CIRCUIT THEORY AND APPLICATIONS, VOL. 10, 299–309 (1982)

CHARACTERIZATION OF INTEGRATION ALGORITHMS FOR THE TIMING ANALYSIS OF MOS VLSI CIRCUITS

GIOVANNI DE MICHELI AND ALBERTO SANGIOVANNI-VINCENTELLI
Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory, University of California, Berkeley, CA 94720, U.S.A

SUMMARY

Displacement techniques used for the timing analysis of VLSI circuits are presented under a new perspective. Their numerical properties such as stability, accuracy, consistency and convergence are investigated.

1. INTRODUCTION

When analogue voltage levels are critical to circuit performance, or where tightly coupled feedback loops are present, standard circuit simulators such as SPICE[1] or ASTAP[2] can be used to analyse the circuit. However, when the size of the circuit becomes large, the cost and the memory requirements of conventional circuit simulators become prohibitive and new techniques have to be used. The timing simulator MOTIS[3] was developed to simulate large scale integrated circuits. The program MOTIS was a revolutionary simulator in two main respects:

(a) It limited severely the types of networks it dealt with (MOS devices with quasi-unidirectional circuit models and a grounded capacitor on every node)
(b) It discarded both sparse Gauss elimination and conventional Newton–Raphson iteration as solution methods.

In MOTIS the backward Euler formula was used to discretize the time derivative operator and a non-linear Gauss–Jacobi like relaxation technique[4] was adopted to decouple the node equations at the non-linear equation level. The algorithms of the timing simulators MOTIS-C[5] and SPLICE[6] perfected this technique. In particular, SPLICE used a non-linear 'Gauss–Seidel like' technique with a selective trace algorithm to exploit the 'latency'[8] of large digital circuits. None of these algorithms carried the iteration of the relaxation methods to convergence: only one sweep was taken. Because of this, the numerical properties such as stability of the integration formulae used to discretize the derivative operator no longer hold. These methods have indeed to be considered as *new integration methods*. Hence a complete analysis of their numerical properties has to be carried out to characterize them.

In this paper we formalize these relaxation or displacement methods and contrast them with a new method proposed by W. Kahan.[9] Then we propose a model to study formally the stability, accuracy, consistency and convergence properties of the methods. Based on this model, we evaluate the various methods and show that the method proposed by Kahan has better stability and accuracy properties.

2. TIMING ANALYSIS ALGORITHMS

Timing analysis programs (e.g. MOTIS and the timing analysis part of the mixed mode simulator SPLICE) assemble the circuit equations of large scale MOS circuits by using node equations.[10] In this paper, we shall assume that node equations can be always written. A sufficient condition for this to happen is that the circuits to be analysed contain voltage controlled current sources, voltage controlled capacitors, voltage controlled two terminal resistors and independent current sources.[10] Moreover, we shall assume

0098-9886/82/040299-11$01.10

Received 6 October 1981
Revised 16 March 1982

that every node in the circuit has a (possibly non-linear) capacitor called grounded capacitor, to either ground or d.c. supply voltage rails. This assumption which is usually satisfied by most practical MOS circuits where these capacitors are used to model the time delay of a signal propagating through the circuit, implies that the node equations of the circuit are also state equations where the state variables are node voltages. Since each node has a capacitor to ground, the node equations have the following form:

$$\mathbf{C}(\mathbf{v})\dot{\mathbf{v}} + f(\mathbf{v}, \mathbf{u}(t)) = \mathbf{0}, \qquad \mathbf{v}(0) = \mathbf{v}_0.$$

$$\mathbf{v} \in R^n; \quad \mathbf{u}: R \to R^m; \quad C(\cdot): R^n \to R^{n \times n}; \quad f(\cdot, \cdot): R^n \times R^m \to R^n \tag{1}$$

$$f(\mathbf{v}, \mathbf{u}(t)) = [f_1(\mathbf{v}, \mathbf{u}(t)), f_2(\mathbf{v}, \mathbf{u}(t)), \ldots, f_n(\mathbf{v}, \mathbf{u}(t))]^T$$

where $\mathbf{v}$ is the vector of node voltages, $\mathbf{u}$ is the vector of independent source waveforms, $\mathbf{C}(\mathbf{v})$ is the non-linear nodal capacitance matrix and $f_i(\mathbf{v}, \mathbf{u}(t))$ is the sum of the currents flowing out of the capacitors connected to node i. In this paper we shall also assume that no floating capacitor (i.e. capacitors connected between two non-ground nodes) is present in the circuit. Therefore $\mathbf{C}(\mathbf{v})$ is a diagonal matrix. We assume also that $\mathbf{C}(\mathbf{v})^{-1}$ exists for all $\mathbf{v}$ of interest. Therefore we can simplify (1) as follows:

$$\dot{\mathbf{v}} + F(\mathbf{v}, \mathbf{u}(t)) = \mathbf{0} \qquad \mathbf{v}(0) = \mathbf{v}_0. \tag{2}$$

where

$$F(\mathbf{v}, \mathbf{u}(t)) = \mathbf{C}(\mathbf{v})^{-1} f(\mathbf{v}, \mathbf{u}(t)) \tag{3}$$

