CR 134577

COMPUTER-AIDED SEMICONDUCTOR DEVICE ANALYSES*

by

Choong-Ki Kim

(a) JUNCTION-GATE FIELD-EFFECT TRANSISTORS

(b) ONE-DIMENSIONAL N^+-N-N^+ STRUCTURES

Vincent R. Lalli
Project Manager
Lewis Research Center
Office of Reliability
& Quality Assurance
21000 Brookpark Rd
Cleveland, OH 44135

December 1969

*This research project is supported by NASA under Contract NGR-33-008-090 and by NSF under Grant GK-1550.

The steady-state operation of a junction-gate field-effect transistor (JFET) with n-channel is described by the following set of equations.

$$\nabla^2 \Psi = -2\alpha[N(x,y) - n(x,y)] \tag{1}$$

$$\nabla \cdot J_n = \nabla \cdot [-\frac{1}{\alpha} n(x,y)\ \mu_n\ \nabla\ \varphi_n] = 0 \tag{2}$$

$$n(x,y) = e^{\Psi - \varphi_n} \tag{3}$$

The symbols used in equations (1) - (3) are

Ψ = electro-static potential normalized by kT/q

φ_n = quasi-Fermi level for electrons normalized by kT/q

N = donor concentration normalized by N_D

n = electron concentration normalized by N_D

N_D = donor concentration of the n-region

μ_n = electron mobility normalized by μ_{no}

μ_{no} = low field electron mobility

V_p = pinch-off voltage = $\frac{qN_D}{2\epsilon\epsilon_o} a^2$

2a = width of the n-region

α = qV_p/kT

J_n = electron current density normalized by $q\mu_{no}N_DV_p/a$

x,y = linear dimensions X,Y normalized by a

Figure 1 shows the model of a JFET to be considered. The region of interest to which equations (1) - (3) are to be applied is a

plane bounded by $X = 0$, $X = L_o$, $Y = a$, and $Y = -a$. The variations of equations (1) - (3) suitable for machine computation are

$$\nabla^2 \Psi = -2\alpha\left[N(x,y) - e^{\Psi-\varphi_n}\right] \tag{4}$$

$$\left.\begin{aligned} \nabla^2 \delta\Psi - 2\alpha e^{\Psi_a-\varphi_n}\,\delta\Psi &= -\nabla^2\Psi_a - 2\alpha\left[N(x,y) - e^{\Psi_a-\varphi_n}\right] \\ \Psi &= \Psi_a + \delta\Psi \end{aligned}\right\} \tag{5}$$

$$\left.\begin{aligned} \varphi_n(x) &= QN_i - \ln\left[e^{-\varphi_n(\ell_o)+QN_i} + \alpha I_D\, W(x)\right] \\ x &\in T_i \end{aligned}\right\} \tag{6}$$

$$I_D = \frac{1}{\alpha}\left[e^{-\varphi_n(o)+QN_1} - e^{-\varphi_n(\ell_o)+QN_1}\right]/W(o) \tag{7}$$

$$\left.\begin{aligned} W(x) &= \int_x^{\ell_o} \exp[-\Psi(t,o)+QN_i+\xi(t)]\,dt \\ x &\in T_i \end{aligned}\right\} \tag{8}$$

$$\xi(x) = -\ln\int_{-1}^{1} \exp\left[\Psi(x,y) - \Psi(x,o) + \ln\mu_n\right] dy \tag{9}$$

where

$$\ell_o = \frac{L_o}{a}$$

$$T_1 = \{x \mid o \le x \le x_1\}$$

$$T_i = \{x \mid x_{i-1} < x \le x_i\} \qquad i = 2,3,\ \cdots,\ M$$

$$0 < x_1 < x_2 < \cdots < x_M = \ell_o$$

QN_i = reference level for φ_n in the subinterval T_i

The field-dependent mobility for n-type silicon is approximated by

$$\mu_n = 1 - \frac{1}{2}\frac{E}{E_c} \qquad E/E_c < 10^{-4}$$

$$\mu_n = \frac{1 - e^{-E/E_c}}{E/E_c} \qquad E/E_c \geq 10^{-4}$$

where E_c = 7.77 KV/cm.

Subroutine BOUND solves equation (4) by the generalized Newton's method along the line $X = 0$ and $X = L_o$. The solutions are then used as the boundary conditions of the two-dimensional problem. Subroutine REF determines the subintervals T_i and the reference levels QN_i. The number of the subintervals is given by the input data. Subroutine FIRSTV solves equation (4) for the whole region of interest by the generalized Newton's method and subroutine COMPV solves equation (5) by successive over-relaxation method. Equations (6) - (9) are computed in subroutine COMPQ. The integration is performed by using Gaussian quadrature DQG32 in subroutine EXINT. The integrand is given by the function subprogram EFCT. The relaxation factor ω is determined in subroutine COME at the 16th iteration.

In the main program, α, doping levels, and the distribution of the donor concentration are first read from the input data. This defines the particular model to be considered. The number

of subintervals, EPS, and the first approximation of φ_n are given for each bias conditions. The potentials (Ψ and φ_n) are normalized by V_p in input and output while they are normalized by kT/q during the computation. EPS is a small constant and used to terminate the iteration scheme. The first approximation of the electro-static potential is arbitrary but the computing time is considerably reduced with a good approximation. The main program gives this approximation by solving the one-dimensional Poisson's equation along y.

$$\frac{d^2\Psi}{dt^2} = -\ 2\alpha\ N(x,t) \qquad 0 \le t \le b$$

where

$$t = 1 - y$$

Using the boundary conditions

$$\frac{d\Psi}{dt}\ (t=b) = 0$$

$$\Psi(t=0) = \Psi(x,o)$$

we obtain

$$\Psi(x,t) = \Psi(x,o) + 2\alpha \int_o^t \eta N(x,\eta)\ d\eta + 2\alpha t \int_t^b N(x,\eta)\ d\eta \qquad 0 \le t \le b$$

$$\Psi(x,t) = \varphi_n(x) + \ln \frac{N(x,1)}{N(x,o)} = \Psi_c \qquad t > b$$

b is determined by

$$\Psi_c = \Psi(x,o) + 2\alpha \int_o^1 \eta N(x,\eta)\ d\eta$$

If

$$\Psi_c > \Psi(x,o) + 2\alpha \int_o^1 \eta N(x,\eta)\, d\eta$$

Ψ is given by

$$\Psi(x,t) = \Psi_c + \left[\Psi(x,o) - \Psi_c\right](t-1)^2$$

The program shown in the following pages is for a JFET with $L_o = 4a$. Depending on the input data for the donor concentration, this program can be used for both the short device ($L/a = 2$) and the graded-channel device. Devices with larger values of L/a can be solved by a program similar to the one shown. Larger dimensions are necessary for these devices.

For a one-dimensional N^+-N-N^+ structure, the pertinent equations are similar to equations (4) - (9).

$$\frac{d^2\Psi}{dx^2} = -2\alpha\left[N(x) - e^{\Psi-\varphi_n}\right] \tag{10}$$

$$\left.\begin{aligned} \frac{d^2}{dx^2}(\delta\Psi) - 2\alpha e^{\Psi_a-\varphi_n}\,\delta\Psi &= -\frac{d^2\Psi_a}{dx^2} - 2\alpha\left[N(x) - e^{\Psi_a-\varphi_n}\right] \\ \Psi &= \Psi_a + \delta\Psi \end{aligned}\right\} \tag{11}$$

$$\varphi_n(x) = QN_i - \ln\left[e^{-\varphi_n(\ell_o)+QN_i} + \alpha JW(x)\right] \quad x \in T_i \tag{12}$$

$$W(x) = \int_{x}^{\ell_o} e^{-\Psi(t)-\ln u_n+QN_i}\, dt \qquad x \in T_i \tag{13}$$

$$J = \frac{1}{\alpha}\,\frac{e^{-\varphi_n(o)+QN_1} - e^{-\varphi_n(\ell_o)+QN_1}}{W(o)} \tag{14}$$

Since the N^+-N junctions are step-junctions, the distribution of the donor concentration is simple and given in the main program. Subroutines are similar to those for a JFET and are self explanatory.

