2 = 1 TO NODE
1520 IF RP1V(K2) = MN THEN 1560
1530 K3 = RP1V(K2): IF MA(K3,J1) = 0 THEN 1560
1540 ZZ = MA(K3,J3)
1550 FOR Z = 1 TO EDGE:MA(K3,Z) = MA(K3,Z) - MA(I3,Z) * ZZ: NEXT
Z
1560 NEXT K2
1570 GOTO 1590
1580 CP1V(N2) = J1:N2 = N2 + 1
1590 IF N1 < (NODE - 1) THEN 1330
1600 FOR I = (J + 1) TO EDGE:CP1V(I) = C1DX(I): NEXT I
1610 PRINT : PRINT : PRINT " MCPIV =": FOR I = 1 TO EDGE: PRINT
CP1V(I):" "": NEXT I: PRINT
1620 PRINT : PRINT " MA ="
1630 FOR I = 1 TO NODE: FOR J = 1 TO EDGE: PRINT MA(RP1V(I),
CP1V(J)):" "": NEXT J: PRINT : NEXT I
1640 FOR I = 1 TO NH: FOR J = 1 TO NH:MT(I,J) = 0: NEXT J: NEXT
I
1650 FOR I = 1 TO NH: FOR J = 1 TO NS:MS(I,J) = 0: NEXT J: NEXT
I
1660 FOR I = 1 TO NS: FOR J = 1 TO NS:MD(I,J) = 0: NEXT J: NEXT
I
1670 FOR I = (VV + 1) TO (NODE - 1):I1 = RP1V(I):I2 = I - VV
1680 FOR J = NODE TO (EDGE - I1):J1 = CP1V(J):J2 = J - VV:MT
(I2,J2) = - MA(I1,J1):MT(J2,I2) = MA(I1,J1): NEXT J
1690 IF I1 = 0 THEN 1710
1700 FOR J = (EDGE - I1 + 1) TO EDGE:J1 = CP1V(J):J2 = J - N
H:MS(I2,J2) = - MA(I1,J1): NEXT J
1710 NEXT I
1720 IF VV = 0 THEN 1760
1730 FOR I = 1 TO VV:I1 = RP1V(I): FOR J = NODE TO (EDGE - I
1):J1 = CP1V(J):I2 = J - VV:MS(I2,I) = MA(I1,J1): NEXT J:
NEXT I
1740 IF I1 = 0 THEN 1760
1750 FOR I = 1 TO VV:I1 = RP1V(I):I2 = I: FOR J = (EDGE - I1
+ 1) TO EDGE:J1 = CP1V(J):J2 = J - NH:MD(I2,J2) = - MA(
I1,J1):MD(J2,I2) = MA(I1,J1): NEXT J: NEXT I
1760 PRINT : PRINT : PRINT " MATRIX T ="
1770 FOR I = 1 TO NH: FOR J = 1 TO NH: PRINT MT(I,J):" "": NEXT
J: PRINT : NEXT I
1780 PRINT : PRINT : PRINT " MATRIX S ="
1790 FOR I = 1 TO NH: FOR J = 1 TO NS: PRINT MS(I,J):" "": NEXT
J: PRINT : NEXT I
1800 PRINT : PRINT : PRINT " MATRIX D ="

```

```

1810 FOR I = 1 TO NS: FOR J = 1 TO NS: PRINT MD(I,J);"  " : NEXT
      J: PRINT : NEXT I
1820 RETURN
2000 REM SOURCE
2005 FRM = 1
2010 FOR I = 1 TO NS:FM(I) = YD(I) + YR(I) * SIN (314.16 *
      TA):DFM(I) = YR(I) * 314.16 * COS (314.16 * TA): NEXT I
2030 RETURN
2100 REM TRANAY
2104 AID = 0: IF TYPE$ = "NT" THEN AID = 1
2108 IF RLC = 0 THEN 2140
2110 FOR I = 1 TO EDGE:ERR(I) = 0:FONT(I) = 0: NEXT I
2120 FOR I = 1 TO NH:K = CPIV(VV + I):FONT(K) = I: NEXT I
2140 FOR I = 1 TO NH:K = I + VV:J = CPIV(K)
2150 XM(I) = CURT(J): IF K < NODE THEN XM(I) = VOLT(J)
2158 NEXT I
2160 FOR I = 1 TO EDGE:MWKT(I) = MCKT(I):WCKT(I) = CKT(I): NEXT
      I
2161 IF NB > 0 THEN 2170
2162 FOR I = (VV + 1) TO (EDGE - II):J = MCKT(I)
2163 IF J = 0 THEN 2166
2164 IF J = 1 THEN 2167
2165 GOTO 2168
2166 MWKT(I) = - 1:WCKT(I) = 1.0E20: GOTO 2168
2167 MWKT(I) = - 1:WCKT(I) = 1.0E - 20: GOTO 2168
2168 NEXT I
2169 GOTO 2177
2170 FOR I = (VV + 1) TO (EDGE - II):J = MCKT(I)
2171 IF J = 0 THEN 2174
2172 IF J = 1 THEN 2175
2173 GOTO 2176
2174 MWKT(I) = - 1:WCKT(I) = 1.0E - 20: GOTO 2176
2175 MWKT(I) = - 1:WCKT(I) = 1.0E20: GOTO 2176
2176 NEXT I
2177 DT = DT / (2 * 15):KA = 0:JD = 0:KB = 0:CONT = 0:FLAG =
      0:G = 0
2178 FOR I = 1 TO NH: FOR J = 1 TO NS:H1(I,J) = 0:H2(I,J) =
      0: NEXT J: NEXT I
2179 DONE = 0:IH = 1:D4 = DT:D5 = D4:TA = TO: INPUT "DFM PROV
      IDED OR NOT (Y/N) ";FRM
2180 GOSUB 3000: REM HMAT
2182 IF AID < > 1 THEN 2186
2184 ENRM = 0:E1 = 0:E2 = 0:SECH = 0:NLSW = 0: FOR I = 1 TO N
      H:KEEP(I) = XM(I): NEXT I
2186 ITER = 0:RTEP = 1
2188 IF KB = 0 THEN GOSUB 2000: REM SOURCE
2190 IF KB < > 0 THEN GOSUB 2000: REM SOURCE
2192 IF TA = TO AND NB > 0 THEN 2196
2194 GOTO 2207
2196 FOR I = 1 TO NS:ZZ = 0
2198 FOR J = 1 TO NB
2200 IF I = MB(J) THEN ZZ = 1
2202 NEXT J
2204 IF ZZ = 0 THEN FM(I) = 0
2206 NEXT I
2207 IF FRM < > 0 THEN 2218

```

