A GUIDE TO THE PROGRAMMING OF NONLINEAR DEVICE MODELS FOR INCLUSION IN McCAE'S BUILT-IN MODEL LIBRARY

J.W. Bandler, Q.J. Zhang and R.M. Biernacki

SOS-90-5-R

February 1990

A GUIDE TO THE PROGRAMMING OF NONLINEAR DEVICE MODELS FOR INCLUSION IN McCAE'S BUILT-IN MODEL LIBRARY

J.W. Bandler, Q.J. Zhang and R.M. Biernacki

Abstract This report is a guide for users of McCAE who want to program their own proprietary nonlinear device models for inclusion in McCAE's built-in model library. We describe the Fortran subroutines related to device models that users must write. These subroutines can be supplied in the form of object code to McCAE authors who will then incorporate them into the overall McCAE system.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants STR0040923, OGP0007239 and OGP0042444 and in part by Optimization Systems Associates Inc.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7, and with Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7.

I. INTRODUCTION

McCAE is a linear/nonlinear microwave circuit simulator for small- and large-signal circuit design and device modeling [1]. As a user of McCAE, you may desire to include your own proprietary device models into McCAE's built-in model library. In this report, we describe the McCAE requirements for programming a nonlinear model in Fortran. Following the instructions in the report, you can create subroutines to set up the model parameters and to implement your model equations. These subroutines can be supplied to McCAE authors in the form of object code. The final integration of your model with the McCAE system will be completed by McCAE authors.

II. FORMULATION OF A NONLINEAR DEVICE MODEL

The device model should contain only the nonlinear part of the device. For example, a FET model contains an extrinsic part and an intrinsic part. The extrinsic part contains linear parasitic elements. The intrinsic part is the nonlinear part of the FET model. Therefore, when creating a nonlinear FET model, only the intrinsic part of the FET should be included. The extrinsic part should be excluded from the nonlinear model and defined separately in McCAE's circuit input file using linear elements. Violation of this arrangement will still produce a correct solution but at the cost of significantly increased computational effort.

Consider a generic representation of a nonlinear device model as shown in Fig. 1. There are n + 1 nodes in the nonlinear model, i.e., nodes 0, 1, 2, ..., n.

Each node in the model may or may not be directly connected to other circuit elements outside the model. A node is called a dangling node if it is not directly connected to other circuit elements.

Unless otherwise explicitly stated in this report, a node always refers to a node in the nonlinear model rather than a node in the overall circuit.

Reference Node

Any node in the model which is not a dangling node can be designated as a reference node. We use node 0 as the reference node as shown in Fig. 1. The model can be considered as an n-port device. Port i is between node i and the reference node.

When a device model is connected to other elements in the overall circuit, the reference node of the model is not necessarily the ground of the overall circuit. In a multi-device circuit, each model has its own reference node.

The Sequence of Nodes

Among all nodes which are not a reference node in the model, dangling nodes should be arranged last. Suppose the number of dangling nodes is n_d, $0 \leq n_d < n$. The dangling nodes should be nodes $n-n_d+1$, $n-n_d+2$, ..., n.

Form of Model Equations

Suppose i is the node index, i = 1, 2, ..., n. Let V_i be the time domain voltage between node i and the reference node. Let Q_i be the total charge in the time domain at node i contributed by the model. This total charge is the sum of the charges of all capacitive elements (in the nonlinear model) which are directly connected to node i. Let I_i be total current in the time domain flowing from node i into the model (see Fig. 1). This total current is the sum of the currents flowing through all resistive elements and controlled current sources (in the nonlinear model) which are directly connected to node i. Let **V** be an n-vector containing voltages V_i, i = 1, 2, ..., n. Let **I** and **Q** be n-vectors containing I_i, i = 1, 2, ..., n and Q_i, i = 1, 2, ..., n, respectively. Let **P** be a vector containing all parameters of the model.

The McCAE simulation driver supplies values of **V** to the model subroutine and expects to receive values of **I** and **Q** in return. Therefore, the mathematical task for programming a nonlinear model is as follows.

Problem: Given **V** and **P**, compute **I = I(V, P)** and **Q = Q(V, P)**.

It is possible that you have a nonlinear capacitance function C(V, **P**) to represent a nonlinear capacitor, where C is the capacitance and V is the voltage across the capacitor. In this case, the capacitance function should be converted into a charge function Q(V, **P**). This can be done by integrating C(y, **P**) over the interval $0 \leq y \leq V$, i.e,

$$Q(V, \mathbf{P}) = \int_0^V C(y, \mathbf{P})\, dy$$

The integration should satisfy the boundary condition Q(0, **P**) = 0. If the capacitance function C(V, **P**) is a piecewise function, the charge function Q(V, **P**) is also piecewise and must be continuous at all the switch conditions. This continuity condition can be satisfied by adding different constant terms (constant w.r.t. V) to different pieces of the piecewise function Q(V, **P**).