JUNCTION-GATE FIELD-EFFECT TRANSISTORS

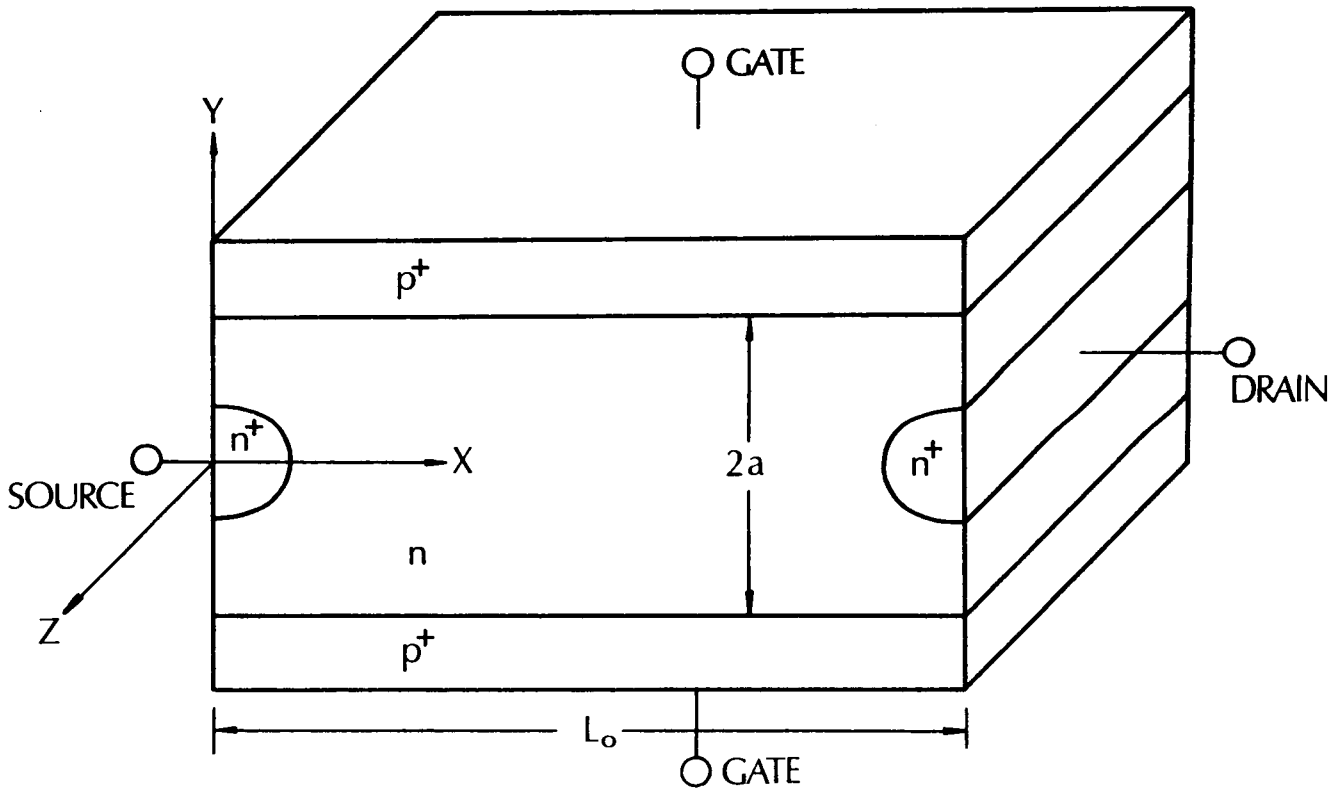

Figure 1. Device Model of Junction-Gate Field-Effect Transistors

```
0001          COMMON  /BLV/V(41,81),BETA,E(41,81) /BLDOD/DOD(21,81)
             1        /BLQ/Q(81),H,EC,DC,ALP  /BLRQ/QN(20),NN(20)
0002          DOUBLE PRECISION  E,H,Q,V,A0,A1,A2,DC,EC,QN,ALP,BETA
        C
        C DIMENSION
        C
0003          LX1 = 41
0004          LXC = 21
0005          LY1 = 81
0006          LYC = 41
0007          LX2 = LX1 - 1
0008          LY2 = LY1 - 1
        C
        C GEOMETRIC PARAMETERS AND DOPING LEVELS
        C
0009          READ( 5,1 )  ALP, BND, BNA, BNI, BNDD
0010        1 FORMAT( F10.1, 1P4E10.2 )
0011          WRITE( 6,2 )  ALP, BND, BNA, BNI, BNDD
0012        2 FORMAT( '1', 'ALPHA = VP/VT =', F6.1 /
             1        '0', 'L/A = LENGTH TO WIDTH RATIO = 2 ' /
             2        '0', 'ND  =', 1PE10.2, 1X, 'PER C.C.' /
             3        '0', 'NA  =', 1PE10.2, 1X, 'PER C.C.' /
             4        '0', 'NI  =', 1PE10.2, 1X, 'PER C.C.' /
             5        '0', 'ND+ =', 1PE10.2, 1X, 'PER C.C.' // )
        C
0013          H = 1. / (LXC - 1.)
0014          A0 = 0.02586 * ALP
0015          A1 =DSQRT(0.24*8.854*A0/(1.601*(BND/1.0E+15)))
0016          A2 = 7.77
0017          VB = ALOG( (BND/BNI)*(BNA/BNI) )*0.02586
0018          A = 1.0 /DSQRT(2.*ALP)
        C
0019          WRITE( 6,3 )  A0, A1, VB, A2, H, A
0020        3 FORMAT( '0','VP=PINCH-OFF VOLTAGE =',F7.3,1X,'VOLTS' /
             1'0','A=HALF OF THE CHANNEL WIDTH =',F6.3,1X,'MICRON'/
             2'0','VB=BUILT IN POTENTIAL AT P-N JUNCTION =',F7.4,1X,
             3'VOLTS'/
             4'0','CRITICAL ELECTRIC FIELD =',F6.3,1X,'KV/CM'/
             5'0','MESH SIZE =',F6.3,1X,'A' /
             6'0','EXTRINSIC DEBYE LENGTH =',F6.3,1X,'A' // )
        C
0021          VB = VB / 0.02586
0022          BETA = 2.*ALP*(H**2)
0023          EC = (2.*H)*(A2*0.1)*A1/0.02586
        C
        C DOPING PROFILE
        C
0024          READ( 5,5 )  ((DOD(I,J),I=1,11),J=1,21)
```

```
0025          READ( 5,5 )  ((DOD(I,J),I=11,21),J=1,21)
0026        5 FORMAT( 11F6.3 )
0027          DO 6  I = 1, LXC
0028          DO 6  J = 22, LYC
0029        6 DOD(I,J) = 1.0
0030          DO 7  J = LYC, LY1
0031          JJ = LY1 + 1 - J
0032          DO 7  I = 1, LXC
0033        7 DOD(I,J) = DOD(I,JJ)
0034          DO 8  I = 1, LXC
0035          DO 8  J = 1, LY1
0036        8 DOD(I,J) = DOD(I,J)*BETA
0037          BETA = DLOG(BETA)
        C
        C BIAS CONDITIONS, NUMBER OF SUBINTERVALS, AND EPS
        C
0038       10 READ( 5,11 )  VGS, VDS, NRQ, EPS
0039       11 FORMAT( 2F6.2, I6, 1PE12.3 )
0040          IF( NRQ .EQ. 0 ) STOP
0041          WRITE( 6,12 )  VGS, VDS
0042       12 FORMAT( '1', 'VGS =',F5.2,1X,'(VP-VB)' /
             1         '0', 'VDS =',F5.2,1X,'(VP-VB)' / )
        C
0043          READ( 5,9)  (Q(J),J=1,81,10)
0044        9 FORMAT( 1P7D10.3 )
0045          DO 156  J = 1, 8
0046          K = (J-1)*10
0047          DO 156  I = 2, 10
0048      156 Q(K+I) = (Q(K+10+1)-Q(K+1))*(I-1)/10. + Q(K+1)
0049          Q(LY1) = 0.0
0050          Q(1) = Q(LY1) - VDS*(ALP-VB)
0051          V(1,1) = Q(1) - VGS*(ALP-VB) - VB
        C
        C BOUNDARY CONDITION AT THE GATE
        C
0052          DO 13  J = 1, LY1
0053          V(1,J) = V(1,1)
0054       13 V(LX1,J) = V(1,J)
        C
        C FIRST APPROXIMATION OF THE QUASI FERMI LEVEL
        C
0055          DO 14  J = 2, LY2
0056       14 Q(J) = Q(1) + ALP*Q(J)
        C
        C FIRST APPROXIMATION OF THE ELECTRO-STATIC POTENTIAL
        C
0057          DO 23  J = 1, LY1
0058          AD = Q(J) + ALOG(DOD(LXC,J)/DOD(1,J))
```

```
0059              DO 15  I = 2, LXC
0060              V(I,J) = V(I-1,J)+((I-2)*DOD(I-1,J)+(I-1)*DOD(I,J))/2.
0061              NB = I
0062              IF( V(I,J) .GE. AO ) GO TO 16
0063           15 CONTINUE
0064           16 IF( NB .EQ. LXC .AND. V(LXC,J) .LT. AO ) GO TO 19
0065              DO 17  I = NB, LXC
0066           17 V(I,J) = AO
0067              A2 = 4*(AO-V(NB-1,J)) / (DOD(NB-1,J)+DOD(NB,J))
0068              A2 = DSQRT(A2 + (NB-1)**2)
0069              A2 = (DOD(NB-1,J)+DOD(NB,J))*(A2-NB+1)/2
0070              V(NB-1,J) = V(NB-1,J) + (NB-2)*A2
0071              NB = NB - 2
0072              DO 18  I = 2, NB
0073              A1 = I - 1
0074              V(I,J) = V(I,J) + A2*A1
0075              DO 18  K = I, NB
0076           18 V(I,J) = V(I,J) + A1*(DOD(K,J)+DOD(K+1,J))/2.
0077              GO TO 21
