# SCREAMER - A SINGLE-LINE PULSED-POWER DESIGN TOOL

M. L. Kiefer and M. M. Widner
Pulsed Power Theory Division 1265
Sandia National Laboratories
Albuquerque, New Mexico 87185

## Abstract

SCREAMER is a special purpose circuit code developed as a design tool for single module accelerators. It is fast, accurate, flexible, and user-friendly. Its development was motivated by the excessive costs and long turn-around times incurred when using the SCEPTRE [1] circuit analysis code to perform simulations of circuits with large numbers of nodes and with nonlinear components. Comparable simulations between SCREAMER running on a VAX 11/780 and SCEPTRE running on a CRAY-1S show that turn-around times and costs can be two orders of magnitude lower when using SCREAMER.

## The Computer Model

The SCREAMER code solves energy transmission problems in circuit elements connected in series. The numerical differencing scheme has one time level, is fully implicit, and is second-order accurate. It has the capability to handle nonlinear circuit components by selecting from various models which have been placed into the code or by using a FORTRAN subroutine written by the user. The circuit topology must be that of a main branch with secondary branches exitting, but not rejoining, the main branch. Each branch consists of circuit elements connected in series.

The basic circuit element is shown in Fig. 1 as part of a general π-section. It consists of a shunt

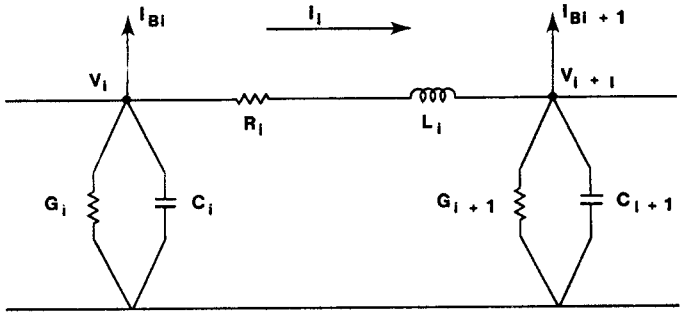


Fig. 1. General circuit element with voltage and current definitions.

resistor, a shunt capacitor, a series resistor, and a series inductor all associated with a single circuit node. All user specified designs are mapped to groups of this element connected in series. Fig. 2 shows the types of secondary branch connections allowed in SCREAMER. An end branch can be connected to any main branch node and a top branch to any pair of adjacent main branch nodes.

Fig. 1 shows how the voltage, $V_i$, and current, $I_i$, at each node i are defined. The quantity $I_{Bi}$ represents the current flowing into a secondary branch, if one is present at that node. For each node i in each branch, SCREAMER's algorithm calculates the

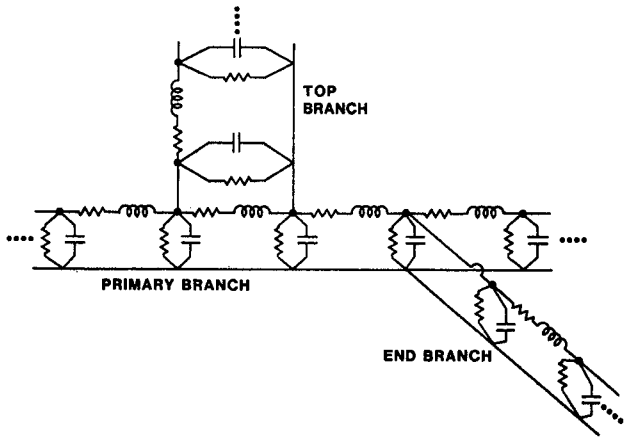


Fig. 2. SCREAMER branch types.

voltage drop between nodes and sets the sum of the currents entering a node to zero as shown in the following equations.

$$V_i - V_{i+1} = R_i I_i + \frac{\partial}{\partial t} L_i I_i$$

$$I_{i-1} - I_i - I_{Bi} = G_i V_i + \frac{\partial}{\partial t} C_i V_i$$

The finite difference representation is given by the following equations. These equations are substituted into the voltage and current equations above to give us the complete finite difference equations. Two time levels are considered: the new time (superscript n), for which the voltages and currents are not known, and the old time (superscript o), for which the voltages and currents are known. By evaluating the equations at the half time step, second order accuracy is obtained.