```

2208 IF KB < > 0 THEN 2210
2209 FOR I = 1 TO NS:DFM(I) = 0: NEXT I: GOTO 2218
2210 REM DERIV
2212 IF KA < 2 THEN 2216
2214 ZZ = D4 + D5: FOR I = 1 TO NS:DFM(I) = ((ZZ + D4) * D5 *
FM(I) - ZZ * ZZ * M1(I) + D4 * D4 * M2(I)) / (ZZ * D4 * D
5): NEXT I
2215 GOTO 2218
2216 FOR I = 1 TO NS:DFM(I) = (FM(I) - M1(I)) / D4: NEXT I
2218 FOR I = 1 TO NH
2220 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 + MT(I,J) * XM(J): NEXT
J
2230 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MS(I,J) * FM(J): NEXT
J
2240 WM(I) = Z1 + Z2
2250 NEXT I
2260 FOR I = 1 TO NS
2270 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 - MS(J,I) * XM(J): NEXT
J
2280 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MD(I,J) * FM(J): NEXT
J
2290 MM(I) = Z1 + Z2
2300 NEXT I
2301 IF KB < > 0 THEN 2304
2302 FOR I = 1 TO NH:DXM(I) = 0:DWM(I) = 0: NEXT I
2303 FOR I = 1 TO NS:DMM(I) = 0: NEXT I: GOTO 2320
2304 IF KA < 2 THEN 2308
2305 ZZ = D4 + D5: FOR I = 1 TO NH:DXM(I) = ((ZZ + D4) * D5 *
XM(I) - ZZ * ZZ * X1(I) + D4 * D4 * X2(I)) / (ZZ * D4 * D
5): NEXT I
2306 GOTO 2310
2308 FOR I = 1 TO NH:DXM(I) = (XM(I) - X1(I)) / D4: NEXT I
2310 FOR I = 1 TO NH
2311 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 + MT(I,J) * DXM(J): NEXT
J
2312 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MS(I,J) * DFM(J): NEXT
J
2313 DWM(I) = Z1 + Z2
2314 NEXT I
2315 FOR I = 1 TO NS
2316 Z1 = 0: FOR J = 1 TO NH:Z1 = Z1 - MT(J,I) * DXM(J): NEXT
J
2317 Z2 = 0: FOR J = 1 TO NS:Z2 = Z2 + MD(I,J) * DFM(J): NEXT
J
2318 DMM(I) = Z1 + Z2
2319 NEXT I
2320 FOR I = 1 TO NH:EM(I) = HM(I,1) * XM(I) + HM(I,2) * WM(
I) + HM(I,3) * DXM(I) + HM(I,4) * DWM(I): NEXT I
2325 IF RLC < = 0 THEN 2360
2330 GOSUB 3500: REM CONVERT
2340 GOSUB 4000: REM ERROR
2350 FOR I = 1 TO RLC:J = MRLC(I):K = FONT(J):EM(K) = ERR(J)
: NEXT I
2360 Z1 = 0:Z2 = 0:Z3 = 0: FOR I = 1 TO NH:Z1 = Z1 + EM(I) *
EM(I):Z2 = Z2 + XM(I) * XM(I):Z3 = Z3 + WM(I) * WM(I): NEXT
I:ENRM = SQR(Z1):XNRM = SQR(Z2):WNRM = SQR(Z3)

```

