Adjoint Transient Sensitivity Computation in Piecewise Linear Simulation

Tuyen V. Nguyen*, Anirudh Devgan*, and Ognen J. Nastov**

*IBM Austin Research Laboratory, Austin, TX 78758

**Dept. of EE & CS, MIT, Cambridge, MA 02139

Abstract

This paper presents a general method for computing transient sensitivities using the adjoint method in event driven simulation algorithms that employ piecewise linear device models. Sensitivity information provides first order assessment of circuit variability with respect to design variables and parasitics. This information is particularly useful for noise stability analysis, timing rule generation, and circuit optimization. Techniques for incorporating adjoint transient sensitivity into ACES, a general piecewise linear simulator, are presented. Sensitivity computation includes algorithms to handle instantaneous charge redistribution due to the discontinuous conductance models of the piecewise linear elements, and the loss of simulation accuracy due to the non-monotonous responses in autonomous adjoint circuits with non-zero initial conditions. Results demonstrate the efficiency and accuracy of the proposed techniques.

1.Introduction

Sensitivity information is extremely useful in assessing the performance and robustness of VLSI circuits, especially as design trends move continuously towards higher performance at lower costs and faster turn around times. These high performance designs require better exploitation of process technology capabilities and more dependence on transistor level analysis and verification. Information on how the design responds to changes in design variables and parasitics is particularly useful for verification and optimization. Transient sensitivity information provides a first order approximation of the design variability. This information can be utilized for noise analysis, timing rule generation, and traditional use in circuit optimization. Noise margin of the circuit can be computed by knowing the sensitivity of the output voltage with respect to the input voltage. The use of transient sensitivity provides information on the more accurate and useful AC noise margin than the traditional DC noise margin. Transient sensitivity information can also be used to reduce the number of required simulations for the generation of timing rules. These timing rules are then used in a static timing analysis environment. Transient sensitivity has been particularly useful in circuit optimization and tuning. Transient sensitivity provides gradient information which reduces the number of simulations required to perform transistor and interconnect sizing, which has implications for timing, power, and manufacturability improvements. Gradients can also be approximated by finite differences instead of transient sensitivity computation. However, in a direct or finite difference scheme, the gradient computation of a given circuit performance with respect to each circuit parameter requires an extra simulation of the perturbed circuit in addition to the simulation of the nominal circuit. The cost of gradient computation, therefore, will quickly escalate as the number of circuit or design parameters increases. Efficiency of the transient sensitivity computation is critical for its application to noise analysis, timing rule generation and circuit optimization. Furthermore, this sensitivity computation should be provided within an efficient circuit/timing simulation environment that can handle large and complex VLSI circuits. In order to address these needs, this paper proposes a method for computing transient sensitivities in Adaptively Controlled Explicit Simulation (ACES)[1], an efficient event driven simulator that employs piecewise linear (PWL) device models.

In general, there are two well known methods for computing sensitivities: the direct[3] and the adjoint[2] methods. The advantages and disadvantages of these methods for sensitivity computation have been discussed in detail [3][4]. In general, the direct method is advantageous when the sensitivities of a large number of circuit responses with respect to a few circuit parameters are desired. On the other hand, the adjoint method is more advantageous when the sensitivities of a few performance functions with respect to a large number of circuit parameters are required. The latter is usually the case in practice, and the adjoint method is the preferred method. However, there are inherent limitations that make the computation of adjoint transient sensitivities expensive in traditional circuit simulation[3].

In such a simulation environment, the original or nominal circuit is simulated forward in time over the time interval of interest. Then, the adjoint circuit, which is constructed from the original circuit, is simulated backward in time. At each time point of this backward simulation, the transpose of the Jacobian matrix, that specifies the state or the appropriate linearized approximation of the original circuit, is used to analyze the resultant adjoint circuit. The adjoint method, therefore, incurs the cost of either storing the Jacobian at each time point of the forward simulation or reconstructing these Jacobians during the backward simulation. Another problem is that, for good error control, the simulation of the adjoint circuit will require a different time grid as compared with the time grid for the original simulation. This time grid mismatch entails the interpolation of the previously stored Jacobians. This interpolation may impact the accuracy of the adjoint simulation. On the other hand, recomputing the Jacobians during the backward simulation will undermine computational efficiency. These problems are the main reasons why the adjoint method has not been widely used for computing transient sensitivities in circuit simulation even though it is the method of choice for computing DC and frequency sensitivities. It is usually more convenient and possibly more efficient to incorporate the direct method for computing transient sensitivities in traditional or relaxation based circuit simulators[3][5]. As a result, the efficiency of the transient sensitivity computation is compromised in these circuit simulation environments.