$$V = \frac{V^n + V^o}{2}$$

$$I = \frac{I^n + I^o}{2}$$

$$\frac{\partial CV}{\partial t} = C \frac{V^n - V^o}{\Delta t} \quad \text{(where C is constant)}$$

$$\frac{\partial LI}{\partial t} = L \frac{I^n - I^o}{\Delta t} \quad \text{(where L is constant)}$$

At time t=0, all initial component values, currents, and voltages are specified. Generally, the user only needs to input a small portion of the component values and to identify a few current and voltage sources to initialize the circuit. Since adjacent elements are coupled, the finite difference representation of the current and voltage equations yields a system of linear equations to be solved simultaneously in order to obtain the voltages and currents at the new time.

In general, R, G, C, and L may be nonlinear components. The user can specify a nonlinear component by choosing one from a SCREAMER model library or by defining one with a FORTRAN subroutine. Typically, SCEPTRE models are added to the SCREAMER library by making a few, minor changes. SCREAMER uses the following procedures for accomodating nonlinear components. Series or shunt resistors may be functions of time, voltage, and current. The value of R or G at the half time step, $(t^n+t^o)/2$, is then used directly in the voltage and current finite difference equations at the appropriate node. The treatment of nonlinear inductors and capacitors is more restrictive. The values of $\partial LI/\partial I$ and $\partial CV/\partial V$ at the half time step are substituted in the finite difference equations for L and C, respectively. With this procedure, L may only be a function of the current passing through it and C only a function of the voltage across it. Currently, SCREAMER contains a generic gas switch model describing shunt or series resistors [2], an active diode model describing shunt resistors [3], a plasma opening switch model describing shunt resistors [4], and a magnetic switch model of a saturable core inductor describing series inductor values [5].

For a single line, the procedure for solving for all $V_i^n$ and $I_i^n$ is straightforward and one simply inverts a block tridiagonal matrix. Note that the number of calculations in this procedure scales linearly in the number of nodes, while a full matrix solution scales as the square of the number of nodes. Complications arise when secondary branches are included because this leads to a few off-diagonal rows and columns. SCREAMER's matrix inversion algorithm is tailored to this sparse, nearly tridiagonal matrix. This results in an efficient algorithm which still scales approximately linearly in the number of nodes. This allows simulations with large numbers of nodes to be performed without an excessive amount of CPU time being used.

## Energy Check

SCREAMER implements an internal energy check on the circuit as an aid to the user in checking the accuracy of the numerical procedure. The total amount of energy input to the circuit by all energy sources is calculated at each time step. Then the amount of energy dissipated in all the circuit resistors is calculated and added to the energy stored in all the circuit capacitors and inductors. The comparison of this amount to the total source input energy gives the energy check. This energy checking also handles nonlinear inductor and capacitor effects by keeping track of the energy used to change the inductance or capacitance.

## Defining a SCREAMER Problem

To define a problem, main and secondary branches must be identified as shown in Fig. 2. These branches are simply series arrangements of circuit elements as shown in Fig. 1. To accomplish this, the user specifies predefined circuit blocks connected in series. This is a compact method of identifying circuit parameters and since there is no node nor circuit component labeling as with the SCEPTRE code, the setup is also very fast. Eleven types of circuit blocks are available in SCREAMER.

Three types of transmission line blocks may be used. The first is the lossless transmission line block. This consists of $\pi$-section segments with no resistance connected in series. To characterize a line, the user specifies a delay time, the input and output impedances, and a resolution time. Based on this resolution time parameter, SCREAMER will break the line into segments with delay times of approximately the resolution time. The impedance may vary over the line in a linear or exponential fashion and SCREAMER uses the interpolated impedance and the fractional time delay when calculating an individual $\pi$-section segment's L and C values. Lossy transmission lines are achieved with two types of magnetically insulated transmission line blocks. The first is a racetrack-type MITL based on geometrical specifications [6,7]. This also uses $\pi$-section segments, but adds nonlinear shunt resistors to simulate the losses. The user specifies the time delay, impedance, gap, circumference, and the resolution time of the line. It is assumed that the gap and circumference are constant and that the gap is much less than the circumference. The second MITL block is based on a perveance description and uses no geometrical parameters at the expense of a preliminary calculation by the user of the line's perveance [8]. This type of MITL is constructed with T-sections with nonlinear shunt resistors. The user specifies the time delay, impedance, perveance, and resolution time.

