Power Series V-I Curve Element
and Optimization Improvements in the
NET-2 Network Analysis Program.

Final Report...
This report was prepared for the Harry Diamond Laboratories, Adelphi, Maryland, by The BDM Corporation under Contract DAAG39-75-C-0158. This document constitutes the final technical report on two tasks related to improvements in the NET-2 Network Analysis Program.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>ii</td>
</tr>
<tr>
<td>1. NONLINEAR RESISTOR MODEL</td>
<td>1</td>
</tr>
<tr>
<td>2. OPTIMIZATION IMPROVEMENTS</td>
<td>4</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>A-1</td>
</tr>
</tbody>
</table>
1. NONLINEAR RESISTOR MODEL

A model for a nonlinear resistor has been developed and implemented in NET-2. The nonlinear resistor has actually been implemented as an algebraic power series in which element voltage is a function of element current. The power series function is defined for non-negative values of current and reflected into the other half plane (non-positive current) as a negative function.

Details of the power series element are given on the following page. The FORTRAN listing of subroutine PSVI, the power series element routine, is given in Appendix A.
Power Series V-I Curve

The PSVI element is a circuit element which may be used to represent a voltage-current characteristic curve in which the voltage across the element is represented by a power series expansion of the current flowing through the element. The voltage across the element is related to the current flowing through the element by the expression:

\[ V = \sum_{i=1}^{n} K_i |I|^i \]

where:
- \( V \) = element voltage
- \( I \) = element current
- \( K_1, K_2, \ldots, K_n \) = power series coefficients
- \( c \) = sign of \( I \)

From the above equation it is seen that the element voltage for negative values of current is the negative of the element voltage for positive values of current, i.e., if \( V = f(I) \) then \( V = -f(-I) \) for \( I > 0 \). This means that the PSVI element behaves exactly like a nonlinear resistor whose resistance is given by:

\[ R = \frac{V}{I} = \sum_{i=1}^{n} K_i |I|^{i-1} \]

The format for the element is:

PSVIn (p) a b K1, K2, ..., Kn

where:
- \( p \) = optional parallel parts designation
- \( a \) and \( b \) = node pair names
- \( K_1, K_2, \ldots, K_n \) = power series coefficients

The power series coefficients may be represented by numbers or mathematical expressions. The element does not contain any intrinsic computational delay.
The voltage across the element is given by

\[ V = e_a - e_b \]

where \( e_a \) and \( e_b \) are the voltages at nodes a and b, respectively. The current flow through the element is given by

\[ I = \frac{V}{R} \]

where \( R \) is the equivalent nonlinear resistance of the element. Element power dissipation \( P \) is given by

\[ P = IV \]

The user may reference the power series coefficients \( K_1, K_2, K_3, \) etc., by using the respective symbolic names \( PSVIN, PSVIN.1, PSVIN.2, \) etc. The element voltage, current and power dissipation may be referenced by using the symbolic names \( V(PSVIN), I(PSVIN), \) and \( P(PSVIN), \) respectively.
2. OPTIMIZATION IMPROVEMENTS

Changes have been made to the optimization routine in NET-2 to permit the user to obtain output information from a partially completed optimization solution when a computer time limit has been reached. The results printed are the best optimized values which have been found up to the moment when the time limit is reached. These output values include the objective function value, the values of the free parameters, and the values of parameters specified by constraint relations.

This output information is identical to that which is provided upon normal completion of the optimization solution. The optimization results are provided whether or not a checkpoint/restart record is written by NET-2.
APPENDIX A