Algorithms used in the timing analysis of MOS and VLSI circuits discretize the derivative operator by the backward Euler[3,6] or trapezoidal formula.[5] In this paper we shall focus on the backward Euler formula

$$\dot{\mathbf{v}}_{k+1} = (\mathbf{v}_{k+1} - \mathbf{v}_k)/h \tag{4}$$

where $h = t_{k+1} - t_k$ and $\mathbf{v}_{k+1}$ and $\mathbf{v}_k$ are the computed voltages of the node voltors at time t_{k+1} and t_k respectively. The solution of the resulting non-linear system of equations:

$$\mathbf{v}_{k+1} - \mathbf{v}_k + hF(\mathbf{v}_{k+1}, \mathbf{u}(t_{k+1})) = 0 \tag{5}$$

is then approximated by one sweep of a displacement technique.

The program MOTIS[3] uses a Gauss–Jacobi like technique which yields the following set of decoupled equations:

$$\begin{aligned} v_{k+1}^1 - v_k^1 + hF_1(v_{k+1}^1, v_k^2, \ldots, v_k^n, u_1(t_{k+1})) &= 0 \\ v_{k+1}^2 - v_k^2 + hF_2(v_k^1, v_{k+1}^2, \ldots, v_k^n, u_2(t_{k+1})) &= 0 \\ v_{k+1}^n - v_k^n + hF_n(v_k^1, v_k^2, \ldots, v_{k+1}^n, u_n(t_{k+1})) &= 0 \end{aligned} \tag{6}$$

The solution of the decoupled non-linear equations (6) is then approximated by taking a single step of a *regula falsi iteration*.[11]

The MOTIS-C and SPLICE programs use a Gauss–Seidel like technique. In SPLICE this technique yields:

$$v_{k+1}^i - v_k^i + hF_i(\tilde{\mathbf{v}}_{k+1,i}, \mathbf{u}(t_{k+1})) = 0, \quad i = 1, 2, \ldots, n \tag{7}$$

where

$$\tilde{\mathbf{v}}_{k+1,i} = [v_{k+1}^1, \ldots, v_{k+1}^i, v_k^{i+1}, \ldots, v_k^n]^T \tag{8}$$

The solution of (7) is then approximated by using one step of the Newton–Raphson algorithm.

Another displacement technique for the solution of (1) has been proposed for a simple circuit in reference 12. This algorithm is a symmetric displacement method reminiscent of the alternating-direction implicit method[11] and is based on a method proposed by Kahan. The basic idea here is to 'symmetrize' the Gauss–Seidel scheme with a method that takes two half steps of size $h/2$ each: one half step is taken

in the usual 'forward' (i.e. lower triangular) direction, the second half step in the backward (i.e. upper triangular) direction. Letting:

$$\bar{\mathbf{v}}_{k,l} = \begin{cases} [v_1^1, \ldots, v_l^l, v_{l-1/2}^{l+1}, \ldots, v_{l-1/2}^n]^T & \text{if } 2l \text{ is odd} \\ [v_{l-1/2}^1, \ldots, v_{l-1/2}^{l-1}, v_l^l, \ldots, v_l^n]^T & \text{if } 2l \text{ is even} \end{cases} \tag{9}$$

the forward step yields:

$$v_{k+1/2}^i - v_k^i + \frac{h}{4} F_i(\bar{\mathbf{v}}_{k+1/2,i}, \mathbf{u}(t_{k+1/2})) + \frac{h}{4} F_i(\bar{\mathbf{v}}_{k+1/2,i-1}, \mathbf{u}(t_{k+1/2})) = 0, \quad i = 1, 2, \quad n \tag{10}$$

and the backward step:

$$v_{k+1}^i - v_k^i + \frac{h}{4} F_i(\bar{\mathbf{v}}_{k+1,i}, \mathbf{u}(t_{k+1})) + \frac{h}{4} F_i(\bar{\mathbf{v}}_{k+1,i+1}, \mathbf{u}(t_{k+1})) = 0, \quad i = n, n-1, \ldots, 1$$

The solution of the decoupled equations is then approximated by taking one step of the Newton–Raphson algorithm. Note that none of these methods solves (5) since only one sweep of the displacement iteration is taken. Therefore the stability and accuracy properties of the integration method used to discretize the derivative operator no longer hold. As a matter of fact, the combination of the discretization formula, of the various relaxation steps and of the Newton–Raphson method form a set of new integration algorithms. These integration methods use an implicit formula to discretize the differential equations, but they do not solve the non-linear equation obtained. Thus, they are somewhat in between explicit and implicit methods. We call these methods 'time advancement' algorithms.

In the sequel we will refer to the 'time advancement' algorithms which use the Gauss–Jacobi, the Gauss–Seidel and modified symmetric Gauss–Seidel displacement step as Gauss–Jacobi, Gauss–Seidel and modified symmetric Gauss–Seidel integration algorithms respectively. In the following section the numerical properties of these 'time advancement' methods will be investigated.