There are four circuit blocks based on the general $\pi$-section shown in Fig. 1. The $\pi$-section circuit block is identical to that in Fig. 1 and the user specifies the component values of the two shunt resistors, the two shunt capacitors, and the series resistor and inductor. The next three blocks are subsets of the $\pi$-section block and are added for convenience. The first is the RC-to-ground block which uses only the first shunt resistor and capacitor portion of the $\pi$-section block while setting the series resistor and inductor as a short circuit. The second is the RL-series block which uses only the series resistor and inductor portion while setting the shunt portions to open circuits. The third is the adder block which sets the shunt and series portions to open circuits. This block is useful for connecting a top branch to the main branch such that the main and top branch voltages are added.

There are four source blocks available. The first two are the voltage-source and current-source blocks and these may only occur as the first block in the main branch. The other two are the end-of-branch-voltage-source and the end-of-branch-current-source blocks and these may only occur as the last block in any branch. Voltage and current waveforms in time are specified by performing a linear interpolation on a table of values, by using a sin-squared function, or by using a polynomial function. By changing a time delay or scaling parameter, these waveforms may be arbitrarily scaled in amplitude or delayed in time. In addition to these source blocks, energy may be placed in the circuit by setting an initial voltage on a capacitor or a lossless transmission line or by placing an initial current in an inductor or a lossless transmission line.

## Circuit Parameter Output

There are four forms of output the user may request for examining a circuit parameter's behavior in time. The first form is the PLOT output request which generates a plot. The second form is the PRINT output request which gives a tabular line printer listing. The third form is the FILE output request which gives an ASCII disk file of the time-parameter coordinates. This is useful for whatever type of post-processing the user may wish to perform. The

fourth form is the TABLE output request which also creates an ASCII disk file of the coordinates. This file is in the form of a SCREAMER compatible source table which may be pasted directly into another SCREAMER deck. This could be used to generate a voltage versus time table from one simulation for use in another simulation as a voltage source specification. Since there is no node or circuit component labeling, the parameters to be output are selected from a large pool of preset parameters using descriptive and easy to remember keywords which are then linked to a specific circuit block in the simulation. For example, the keyword POUT refers to the power flowing out of a specified circuit block.

## Setup Parameters

Before a SCREAMER problem can begin execution, the user sets a few initial parameters. These are the iteration time step, the total simulation time, the default transmission line resolution time, and the number of times to print out the complete status of the circuit during the simulation. Also, the user can specify whether all or just one of the time iteration cycles should be performed. This is useful in checking an initial run of a complicated setup before performing the full simulation. In addition, the user has a choice of whether or not to place grids on the plots, to provide a detailed echo of SCREAMER's arrays, and the maximum number of points to be stored for each PLOT, PRINT, or FILE form of output. This is useful if the user has a slow graphics device and needs to limit the amount of points to speed up output.

## Comparisons with SCEPTRE

Figs. 3 and 4 show comparisons between SCREAMER and SCEPTRE for the DEMON [9] and COMET [10] accelerator modules at Sandia National Laboratories.

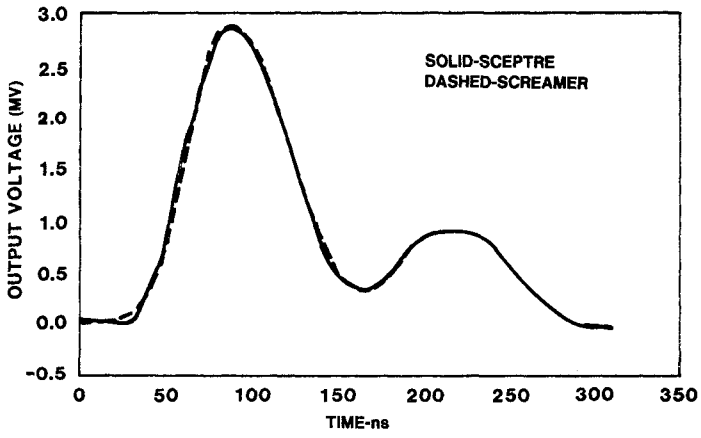


Fig. 3. Comparison between SCREAMER and SCEPTRE for the DEMON module simulation.