The key to resolve the above problems for the adjoint method is to take advantage of the simplified device models and the event driven nature of the more efficient circuit/timing simulators[4][6]. The event driven nature of these simulators allows one to store only a few circuit elements at each event time to reconstruct the Jacobians during the adjoint simulation. Moreover, the adjoint simulation can traverse the same event queue, which comes from the original simulation, except backward in time. The simplified models eliminate the need for interpolating the Jacobians. These advantages hold for ACES. This paper will also address two specific problems in incorporating adjoint transient sensitivity computation into ACES: the discontinuity of the device models and the non-monotonic responses in the adjoint circuit. The piecewise linear models are continuous in the original ACES simulation. However, the device models in the adjoint circuit, as derived from these PWL models are discontinuous. This discontinuity gives rise to the impulse conductance in the derivative space of the adjoint circuit. An algorithm has been developed to handle this type of impulse conductance to ensure the accuracy of the adjoint simulation. The next problem is the non-monotonic responses of the autonomous adjoint circuit with non-zero initial conditions. This problem is similar to the problem of underdamped RLC circuit responses, which can be handled by a special procedure that uses macromodeling of the linear circuits [1]. In this paper, we will develop a more general method that essentially makes the numerical integration in ACES less damping and allows it to capture these non-monotonic responses.

The paper is organized as follows: a brief overview of the adjoint method in ACES is presented in Section 2 together with the description of typical circuit elements in the adjoint circuit. Section 3 discusses the algorithms to handle problems specific to adjoint computation in ACES. Section 4 presents some results to demonstrate the effectiveness of the new algorithms. Section 5 concludes the paper.

2. Overview of Adjoint Sensitivity Computation in ACES

An overview of the procedure for computing transient sensitivities in ACES using the adjoint method is shown in Fig. 1. First, the original circuit is simulated forward in time. The events due to the PWL elements are stored in an event queue during the original simulation. At each PWL event, the index of the parent partition and all the necessary information about the PWL element causing the event are stored. Then the adjoint circuit is constructed from the original circuit and simulated backward in time. The inputs of the adjoint circuit are derived from the specified circuit performance function. These auxiliary excitations constitute the inputs for the adjoint simulation. The same partitions of the original circuit can be used for the adjoint simulation. During the adjoint simulation, the PWL event queue is traversed backward in time. At each event time, the information about the PWL element and its parent partition are retrieved for the analysis of the adjoint circuit at that particular time point. Finally, the results of the adjoint simulation are convolved with the results from the original simulation to provide the required sensitivities. A detailed example is presented in Section 4 to illustrate the adjoint sensitivity computation in ACES.

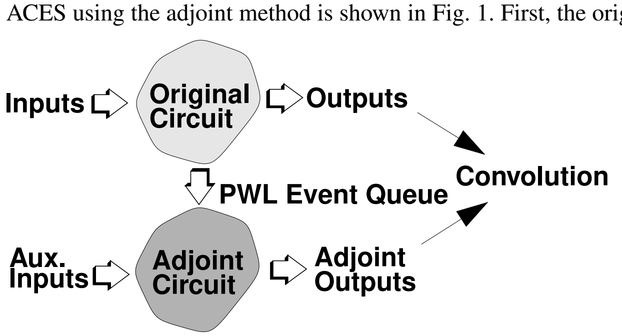


Fig. 1*: An overview of the adjoint method for computing transient sensitivities in ACES*

In the following subsections, the derived models in the adjoint circuit of some typical circuit elements used in ACES and the selection of the excitations for the adjoint circuit will be described. The following notations will be used throughout the paper. The voltages and currents in the in original circuit are denoted by v and i while the voltages and currents in the adjoint circuit are denoted by $\hat{v}$ and $\hat{i}$. The temporal variable in the original circuit is denoted by t, and in the adjoint circuit by $\tau = t_0 + t_f - t$, where t_0 and t_f are the initial and final times of the simulation.