Before leaving this section, we remark that the 'time advancement' schemes described above can be applied to any circuit whose equations are written as in (2). However, since the main issue here is speed of computation, we focused on cases where equations of the form (2) can be assembled directly from the input description of the circuit. In general, formulating the equations of a circuit in form (2) is expensive in terms of computing.

In some interesting theoretical papers, Sandberg[18] and Roska[19,20] have tackled the important problem of the uniqueness of the solution of the non-linear algebraic equations obtained by discretizing the differential equations describing the behaviour of the circuit. The 'time advancement' schemes presented here have the nice property that since no equations are solved, the 'well-posedness' of the computation involved is almost always ensured. We have only to make sure that one step of the Newton–Raphson iteration can be taken on (6), (7) or (10), (11). Therefore, for the Gauss–Jacobi integration algorithm, the diagonal entries of the Jacobian of F evaluated at $\mathbf{v}_k, \mathbf{u}(t_{k+1})$ must all be non-zero. For the Gauss–Seidel integration algorithm, the diagonal entries of the Jacobian of F evaluated at $\bar{\mathbf{v}}_{k+1,i} \triangleq [v_{k+1}^1, \ldots, v_{k+1}^{i-1}, v_k^i, \ldots, v_k^n]^T$ must all be non-zero. Finally, for the modified symmetric Gauss–Seidel integration algorithm, the condition for the Gauss–Seidel integration algorithm must be satisfied at $k+\frac{1}{2}$ and in addition, an analogous condition must be satisfied for the backward step. Notice that these conditions are indeed very mild.

3. NUMERICAL PROPERTIES OF TIMING ANALYSIS ALGORITHMS

The numerical properties of an integration method, such as stability, are studied on test problems[13,14], which are simple enough to allow a theoretical analysis but still so general that one can have insight about how the method behaves in general. For the commonly used multistep methods, the test problem consists of a linear time-invariant asymptotically stable autonomous differential equation. Unfortunately this simple test problem cannot be used to evaluate the displacement techniques introduced in Section

2. In fact, each variable of the system of differential equations is treated differently according to the ordering in which equations are processed. Hence a more complex test problem is needed. The test problem we choose is a linear time-invariant asymptotically stable system of autonomous differential equations, i.e.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \qquad \mathbf{x}(0) = \mathbf{x}_0. \tag{12}$$

where $\mathbf{A} \in R^{n \times n}$ and the set of eigenvalues (spectrum) of $\mathbf{A}$, $\sigma(\mathbf{A})$, is in the open left half complex plane, i.e., $\sigma(\mathbf{A}) \in C_0^-$. In circuit theoretic terms, we consider as test circuits linear circuits whose natural frequencies are in the open left half plane and which satisfy the assumptions described in Section 2. Let $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, where $\mathbf{L}$ is strictly lower triangular, $\mathbf{D}$ is diagonal and $\mathbf{U}$ is strictly upper triangular. The displacement methods presented in Section 2 applied to the test system (12) yield the following recursive relations:

(a) Gauss–Jacobi integration algorithm:

$$[\mathbf{I} - h\mathbf{D}]\mathbf{x}_{k+1} = [\mathbf{I} + h(\mathbf{L} + \mathbf{U})]\mathbf{x}_k. \tag{13}$$

$$\mathbf{x}_{k+1} = \mathbf{M}_{GJ}(h)\mathbf{x}_k. \tag{14}$$

where $\mathbf{I}$ is the identity matrix and

$$\mathbf{M}_{GJ}(h) = [\mathbf{I} - h\mathbf{D}]^{-1}[\mathbf{I} + h(\mathbf{L} + \mathbf{U})]$$

(b) Gauss-Seidel integration algorithm:

$$[\mathbf{I} - h(\mathbf{D} + \mathbf{L})]\mathbf{x}_{k+1} = [\mathbf{I} + h\mathbf{U}]\mathbf{x}_k$$

$$\mathbf{x}_{k+1} = \mathbf{M}_{GS}(h)\mathbf{x}_k \tag{17}$$

where

$$\mathbf{M}_{GS}(h) = [\mathbf{I} - h(\mathbf{D} + \mathbf{L})]^{-1}[\mathbf{I} + h\mathbf{U}]$$

(c) Modified symmetric Gauss–Seidel integration algorithm:
Let:

$$\mathbf{A}_L = \mathbf{L} + \tfrac{1}{2}\mathbf{D} \qquad \mathbf{A}_U = \mathbf{U} + \tfrac{1}{2}\mathbf{D} \tag{19}$$