0078           19 A2 = V(1,J) - AO
0079              DO 20  I = 2, LXC
0080              A1 = H*(I-1)
0081           20 V(I,J) = AO + A2*(A1-1.)**2
0082           21 DO 22  I = 2, LXC
0083              II = LX1 + 1 - I
0084           22 V(II,J) = V(I,J)
0085           23 CONTINUE
            C
            C BOUNDARY CONDITION AT THE SOURCE
            C
0086              WRITE( 6,30 )
0087           30 FORMAT( //// '0', 'BOUNDARY CONDITION AT THE SOURCE'//)
0088              CALL   BOUND(1,LXC)
            C
            C BOUNDARY CONDITION AT THE DRAIN
            C
0089              WRITE( 6,31 )
0090           31 FORMAT( //// '0', 'BOUNDARY CONDITION AT THE DRAIN'//)
0091              CALL   BOUND(LY1,LXC)
            C
            C PREPERATION OF THE ITERATION
            C
0092              CALL   REF(VDS,NRQ,LY1)
0093              CALL   FIRSTV(LX1,LX2,LXC,LY2)
            C
            C ITERATION
            C
0094           40 FORMAT( '1' )
```

```
0095              K = 1
0096           41 WRITE( 6,40 )
0097           42 CALL   COMPQ(LXC,LY1,LY2,NRQ)
0098              WRITE( 6,43 )  K, DC
0099           43 FORMAT( '0', 2X, 'K  =',I3,3X,'DRAIN CURRENT =',
                 1         1PD11.4 )
0100              CALL   COMPV(LX1,LX2,LXC,LY1,LY2,RMAX)
0101              IF( RMAX .LT. EPS ) GO TO 50
0102              IF( K .GE. 50 ) GO TO 50
0103              K = K + 1
0104              IF( K.EQ. 19.OR.K.EQ. 37 ) GO TO 41
0105              GO TO 42
            C
            C COMPUTE FOR THE OUTPUT
            C
0106           50 DO 53  I = 1, LXC
0107              DO 53  J = 1, LY1
0108           53 E(I,J) =DEXP(V(I,J)-Q(J))
            C
0109              AO = Q(1)
0110              DO 54  J = 1, LY1
0111           54 Q(J) = (Q(J)-AO) / ALP
            C
0112              AO = V(1,1)
0113              DO 55  J = 1, LY1
0114              DO 55  I = 1, LXC
0115           55 V(I,J) = (V(I,J)-AO) / ALP
            C
            C OUTPUT
            C
0116              WRITE( 6,61 )
0117           61 FORMAT( '1','ELECTRO-STATIC POTENTIAL',5X,'PAGE 1'/)
0118              WRITE( 6,66 )  ((V(I,J),I=1,21),J,J=1,41)
0119              WRITE( 6,62 )
0120           62 FORMAT( '1','ELECTRO-STATIC POTENTIAL',5X,'PAGE 2'/)
0121              WRITE( 6,66 )  ((V(I,J),I=1,21),J,J=41,81)
            C
0122              WRITE( 6,63 )
0123           63 FORMAT( '1', 'ELECTRON DENSITY', 5X, 'PAGE 1' / )
0124              WRITE( 6,66 )  ((E(I,J),I=1,21),J,J=1,41)
0125              WRITE( 6,64 )
0126           64 FORMAT( '1', 'ELECTRON DENSITY', 5X, 'PAGE 2' / )
0127              WRITE( 6,66 )  ((E(I,J),I=1,21),J,J=41,81)
0128           66 FORMAT( ' ', 21F6.3,I3 )
            C
0129              WRITE( 6,65 )  (K,K=1,10)
0130           65 FORMAT( '1', 'QUASI FERMI LEVEL' / '0', 1X, 10I12 / )
0131           67 FORMAT( '0', I4, 1X, 1P10D12.3 )
```

```
0132          DO 68  K = 1, 8
0133          I = 10*(K-1)
0134       68 WRITE( 6,67 )  I, (Q(I+J),J=1,10)
0135          I = 80
0136          WRITE( 6,67 )  I, Q(I+1)
0137          GO TO 10
0138          END
```

TOTAL MEMORY REQUIREMENTS 00174A BYTES

```
0001          SUBROUTINE   REF(VDS,N,LY1)
        C
0002          COMMON /BLQ/Q(81) /BLRQ/QN(20),NN(20)
0003          DOUBLE PRECISION  Q,QN,A0,A1,A2
        C
0004          IF( VDS .EQ. 0.0 ) RETURN
0005          NN(1) = 1
0006          M = N - 1
0007          A1 = (Q(LY1)-Q(1))/M
0008          A0 = A1 / 2.
0009          QN(1) = Q(1) + A0
0010          DO 2  I = 2, M
0011          QN(I) = QN(I-1) + A1
0012          NN(I) = NN(I-1) + 1
0013          K = NN(I) + 1
0014          A2 = QN(I-1) + A0
0015          DO 1  J = K, LY1
0016          IF( Q(J) .LE. A2 ) NN(I) = J
0017          IF( Q(J) .GT. A2 ) GO TO 2
0018        1 CONTINUE
0019        2 CONTINUE
0020          NN(N) = LY1
        C
0021          WRITE( 6,21 )  (K,K=1,10)
0022       21 FORMAT( '1', 'FIRST APPROXIMATION OF THE QUASI FERMI
             1              LEVEL' / '0', 1X, 10I12 / )
0023          DO 22  I = 1, 8
0024          K = 10*(I-1)
0025       22 WRITE( 6,23 )  K, (Q(K+J),J=1,10)
0026       23 FORMAT( '0', I4, 1X, 1P10D12.3 )
0027          K = 80
0028          WRITE( 6,23 )  K, Q(K+1)
0029          WRITE( 6,24 )
0030       24 FORMAT( ///'0', 9X, 'J', 7X, 'NN(J)', 6X, 'QN(J)' // )
0031          DO 25  I = 1, M
0032       25 WRITE( 6,26 )  I, NN(I), QN(I)
0033       26 FORMAT( ' ', 2I10, 1PE15.3 )
0034          WRITE( 6,26 )  N, NN(N)
0035          RETURN
0036          END
```