```

2370 IF AID < > 1 THEN 2400
2380 E2 = E1:E1 = ENRM:ENRM = 0:XNRM = 0:WNRM = 0
2390 FOR I = 1 TO NH:ENRM = ENRM + ABS (EM(I)):XNRM = XNRM +
ABS (XM(I)):WNRM = WNRM + ABS (WM(I)): NEXT I
2400 IF ENRM < = (XNRM + WNRM) * 1.0E - 6 THEN 2852
2405 IF TYPE# = "LT" AND ITER > 0 THEN 2600
2410 IF AID < > 1 THEN 2540
2415 IF SECH = 0 THEN E0 = ENRM
2420 ON NLSW GOTO 2470,2470,2600
2425 IF SECH < > 0 THEN 2440
2430 IF FLAG = 0 THEN 2550
2435 IF FLAG < > 0 THEN 2570
2440 IF E0 < ENRM THEN 2455
2445 NLSW = 2
2450 RTEP = RTEP * 2: GOTO 2620
2455 NLSW = 1
2460 RTEP = RTEP / 2: GOTO 2620
2470 Z6 = E1 * 1.0E - 5:Z7 = ABS (ENRM - E1)
2480 IF Z7 < Z6 THEN 2510
2490 IF NLSW = 1 THEN 2720
2500 IF NLSW < > 1 THEN 2740
2510 IF NLSW = 1 THEN 2730
2520 IF NLSW < > 1 THEN 2750
2540 IF FLAG = 1 THEN 2600
2550 GOSUB 4500: REM UJADM
2560 FLAG = 1: IF RLC = 0 THEN 2590
2570 GOSUB 5000: REM NUJADM
2580 IF JD < > 1 THEN 2590
2585 JD = 0: GOTO 2906
2590 GOSUB 5500: REM FACTOR
2600 GOSUB 6000: REM FBSUB
2610 IF AID < > 1 THEN 2650
2620 SECH = SECH + 1
2640 FOR I = 1 TO NH:XM(I) = KEEP(I): NEXT I
2650 FOR I = 1 TO NH:XM(I) = XM(I) - RTEP * ELXM(I): NEXT I
2660 IF AID < > 1 THEN 2690
2670 IF NLSW < > 4 THEN 2218
2680 NLSW = 0:SECH = 0:ENRM = 0:E1 = 0:E2 = 0:RTEP = 1: FOR I
= 1 TO NH:KEEP(I) = XM(I): NEXT I
2690 ITER = ITER + 1
2700 IF ITER < = 100 THEN 2218
2710 PRINT : PRINT : PRINT " ITER>100:STOP": END
2720 IF E0 < ENRM THEN 2460
2730 RANL = 0:RENL = RTEP:RCNL = RTEP * 2:EA = E0:EB = ENRM:E
C = E1: GOTO 2770
2740 IF ENRM < E1 THEN 2450
2750 RCNL = RTEP:RENL = RTEP / 2:EA = E2:EB = E1:EC = ENRM
2760 RANL = RENL / 2: IF SECH = 2 THEN RANL = 0
2770 EA = EA * (RCNL - RENL):ENRM = EB * (RANL - RCNL):EC = E
C * (RENL - RANL)
2780 RTEP = (EA * (RCNL + RENL) + ENRM * (RANL + RCNL) + EC *
(RENL + RANL)) / (2 * (EA + ENRM + EC))
2790 NLSW = 3: GOTO 2640
2800 IF ENRM < = EB THEN 2820
2810 RTEP = RENL:NLSW = 4: GOTO 2640
2820 ITER = ITER + 1: IF ITER < = 100 THEN 2840

```

```

2830 PRINT : PRINT " ITER>100:STOP": END
2840 NLSW = 0:SECH = 0:EO = ENRM:E1 = 0:E2 = 0:RTEP = 1: FOR
  I = 1 TO NH:KEEP(I) = XM(I): NEXT I
2850 GOTO 2570
2852 IF KA < = 15 AND KA < > 0 THEN 2856
2854 GOTO 2860
2856 KA = KA + 1:DT = DT * 2:CONT = 0:FLAG = 0
2857 IF KA < 16 THEN 2922
2858 TA = DT
2860 IF RLC < > 0 THEN 2872
2870 GOSUB 3500: REM CONVERT
2872 FOR I = 1 TO EDGE:VF(KB, I) = VOLT(I):CF(KB, I) = CURT(I)
  : NEXT I
2873 PRINT "VOLT(3) = ";VOLT(3)
2874 IF MODE# < > "AU" THEN KB = KB + 1
2876 IF MODE# < > "AU" THEN 2922
2878 IF KB > 0 THEN TM(KB) = TA
2880 KB = KB + 1
2882 IF ABS(TA - TD(IH)) > DT * 1.0E - 5 THEN 2922
2884 TA = TD(IH)
2886 IF NVD < = 0 THEN 2892
2888 FOR I = 1 TO EDGE:VD(I) = 0:KV(I) = 0: NEXT I
2890 FOR I = 1 TO NVD:J = MVD(I):VD(J) = VR(I, IH):KV(J) = KT
  (I, IH): NEXT I
2892 IF NID < = 0 THEN 2896
2894 FOR I = 1 TO EDGE:KI(I) = 0:ID(I) = 0: NEXT I
2895 FOR I = 1 TO NID:J = MID(I):ID(J) = IR(I, IH):KI(J) = KU
  (I, IH): NEXT I
2896 GOSUB 6200: REM FUNC6
2898 GP = GP + G
2899 IF IH < NTD THEN 2902
2900 IF GP < = 1.0E - 6 THEN DONE = 1
2901 GOTO 2948
2902 IF IL < > 0 THEN 2910
2903 IF AID = 1 THEN JD = 1
2904 IF AID = 1 THEN 2570
2906 GOSUB 6500: REM GRAD6
2908 FOR I = 1 TO NP:DS(I) = DS(I) + DGDP(I): NEXT I
2910 IF IH < NTD THEN 2920
2912 IF IL < > 0 THEN 2916
2914 GOSUB 8000: REM NORMAL
2916 FOR I = 1 TO EDGE:VOLT(I) = VF(0, I):CURT(I) = CF(0, I): NEXT
  I
2918 DONE = 0: GOTO 2948
2920 IH = IH + 1
2922 TA = TA + DT:D5 = D4:D4 = DT
2924 FOR I = 1 TO NS:N2(I) = M1(I):M1(I) = FM(I): NEXT I
2926 FOR I = 1 TO NH:X2(I) = X1(I):X1(I) = XM(I): NEXT I
2928 IF MODE# < > "AU" THEN 2938
2930 IF ABS(TA - TD(IH)) * 100 < DT THEN TA = TD(IH)
2932 IF ABS(TA - TD(IH)) * 100 < DT THEN 2940
2934 IF TA < TD(IH) THEN 2940
2936 D4 = TD(IH) - TA + DT:TA = TD(IH):CONT = 0:FLAG = 0: GOTO
  2942
2938 IF TA > TD THEN 2948
2940 CONT = CONT + 1: IF CONT < = 2 THEN FLAG = 0

```