Forward step:

$$\left[\mathbf{I} - \frac{h}{4}(2\mathbf{L} + \mathbf{D})\right]\mathbf{x}_{k+1/2} = \left[\mathbf{I} + \frac{h}{4}(\mathbf{D} + 2\mathbf{U})\right]\mathbf{x}_k$$

$$\left[\mathbf{I} - \frac{h}{2}\mathbf{A}_L\right]\mathbf{x}_{k+1/2} = \left[\mathbf{I} + \frac{h}{2}\mathbf{A}_U\right]\mathbf{x}_k$$

$$\mathbf{x}_{k+1/2} = \left[\mathbf{I} - \frac{h}{2}\mathbf{A}_L\right]^{-1}\left[\mathbf{I} + \frac{h}{2}\mathbf{A}_U\right]\mathbf{x}_k$$

Backward step:

$$\left[\mathbf{I} - \frac{h}{4}(\mathbf{D} + 2\mathbf{U})\right]\mathbf{x}_{k+1} = \left[\mathbf{I} + \frac{h}{2}(2\mathbf{L} + \mathbf{D})\right]\mathbf{x}_{k+1/2}$$

$$\mathbf{x}_{k+1} = \left[\mathbf{I} - \frac{h}{2}\mathbf{A}_U\right]^{-1}\left[\mathbf{I} + \frac{h}{2}\mathbf{A}_L\right]\mathbf{x}_{k+1/2}$$

Combining (22) and (24) we obtain:

$$\mathbf{x}_{k+1} = \mathbf{M}_S(h)\mathbf{x}_k$$

where

$$\mathbf{M}_S(h)=\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_U\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_U\right]$$

The matrices $\mathbf{M}_{GJ}(h)$, $\mathbf{M}_{GS}(h)$ and $\mathbf{M}_S(h)$ are called the companion matrices of the methods. If we denote by $\mathbf{M}(h)$ the generic companion matrix of a method, we have:

$$\mathbf{x}_k=[\mathbf{M}(h)]^k\mathbf{x}_0.$$

We define next the numerical properties of the integration algorithms described by (27) following the outlines of one-step integration methods applied to ordinary differential equations.[13]

Definition 1. (*consistency*)

An integration algorithm is consistent if its companion matrix can be expanded in power series as a function of the step-size h as:

$$\mathbf{M}(h)=\mathbf{I}+h\mathbf{A}+O(h^2)$$

Definition 2 (*stability*)

An integration algorithm is stable if $\exists\delta>0$, $\exists N>0$ such that $\forall\mathbf{x}_0\in R^n$, $\exists\bar{k}>0$

$$\|\mathbf{x}_k\|<N \quad \forall k\geq\bar{k} \quad \forall h\in[0,\delta), \tag{29}$$

where $\mathbf{x}_k$ is the sequence generated by the algorithm applied to the test problem according to (27).

Definition 3 (*convergence*)

Let $\mathbf{x}(t)$ be the exact solution of the test problem. An integration algorithm is convergent if the sequence of the computed solution converges uniformly to $\mathbf{x}(t)$ as the step-size h tends to zero.

Theorem 1

The Gauss–Jacobi, Gauss–Seidel and modified symmetric Gauss–Seidel integration algorithms are consistent.

Proof. (a) Let us consider the Gauss–Jacobi integration algorithm first. To expand the companion matrix given by (15) in a power series as a function of the step-size h, we compute

$$\frac{d}{dh}\mathbf{M}_{GJ}(h)=[\mathbf{I}-h\mathbf{D}]^{-1}\mathbf{D}[\mathbf{I}-h\mathbf{D}]^{-1}[\mathbf{I}+h(\mathbf{L}+\mathbf{U})]+[\mathbf{I}-h\mathbf{D}]^{-1}(\mathbf{L}+\mathbf{U}) \tag{30}$$

and

$$\frac{d}{dh}\mathbf{M}_{GJ}(0)=\mathbf{D}+\mathbf{L}+\mathbf{U}=\mathbf{A} \tag{31}$$

where $\frac{d}{dh}\mathbf{M}_{GJ}(0)$ is the derivative of $\mathbf{M}_{GJ}(h)$ evaluated at $h=0$. It follows that

$$\mathbf{M}_{GJ}(h)=\mathbf{I}+h\mathbf{A}+O(h^2) \tag{32}$$

(b) The consistency of the Gauss–Seidel integration algorithm follows, '*mutatis mutandis*', by a similar argument.

(c) For the modified symmetric Gauss–Seidel integration algorithm, we have:

$$\frac{d}{dh}\mathbf{M}_S(h) = \left[\mathbf{I}-\frac{h}{2}\mathbf{A}_U\right]^{-1}\frac{1}{2}\mathbf{A}_U\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_U\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_U\right]$$
$$+\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_U\right]^{-1}\frac{1}{2}\mathbf{A}_L\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_U\right]$$
$$+\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_U\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\frac{1}{2}\mathbf{A}_L\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_U\right]$$
$$+\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_U\right]^{-1}\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\frac{1}{2}\mathbf{A}_U$$

and

$$\frac{d}{dh}\mathbf{M}_S(0) = \tfrac{1}{2}\mathbf{A}_U+\tfrac{1}{2}\mathbf{A}_L+\tfrac{1}{2}\mathbf{A}_L+\tfrac{1}{2}\mathbf{A}_U = \mathbf{A}$$

Hence:

$$\mathbf{M}_S(h) = \mathbf{I}+h\mathbf{A}+O(h^2) \tag{35}$$

The definition of stability requires the boundness of the sequence at $\mathbf{x}_k$ for small values of the stepsize h. The following proposition relates the boundness of the sequence $\mathbf{x}_k$ to the spectrum of $\mathbf{M}(h)$.