2.1. Piecewise Linear Adjoint Circuit Elements

In this subsection, we will derive the characteristic equations of the elements in the adjoint circuit corresponding to the PWL devices, which model all the resistive elements in ACES. The adjoint characteristic equations for other circuit elements can be found in [2][4]. In order to simplify the discussion, consider a two terminal device, the PWL characteristic of which is shown in Fig. 2. Formally, the i-v

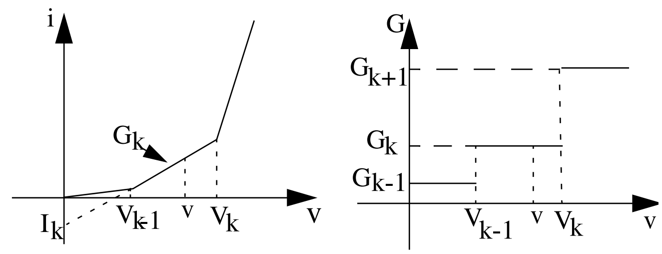


Fig. 2*: The i-v characteristic and the conductance of a two terminal PWL element*

characteristic of a PWL two terminal resistive element can be written as

$$i = \sum_{k=-\infty}^{\infty} (G_k v + I_k)[u(v - V_k) - u(v - V_{k+1})] \tag{1}$$

where V_k and V_{k+1} are the breakpoints in the PWL model, $(G_k v + I_k)$ is the equation representing the k^{th} linear segment of the model, and $u(v)$ is the unit step function. Note that the index is extended to infinity to simplify the notation. Changing the index in Eq.(1) yields

$$\begin{aligned} i &= \sum_{k=-\infty}^{\infty} (G_k v + I_k) u(v - V_k) - \sum_{k=-\infty}^{\infty} (G_{k-1} v + I_{k-1}) u(v - V_k) \\ &= \sum_{k=-\infty}^{\infty} (\Delta G_k v + \Delta I_k) u(v - V_k) \end{aligned} \tag{2}$$

where $\Delta G_k = G_k - G_{k-1}$ and $\Delta I_k = I_k - I_{k-1}$. Hence, the conductance of the PWL model can be written as

$$G = \frac{\partial i}{\partial v} = \sum_{k=-\infty}^{\infty}(\Delta G_k)u(v - V_k) + \sum_{k=-\infty}^{\infty}(\Delta G_k v + \Delta I_k)\delta(v - V_k)$$
$$= \sum_{k=-\infty}^{\infty}(\Delta G_k)u(v - V_k) \tag{3}$$

where $\delta(v)$ is the Dirac delta function and $(\Delta G_k v + \Delta I_k)\delta(v - V_k) = 0$ due to the continuity of the current at the breakpoints of the PWL model.

It can be shown [4] that for a general two terminal device, $i(t) = i(v(t))$, the corresponding element in the adjoint circuit is described by

$$\hat{i}(\tau) = \left.\frac{\partial i}{\partial v}\right|_{\hat{v}(\tau)}\hat{v}(\tau) = G(\hat{v}(\tau))\hat{v}(\tau) \tag{4}$$

In ACES, it is more convenient and efficient to operate in the derivative space [1]. The corresponding element of the two terminal device described by Eq.(4) in the derivative space is given by

$$\frac{d\hat{i}}{d\tau} = G\frac{d\hat{v}}{d\tau} + \frac{\partial G}{\partial \tau}\hat{v} \tag{5}$$

Therefore, in the adjoint circuit, the impulse conductance of a PWL two terminal device in the derivative space can be written as

$$\frac{dG}{d\tau} = \frac{\partial G}{\partial \hat{v}}\frac{d\hat{v}}{d\tau} = \left\{\sum_{k=-\infty}^{\infty}(\Delta G_k)\delta(\hat{v} - V_k)\right\}\dot{\hat{v}}(\tau)$$
$$= \left\{\sum_{k=-\infty}^{\infty}(\Delta G_k)\delta[\dot{\hat{v}}(\tau)(\tau - \tau_k)]\right\}\dot{\hat{v}}(\tau) = \sum_{k=-\infty}^{\infty}(\Delta G_k)\delta(\tau - \tau_k) \tag{6}$$