FORTRAN Listing of Subroutine PSVI

SUBROUTINE PSVI
COMMON/Q(1)
COMMON/P/LOC(120), IXP(297)
COMMON/C277/L, LD, N, I, IX7700(3)
COMMON/C277R/P, RJZ, V, RJ, D, S, RM, B, X7700(22)
COMMON/C278/1ALPHA, V1, IBETA
COMMON/C260/IRUN, X6D00(17)
COMMON/C273/Y, Y1, ICP, J, K, IX7301
COMMON/C200I/IX200, LAM, LAM1, LDYN, IX201(6)
COMMON/C2LOC/LGAM, IXLOC1(2), LOCN, IXLOC2(4), LOCV, IXLOC3(2), LOCB,
1 IXLOC4(20)
COMMON/C26F/M, IX6F00(12)
COMMON/C26D00/IX6D0(6), ICALC, IX6E00(3)
COMMON/C26E/IConv, IX6E00(2)
COMMON/C10800/XTD0(9), TMD016, XTD02
DIMENSION ITAB(3)
DATA ITAB/1, 2, 3/
EQUIVALENCE (118), (119)
CALL ENTRY PSVI
C *IF(NOT RESPONSE VARIABLE CALCULATION) THEN
IF(IRUN.EQ.8)GO TO 8000
C *DO FOR ALL PSVI ELEMENTS
500  L=Q(JG118+LAM)
     L=L+JG119
C *INITIAL SETUP
LD=LDYN+TMD016*(Q(L)-118)
     J=Q(L+1)
     J=Q(LOCN+J)
     K=Q(L+2)
     K=Q(LOCN+K)
     N=Q(L+3)
C *CASE(IRUN)
      GO TO(1000, 9000, 3000, 9000, 5000, 6000, 9000, 9000, 6500), IRUN
C *IRUN=1, STORE B AND H TERMS
1000  P=1,
     IF(AND(Q(L), 10B), NE, 0)P=Q(L+N+4)
     RJZ=ARS(Q(LD))
     RM=N*Q(L+N+3)
1300  IF(N, EQ, 1)GO TO 1400
     N=N+1
     RM=RM*RJZ+N*Q(L+N+3)
     GO TO 1300
1400  RM=ABS(RM)+2.
     Y=P/RM
C *STORE H TERMS
     CALL STORY
     W(LD+2)=RM

A-1
C *IF(TIME_DOMAIN) THEN
  IF(ICALC,NE,0)GO TO 7000
C *STORE B TERMS
  B=Y*Q(LD+1)-P*Q(LD)
  IF(Q(LGAM+J),GT,0.)Q(LOCB+J)=Q(LOCB+J)+B
  IF(Q(LGAM+K),GT,0.)Q(LOCB+K)=Q(LOCB+K)-B
  GO TO 7000
C *IRUN=3, INITIALIZE DC
3000  Q(LD)=0.
5000  Q(LD+1)=0.
  GO TO 7000
C *IRUN=5, CONVERGENCE CHECK
5000  V1=0(LDCV+J)-Q(LOCB+K)
  RJ=Q(LD)+(V1-Q(LD+1))/Q(LD+2)
C *CALCULATE V = F(J)
  v=0
  s=SIGN(1.,RJ)
  RJZ=ABS(RJ)
  I=N
5100  IF(I.EQ,0)GO TO 5200
  V=(V+Q(L+I+3))*RJZ
  I=I-1
  GO TO 5100
5200  V=S*V
  IF(ABS(V1),LT,1E-10.AND.,ABS(V),LT,1E-10)GO TO 5400
  D=ABS(V1+V)
  IF(S.EQ,0)GO TO 5300
  IF(ABS(V1-V)/D.LE.,001)GO TO 5400
5300  ICONV=0
5400  Q(LD)=RJ
  Q(LD+1)=V
  GO TO 7000
C *IRUN=6, SPARSE MATRIX SETUP
6000  CALL SKELE(ITAB(1),2)
  GO TO 7000
C *IRUN=9, NODAL CONNECTIVITY
6500  IF(M,NE,0)RETURN
  CALL NODCON(ITAB(1),2,0)
  GO TO 7000
7000  IF(LAM,EQ,LAM1)GO TO 9000
  LAM=LAM+1
  GO TO 500
C *ELSE
C *CALCULATE RESPONSE VARIABLES
8000  V1=Q(LD+1)
  IF(IALPHA,NE,10)GO TO 9000
  IF(IALPHA,NE,12)V1=V1*Q(LD)
  IF(IALPHA,NE,11)V1=Q(LD)
9000  RETURN
END