Proposition 1[15]

The sequence of vectors $\{\mathbf{x}_k\}$ defined by (27) is bounded for a given value of the stepsize $\bar{h}$ if and only if the spectrum of $\mathbf{M}(\bar{h})$ is contained in the unit ball $B(0, 1)$, i.e. $\sigma(\mathbf{M}(\bar{h})) \subseteq B(0, 1)$ and no multiple zero of the minimal polynomial of $\mathbf{M}(h)$ has modulus equal to one.

In the sequel we restrict our analysis to the cae in which the step-size is constant. From Proposition 1 it is immediate to derive the following theorem:

Theorem 2

An integration algorithm is stable if and only if $\exists\delta > 0$ such that $\forall h \in [0, \delta)$ the spectrum of $\mathbf{M}(h)$ is contained in the unit ball $B(0, 1)$ and no multiple zero of the minimal polynomial of $\mathbf{M}(h)$ has modulus equal to one.

Theorem 3

The Gauss–Jacobi, Gauss–Seidel and modified symmetric Gauss–Seidel integration algorithms are stable.

Proof. From the consistency of the above mentioned algorithms we have

$$\mathbf{M}(h) = \mathbf{I}+h\mathbf{A}+O(h^2) \tag{36}$$

By the spectral mapping theorem[15]

$$\sigma(\mathbf{M}(h)) = \{\xi_i | \xi_i = 1+h\lambda_i+O(h^2);\ \lambda_i \in \sigma(\mathbf{A});\ i = 1, 2, \ldots, \sigma\} \tag{37}$$

From (37) we have:

$$|\xi_i| = |1+h\lambda_i+O(h^2)|, \quad i = 1, 2, \ldots, \sigma \tag{38}$$

and

$$|\xi_i|^2 = [1+h\ \mathrm{Re}\ (\lambda_i)]^2+[h\ \mathrm{Im}\ (\lambda_i)]^2+O(h^2) \tag{39}$$

Since $\mathbf{M}(0) = \mathbf{I}$, its eigenvalues are all 1, and 1 is a simple zero of the minimal polynomial of the identity matrix. Therefore from Theorem 2 it is sufficient to show that:

$$\sigma(\mathbf{M}(h)) \subset B(0, 1) \quad \forall h \in (0, \delta) \tag{40}$$

i.e. from (39)

$$|\xi_i|^2 < 1 \quad \forall h \in (0, \delta) \quad i = 1, 2, \ldots, \sigma \tag{41}$$

From (41), we have:

$$2\mathrm{Re}(\lambda_i) + h(\mathrm{Re}^2(\lambda_i) + \mathrm{Im}^2(\lambda_i)) + O(h) < 0 \quad i = 1, 2, \ldots, \sigma$$

$$2\mathrm{Re}(\lambda_i) + O(h) < 0 \quad i = 1, 2, \ldots, \sigma$$

Since by assumption $\mathrm{Re}(\lambda_i) < 0$, $i = 1, 2, \ldots, \sigma$, $\exists \delta > 0$, such that $\forall h \in (0, \delta)$,

$$\sigma(\mathbf{M}(h)) \subset B(0, 1).$$

Corollary 1

The Gauss–Jacobi; Gauss–Seidel and modified symmetric Gauss–Seidel integration algorithms are convergent.

Proof. Follows from Theorems 1 and 3 and the classical convergence theorem.

For computational efficiency, it would be highly desirable that the stepsize be limited only by accuracy considerations as in the case of the implicit backward differentiation formulae.[13] In the case of classical multistep methods, the concept of A-stability[14] and stiff-stability[13] have been introduced to test the 'unconditional' stability of multistep methods. For the 'time-advancement' techniques introduced in this paper, it would make sense to define a similar concept. Unfortunately, general results of 'unconditional' stability are not available for the test problem previously defined, but only for a subclass, the subclass characterized by a symmetric A matrix. In circuit theoretic terms, we are now considering linear circuits whose node equations yield a symmetric nodal admittance matrix when only the resistive part of the circuit is considered. Moreover it is required that this matrix remain symmetric when premultiplied by $\mathbf{C}^{-1}$, the diagonal matrix of the grounded capacitors. A sufficient condition for this to occur is that the circuit consists of two terminal linear resistors and capacitors, and that the grounded capacitors be equal. The case of unequal grounded capacitors can also be included in this class provided that a scaling of the rows of the matrix is performed.