where $\hat{v} - V_k = \dot{\hat{v}}(\tau)(\tau - \tau_k)$ with τ_k being the event time when the breakpoint V_k is reached during the simulation, and $\delta[\dot{\hat{v}}(\tau)(\tau - \tau_k)] = [\delta(\tau - \tau_k)]/\dot{\hat{v}}(\tau)$ by the scaling property of the impulse function. In other words, $(dG)/(d\tau)$ is a train of impulses, which is zero everywhere except at the event times τ_k 's corresponding to the element reaching one of its breakpoints. This derivation can be extended in a straightforward manner to multi-terminal PWL elements.

2.2. Adjoint Circuit Excitations

The excitations in the adjoint circuit are derived from a specified circuit performance function. In general, the circuit performance will be a function of circuit variables such as currents and voltages. The performance function can be assumed to be of the form

$$\Phi(\mathbf{v}_I, \mathbf{i}_V) = \int_{t_0}^{t_f} f(\mathbf{v}_I, \mathbf{i}_V)dt \tag{7}$$

where $\mathbf{v}_I$ denotes the vector of all voltage responses and $\mathbf{i}_V$ denotes the vector of all current responses. This formulation is sufficiently general to express all the interesting performance functions in practice. The performance function variation can then be expanded as

$$\delta\Phi = \int_{t_0}^{t_f}\left[\sum_V \frac{\partial f}{\partial v_I}\delta v_I + \sum_I \frac{\partial f}{\partial i_V}\delta i_V\right]dt \tag{8}$$

If we assume that $\delta v_V = 0$ and $\delta i_I = 0$, and the adjoint circuit excitations are chosen such that $\hat{i}_I = \frac{\partial f}{\partial v_I}$ and $\hat{v}_V = -\frac{\partial f}{\partial i_V}$, then the left hand side of (8) is exactly the performance function variation $\delta\Phi$.

For example, suppose that the sensitivity, $\delta v_n(t_k)$, of a given node voltage at a particular time point, $v_n(t_k)$, is desired. First attach a zero valued current source I_n to node n to measure the node voltage in the original circuit, i.e. $v_{In}(t) = v_n(t)$. In other words,

$$\Phi(\mathbf{v}_I, \mathbf{i}_V) = v_{In}(t_k) = \int_{t_0}^{t_f} f(\mathbf{v}_I, \mathbf{i}_V)dt = \int_{t_0}^{t_f} v_{In}(t)\delta(t - t_k)dt$$

Then, $\hat{v}_V = -\frac{\partial f}{\partial i_V} = 0$ and $\hat{i}_{In} = \frac{\partial f}{\partial v_{In}} = \delta(t - t_k)$. For other circuit performance functions such as delay or power, the excitations in the adjoint circuit can be derived in a similar manner.

3.Adjoint Sensitivity Algorithms

This section discusses the additional algorithms that have been developed to address the specific problems of incorporating adjoint sensitivity computation into ACES: charge redistribution due to impulse conductances and non-monotonic responses of the adjoint circuit. A brief overview of the problems is presented before a detailed description in the next two subsections. The problem of impulse conductances arises from the nature of the PWL models used in ACES. As shown in the previous section, the BCR of a multi-terminal resistive element in the adjoint circuit is the transpose of the conductance matrix, $\frac{\partial i}{\partial v}$, of the corresponding element in the original circuit. Since the resistive elements in the original circuit are modeled by PWL i-v characteristics, the corresponding element in the adjoint circuit will be modeled by piecewise constant conductance. One more level of differentiation to operate in the derivative space yield the impulse conductances described by Eq.(6). These impulse conductances will cause instantaneous charge redistribution in the derivative space. An algorithm is presented in subsection 3.1. to compute accurately the charge redistribution.

The problem of non-monotonic responses of autonomous adjoint circuits arises when a voltage sensitivity at a specified time point is desired. As discussed in subsection 2.2., the excitation corresponding to this performance function in the adjoint circuit is an impulse current source. The effect of such an impulse excitation is to deposit non-zero initial conditions on node charges. The adjoint circuit is then autonomous or not driven by excitations. The responses of such an autonomous circuit with non-zero initial conditions are usually non-monotonic, and this type of non-monotonic response is not captured by the current quiescence algorithm in ACES [1]. A modified quiescence algorithm will be proposed in subsection 3.2. to handle this case.