The Sequence of Model Parameters

Suppose the total number of parameters including time delay parameters in the model is n_p. Suppose the number of time delay parameters in the model is n_{tau}. We require that the time delay parameters be arranged before other parameters in the model. Therefore, the jth parameter is a time delay parameter if $1 \leq j \leq n_{tau}$.

III. FORTRAN SUBROUTINES FOR A DEVICE MODEL

Source Code Template

```
      SUBROUTINE user_model(MT,NT,V,VTAU,A)

C     compute the nonlinear currents and charges of the model

C     MT :    an integer to define the first dimension of matrices
C             V(*, *), VTAU(*, *) and A(*, *).  MT is an input argument.
C     NT :    number of time samples (input argument).
C     V :     time domain voltages (input argument).
C     VTAU :  delayed time domain voltages (input argument).
C     A :     time domain currents or charges (output argument).
```

```
C       You should substitute n and n_tau in the next line by the actual
C       number of nodes (excluding the reference node) and the number of
C       time delay parameters in your model.

        REAL  V(MT, n), VTAU(MT, n_tau), A(MT, n)

C       The following common block includes all model parameters of
C       your model.  Here we use an example of 7 arbitrary parameters.

        COMMON /user_modelK/TAU_A, TAU_B, A1, A2, A3, GAMMA, BETA
        REAL                TAU_A, TAU_B, A1, A2, A3, GAMMA, BETA

C       Local variables for the model

        INTEGER L

        DO 40 L=1,NT

           ........

C          Compute the sum of currents at each node contributed from
C          all resistive elements and controlled current sources
C          within the model which are directly connected to the node.

           A(L,1) =
           A(L,2) =
           ......

           A(L,n) =

C          Compute the sum of charges at each node contributed from
C          all capacitive elements within the model which are directly
C          connected to the node.

           A(L,n+1) =
           A(L,n+2) =
           ......

           A(L,n+n) =
 40     CONTINUE
        RETURN
        END


        SUBROUTINE user_modelA(P, NP)

C       Assign values to model parameters
C       This is an example of 7 arbitrary parameters

C       NP :  number of parameters in the model (input argument)
C       P :   contains values of all parameters for this model (input
```

```
C             argument)

      INTEGER NP
      REAL P(NP)
      COMMON /user_modelK/TAU_A, TAU_B, A1, A2, A3, GAMMA, BETA
      REAL                TAU_A, TAU_B, A1, A2, A3, GAMMA, BETA

      TAU_A = P(1)
      TAU_B = P(2)
      A1    = P(3)
      A2    = P(4)
      A3    = P(5)
      GAMMA = P(6)
      BETA  = P(7)
      RETURN
      END
```

Description of the Variables

We describe here the notation used in the source code template. Suppose l is the index for time samples, l = 1, 2, ..., NT, where NT is the total number of time samples. Suppose i is the index for nodes in the model, i = 1, 2, ..., n. Suppose j is the index for time delay parameters in the model, j = 1, 2, ..., n_{tau}. V(l, i) contains the voltage between node i and the reference node at time point l. VTAU(l, j) contains the voltage delayed from the lth time point by the jth time delay parameter. VTAU(*, j) can be the voltage across any predetermined node pair in the model. Such a node pair may include the reference node. A(l, i) contains the total current flowing from node i into the model at time sample l. A(l, i+n) contains the total charge at node i contributed by the model at time point l. Both arrays V(*, *) and VTAU(*, *) are computed by McCAE before calling the "user_model" subroutine. McCAE also initializes or updates all model parameters in the common block "user_modelK" (i.e., TAU_A, TAU_B, A1, A2, A3, GAMMA and BETA in the above example of source code listing) by calling routine "user_modelA" before entering routine "user_model".

MT, NT, V and VTAU are all input arguments of the subroutine "user_model". Their contents should not be altered in the subroutine.

The task of subroutine "user_model" is to compute A(*, *) from the given quantities in V(*, *), VTAU(*, *) and model parameters, e.g., TAU_A, TAU_B, A1, A2, A3, GAMMA and BETA.

The symbols n and n_{tau} in the source code template should be substituted by the actual number of nodes excluding the reference node and the actual number of time delay parameters.

The task of subroutine "user_modelA" is simply to initialize or update the model parameters from given values in P(*). The sequence of the parameters is such that time delay parameters (e.g., TAU_A and TAU_B) are ranked before other parameters (e.g., A1, A2, A3, GAMMA and BETA). NP and P are input arguments of the subroutine "user_modelA". Their contents should not be altered in the routine.

The common block "user_modelK" is not used elsewhere in McCAE except in your subroutines "user_model", "user_modelA" and any additional intermediate subroutines that you may write.