TOTAL MEMORY REQUIREMENTS 000540 BYTES

```
0001          SUBROUTINE   FIRSTV(LX1,LX2,LXC,LY2)
        C
0002          COMMON  /BLV/V(41,81),BETA /BLQ/Q(81) /BLDOD/DOD(21,81)
0003          DOUBLE PRECISION  V,BETA,Q,A0,A1,A2,ER1
        C
0004          KV = 1
0005          OME = 1.0
0006        1 A2 = 0.0
0007          EER = 0.0
0008          DO 3  I = 2, LX2
0009          IF( I .LE. LXC ) II = I
0010          IF( I .GT. LXC ) II = LX1 + 1 - I
0011          DO 3  J = 2, LY2
0012          A0 = V(I+1,J)+V(I-1,J)+V(I,J+1)+V(I,J-1)
0013          A1 =DEXP(V(I,J)-Q(J)+BETA)
0014          A0 = (A0-4.*V(I,J)) + (DOD(II,J)-A1)
0015          A1 = A1 + 4.
0016          A0 = OME*(A0/A1)
0017          V(I,J) = V(I,J) + A0
0018          B = DABS(A0)
0019          IF( B .GT. EER ) EER = B
0020        3 A2 = A2 + A0**2
        C
0021          IF( EER .LT. 0.005 ) GO TO 4
0022          IF( KV .GE. 200 ) GO TO 4
0023          CALL   COME(KV,A2,ER1,OME)
0024          KV = KV + 1
0025          GO TO 1
        C
0026        4 WRITE( 6,5 )  KV, OME, EER
0027        5 FORMAT( ///'0','KV =',I3,3X,'OME =',F5.2,3X,'EER =',
             1            1PE10.2 )
0028          RETURN
0029          END
```

TOTAL MEMORY REQUIREMENTS 00048C BYTES

```
0001          SUBROUTINE   COMPQ(LXC,LY1,LY2,N)
        C
0002          COMMON  /BLV/V(41,81) /BLQ/Q(81),H,EC,DC,ALP
             1        /BLRQ/QN(20),NN(20)
0003          DOUBLE PRECISION  V,Q,H,EC,DC,ALP,QN,X(81),Y(81),
             1                  XX(81),AO,A1,A2
        C
0004          M = N - 1
0005          IF( Q(1) .EQ. Q(LY1) ) GO TO 100
0006          DO 3  J = 1, LY1
0007          DO 1  I = 1, LXC
0008          K = LXC + 1 - I
0009          IF( J .EQ. 1 ) AO = 4.*V(I,J+1)-3.*V(I,J)-V(I,J+2)
0010          IF( J .EQ. LY1 ) AO = 3.*V(I,J)+V(I,J-2)-4.*V(I,J-1)
0011          IF( J .GT. 1 .AND. J .LT. LY1 ) AO = V(I,J+1)-V(I,J-1)
0012          A1 =DABS(AO/EC)
0013          IF( A1 .LT. 1.0D-4 ) A2 = 1. - A1/2.
0014          IF( A1 .GE. 1.0D-4 ) A2 = (1.-DEXP(-A1))/A1
0015          A2 = DLOG(A2)
0016        1 XX(K) = V(I,J) - V(LXC,J) + A2
0017          CALL   EXINT(H,1,LXC,XX,Y,0.0,LXC)
0018        3 X(J) = -DLOG(Y(1))
        C
0019          Y(LY1) = 0.0
0020          DO 7  I = 1, M
0021          K = N - I
0022          JS = NN(K)
0023          JL = NN(K+1)
0024          DO 5  J = JS, JL
0025        5 XX(J) = -V(LXC,J) + X(J) + QN(K)
0026          CALL   EXINT(H,JS,JL,XX,Y,Y(JL),LY1)
0027          IF( K .EQ. 1 ) GO TO 7
0028          IF( Y(JS) .EQ. 0.0 ) AO = -1000.
0029          IF( Y(JS) .NE. 0.0 ) AO = DLOG(Y(JS))
0030          Y(JS) = DEXP(QN(K-1)-QN(K)+AO)
0031        7 CONTINUE
        C
0032          DO 9  J = 1, LY2
0033          AO = Y(J)
0034          IF( AO .EQ. 0.0 ) Y(J) = -1000.
0035          IF( AO .NE. 0.0 ) Y(J) = DLOG(Y(J))
0036        9 CONTINUE
        C
0037          AO = -Q(LY1) + QN(1) - Y(1)
0038          A1 = -Q(1) + QN(1) - Y(1)
0039          DO 11  J = 2, LY2
0040          DO 10  I = 1, M
0041          IF( J .GT. NN(I) .AND. J .LE. NN(I+1) ) K = I
```

```
0042          10 CONTINUE
0043             Q(J) =DEXP(QN(K)-Q(LY1))
0044             Q(J) = Q(J) - DEXP(A0 + Y(J))
0045             Q(J) = Q(J) + DEXP(A1 + Y(J))
0046          11 Q(J) = QN(K) - DLOG(Q(J))
        C
0047             DC = DEXP(QN(1)-Q(1)-Y(1)) - DEXP(QN(1)-Q(LY1)-Y(1))
0048             DC = 2.*DC/ALP
0049             RETURN
0050         100 DC = 0.0
0051             RETURN
0052             END
```

TOTAL MEMORY REQUIREMENTS 001052 BYTES

```
0001          SUBROUTINE   COMPV(LX1,LX2,LXC,LY1,LY2,RMAX)
        C
0002          COMMON  /BLV/V(41,81),BETA,E(41,81) /BLQ/Q(81)
            U         /BLDOD/DOD(21,81)
0003          DOUBLE PRECISION  D(41,81),R(40,80),E,Q,V,A0,A1,A2,
            1                   ER1,BETA
        C
0004          RMAX = 0.0
0005          DO 2  I = 2, LX2
0006          IF( I .LE. LXC ) II = I
0007          IF( I .GT. LXC ) II = LX1 + 1 - I
0008          DO 2  J = 2, LY2
0009          A0 = V(I+1,J)+V(I-1,J)+V(I,J+1)+V(I,J-1)-4.*V(I,J)
0010          E(I,J) =DEXP(V(I,J)-Q(J)+BETA)
0011          R(I,J) = A0 + (DOD(II,J) - E(I,J))
0012          B = DABS(R(I,J))/DOD(II,J)
0013          IF( B .GT. RMAX ) RMAX = B
0014        2 CONTINUE
0015          DO 3  I = 1, LX1
0016          DO 3  J = 1, LY1
0017        3 D(I,J) = 0.0
        C
0018          KD = 1
0019          OME = 1.0
0020       10 DMAX = 0.0
0021          A2 = 0.0
0022          EER = 0.0
0023          DO 4  I = 2, LX2
0024          DO 4  J = 2, LY2
0025          A0 = D(I+1,J) + D(I-1,J) + D(I,J+1) + D(I,J-1)
0026          A1 = 4. + E(I,J)
0027          A0 = OME*(A0-(A1*D(I,J)-R(I,J)))/A1
0028          D(I,J) = D(I,J) + A0
0029          B = DABS(A0)
0030          IF( B .GT. EER ) EER = B
0031          B = DABS(D(I,J))
0032          IF( B .GT. DMAX ) DMAX = B
0033        4 A2 = A2 + A0**2
        C
0034          IF( EER/DMAX .LT. 1.0E-02 ) GO TO 20
0035          IF( KD .GE. 100 ) GO TO 20
0036          CALL   COME(KD,A2,ER1,OME)
0037          KD = KD + 1
0038          GO TO 10
        C
0039       20 WRITE( 6,100 )  KD, OME, EER, DMAX, RMAX
0040      100 FORMAT( ' ',2X,'KD =',I3,3X,'OME =',F5.2,3X,'EER =',
            1  1PE10.2,3X,'DMAX =',1PE10.2,3X,'RMAX =',1PE10.2 )
```

```
0041          DO 5  I = 2, LX2
0042          DO 5  J = 2, LY2
0043        5 V(I,J) = V(I,J) + D(I,J)
0044          RETURN
0045          END
```