3.1. Instantaneous Charge Redistribution in the Adjoint Circuit

In general, using the charge formulation, the circuit equation of the original circuit can be written as

$$\dot{q}_n(v_n) = i_{pwl}(v_n) + I_s \quad (9)$$

where q_n denotes the vector of node charges, i_{pwl} is the vector of currents due to the PWL elements, and I_s is the vector of current sources. For ACES, it is more convenient to perform the simulation in the derivative space [1]. Differentiating Eq. (9) with respect to time yields

$$\ddot{q}_n(v_{n)}) = G_{pwl}\dot{v}_n(t) + \dot{I}_s \quad (10)$$

where $G_{pwl} = \dfrac{\partial i_{pwl}}{\partial v_{pwl}}$ is the piecewise constant conductances of the PWL elements. It can be shown that the state equation for the adjoint circuit can then be written as

$$\frac{d}{d\tau}\hat{q}(\hat{v}_n) = G^T_{pwl}(\hat{v}_n)\hat{v}_n(\tau) + \hat{I}_s \quad (11)$$

In the derivative space, Eq. (11) becomes

$$\frac{d^2}{d\tau^2}\hat{q}(\hat{v}_n) = G^T_{pwl}(\hat{v}_n)\frac{d\hat{v}_n}{d\tau} + \left(\frac{dG_{pwl}}{d\tau}\right)\hat{v}_n(\tau) + \frac{d\hat{I}_s}{d\tau} \quad (12)$$

where

$$\frac{dG_{pwl}}{d\tau} = \left(\frac{\partial G_{pwl}}{\partial v_{pwl}}\right)\left(\frac{d\hat{v}_{pwl}}{d\tau}\right) \quad (13)$$

represents a train of impulses because G_{pwl} are piecewise constant functions of voltages (and time). In other words, $(dG_{pwl})/(d\tau)$ is a train of impulses, which is zero everywhere except at the event times τ_k 's corresponding to the element reaching one of its breakpoints. Therefore, between two successive PWL events τ_{e-1} and τ_e, Eq. (12) simplifies to

$$\frac{d^2}{d\tau^2}\hat{q}(\tau) = G_{pwl}(\tau)\frac{d\hat{v}_n}{d\tau} + \frac{d\hat{I}_s}{d\tau} \qquad \text{for} \quad \tau \in (\tau^+_{e-1}, \tau^-_e) \quad (14)$$

The numerical solution of this equation can be computed using standard ACES algorithm.

Next we need to compute the charge redistribution due to the impulse conductance at each PWL event τ_e. There is a more direct way to derive Eq. (6) for each PWL event, which is also more convenient for implementation. Note that at each event time τ_e corresponding to a PWL element changing its region, the conductance of the element can be written as

$$\begin{aligned} G(\tau) &= G(\tau^-_e) + [G(\tau^+_e) - G(\tau^-_e)]u(\tau - \tau_e) \\ &= G(\tau^-_e) + \Delta G(\tau_e)u(\tau - \tau_e) \qquad \text{for} \quad \tau \in (\tau^-_e, \tau^+_e) \end{aligned} \quad (15)$$

where $\tau^-_e = \lim_{\varepsilon \to 0}(\tau_e - \varepsilon)$ and $\tau^+_e = \lim_{\varepsilon \to 0}(\tau_e + \varepsilon)$ with $\varepsilon > 0$, and $\Delta G(\tau_e) = G(\tau^+_e) - G(\tau^-_e)$. Hence,

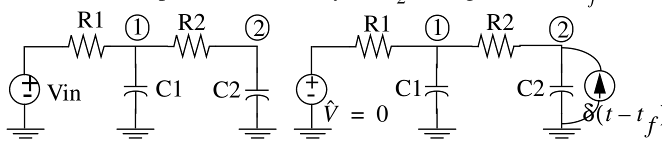
$$\frac{\partial G}{\partial \tau} = \Delta G(\tau_e)\delta(\tau - \tau_e) \qquad \text{for} \quad \tau \in (\tau^-_e, \tau^+_e) \quad (16)$$