Definition 4. ($\tilde{A}$-stability)

An integration method is $\tilde{A}$-stable if $\exists N > 0$ such that $\forall \mathbf{x}_0 \in R^n$, $\exists \bar{k}$

$$\|\mathbf{x}_k\| < N \quad \forall k \geq \bar{k} \quad \forall h \in [0, \infty). \tag{45}$$

where $\{\mathbf{x}_k\}$ is the sequence generated by the method applied to the test problem (12) with A symmetric.

Theorem 4

The modified symmetric Gauss–Seidel method is $\tilde{A}$-stable.

Proof. Since A is symmetric and $\sigma(\mathbf{A}) \in C_0^-$, A is a negative definite matrix. For $h = 0$, $\mathbf{M}_S(0) = \mathbf{I}$, the eigenvalues of $\mathbf{M}_S(0)$ are all 1, and 1 is a simple zero of the minimal polynomial. Hence we need only to see where the eigenvalues of $\mathbf{M}_S(h)$ lie when $h \in (0, \infty)$. Let us apply to $\mathbf{M}_S(h)$ a similarity transformation:

$$\tilde{\mathbf{M}}_S(h) = \left[\mathbf{I} - \frac{h}{2}\mathbf{A}_U\right]\mathbf{M}_S(h)\left[\mathbf{I} - \frac{h}{2}\mathbf{A}_U\right]^{-1} \tag{46}$$

and factorize $\mathbf{M}_S$ as:

$$\tilde{\mathbf{M}}_S(h) = \mathbf{P}(h)\mathbf{Q}(h) \tag{47}$$

where

$$\mathbf{P}(h)=\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1} \tag{48}$$

$$\mathbf{Q}(h)=\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_U\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_U\right]^{-1} \tag{49}$$

Now:

$$\|\mathbf{P}(h)\|_2^2=\max_{x\neq 0}\frac{\left\langle\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\mathbf{x},\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]\mathbf{x}\right\rangle}{\langle\mathbf{x},\mathbf{x}\rangle} \tag{50}$$

$$\mathbf{y}=\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]^{-1}\mathbf{x} \tag{51}$$

Then:

$$\|\mathbf{P}(h)\|_2^2=\max_{y\neq 0}\frac{\left\langle\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\mathbf{y},\left[\mathbf{I}+\frac{h}{2}\mathbf{A}_L\right]\mathbf{y}\right\rangle}{\left\langle\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]\mathbf{y},\left[\mathbf{I}-\frac{h}{2}\mathbf{A}_L\right]\mathbf{y}\right\rangle} \tag{52}$$

$$=\max_{y\neq 0}\frac{\langle\mathbf{y},\mathbf{y}\rangle+\frac{h}{2}\langle\mathbf{y},\mathbf{A}\mathbf{y}\rangle+\frac{h^2}{4}\langle\mathbf{y},\mathbf{A}_U\mathbf{A}_L\mathbf{y}\rangle}{\langle\mathbf{y},\mathbf{y}\rangle-\frac{h}{2}\langle\mathbf{y},\mathbf{A}\mathbf{y}\rangle+\frac{h}{4}\langle\mathbf{y},\mathbf{A}_U\mathbf{A}_L\mathbf{y}\rangle} \tag{53}$$

Since $\forall\mathbf{y}$, $\langle\mathbf{A}_L\mathbf{y},\mathbf{A}_L\mathbf{y}\rangle>0$, and $\mathbf{A}$ is negative definite

$$\|\mathbf{P}(h)\|_2^2<1 \quad \forall h\in(0,\infty). \tag{54}$$

Hence:

$$\|\mathbf{P}(h)\|_2<1 \quad \forall h\in(0,\infty). \tag{55}$$

It can be proved in a similar way that

$$\|\mathbf{Q}(h)\|_2<1 \quad \forall h\in(0,\infty). \tag{56}$$

Hence:

$$\|\tilde{\mathbf{M}}_S(h)\|\leq\|\mathbf{P}(h)\|\|\mathbf{Q}(h)\|<1 \quad \forall h\in(0,\infty). \tag{57}$$

and:

$$\sigma(\tilde{\mathbf{M}}_S(h))=\sigma(\mathbf{M}_S(h))\subset B(0,1) \quad \forall h\in(0,\infty).$$

Remark

Note that we cannot prove any $\tilde{A}$ stability result for the Gauss–Jacobi and the Gauss–Seidel integration methods. In our practical experiments, we have seen that when applied to circuit problems, the modified symmetric Gauss–Seidel method is indeed 'more stable' than the other two methods.

Now we are going to discuss the accuracy of the integration methods presented in this paper. Once more, we are going to define accuracy in terms of the test problem (12).

Definition 5

Let $\mathbf{x}(t_k)$ be the exact value of the solution of the test problem at time t_k. Let $\mathbf{x}_k$ be the computed solution at time t_k assuming $\mathbf{x}_{k-1} = \mathbf{x}(t_{k-1})$ i.e. that no error has been made in computing the previous time point-value of $\mathbf{x}$. Letting $h = t_k - t_{k-1}$, the local truncation error is defined to be

$$\varepsilon = \|\mathbf{x}(t_k) - \mathbf{x}_k\| \tag{58}$$

If $\varepsilon = O(h^{r+1})$, r is said to be the order of the integration method.[13]

Theorem 5

The Gauss–Jacobi and Gauss–Seidel integration methods are first order integration algorithms.