TOTAL MEMORY REQUIREMENTS 00D33C BYTES

```
0001          SUBROUTINE   BOUND(J,NDIM)
        C
0002          COMMON  /BLV/V(41,81),BETA /BLQ/Q(81) /BLDOD/DOD(21,81)
0003          DOUBLE PRECISION  A0,A1,A2,V,Q,BETA,ER1
        C
0004          MDIM = NDIM - 1
0005          OME = 1.0
0006          K = 1
0007        1 A2 = 0.0
0008          EER = 0.0
0009          DO 2  I = 2, MDIM
0010          A0 = V(I+1,J)+V(I-1,J)-2.*V(I,J)
0011          A1 = DEXP(V(I,J)-Q(J)+BETA)
0012          A0 = A0 + (DOD(I,J) - A1)
0013          A1 = 2. + A1
0014          A0 = OME*(A0/A1)
0015          V(I,J) = V(I,J) + A0
0016          B = DABS(A0)
0017          IF( B .GT. EER ) EER = B
0018        2 A2 = A2 + A0**2
        C
0019          IF( EER .LT. 1.0E-04 .OR. K .GE. 150 ) GO TO 5
0020          CALL   COME(K,A2,ER1,OME)
0021          K = K + 1
0022          GO TO 1
        C
0023        5 WRITE( 6,20 )   K, OME, EER
0024       20 FORMAT( ' ', 'KV =',I3,5X,'OME =',F5.2,5X,'EER =',
             1                1PE10.2 // )
        C
0025          WRITE( 6,21 )  ((V(7*(K-1)+I,J),I=1,7),K=1,3)
0026       21 FORMAT( ' ', 1P7E15.3 )
0027          DO 30  I = 2, MDIM
0028          II = MDIM*2 + 2 - I
0029       30 V(II,J) = V(I,J)
0030          RETURN
0031          END
```

TOTAL MEMORY REQUIREMENTS 000540 BYTES

```
0001          SUBROUTINE   EXINT(H,IS,IF,X,Y,YO,NDIM)
        C
        C THIS SUBROUTINE INTEGRATES EXP(X) FROM I, WHICH IS BETWEEN
        C IS AND IF, TO IF AND STORES (RESULT+YO) AT Y(I). H IS THE
        C MESH SIZE AND NDIM IS THE MAXIMUM DIMENSION OF X AND Y.
        C
0002          DOUBLE PRECISION  A,H,X,Y,A0,A1,A2,YO,H2I,HSQ2I,EFCT
0003          DIMENSION  X(NDIM), Y(NDIM)
0004          COMMON  /BLINT/A1,A2
0005          EXTERNAL   EFCT
        C
0006          H2I = 1.0 / (2.*H)
0007          HSQ2I = 1.0 / (2.*H**2)
0008          Y(IF) = YO
0009          LAST = IF - IS
0010          DO 1  II = 2, LAST
0011          I = IF + 1 - II
0012          A0 = X(I)
0013          A1 = (X(I+1)-X(I-1))*H2I
0014          A2 = (X(I+1)+X(I-1)-2.*X(I))*HSQ2I
0015          CALL   DQG32(0.0,H,EFCT,A)
0016        1 Y(I) = Y(I+1) + DEXP(A0+DLOG(A))
0017          CALL   DQG32(-H,0.0,EFCT,A)
0018          Y(IS) = Y(IS+1) + DEXP(A0+DLOG(A))
0019          RETURN
0020          END
```

TOTAL MEMORY REQUIREMENTS 000368 BYTES

```
0001          FUNCTION   EFCT(Z)
        C
0002          DOUBLE PRECISION  A1,A2,Z,EFCT
0003          COMMON  /BLINT/A1,A2
0004          EFCT = DEXP(A1*Z + A2*Z**2)
0005          RETURN
0006          END
```

TOTAL MEMORY REQUIREMENTS 000166 BYTES

```
0001          SUBROUTINE   COME(KDV,ER,ERP,OME)
        C
0002          DOUBLE PRECISION  A,ERP,ER
        C
0003          IF( KDV .GT. 16 ) GO TO 2
0004          A = DSQRT(ER)
0005          IF( KDV .LE. 15 ) GO TO 1
0006          AA = A/ERP
0007          BB = (AA+OME-1.)/(OME*SQRT(AA))
0008          OME = 2./(1.+SQRT(1.-BB**2))
0009        1 ERP = A
0010        2 RETURN
0011          END
```