Substituting Eq.(16) into Eq.(12) and integrating the resulting equation from τ^-_e to τ^+_e yields the following equation for updating $\hat{q}$ at each PWL event,

$$\dot{\hat{q}}_n(\tau^+_e) = \dot{\hat{q}}_n(\tau^-_e) + \Delta G(\tau_e)\hat{v}_n(\tau_e) \quad (17)$$

3.2. Non-monotonic Responses in the Autonomous Adjoint Circuit

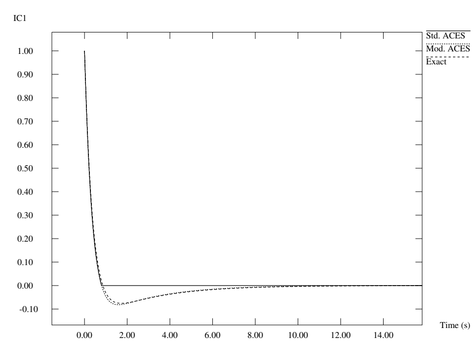
The problem of non-monotonic responses in the adjoint circuit will be illustrated in this subsection by considering a simple 2RC example as shown in Fig. 3. The associated adjoint circuit is constructed to compute the sensitivity of v_2 at a given time t_f. The



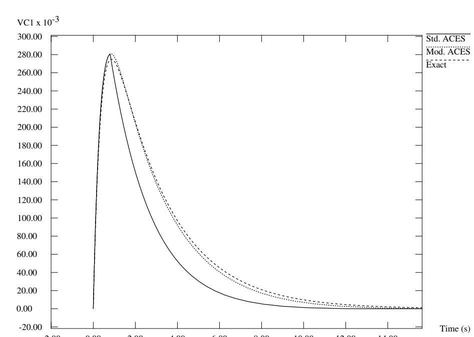
(a) Original Circuit (b) Adjoint Circuit

Fig. 3: *A simple 2RC example: (a) the original circuit and (b) the adjoint circuit to compute the sensitivity of the voltage at node 2.*

effect of the impulse source is to deposit a charge of 1 on C2. In other words, the above circuit is equivalent to the original 2RC circuit without any excitations, but with non-zero initial conditions, $v_1 = 0$ and $v_2 = 1$. The sensitivity with respect to various circuit parameters can then be computed by convolving appropriate simulation results of the original circuit shown in Fig. 3(a) with simulation results of the adjoint circuit shown in Fig. 3(b). Fig. 4 shows the cur-



(a) Current waveforms (IC1)



(b) Voltage Waveforms (VC1)

Fig. 4: *The current and voltage waveforms at node 1 obtained by (a) Exact solution, (b) ACES with standard quiescence algorithm, and (c) ACES with modified quiescence algorithm*

rent and voltage waveforms at node 1 of the adjoint circuit obtained by exact solution, ACES with standard quiescence algorithm, and ACES with modified quiescence algorithm. The ACES simulations were performed with an error tolerance of 0.001. As shown in Fig. 4, both the current and voltage at node 1 approach steady state non-monotonically. In the standard ACES algorithm, when the current IC1 becomes zero at $t = 0.8s$, the (capacitance) state V1 is put into quiescence. This corresponds to the peak of the voltage waveform when the voltage slope becomes zero. As a result, the standard ACES algorithm missed the negative currents for $t > 0.8s$ resulting in a corresponding loss of accuracy for the voltage waveform.