```

2942 IF KA < > 0 THEN 2182
2944 FOR I = 1 TO EDGE:WCKT(I) = CKT(I):MWKT(I) = MCKT(I): NEXT
I
2946 KA = KA + 1: GOTO 2180
2948 KB = KB - 1
2990 RETURN
3000 REM HMAT
3010 FOR I = 1 TO NH: FOR J = 1 TO 4:HM(I,J) = 0: NEXT J: NEXT
I
3020 FOR I = 1 TO (NODE - 1 - VV)
3030 K = I + VV:I1 = CP1V(K): IF MWKT(I1) > 1 THEN 3100
3040 IF MWKT(I1) = - 1 THEN 3070
3050 IF MWKT(I1) = 0 THEN 3080
3060 IF MWKT(I1) = 1 THEN 3090
3065 PRINT " CHECK CKT:STOP": END
3070 HM(I,1) = 1:HM(I,2) = - WCKT(I1): GOTO 3100
3080 HM(I,1) = 1:HM(I,4) = - WCKT(I1): GOTO 3100
3090 HM(I,2) = 1:HM(I,3) = - WCKT(I1)
3100 NEXT I
3110 FOR I = (NODE - VV) TO NH
3120 K = I + VV:I1 = CP1V(K): IF MWKT(I1) > 1 THEN 3200
3130 IF MWKT(I1) = - 1 THEN 3170
3140 IF MWKT(I1) = 0 THEN 3180
3150 IF MWKT(I1) = 1 THEN 3190
3160 PRINT " CHECK CKT:STOP": END
3170 HM(I,1) = - WCKT(I1):HM(I,2) = 1: GOTO 3200
3180 HM(I,2) = 1:HM(I,3) = - WCKT(I1): GOTO 3200
3190 HM(I,1) = 1:HM(I,4) = - WCKT(I1): GOTO 3200
3200 NEXT I
3210 RETURN
3500 REM CONVERT
3510 KSW = 0: IF TYPE$ = "LT" OR TYPE$ = "NT" THEN KSW = 1
3520 K = VV
3530 FOR I = 1 TO EDGE:J = CP1V(I)
3540 IF I < = VV THEN 3600
3550 IF I < NODE THEN 3630
3560 IF I < . = (EDGE - I1) THEN 3660
3570 K = K + 1:VOLT(J) = MM(K):CURT(J) = FM(K)
3580 IF KSW < > 1 THEN 3700
3590 DV(J) = DMM(K):DC(J) = DFM(K): GOTO 3700
3600 VOLT(J) = FM(I):CURT(J) = MM(I)
3610 IF KSW < > 1 THEN 3700
3620 DV(J) = DFM(I):DC(J) = DMM(I): GOTO 3700
3630 L = I - VV:VOLT(J) = XM(L):CURT(J) = WM(L)
3640 IF KSW < > 1 THEN 3700
3650 DV(J) = DXM(L):DC(J) = DWM(L): GOTO 3700
3660 L = I - VV:VOLT(J) = WM(L):CURT(J) = XM(L)
3670 IF KSW < > 1 THEN 3700
3680 DV(J) = DWM(L):DC(J) = DXM(L): GOTO 3700
3700 NEXT I
3710 RETURN
4000 REM ERROR
4010 FOR I = 1 TO RLC:C(MRLC(I)) = 1.0E - 9 * ( EXP (VOLT(MR
LC(I)) * FU) - 1): NEXT I
4030 ERR(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(
2))

```