TOTAL MEMORY REQUIREMENTS 00024C BYTES

ONE-DIMENSIONAL N^+-N-N^+ STRUCTURES

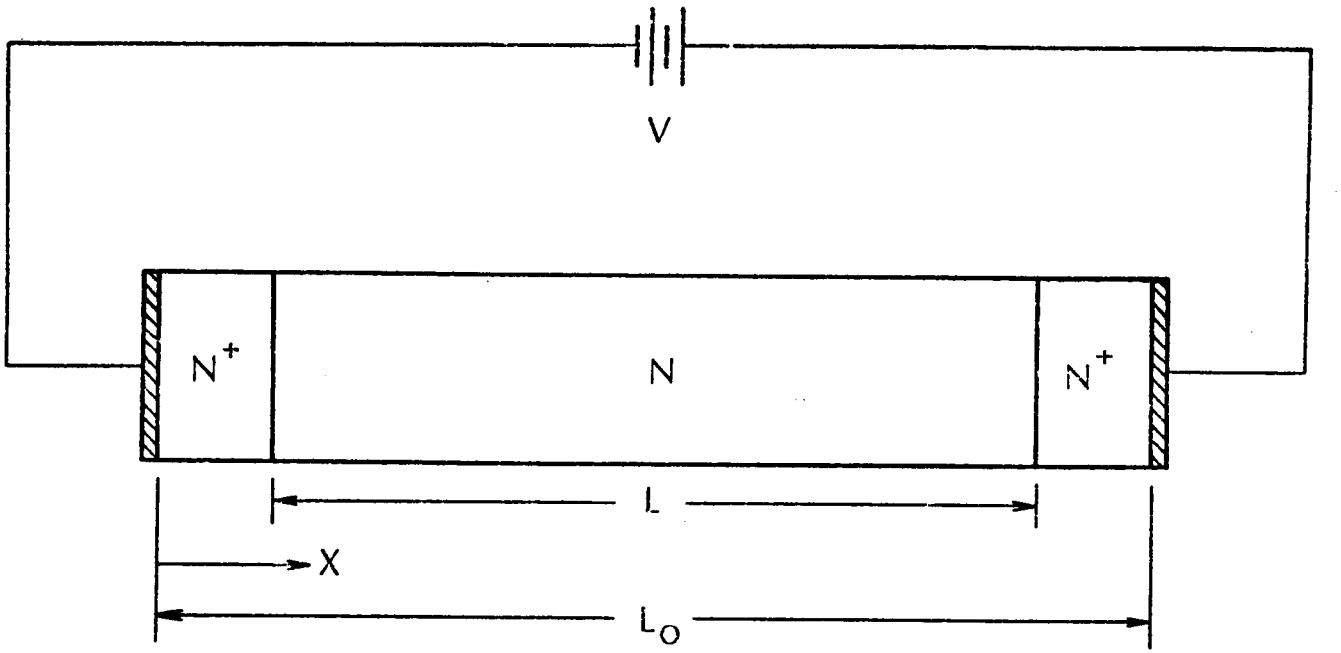

Figure 2. One-Dimensional N^+-N-N^+ Structure

```
0001          COMMON   /BLV/V(341),HR,BETAS,BETAL,BETASL,BETALL
             1         /BLE/E(341) /BLQ/Q(341),HS,HL,EN,ALP,DC
             2         /BLRQ/QN(20),NN(20) /BLDOD/DOD(341)
             3         /BLAUX/X(341),XX(341),Y(341)
0002          DOUBLE PRECISION  V,HR,BETAS,BETAL,BETASL,BETALL,Q,HS,
             1                  HL,EN,ALP,DC,QN,E,A0,A1,A2,A3,A4,DR,
             2                  VB,BND,BNDD,DOD,VCE,X,Y,XX
        C
0003          LXE = 41
0004          LXC = 291
0005          LXT = 331
0006          LXM = 1 + (LXT-1) / 2
0007          LXE1 = LXE - 1
0008          LXE2 = LXE + 1
0009          LXC1 = LXC - 1
0010          LXC2 = LXC + 1
0011          LXT1 = LXT - 1
        C
0012          READ( 5,1 )  ALP, BND, BNDD
0013        1 FORMAT( 1PD10.1, 1P2D10.1 )
0014          DR = BNDD/BND
0015          DO 2  I = 1, LXE1
0016          DOD(I) = DR
0017          II = LXT + 1 - I
0018        2 DOD(II) = DR
0019          DO 3  I = LXE, LXC
0020        3 DOD(I) = 1.0
0021          DOD(LXE1) = 1. + (DR-1.) / 2.
0022          DOD(LXC2) = DOD(LXE1)
0023          WRITE( 6,4 )  (DOD(I),I=1,LXT)
0024        4 FORMAT( '0', 1P10D11.3 )
0025          WRITE( 6,5 )  ALP, BND, BNDD
0026        5 FORMAT( '1', 'ALPHA =', F6.1 /
             1         '0', 'ND  =', 1PE8.1, 1X, 'PER C.C.' /
             2         '0', 'ND+ =', 1PE8.1, 1X, 'PER C.C.' // )
0027          HL = 1.0D 00 / (LXC-LXE)
0028          HR = 25.0 / DSQRT(DR)
0029          HS = HL * HR
0030          A0 = 0.02586 * ALP
0031          A1 = DSQRT(0.24*8.854*A0/(1.601*(BND/1.0E+15)))
0032          A2 = A1*(LXE-1.)*HR/(LXC-LXE)
0033          A3 = DSQRT(2.*ALP) * HL
0034          A4 = DSQRT(2.*ALP*DR) * HS
0035          VB = 0.02586*DLOG(DR)
0036          EN = 0.2586 / A1
0037          WRITE( 6,6 )  A0,A1,A2,A3,A4,HR,VB
0038        6 FORMAT('0','VO = CROSSOVER VOLTAGE =',F7.3,' VOLTS' /
             1        '0','SIZE OF THE N REGION =',F6.3,' MICRONS' /
```

```
                2        '0','SIZE OF THE N+ REGION =',F6.3,' MICRONS' /
                3        '0','(MESH SIZE/DEBYE LENGTH) IN N  REGION =',
                4        F7.4 /
                5        '0','(MESH SIZE/DEBYE LENGTH) IN N+ REGION =',
                6        F6.3 /
                7        '0','RATIO OF TWO MESH SIZES =', F6.3 /
                8        '0','BUILT-IN POTENTIAL AT THE N-N+ JUNCTION ='
                9        ,F6.3,' VOLTS' // )
          C
0039            VB = DLOG(DR)
0040            BETAL = 2.*ALP*(HL**2)
0041            BETAS = 2.*ALP*(HS**2)
0042            DO 7  I = 1, LXE1
0043            II = LXT + 1 - I
0044            DOD(I) = DOD(I)*BETAS
0045          7 DOD(II) = DOD(I)
0046            DO 8  I = LXE, LXC
0047          8 DOD(I) = DOD(I) * BETAL
0048            BETASL = DLOG(BETAS)
0049            BETALL = DLOG(BETAL)
          C
          C READ AND WRITE THE BIAS CONDITION
0050         10 READ( 5,11 )  VCE, NRQ, EPS
0051         11 FORMAT( 1PD10.2, I10, 1PE10.2 )
0052            IF( NRQ .EQ. 0 ) STOP
0053            WRITE( 6,12 )  VCE
0054         12 FORMAT( '1', 'VCE =', F5.2, 1X, 'V0' / )
          C
0055            Q(1) = 0.0
0056            Q(LXT) = VCE*ALP
0057            V(1) = Q(1) + VB
0058            V(LXT) = Q(LXT) + VB
          C
          C INITIAL DISTRIBUTION OF Q AND V
0059            DO 13  I = 2, LXE
0060            Q(I) = Q(1)
0061         13 V(I) = V(1)
0062            A0 = LXC - LXE
0063            IF( VCE .LT. 1.0 ) A2 = 1.0
0064            IF( VCE .GE. 1.0 ) A2 = 1.5
0065            A0 = (Q(LXT)-Q(1))/(A0**A2)
0066            DO 14  I = LXE2, LXC1
0067            A1 = I - LXE
0068            Q(I) = Q(1) + A0*A1**A2
0069         14 V(I) = Q(I)
0070            DO 15  I = LXC, LXT1
0071            Q(I) = Q(LXT)
0072         15 V(I) = V(LXT)
```

```
0073          CALL   REF(VCE,NRQ,LXE,LXC,LXT)
        C
        C ITERATION
0074          KV1 = 10
0075          K = 1
0076       21 WRITE( 6,22 )
0077       22 FORMAT( '1' )
0078       23 CALL   COMPQ(LXT,LXE,LXC,LXT1,NRQ)
0079          WRITE( 6,24 )  K, DC
0080       24 FORMAT( '0',2X,'K  =',I3,3X,'CURRENT =',1PD11.4 )
0081          CALL   FIRSTV(LXT1,LXE,LXC,KV)
0082          IF( KV1 .LT. 10 .AND. KV .LT. 10 ) GO TO 25
0083          IF( K .GE. 15 ) GO TO 25
0084          KV1 = KV
0085          K = K + 1
0086          IF( K .EQ. 19 .OR. K .EQ. 37 ) GO TO 21
0087          GO TO 23
        C
0088       25 K = 1
0089       26 WRITE( 6,22 )
0090       27 CALL   COMPQ(LXT,LXE,LXC,LXT1,NRQ)
0091          WRITE( 6,24 ) K, DC
0092          CALL   COMPV(LXT,LXT1,LXE,LXC,RMAX)
0093          IF( RMAX .LT. EPS ) GO TO 30
0094          IF( K .GE. 200 ) GO TO 30
0095          K = K + 1
0096          IF(K.EQ.19.OR.K.EQ.37.OR.K.EQ.55.OR.K.EQ.73.OR.K.EQ.
             1   91.OR.K.EQ.109.OR.K.EQ.127.OR.K.EQ.145.OR.K.EQ.163
             3   .OR.K.EQ.181) GO TO 26
0097          GO TO 27
        C
        C OUTPUT
0098       30 A0 = V(1) - VB
0099          A1 = Q(1)
0100          DO 31  I = 1, LXT
0101          XX(I) = DEXP(V(I)-Q(I))
0102          V(I) = (V(I)-A0)/ALP
0103       31 Q(I) = (Q(I)-A1)/ALP
0104          WRITE( 6,32 )
0105       32 FORMAT( '1',6X,'ELECTRO-STATIC',9X,'QUASI-FERMI',12X,
             1         'ELECTRON',12X,'ELECTRIC'/' ',11X,'POTENTIAL',
             2         11X,'POTENTIAL',13X,'DENSITY',15X,'FIELD' /
             3         '0',18X,'V0',18X,'V0',18X,'N0',15X,'KV/CM'//)
0106          DO 33  I = 1, LXE
0107       33 WRITE( 6,34 )  I,V(I),Q(I),XX(I),E(I)
0108       34 FORMAT( ' ', I3, F17.5, 3F20.5 )
0109          WRITE( 6,32 )
0110          DO 35  I = LXE, LXC
```

```
0111          35 WRITE( 6,34 )  I,V(I),Q(I),XX(I),E(I)
0112             WRITE( 6,32 )
0113             DO 36  I = LXC, LXT
0114          36 WRITE( 6,34 )  I,V(I),Q(I),XX(I),E(I)
0115             GO TO 10
0116             END
```