This problem illustrates the overdamping nature of the standard quiescence algorithm in ACES. One way to resolve this problem is to make the quiescence algorithm less damping. Note that the goal of the quiescence algorithm in ACES is to ensure the stability of the underlying explicit simulation algorithm by not allowing the state variables to cross their steady state values. However, one needs to distinguish between stable and unstable equilibrium (or steady) state. The peak of the voltage waveform in Fig. 4(b) represents an unstable equilibrium point, where a small perturbation in either direction will move it out of equilibrium. On the other hand, the "true" steady state at $t \to \infty$ is a stable equilibrium. Hence, the solution to the problem of unstable equilibrium is not to put a given state into quiescence when it first reaches quiescence (or equilibrium point). We simply mark the state to be in potential quiescence (PQS). Note we do not ignore the computation of the quiescence (QS) time step. It is still used as a time step limiting mechanism to ensure that all quiescence (or equilibrium) points are captured. However, the state is not put into QS the first time it reaches an equilibrium point. Recall that in ACES once a state is in quiescence it is in quiescence at all successive time points until an input source changes its PWL segment or a PWL element changes its PWL region. In the modified quiescence algorithm, a state will stay in PQS unless the circuit has an event due an input source, a PWL element, or a local truncation error (LTE) step. These events indicate that the equilibrium point for the given state may not be stable. For input source or PWL events, all states should be kicked out of PQS or QS. For LTE events, an additional check is performed to make sure that the time derivative (slope) of a given PQS state has changed sign before it is kicked out of PQS. Otherwise, the state will be kept in PQS. Note that a state is still in PQS if other states enters PQS or QS. These conditions indicate that the circuit is approaching steady state; hence, the given state should be kept in PQS. The state will enter QS when it has a second QS time step. The modified quiescence algorithm is summarized below

```
if (event due to the QS time step of a state S) {
    if (state S is non-quiescent)
        mark S in PQS
    else /* state S already in PQS */
        mark S in QS
}
else if (event due to an LTE time step)
    mark PQS states non-quiescent if slopes changed signs
else /* event due to an input source or a PWL element */
    mark all PQS and QS states non-quiescent
```

The results in Fig. 4 shows that the modified quiescence algorithm indeed captures all the equilibrium points correctly and helps ensure the accuracy of the simulation.

4.Results

The first example is a simple CMOS circuit shown in Fig. 5. This

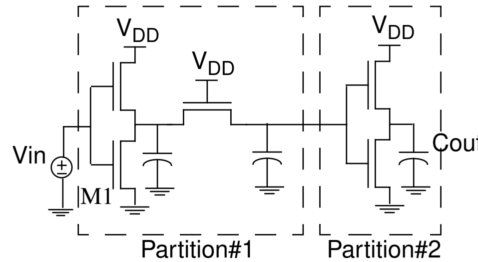
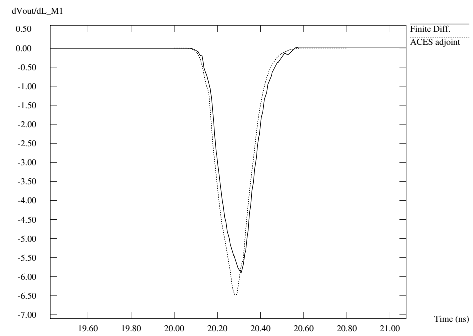


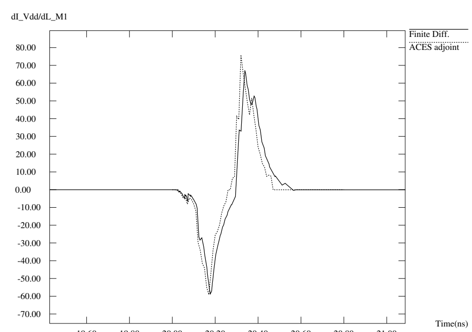
Fig. 5: *A simple two inverter circuit to demonstrate adjoint sensitivity computation with circuit partitioning.*

example is to demonstrate adjoint sensitivity computation with circuit partitioning. In the original circuit, the first inverter and the pass transistor is grouped into one partition, and the second inverter is grouped into a second partition. Suppose that the sensitivities of the output node with respect to various circuit parameters are desired. Then the information flow in the adjoint circuit is from the output to the input in opposition to the information flow from the input to the output for the original circuit. Since the controlling flow of the equivalent models of the MOSFETs in the adjoint circuit is also reversed, it can be argued that the adjoint circuit should be repartitioned to group the second inverter and the pass transistor into one partition and the first inverter into another partition. However, it is advantageous to retain the same partitioning to avoid the cost of partitioning and setting up the partitions. Moreover, the MOSFET model in the adjoint circuit does not have the nice property of the gate current being always zero [4]. Hence, there is no DC decoupling in the adjoint circuit as compared to the original circuit. Therefore, there are no clear advantages in repartitioning the circuit. In addition, the partitioning should affect only the efficiency and not the accuracy of the simulation. The sensitivity waveforms of the output voltage and the current through Vdd with respect to the length of the NMOS transistor (M1) of the first inverter as obtained by direct (finite) difference and ACES adjoint computation are shown in Fig. 6. Note that the adjoint waveforms are obtained by multiple adjoint simulations with the voltage or current at a set of timepoints as the performance function. These simulations are performed with an error tolerance of 0.01.