```

4040 ERR(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(
1))
4050 RETURN
4500 REM UJADM
4505 DK = D4 * (D4 + D5) / (2 * D4 + D5): IF KA < 2 THEN DK =
D4
4510 FOR I = 1 TO NH: FOR J = 1 TO NH: UJAC(I,J) = 0: WU(I,J) =
0: NEXT J: NEXT I
4520 FOR I = 1 TO NH: WU(I,I) = HM(I,4) / DK + HM(I,2): NEXT
I
4530 FOR I = 1 TO NH: FOR J = 1 TO NH: Z1 = 0: FOR K = 1 TO N
H: Z1 = Z1 + WU(I,K) * MT(K,J): NEXT K: UJAC(I,J) = Z1: NEXT
J: NEXT I
4540 FOR I = 1 TO NH: UJAC(I,I) = UJAC(I,I) + HM(I,1) + HM(I,
3) / DK: NEXT I
4550 RETURN
5000 REM NUJADM
5010 KSW = 0: IF TYPE$ = "LT" OR TYPE$ = "NT" THEN KSW = 1
5020 FOR I = 1 TO NH: FOR J = 1 TO NS: H1(I,J) = 0: H2(I,J) =
0: NEXT J: NEXT I
5030 FOR I = 1 TO EDGE: EV(I) = 0: EI(I) = 0: NEXT I
5040 FOR I = 1 TO RLC
5050 I1 = MRLC(I)
5060 FOR J = 1 TO NH: J2 = J + VV: J1 = CPIV(J2)
5070 FOR K = 1 TO EDGE: V1(K) = VOLT(K): NEXT K
5080 V1(J1) = 1.01 * VOLT(J1): IF VOLT(J1) = 0 THEN V1(J1) =
0.001
5090 C(MRLC(1)) = 1.0E - 9 * ( EXP (V1(MRLC(1)) * FU) - 1)
5100 C(MRLC(2)) = 1.0E - 9 * ( EXP (V1(MRLC(2)) * FU) - 1)
5110 EV(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2
))
5120 EV(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1
))
5130 D8 = V1(J1) - VOLT(J1)
5140 FOR K = 1 TO EDGE: C1(K) = CURT(K): NEXT K
5150 C1(J1) = 1.01 * CURT(J1): IF CURT(J1) = 0 THEN C1(J1) =
0.001
5160 C(MRLC(1)) = 1.0E - 9 * ( EXP (VOLT(MRLC(1)) * FU) - 1)
5170 C(MRLC(2)) = 1.0E - 9 * ( EXP (VOLT(MRLC(2)) * FU) - 1)
5180 EI(MRLC(1)) = C1(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
5190 EI(MRLC(2)) = C1(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1))

5200 D9 = C1(J1) - CURT(J1)
5201 IF KEW < > 1 THEN 5210
5202 FOR K = 1 TO EDGE: YB(K) = DV(K): NEXT K: YB(J1) = 1.01 *
DV(J1): IF DV(J1) = 0 THEN YB(J1) = 0.001
5203 C(MRLC(1)) = 1.0E - 9 * ( EXP (VOLT(MRLC(1)) * FU) - 1):
C(MRLC(2)) = 1.0E - 9 * ( EXP (VOLT(MRLC(2)) * FU) - 1): Y
D(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
: YD(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1
))
5204 D6 = YB(J1) - DV(J1)
5205 FOR K = 1 TO EDGE: YC(K) = DC(K): NEXT K: YC(J1) = 1.01 *
DC(J1): IF DC(J1) = 0 THEN YC(J1) = 0.001
5206 C(MRLC(1)) = 1.0E - 9 * ( EXP (VOLT(MRLC(1)) * FU) - 1):

```

```

C(MRLC(2)) = 1.0E - 9 * ( EXP (VOLT(MRLC(2)) * FU) - 1):Y
E(MRLC(1)) = CURT(MRLC(1)) - C(MRLC(1)) + FV * C(MRLC(2))
:YE(MRLC(2)) = CURT(MRLC(2)) - C(MRLC(2)) + FW * C(MRLC(1))
))
5207 D7 = YC(J1) - DC(J1)
5210 IF J2 < NODE THEN 5230
5220 W1(J) = (EI(I1) - ERR(I1)) / D9:W2(J) = (EV(I1) - ERR(I1)) / D8
5224 IF KSW < > 1 THEN 5240
5226 W3(J) = (YE(I1) - ERR(I1)) / D7:W4(J) = (YD(I1) - ERR(I1)) / D6
5228 GOTO 5240
5230 W1(J) = (EV(I1) - ERR(I1)) / D8:W2(J) = (EI(I1) - ERR(I1)) / D9
5232 IF KSW < > 1 THEN 5240
5234 W3(J) = (YD(I1) - ERR(I1)) / D6:W4(J) = (YE(I1) - ERR(I1)) / D7
5236 GOTO 5240
5240 NEXT J
5250 IF MODE# = "AU" AND ICC = 1 THEN 5270
5260 GOTO 5280
5270 J1 = PONT(I1)
5272 FOR J = 1 TO NS
5274 Z1 = 0: FOR K = 1 TO NH:Z1 = Z1 + W2(K) * MS(K,J): NEXT K:H1(J1,J) = Z1
5275 IF KSW < > 1 THEN 5279
5276 Z1 = 0: FOR K = 1 TO NH:Z1 = Z1 + W4(K) * MS(K,J): NEXT K:H2(J1,J) = Z1
5279 NEXT J
5280 FOR K = 1 TO NH:K1 = PONT(I1):Z1 = 0: FOR Z = 1 TO NH:Z1 = Z1 + W2(Z) * MT(Z,K): NEXT Z:UJAC(K1,K) = W1(K) + Z1
5282 IF KSW = 0 THEN 5288
5285 Z1 = 0: FOR Z = 1 TO NH:Z1 = Z1 + (W2(Z) + W4(Z) / DK) * MT(Z,K): NEXT Z:UJAC(K1,K) = W1(K) + W3(K) / DK + Z1
5288 NEXT K
5300 NEXT I
5310 RETURN
5500 REM FACTOR
5510 FOR I = 1 TO NH: FOR J = 1 TO NH:A(I,J) = UJAC(I,J):W(I,J) = FAUJ(I,J): NEXT J:ZF(I) = IF(I): NEXT I
5515 IF CODE < > 1 THEN 5540
5520 FOR I = 1 TO NH: FOR J = 1 TO NH:A(I,J) = UT(I,J):W(I,J) = FJ(I,J): NEXT J:ZF(I) = YY(I): NEXT I
5540 FOR I = 1 TO NH: FOR J = 1 TO NH:W(I,J) = A(I,J): NEXT J: NEXT I
5550 FOR I = 1 TO NH:ZF(I) = I:MX = 0
5560 FOR J = 1 TO NH
5570 IF MX < ABS(W(I,J)) THEN MX = ABS(W(I,J))
5580 NEXT J
5590 IF MX = 0 THEN END
5600 D1(I) = MX
5610 NEXT I
5620 N1 = NH - 1
5630 IF N1 = 0 THEN RETURN
5640 FOR K = 1 TO N1:J = K:KP1 = K + 1:I1 = ZF(K):CX = ABS(W(I1,K)) / D1(I1)

```