The agreement is reasonable. For the DEMON simulation, SCREAMER used 2.5 minutes of CPU time on a VAX 11/780 at a cost of $1 with about 5 minute turn-around. SCEPTRE used 30 minutes of CPU time on a CRAY-1S at a cost of $500 with 1-2 day turn-around. For the COMET simulation, SCREAMER used 5 minutes of VAX CPU time at a cost of $2 with about 10 minute turn-around. SCEPTRE used 42 minutes of CRAY CPU time at a cost of $700 with 1-2 day turn-around. By using a smaller time step in SCREAMER, the SCEPTRE results of Fig. 4 were exactly reproduced. This run required

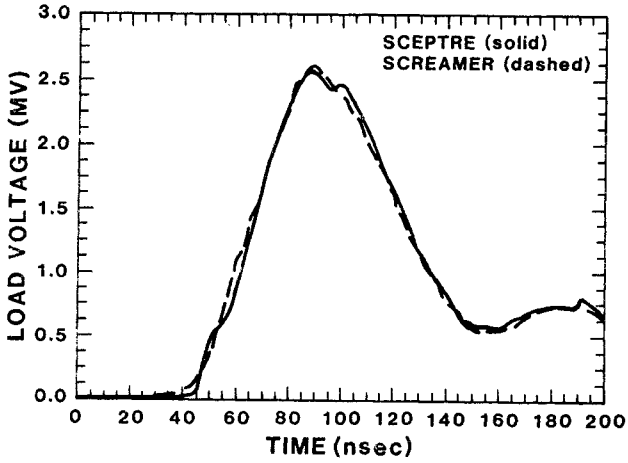


Fig. 4. Comparison between SCREAMER and SCEPTRE for the COMET module simulation.

65 VAX CPU minutes. Such a comparison was attempted with a PBFA-II Convolute module simulation. SCREAMER used 17 VAX CPU minutes at a cost of $6. SCEPTRE was unable to finish the problem in a reasonable amount of time, but it was estimated substantially more than one hour of CRAY CPU time would have been used at a cost of well over $1000 with a turn-around time of one week. The reasons for these large differences are due to SCEPTRE's inability to implement sparse matrix techniques, its complicated treatment of nonlinear elements, and its use of a variable time step. Generally, SCREAMER has been found to have turn-around times and costs of 1 to 2 orders of magnitude lower than SCEPTRE on comparable pulsed power simulations, in spite of the fact that SCREAMER runs on a VAX computer while SECPTRE runs on a CRAY computer.

## References


[1] J. C. Bowers and S. R. Sedore, SCEPTRE: A Computer Program for Circuit and System Analysis. Englewood Cliffs, NJ: Prentice Hall, 1971.

[2] E. L. Neau, Sandia National Laboratories (private communications).

[3] S. A. Slutz, Sandia National Laboratories (private communications).

[4] C. W. Mendel and D. H. McDaniel, Sandia National Laboratories (private communications).

[5] M. Stockton, E. L. Neau, and J. P. VanDevender, "Pulsed Power Switching Using Saturable Core Inductors," J. Appl. Phys. 53, 2756 (1982).

[6] K. D. Bergeron and J. W. Poukey, Appl. Phys. Lett. 27, 58 (1975).

[7] M. M. Widner and G. R. Montry, "Simulations of Single-Line Magnetically Insulated Power Flow," Particle Beam Fusion Progress Report, January 1984 Through June 1984, SAND84-1336 (Albuquerque, NM: Sandia National Laboratories, 1985).

[8] L. X. Schneider and D. H. McDaniel, Sandia National Laboratories (private communications).

[9] B. N. Turman, E. L. Neau, J. F. Seamen, and D. Green, "Pulse-Compression Testing for PBFA II," Particle Beam Fusion Progress Report, January 1984 Through June 1984, SAND84-1336 (Albuquerque, NM: Sandia National Laboratories, 1985).

[10] E. L. Neau, T. L. Woolston, K. J. Penn, B. R. Sujka, and W. D. Helgeson, "Recent Results From High-Power Magnetic Switching," Particle Beam Fusion Progress Report, January 1984 Through June 1984, SAND84-1336 (Albuquerque, NM: Sandia National Laboratories, 1985).

This work performed at Sandia National Laboratories supported by the U. S. Department of Energy under Contract No. DE-AC04-76DP00789.