TOTAL MEMORY REQUIREMENTS 0011DE BYTES

```
0001          SUBROUTINE   REF(VCE,N,LXE,LXC,LXT)
        C
0002          COMMON /BLQ/Q(341) /BLRQ/QN(20), NN(20)
0003          DOUBLE PRECISION  Q,QN,A0,A1,A2
        C
0004          WRITE( 6,20 )
0005       20 FORMAT( '0', /// )
0006          IF( VCE .EQ. 0.0 ) RETURN
0007          NN(1) = 1
0008          NN(2) = LXE
0009          QN(1) = Q(1)
0010          M = N - 2
0011          A1 = (Q(LXT)-Q(1))/(M-1)
0012          A0 = A1 / 2.
0013          QN(2) = Q(1) + A0
0014          DO 2  I = 3, M
0015          QN(I) = QN(I-1) + A1
0016          K = NN(I-1) + 1
0017          A2 = QN(I-1) + A0
0018          DO 1  J = K, LXC
0019          IF( Q(J) .LE. A2 ) NN(I) = J
0020          IF( Q(J) .GT. A2 ) GO TO 2
0021        1 CONTINUE
0022        2 CONTINUE
0023          NN(N-1) = LXC
0024          NN(N) = LXT
0025          QN(N-1) = Q(LXT)
0026          WRITE( 6,10 )
0027       10 FORMAT( '0', 9X, 'I', 7X, 'NN(I)',6X, 'QN(I)' // )
0028          M = N - 1
0029          DO 11  I = 1, M
0030       11 WRITE( 6,12 )  I, NN(I), QN(I)
0031       12 FORMAT( ' ', 2I10, 1PE15.3 )
0032          WRITE( 6,12 )  N, NN(N)
0033          WRITE( 6,20 )
0034          RETURN
0035          END
```

TOTAL MEMORY REQUIREMENTS 000478 BYTES

```
0001          SUBROUTINE   FIRSTV(LXT1,LXE,LXC,KV)
        C
0002          COMMON  /BLV/V(341),HR,BETAS,BETAL,BETASL,BETALL
             1        /BLQ/Q(341) /BLDOD/DOD(341)
0003          DOUBLE PRECISION  V,BETAS,BETAL,BETASL,BETALL,HR,Q,A0,
             1                  A1,A2,ER1,DOD,C1,C2,C3,ER
        C
0004          C1 = 2./(1.+HR)
0005          C2 = C1/HR
0006          C3 = 2./HR
0007          KV = 1
0008          OME = 1.5
0009        1 ER = 0.0
0010          EER = 0.0
0011          DO 8  I = 2, LXT1
0012          IF( I .LT. LXE .OR. I .GT. LXC ) GO TO 2
0013          IF( I .EQ. LXE ) GO TO 4
0014          IF( I .EQ. LXC ) GO TO 5
0015          A1 = DEXP(V(I)-Q(I)+BETALL)
0016          GO TO 3
0017        2 A1 = DEXP(V(I)-Q(I)+BETASL)
0018        3 A0 = V(I+1) + V(I-1) - 2.*V(I)
0019          A2 = A1 + 2.
0020          GO TO 7
0021        4 A0 = C1*V(I+1) + C2*V(I-1) - C3*V(I)
0022          A1 = DEXP(V(I)-Q(I)+BETALL)
0023          GO TO 6
0024        5 A0 = C2*V(I+1) + C1*V(I-1) - C3*V(I)
0025          A1 = DEXP(V(I)-Q(I)+BETALL)
0026        6 A2 = A1 + C3
0027        7 A0 = DOD(I) - A1 + A0
0028          A0 = OME*(A0/A2)
0029          V(I) = V(I) + A0
0030          B = DABS(A0/V(I))
0031          IF( B .GT. EER ) EER = B
0032        8 ER = ER + A0**2
        C
0033          IF( EER .LT. 0.01 .OR. KV .GE. 100 ) GO TO 10
0034          CALL   COME(KV,ER,ER1,OME)
0035          KV = KV + 1
0036          GO TO 1
        C
0037       10 WRITE( 6,11 )  KV, OME, EER
0038       11 FORMAT( ' ', 2X, 'KV =',I3,3X,'OME =',F5.2,3X,'EER =',
             1        1PE10.2 )
0039          RETURN
0040          END
```

TOTAL MEMORY REQUIREMENTS 000580 BYTES

```
0001          SUBROUTINE   COMPQ(LXT,LXE,LXC,LXT1,N)
        C
0002          COMMON  /BLQ/Q(341),HS,HL,EN,ALP,DC /BLV/V(341),HR
             1        /BLE/E(341) /BLRQ/QN(20),NN(20) /BLAUX/X(341),
             2        XX(341),Y(341)
0003          DOUBLE PRECISION  Q,HS,HL,HR,EN,DC,V,QN,XX,X,Y,A0,
             1                  A1,A2,A3,E,ALP,H,HS2,HL2,SUMH
        C
0004          HS2 = 2.*HS
0005          HL2 = 2.*HL
0006          SUMH = HL + HS
0007          A1 = HR / SUMH
0008          A2 = 1. / (HR*SUMH)
0009          A3 = (HL-HS) / (HS*HL)
0010          DO 1  I = 1, LXT
0011          IF( I .EQ. 1 ) A0 = (4.*V(I+1)-3.*V(I)-V(I+2))/HS2
0012          IF( I .EQ. LXT ) A0 = (3.*V(I)-4.*V(I-1)+V(I-2))/HS2
0013          IF( I .GT. 1 .AND. I .LT. LXE )A0=(V(I+1)-V(I-1))/HS2
0014          IF( I.GT.LXC.AND.I.LT.LXT ) A0 = (V(I+1)-V(I-1))/HS2
0015          IF( I.GT.LXE.AND.I.LT.LXC ) A0 = (V(I+1)-V(I-1))/HL2
0016          IF( I .EQ. LXE ) A0 = A1*V(I+1) - A2*V(I-1) + A3*V(I)
0017          IF( I .EQ. LXC ) A0 = A2*V(I+1) - A1*V(I-1) - A3*V(I)
0018        1 E(I) = A0*EN
0019          IF( Q(1) .EQ. Q(LXT) ) GO TO 100
0020          M = N - 1
0021          DO 11  I = 1, LXT
0022          A0 = E(I) / 7.77
0023          IF( A0 .LT. 1.0D-03 ) A1 = 1.0 - A0/2. + A0**2/6.
0024          IF( A0 .GE. 1.0D-03 ) A1 = (1. - DEXP(-A0)) / A0
0025       11 X(I) = -V(I) - DLOG(A1)
        C
0026          Y(LXT) = 0.0
0027          DO 3  I = 1, M
0028          K = N - I
0029          JS = NN(K)
0030          JL = NN(K+1)
0031          DO 2  J = JS, JL
0032        2 XX(J) = X(J) + QN(K)
0033          IF( K .EQ. 1 .OR. K .EQ. M ) H = HS
0034          IF( K .NE. 1 .AND. K .NE. M ) H = HL
0035          CALL   EXINT(H,JS,JL,XX,Y,Y(JL),LXT)
0036          IF( K .EQ. 1 ) GO TO 3
0037          IF( Y(JS) .EQ. 0.0 ) A0 = -1000.
0038          IF( Y(JS) .NE. 0.0 ) A0 = DLOG(Y(JS))
0039          Y(JS) = DEXP(QN(K-1)-QN(K)+A0)
0040        3 CONTINUE
        C
0041          DO 4  I = 1, LXT1
```

```
0042          AO = Y(I)
0043          IF( AO .EQ. 0.0 ) Y(I) = -1000.
0044          IF( AO .NE. 0.0 ) Y(I) = DLOG(AO)
0045        4 CONTINUE
        C
0046          AO = -Q(LXT) + Q(1) - Y(1)
0047          DO 6  I = 2, LXT1
0048          DO 5  J = 1, M
0049          IF( I .GT. NN(J) .AND. I .LE. NN(J+1) ) K = J
0050        5 CONTINUE
0051          Q(I) = DEXP(QN(K)-Q(LXT))
0052          Q(I) = Q(I) - DEXP(AO + Y(I))
0053          Q(I) = Q(I) + DEXP(-Y(1)+Y(I))
0054        6 Q(I) = QN(K) - DLOG(Q(I))
0055          DC = ( DEXP(-Y(1)) - DEXP( AO ) ) * 16. / (9.*ALP)
0056          RETURN
        C
0057      100 DC = 0.0
0058          RETURN
0059          END
```