```

5650 FOR I = KP1 TO NH: I1 = ZP(I): MX = ABS (W(I1,K)) / D1(I
1)
5660 IF MX < = CX THEN 5680
5670 CX = MX: J = I
5680 NEXT I
5690 IF CX = 0 THEN END
5700 ZZ = ZP(J): ZP(J) = ZP(K): ZP(K) = ZZ
5710 FOR I = KP1 TO NH: I1 = ZP(I): W(I1,K) = W(I1,K) / W(ZZ,K
): R1 = - W(I1,K)
5720 FOR J = KP1 TO NH
5730 W(I1,J) = R1 * W(ZZ,J) + W(I1,J)
5740 NEXT J
5750 NEXT I
5760 NEXT K
5770 IF W(I1,NH) < > 0 THEN 5785
5780 PRINT " CAN NOT BE FACTORIZED"
5785 IF CODE = 1 THEN 5795
5790 FOR I = 1 TO NH: FOR J = 1 TO NH: UJAC(I,J) = A(I,J): FAU
J(I,J) = W(I,J): NEXT J: IP(I) = ZP(I): NEXT I
5792 GOTO 5800
5795 FOR I = 1 TO NH: FOR J = 1 TO NH: UT(I,J) = A(I,J): FJ(I,
J) = W(I,J): NEXT J: YY(I) = ZP(I): NEXT I
5800 RETURN
6000 REM FBSUB
6010 FOR I = 1 TO NH: FOR J = 1 TO NH: W(I,J) = FAUJ(I,J): NEXT
J: B(I) = EM(I): X(I) = ELXM(I): ZP(I) = IP(I): NEXT I
6012 IF CODE < > 1 THEN 6020
6015 FOR I = 1 TO NH: FOR J = 1 TO NH: W(I,J) = FJ(I,J): NEXT
J: B(I) = ETSI(I): X(I) = YZV(I): ZP(I) = YY(I): NEXT I
6020 IF NH > 1 THEN 6040
6030 X(1) = B(1) / W(1,1): RETURN
6040 I1 = ZP(1): X(1) = B(I1)
6050 FOR K = 2 TO NH: I1 = ZP(K): KP1 = K - 1
6060 ZZ = 0: FOR J = 1 TO KP1: ZZ = ZZ + W(I1,J) * X(J): NEXT
J
6070 X(K) = B(I1) - ZZ
6080 NEXT K
6090 X(NH) = X(NH) / W(I1,NH): N1 = NH - 1
6100 FOR K = N1 TO 1 STEP - 1: I1 = ZP(K): KP1 = K + 1
6110 ZZ = 0: FOR J = KP1 TO NH: ZZ = ZZ + W(I1,J) * X(J): NEXT
J
6120 X(K) = (X(K) - ZZ) / W(I1,K)
6130 NEXT K
6135 IF CODE = 1 THEN 6145
6140 FOR I = 1 TO NH: FOR J = 1 TO NH: FAUJ(I,J) = W(I,J): NEXT
J: EM(I) = B(I): ELXM(I) = X(I): IP(I) = ZP(I): NEXT I
6142 GOTO 6150
6145 FOR I = 1 TO NH: FOR J = 1 TO NH: FJ(I,J) = W(I,J): NEXT
J: ETSI(I) = B(I): YZV(I) = X(I): YY(I) = ZP(I): NEXT I
6150 RETURN
6200 REM FUNC6
6210 IF NID = 0 THEN 6250
6220 IF NVD = 0 THEN 6270
6230 G = 0: FOR I = 1 TO EDSE: G = G + ((VOLT(I) - VD(I)) ^ 2)
* KV(I) + ((CURT(I) - ID(I)) ^ 2) * KI(I): NEXT I
6240 GOTO 6300

```