Restrictions of Notation in Your Program

Subroutine names "user_model", "user_modelA" and the common block name "user_modelK" can be arbitrary as long as they do not coincide with existing subroutine names and common block names in McCAE. Consequently, it is required that you contact McCAE authors to finalize the word "user_model". The name for the second subroutine listed above (i.e., "user_modelA") should be formed by appending character "A" to the finalized word "user_model". It is also required that the names of any additional subroutines and common blocks you create be formed by appending an arbitrary letter to the end of the word "user_model", e.g., "user_modelB", "user_modelD", etc.

The notation of variables in the subroutine "user_model" (e.g., NT, V, VTAU, A) and the notation of model parameters in the common block "user_modelK" (e.g., TAU_A, TAU_B, A1, A2, A3, GAMMA and BETA) can be arbitrary.

IV. SUMMARY OF INFORMATION NEEDED BY McCAE

In order to include your model in McCAE's built-in model library, McCAE authors require at least the following information from you.

1. The word "user_model"
2. The number of nodes in the model excluding the reference node (i.e., n).
3. The number of dangling nodes in the model (i.e., n_d).
4. A list of all parameters of the model in exactly the same sequence as in the subroutine "user_modelA" (e.g., TAU_A, TAU_B, A1, A2, A3, GAMMA and BETA). Verbal definitions and the default value of each parameter are also desirable.
5. The number of time delay parameters in the model (i.e., n_{tau}).
6. For each time delay parameter, indicate the pair of nodes across which the voltage should be delayed.
7. Subroutines "user_model" and "user_modelA". They may be in the form of object code.

V. CONCLUSIONS

This report provides instructions for you, a McCAE user to program your own proprietary nonlinear device models for inclusion in McCAE's built-in model library. In addition to the two standard subroutines "user_model" and "user_modelA" illustrated in the report, you can create as many intermediate subroutines as necessary to implement a complicated model with large equations. These subroutines can be supplied to McCAE authors in the form of object code. The final integration of your model with the McCAE system will be completed by McCAE authors. Your model will enjoy all the small- and large-signal simulation and optimization features of McCAE.

REFERENCE

[1] *McCAE*, Simulation Optimization Systems Research Laboratory and Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7, 1989.

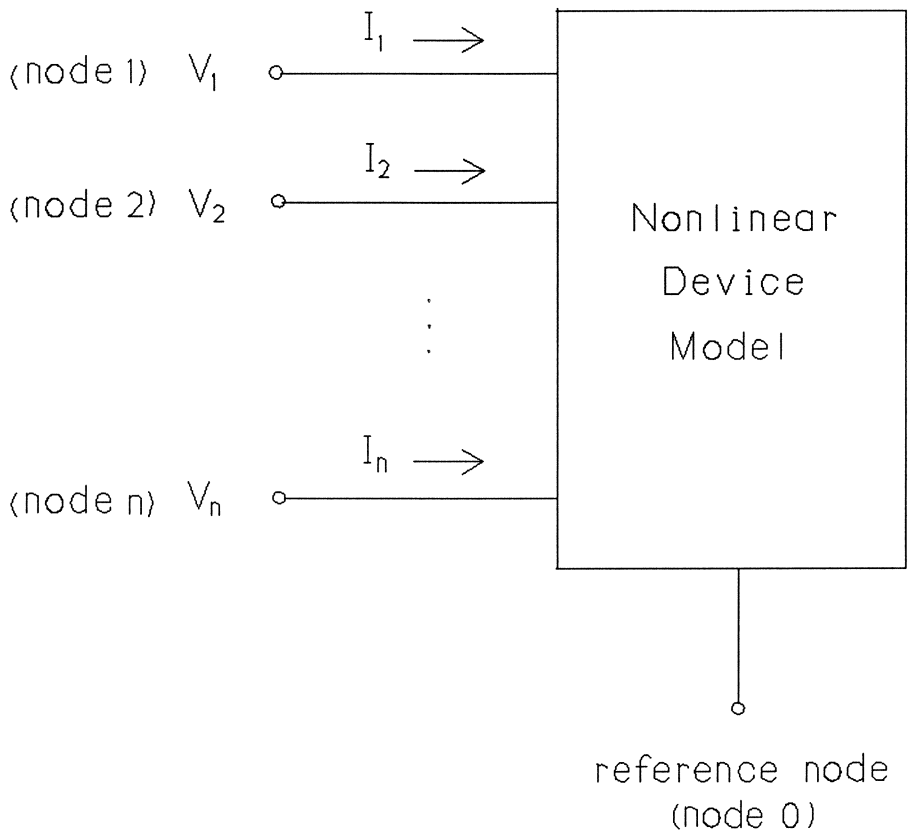

Fig. 1. A generic representation of a nonlinear device model. There are n + 1 nodes in the model. Node 0 is the reference node. The model can be considered an n-port device where port i is between node i and the reference node, i = 1, 2, ..., n. V_i is the voltage between node i and the reference node. I_i is the total current flowing out of node i into various branches in the nonlinear model.