TOTAL MEMORY REQUIREMENTS 000A98 BYTES

```
0001          SUBROUTINE   COMPV(LXT,LXT1,LXE,LXC,RMAX)
        C
0002          COMMON  /BLV/V(341),HR,BETAS,BETAL,BETASL,BETALL
             1        /BLQ/Q(341) /BLDOD/DOD(341) /BLAUX/X(341),
             2        XX(341),Y(341)
0003          DOUBLE PRECISION  V,HR,BETAS,BETAL,BETASL,BETALL,Q,DOD
             1                  ,X,XX,Y,C1,C2,C3,A0,ER,ER1
        C
0004          C1 = 2. / (1.+HR)
0005          C2 = C1 / HR
0006          C3 = 2./HR
0007          RMAX = 0.0
0008          DO 6  I = 2, LXT1
0009          IF( I .EQ. LXE ) GO TO 1
0010          IF( I .EQ. LXC ) GO TO 2
0011          A0 = V(I+1) + V(I-1) - 2.*V(I)
0012          IF( I .LT. LXE .OR. I .GT. LXC ) GO TO 4
0013          GO TO 3
0014        1 A0 = C1*V(I+1) + C2*V(I-1) - C3*V(I)
0015          GO TO 3
0016        2 A0 = C2*V(I+1) + C1*V(I-1) - C3*V(I)
0017        3 X(I) = DEXP(V(I)-Q(I)+BETALL)
0018          GO TO 5
0019        4 X(I) = DEXP(V(I)-Q(I)+BETASL)
0020        5 XX(I) = DOD(I) - X(I) + A0
0021          B = DABS(XX(I)) / DOD(I)
0022          IF( B .GT. RMAX ) RMAX = B
0023        6 CONTINUE
        C
0024          DO 10  I = 1, LXT
0025       10 Y(I) = 0.0
        C
0026          KD = 1
0027          OME = 1.5
0028       20 DMAX = 0.0
0029          ER = 0.0
0030          EER = 0.0
0031          DO 25 I = 2, LXT1
0032          IF( I .EQ. LXE ) GO TO 21
0033          IF( I .EQ. LXC ) GO TO 22
0034          A0 = Y(I+1) + Y(I-1)
0035          A1 = 2. + X(I)
0036          GO TO 24
0037       21 A0 = C1*Y(I+1) + C2*Y(I-1)
0038          GO TO 23
0039       22 A0 = C2*Y(I+1) + C1*Y(I-1)
0040       23 A1 = C3 + X(I)
0041       24 A0 = OME*(XX(I)-A1*Y(I)+A0) / A1
```

```
0042              Y(I) = Y(I) + AO
0043              B = DABS(AO)
0044              IF( B .GT. EER ) EER = B
0045              B = DABS(Y(I))
0046              IF( B .GT. DMAX ) DMAX = B
0047           25 ER = ER + AO**2
            C
0048              IF( EER/DMAX .LT. 1.0E-02 ) GO TO 30
0049              IF( KD .GE. 100 ) GO TO 30
0050              CALL   CCME(KD,ER,ER1,OME)
0051              KD = KD + 1
0052              GO TO 20
            C
0053           30 WRITE( 6,40 )  KD,OME,EER,DMAX,RMAX
0054           40 FORMAT( ' ',2X,'KD =',I3,3X,'OME =',F5.2,3X,'EER =',
                 1         1PE10.2,3X,'DMAX =',1PE10.2,3X,'RMAX =',
                 2         1PE10.2 )
0055              DO 50  I = 2, LXT1
0056           50 V(I) = V(I) + Y(I)
0057              RETURN
0058              END
```

TOTAL MEMORY REQUIREMENTS 000708 BYTES

```
0001          SUBROUTINE   EXINT(H,IS,IF,X,Y,YO,NDIM)
        C
        C THIS SUBROUTINE INTEGRATES EXP(X) FROM I, WHICH IS BETWEEN
        C IS AND IF, TO IF AND STORES (RESULT+YO) AT Y(I). H IS THE
        C MESH SIZE AND NDIM IS THE MAXIMUM DIMENSION OF X AND Y.
        C
0002          DOUBLE PRECISION  A,H,X,Y,AO,A1,A2,YO,H2I,HSQ2I,EFCT
0003          DIMENSION  X(NDIM), Y(NDIM)
0004          COMMON  /BLINT/A1,A2
0005          EXTERNAL   EFCT
        C
0006          H2I = 1.0 / (2.*H)
0007          HSQ2I = 1.0 / (2.*H**2)
0008          Y(IF) = YO
0009          LAST = IF - IS
0010          DO 1  II = 2, LAST
0011          I = IF + 1 - II
0012          AO = X(I)
0013          A1 = (X(I+1)-X(I-1))*H2I
0014          A2 = (X(I+1)+X(I-1)-2.*X(I))*HSQ2I
0015          CALL   DQG32(0.0,H,EFCT,A)
0016        1 Y(I) = Y(I+1) + DEXP(AO+DLOG(A))
0017          CALL   DQG32(-H,0.0,EFCT,A)
0018          Y(IS) = Y(IS+1) + DEXP(AO+DLOG(A))
0019          RETURN
0020          END
```

TOTAL MEMORY REQUIREMENTS 0003A8 BYTES

FORTR

000

000 TOT

000
000
000
000
000
000
000
001
001

```
0001          FUNCTION   EFCT(Z)
C
0002          DOUBLE PRECISION  A1,A2,Z,EFCT
0003          COMMON  /BLINT/A1,A2
0004          EFCT = DEXP(A1*Z + A2*Z**2)
0005          RETURN
0006          END
```

```
0001          FUNCTION   EFCT(Z)
        C
0002          DOUBLE PRECISION  A1,A2,Z,EFCT
0003          COMMON  /BLINT/A1,A2
0004          EFCT = DEXP(A1*Z + A2*Z**2)
0005          RETURN
0006          END
```

TOTAL MEMORY REQUIREMENTS 000166 BYTES

```
0001          SUBROUTINE   COME(KDV,ER,ERP,OME)
        C
0002          DOUBLE PRECISION  A,ERP,ER
        C
0003          IF( KDV .GT. 16 ) GC TO 2
0004          A = DSQRT(ER)
0005          IF( KDV .LE. 15 ) GO TO 1
0006          AA = A/ERP
0007          BB = (AA+OME-1.)/(OME*SQRT(AA))
0008          OME = 2./(1.+SQRT(1.-BB**2))
0009        1 ERP = A
0010        2 RETURN
0011          END
```

TOTAL MEMORY REQUIREMENTS 00024C BYTES