```

6250 G = 0: FOR I = 1 TO EDGE:G = G + ((VOLT(I) - VD(I)) ^ 2)
      * KV(I): NEXT I
6255 PRINT "*****6275*****": FOR I = 1 TO NH: PRINT VO
      LT(I),VD(I),KV(I): NEXT I
6260 GOTO 6300
6270 G = 0: FOR I = 1 TO EDGE:G = G + ((CURT(I) - ID(I)) ^ 2)
      * KI(I): NEXT I
6300 RETURN
6500 REM GRADG
6505 FOR I = 1 TO EDGE:CKT(I) = WCKT(I):MCKT(I) = MWKT(I): NEXT
      I
6510 FOR I = 1 TO NH: FOR J = 1 TO NH:UT(I,J) = UJAC(J,I): NEXT
      J: NEXT I
6520 FOR I = 1 TO NP:DGDP(I) = 0: NEXT I
6530 GOSUB 7000: REM DHDF
6540 CODE = 1: GOSUB 5500: REM FACTOR
6542 CODE = 0
6545 FOR Z = 1 TO 10:YA(Z) = 0: NEXT Z
6550 IF NVD = 0 THEN 6580
6560 FLAG = 1:ND = NVD: FOR I = 1 TO NVD:YA(I) = MVD(I): NEXT
      I
6570 FOR I = 1 TO EDGE:TP$(I) = VT$(I): NEXT I: GOTO 6600
6580 FLAG = 0:ND = NID: FOR I = 1 TO NID:YA(I) = MID(I): NEXT
      I
6590 FOR I = 1 TO EDGE:TP$(I) = IY$(I): NEXT I
6600 FOR IB = 1 TO ND
6610 FOR J = 1 TO NH:ETSI(J) = 0: NEXT J
6620 FC = 0:JZ = YA(IB):IK = CH(JZ)
6630 IF TP$(JZ) = "X" THEN 6690
6640 IF TP$(JZ) = "W" THEN 6700
6650 FOR Z = 1 TO NH:ETSI(Z) = - MS(Z,IK): NEXT Z
6660 IF ICC = 0 THEN 6730
6670 FC = 1: FOR Z = 1 TO NS:ESD(Z) = MD(IK,Z): NEXT Z
6680 GOTO 6730
6690 ETSI(IK) = 1: GOTO 6730
6700 FOR Z = 1 TO NH:ETSI(Z) = MT(IK,Z): NEXT Z
6710 IF ICC = 0 THEN 6730
6720 FC = 1: FOR Z = 1 TO NS:ESD(Z) = MS(IK,Z): NEXT Z
6725 GOTO 6730
6730 GOSUB 7500: REM SENSTY
6740 VIM = (CURT(JZ) - ID(JZ)) * KI(JZ): IF FLAG = 1 THEN VIM
      = (VOLT(JZ) - VD(JZ)) * KV(JZ)
6750 FOR Z = 1 TO NP:DGDP(Z) = DGDP(Z) + VIM * SI(Z): NEXT Z

6760 NEXT IB
6770 IF FLAG < > 0 AND NID < > 0 THEN 6580
6800 RETURN
7000 REM DHDF
7010 KW = 0: IF TYPE$ = "NT" OR TYPE$ = "LT" THEN KW = 1
7020 FOR I = 1 TO NH: FOR J = 1 TO NP:HDF(I,J) = 0: NEXT J: NEXT
      I
7030 FOR J = 1 TO NP
7040 FOR I = 1 TO NH:K = CPIV(I + VV)
7050 IF K < > MPAR(J) THEN 7150
7060 IF I < (NODE - VV) THEN 7110
7065 HDF(I,J) = - DWM(I)

```