Proof. From (58) we have:

$$\varepsilon = \|\mathbf{x}(t_k) - \mathbf{x}_k\| \tag{59}$$

$$= \|(\mathbf{e}^{h\mathbf{A}} - \mathbf{M})\mathbf{x}_{k-1}\| \tag{60}$$

By expanding $\mathbf{e}^{h\mathbf{A}}$ in power series of h and by Theorem 1,

$$\varepsilon = \|\{\mathbf{I} + h\mathbf{A} + O(h^2) - \mathbf{I} - h\mathbf{A} - O(h^2)\}\mathbf{x}_{k-1}\|$$
$$= O(h^2)$$

Theorem 6

The modified symmetric Gauss–Seidel algorithm is a second-order integration algorithm.

Proof. Since the matrices $[\mathbf{I} + \frac{h}{2}\mathbf{A}_L]$ and $[\mathbf{I} + \frac{h}{2}\mathbf{A}_L]^{-1}$ commute, then:

$$\mathbf{M}_S = \left[\mathbf{I} - \frac{h}{2}\mathbf{A}_U\right]^{-1}\left[\mathbf{I} - \frac{h}{2}\mathbf{A}_L\right]^{-1}\left[\mathbf{I} + \frac{h}{2}\mathbf{A}_L\right]\left[\mathbf{I} + \frac{h}{2}\mathbf{A}_U\right]$$
$$= \left[\mathbf{I} - \frac{h}{2}\mathbf{A} + \frac{h^2}{4}\mathbf{A}_L\mathbf{A}_U\right]^{-1}\left[\mathbf{I} + \frac{h}{2}\mathbf{A} + \frac{h^2}{4}\mathbf{A}_L\mathbf{A}_U\right]$$
$$= \mathbf{I} + h\mathbf{A} + \frac{h^2}{2}\mathbf{A} + O(h^3).$$

Hence

$$\varepsilon = \|(\mathbf{e}^{h\mathbf{A}} - \mathbf{M}_S)\mathbf{x}_{k-1}\| = O(h^3) \tag{65}$$

In circuit analysis, another important criterion for evaluating the accuracy of an integration method, is what we call 'waveform accuracy'. In general, the computed solution of a system of differential equations is the superposition of a principal solution and parasitic solutions.[13] Parasitic solutions are generated by the numerical approximations of the integration methods. In particular, an nth order integration algorithm yield $n-1$ parasitic solutions when applied to the test problem. For the algorithms we are dealing with in this paper, the displacement technique used introduce spurious components also that we shall call numerical solution components.

Proposition 2

Oscillatory numerical solution components are present in the computer solution if the spectrum of the companion matrix $\mathbf{M}(h)$ contains complex conjugate eigenvalues.

If the original system to be analysed does not contain an oscillatory component, the presence of such a component in the computed solution can be misleading in the evaluation of the performances of the

system.[16] Therefore we introduce a subclass of the test problem, characterized by $\sigma(\mathbf{A}) \subset R_0^-$: i.e. the set of test problems which does not have an oscillatory component in the solution, and we look for bounds on the oscillatory components of the computed solutions.

Theorem 1 gives a bound on the oscillatory components of all the methods. In particular it is obvious that, by choosing an appropriately small step-size h, the numerical solution oscillatory components can be made negligible with respect to the principal solution.

If we restrict the class of the test problems to the subclass characterized by a symmetric $\mathbf{A}$ matrix, then we can prove a much stronger result for the modified symmetric Gauss–Seidel integration method.

Theorem 7

If $\mathbf{A}$ is a real symmetric matrix, the spectrum of the companion matrix of the modified symmetric Gauss–Seidel integration method is real, i.e. no oscillatory parasitic components are present in the computed solution.

Proof. Let us factorize matrix $\mathbf{M}_S$ as in (63)

$$\mathbf{M}_S = \mathbf{PQ} \tag{66}$$

$$\mathbf{P} = \left[\mathbf{I} - \frac{h}{2}\mathbf{A} + \frac{h^2}{4}\mathbf{A}_L\mathbf{A}_U\right]^{-1} \tag{67}$$

$$\mathbf{Q} = \left[\mathbf{I} + \frac{h}{2}\mathbf{A} + \frac{h^2}{4}\mathbf{A}_L\mathbf{A}_U\right]$$