A set of realistic test circuits are then used to assess the performance of the adjoint analysis in ACES. One adjoint analysis was performed for each circuit. Table 1 summarizes the run times of the ACES simulation and the associated adjoint simulation.



(a) Voltage sensitivity waveforms



(b) Current sensitivity waveforms

Fig. 6*: The sensitivity waveforms of the output voltage and the current through Vdd with respect to the length of the NMOS transistor M1 of the circuit shown in Fig. 5.*

Table 1: Simulation run times for a number of test circuits

Circuit	No. of MOSFETs	Original simulation	Adjoint simulation
2bit error control circuit	104	3.81s	0.34s
Counter circuit	220	20.5s	2.2s
4bit ALU	282	6.64s	0.58s

Note that only the event processing of the original ACES simulation and the adjoint simulation are shown. The costs of setting up the circuit and computing the convolutions are not included. The cost of the convolution varies depends upon the number of timepoints in the waveforms. On the average, the cost of the adjoint simulation is about 10% of the cost of the original simulation with 0.01-0.02% overhead for each convolution. Note that these results are obtained with an error tolerance of 0.01 on an IBM RS600 390 workstation with 256M of RAM. A set of typical sensitivity waveforms obtained by finite difference and ACES adjoint computation is shown in Fig. 7.

5.Conclusions

A general method for computing adjoint transient sensitivities in event driven simulation algorithms employing piecewise linear models such as ACES has been presented. Accuracy and efficiency of the algorithms has been demonstrated on various examples. The cost of the adjoint simulation is only about 10% of the original nominal simulation. The sensitivities of both voltage and current responses with respect to circuit parameters can be computed. Sensitivities of general circuit performances which are functions of these circuit responses can be obtained by applying the chain rule. Two additional algorithms have been developed to address the charge redistribution from the impulse conductances due to the discontinuity in the PWL device models, and the non-monotonic responses of the autonomous adjoint circuit.

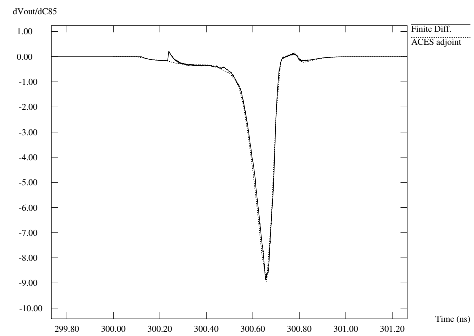


Fig. 7*: A set of sensitivity waveforms as obtained by finite (direct) difference and ACES adjoint computation.*

Acknowledgments

The authors would like to thank Dr. David LaPotin for his support during the course of this work, Dr. David Winston for his help with the interface code, and Dr. Ali Sadigh for his helpful discussion with the quiescence algorithm.

References

[1] A. Devgan and R. A. Rohrer, "Adaptively Controlled Explicit Simulation," *IEEE Trans. Computer-Aided Design*, Val. 13, pp.746-762, June 1994.

[2] S. W. Director and R. A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities," *IEEE Tran. Circuit Theory*, vol.16, pp. 318-323, August 1969.

[3] D. A. Hocevar, P. Yang, T. N. Trick, and B. D. Epler, "Transient Sensitivity Computation for MOSFET Circuits," *IEEE Trans. Computer-Aided Design*, vol. 4, pp. 609-620, Oct. 1985.

[4] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity Computation in Piecewise Approximate Circuit Simulation," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 171-183, Feb. 1991.

[5] C-J Chen and W-S Feng, "Relaxation-Based Transient Sensitivity Computation for MOSFET Circuits," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 173-185, Feb. 1995.

[6] T. V. Nguyen, P. Feldmann, S. W. Director, and R. A. Rohrer, "SPECS Simulation Validation with Efficient Transient Sensitivity Computation," *Proc. ICCAD*, 1989, pp. 252-255.