```

7070 IF MCKT(K) = - 1 THEN HDP(I,J) = - XM(I)
7080 IF MCKT(K) = 0 THEN HDP(I,J) = - DXM(I)
7100 GOTO 7160
7110 HDP(I,J) = - DXM(I): IF MCKT(K) = - 1 THEN HDP(I,J) =
- WM(I)
7120 IF MCKT(K) = 0 THEN HDP(I,J) = - DWM(I)
7140 GOTO 7160
7150 NEXT I
7160 NEXT J
7170 FOR I = 1 TO NH: FOR J = 1 TO NP: HDP(I,J) = - HDP(I,J)
: NEXT J: NEXT I
7180 IF ICC = 0 THEN 7400
7190 FOR I = 1 TO NH: FOR J = 1 TO NS: HS(I,J) = HM(I,2) * MS
(I,J): NEXT J: NEXT I
7200 IF RLC < = 0 THEN 7220
7210 FOR I = 1 TO NH: FOR J = 1 TO NS: HS(I,J) = HS(I,J) + H1
(I,J): NEXT J: NEXT I
7220 FOR I = 1 TO NH
7230 FOR J = 1 TO NP: ZZ = 0: FOR Z = 1 TO NS: ZZ = ZZ + HS(I,
Z) * FDP(Z,J): NEXT Z: HF(I,J) = ZZ: NEXT J
7240 NEXT I
7250 FOR I = 1 TO NH: FOR J = 1 TO NP: HDP(I,J) = HDP(I,J) -
HF(I,J): NEXT J: NEXT I
7260 IF KW = 0 THEN 7400
7270 FOR I = 1 TO NH: FOR J = 1 TO NS: HS(I,J) = HM(I,4) * MS
(I,J): NEXT J: NEXT I
7280 IF RLC < = 0 THEN 7300
7290 FOR I = 1 TO NH: FOR J = 1 TO NS: HS(I,J) = HS(I,J) + H2
(I,J): NEXT J: NEXT I
7300 FOR I = 1 TO NH
7310 FOR J = 1 TO NP
7320 ZZ = 0: FOR Z = 1 TO NS: ZZ = ZZ + HS(I,Z) * DDP(Z,J): NEXT
Z: HF(I,J) = ZZ
7330 NEXT J
7340 NEXT I
7350 FOR I = 1 TO NH: FOR J = 1 TO NP: HDP(I,J) = HDP(I,J) -
HF(I,J): NEXT J: NEXT I
7400 RETURN
7500 REM SENSTY
7510 CODE = 1: GOSUB 6000: REM FBSUB
7515 CODE = 0
7520 FOR Z = 1 TO NP: ZZ = 0: FOR ZK = 1 TO NH: ZZ = ZZ + YZV(
ZK) * HDP(ZK,Z): NEXT ZK: SI(Z) = ZZ: NEXT Z
7530 IF FC = 0 THEN 7550
7540 FOR Z = 1 TO NP: ZZ = 0: FOR ZK = 1 TO NS: ZZ = ZZ + ESD(
ZK) * FDP(ZK,Z): NEXT ZK: SI(Z) = SI(Z) + ZZ: NEXT Z
7550 RETURN
8000 REM NORMAL
8010 K = 1: CST = 0.1: RM = ABS (DGDF(1))
8012 IF G < = 1.0E - 3 THEN CST = 0.05
8014 IF G < = 1.0E - 4 THEN CST = 0.025
8016 IF G < = 1.0E - 5 THEN CST = 0.003
8020 IF NP < 2 THEN 8070
8030 FOR I = 2 TO NP: RN = ABS (DGDF(I))
8035 PRINT DGDF(1), DGDF(2)
8040 IF RM > = RN THEN 8060

```

```

8050 RM = RN:K = I
8060 NEXT I
8070 PCKT = WCKT(MPAR(K))
8080 FOR I = 1 TO NF:DGDF(I) = DGDF(I) * PCKT * CST / RM: NEXT
  I
8085 PRINT "L8085 "; DGDF(1), DGDF(2)
8086 PRINT "L8086 "; WCKT(5)
8100 RETURN
8500 REM FLOT
8510 ZU = PT(0):ZL = PT(0)
8520 FOR Z = 1 TO KB
8530 IF ZU < PT(Z) THEN ZU = PT(Z)
8540 IF ZL > PT(Z) THEN ZL = PT(Z)
8550 NEXT Z
8560 IF ZL < = 0 THEN 8580
8570 IZ = 12:UML = ZU:ZX = ZL / ZU:ZI = 40 * ZX + 24:ZA = 52:
  GOTO 8640
8580 IF ZU > = 0 THEN 8600
8590 IZ = 52:UML = - ZL:ZX = ZU / ZL:ZA = 52 - 40 * ZX:ZI =
  12: GOTO 8640
8600 UML = ZU - ZL: IF UML = 0 THEN IZ = 37
8610 IF UML < > 0 THEN ZX = ZL / UML
8620 IF UML < > 0 THEN IZ = 12 - 40 * ZX
8630 ZA = 52:ZI = 12
8640 PRINT : PRINT SPC( 24);PL$;"( ";J;" )"; SPC( 9);"VERSU
  S"; SPC( 15);"TIME"
8650 PRINT TAB( ZI - 3);"MINIMUM"; TAB( ZA - 3);"MAXIMUM"
8660 PRINT TAB( ZI - 5);ZL; TAB( ZA - 5);ZU
8670 PRINT TAB( 6);"TIME"; TAB( ZI);"!"; TAB( ZA);"!"; SPC(
  9);PL$;"( ";J;" )"
8680 TA = TO:Z = 0:N = 1
8685 ZX = PT(Z) / UML
8690 ZA = 40 * ZX + IZ
8700 PRINT Z; TAB( 5);TA; TAB( IZ);"!"; TAB( ZA);"*"; SPC( 4
  );PT(Z)
8710 Z = Z + 1
8720 IF Z > = KB THEN RETURN
8730 IF MODE$ = "AY" THEN TA = TA + DT
8740 IF MODE$ < > "AY" THEN TA = TM(Z)
8750 GOTO 8685
8800 RETURN
9000 REM MAP
9005 FOR I = 1 TO EDGE
9010 FOR J = 1 TO EDGE
9015 IF I < > CPIV(J) THEN 9055
9020 IF J < = VV THEN 9040
9025 IF J < NODE THEN 9045
9030 IF J < = (EDGE - II) THEN 9050
9035 VT$(I) = "N":IY$(I) = "F":CH(I) = J - (NH + VV): GOTO 90
  60
9040 VT$(I) = "F":IY$(I) = "M":CH(I) = J: GOTO 9060
9045 VT$(I) = "X":IY$(I) = "W":CH(I) = J - VV: GOTO 9060
9050 VT$(I) = "W":IY$(I) = "X":CH(I) = J - VV: GOTO 9060
9055 NEXT J
9060 NEXT I
9065 RETURN

```