Since $\mathbf{A}_L\mathbf{A}_U$ is a positive semidefinite symmetric matrix and $-\mathbf{A}$ is symmetric and positive definite it follows that $\mathbf{P}$ is a symmetric positive definite matrix. The matrix $\mathbf{Q}$ is the sum of symmetric matrices, hence symmetric. Since

$$\mathbf{P} = \sum_{i=1}^{\sigma} \lambda_i \mathbf{R}_i \tag{69}$$

where λ_i are the eigenvalues and $\mathbf{R}_i$ are the residues of matrix $\mathbf{P}$, then

$$\mathbf{P}^{1/2} = \sum_{i=1}^{\sigma} \sqrt{\lambda_i}\,\mathbf{R}_i$$

$\mathbf{P}^{1/2}$ is a symmetric matrix, since the residues $\mathbf{R}_i$ are symmetric matrices. Let us consider now the similarity transformation:

$$\tilde{\mathbf{M}}_S = \mathbf{P}^{-1/2}\mathbf{M}_S\mathbf{P}^{1/2} \tag{71}$$

$$= \mathbf{P}^{1/2}\mathbf{Q}\mathbf{P}^{1/2} \tag{72}$$

The matrix $\tilde{\mathbf{M}}_S$ is symmetric and therefore has real eigenvalues. Then by similarity also $\mathbf{M}_S$ has real eigenvalues.

4. CONCLUSIONS

We have investigated the numerical properties of certain displacement techniques used for the timing analysis of VSLI, MOS circuits: the Gauss–Jacobi method used in MOTIS, the Gauss–Seidel method used in MOTIS-C and SPLICE, and a method proposed by Kahan called here modified symmetric Gauss–Seidel method. The algorithms have been discussed for circuits containing no floating capacitors. We have shown that from stability and accuracy viewpoint, the modified symmetric Gauss–Seidel integration algorithm outperforms the other two methods. When floating capacitors are present, the algorithms have to be modified to deal with the additional coupling between equations introduced by

the capacitors. The analysis of the modified algorithms is complex and is carried out in Reference 17, where experimental results are also presented and discussed.

ACKNOWLEDGEMENT

Research sponsored by the Defense Advanced Research Projects Agency (DoD), and by Rotary International. Monitored by Naval Electronic Systems Command under Contract No. N00039-K-0251.

REFERENCES

1. W. Nagel, 'SPICE2, a computer program to simulate semiconductor circuits'. University of California, Berkeley, *ERL Memo No. ERL-M520*, May 1975.
2. 'Advanced statistical analysis program (ASTAP)', *Program Reference Manual*, Pub. No. SH20-1118-0, IBM Corp. Data Proc. Div., White Plains, N.Y. 10604.
3. B. R. Chawla, H. K. Gummel and P. Kozak, 'MOTIS—an M.O.S. timing simulator', IEEE *Trans Circuits and Systems* CAS 22, 901–909, (1975).
4. M. Ortega and W. C. Reinboldt, '*Iterative Solutions of Non-Linear Equations in Several Variables*' Academic Press 1970.
5. S. P. Fan, M. Y. Hsueh, A. R. Newton, D. O. Pederson, 'MOTIS-C a new circuit simulator for MOS LSI circuits', *Proc. IEEE ISCAS*, 700–703 (1977).
6. A. R. Newton, 'The simulation of Large Scale Integrated Circuits', *Memorandum UCB/ERL M78/52*, July 1978.
7. A. R. Newton, 'Timing, logic and mixed-mode simulation for large M.O.S. integrated circuits,' *NATO ASI notes*, Sogeta, Urbino, Italy, 1980.
8. N. B. G. Rabbat, A. L. Sangiovanni-Vincentelli and H. Y. Hsieh, 'A multilevel Newton Algorithm with macro modeling and latency for the analysis of large scale nonlinear circuits in the time domain', *IEEE Trans. Circuits and Systems*, CAS-26, 733–741 (1979).
9. W. Kahan, private notes.
10. C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*, McGraw-Hill, 1969.
11. E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Wiley, 1966.
12. G. De Micheli, A. R. Newton and A. L. Sangiovanni-Vincentelli, 'New algorithms for timing analysis of large circuits', *International Symposium on Circuits and Systems*, Houston, April 1980.
13. C. W. Gear, *Numerical Initial Value Problems for Ordinary Differential Equations*, Prentice-Hall, 1971.
14. G. Dahlquist and A. Bjorck, *Numerical Methods*, Prentice-Hall, 1974.
15. C. A. Desoer, *Notes for a Second Course on Linear Systems*, D. Van Nostrand, 1970.
16. A. R. Newton, 'The timing analysis of floating capacitors for timing simulation', *Proc. 13th Asilomar Conference on Circuits Systems and Computers*, Asilomar, November 1979.
17. G. De Micheli, A. R. Newton and A. L. Sangiovanni-Vincentelli, 'Symmetric displacement algorithms for the timing analysis of VLSI MOS circuits', (in preparation).
18. I. W. Sandberg, 'Theorems on the computation of the transient response of nonlinear networks containing transistors and diodes', *Bell Syst. Tech. J.*, **49**, 1739–1776 (1970).
19. T. Roska and J. Klimo, 'On the solvability of DC equations and the implicit integration formula', *Int. J. Circuit Theory Appl.*, **1**, 273–280 (1973).
20. T. Roska, 'The limits of modeling of nonlinear circuits', *IEEE Trans. Circuits Systems*, CAS-28, 212–216 (1981).