January 1988

UILU-ENG-88-2209
DAC-11

COORDINATED SCIENCE LABORATORY
College of Engineering

ON THE CONVERGENCE OF BLOCK TIME-POINT RELAXATION METHODS FOR CIRCUIT SIMULATION

Madhav P. Desai
Ibrahim N. Hajj

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ILU-ENG-88-2209 (DAC-11)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois 6b. OFFICE SYMBOL (*If applicable*) N/A	7a. NAME OF MONITORING ORGANIZATION Semiconductor Research Corporation
6c. ADDRESS (*City, State, and ZIP Code*) 1101 W. Springfield Avenue Urbana, IL 61801	7b. ADDRESS (*City, State, and ZIP Code*) Research Triangle Park, NC 27709
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Semiconductor Research Corporation 8b. OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER SRC 86-12-109
8c. ADDRESS (*City, State, and ZIP Code*) Research Triangle Park, NC 27709	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)
On the Convergence of Block Time-point Relaxation Methods for Circuit Simulation

12. PERSONAL AUTHOR(S)
Desai, Madhav P. and Hajj, Ibrahim, N.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
Technical	FROM _____ TO _____	1988 January	36

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	VLSI circuits, circuit simulation, timing, relaxation methods, Gauss-Seidel, Gauss-Jacobi, time-point relaxation, convergence, partitioning

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Time-point relaxation methods have been in use in circuit simulators for a number of years. Typically, block time-point relaxation methods are used, where the circuit being analyzed is partitioned into several subcircuits or blocks, each consisting of several nodes. Before a relaxation method is used in a circuit simulator, its convergence properties need to be ascertained. Previously known sufficient conditions are too restrictive, in the sense that they do not take into account the effect of circuit topology on the convergence properties of relaxation algorithms. In this report, we develop new sufficient conditions for the convergence of the block Gauss-Seidel-Newton and the block Newton-Gauss-Seidel algorithms, when applied to circuit simulation. For a given partitioning of a circuit, we define a set of feedback nodes which capture the topology of the partitioned circuit to a certain extent. We then show that the G-S-N and the N-G-S algorithms converge under certain mild assumptions if there is a path consisting of capacitors from each feedback node to ground, and if the time-step of the (implicit) integration formula is small enough. We also provide an example of a linear circuit on which it is also necessary that the sufficient condition derived herein be satisfied to obtain convergence of the G-S-N and N-G-S algorithms. As corollaries to our main result, we obtain improved sufficient conditions for the Gauss-Jacobi-Newton and the Newton-Gauss-Jacobi algorithms and for circuits partitioned in the Bordered Block Diagonal Form.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (*Include Area Code*) / 22c. OFFICE SYMBOL

On the Convergence of Block Time-point Relaxation Methods for Circuit Simulation

Madhav P. Desai and Ibrahim N. Hajj

Coordinated Science Laboratory

and the Department of Electrical Engineering

University of Illinois

Urbana-Champaign

Abstract

Time-point relaxation methods have been in use in circuit simulators for a number of years. Typically, block time-point relaxation methods are used, where the circuit being analyzed is partitioned into several subcircuits or blocks, each consisting of several nodes. Before a relaxation method is used in a circuit simulator, its convergence properties need to be ascertained. Previously known sufficient conditions are too restrictive, in the sense that they do not take into account the effect of circuit topology on the convergence properties of relaxation algorithms. In this paper, we develop new sufficient conditions for the convergence of the block Gauss-Seidel-Newton and the block Newton-Gauss-Seidel algorithms, when applied to circuit simulation. For a given partitioning of a circuit, we define a set of feedback nodes which capture the topology of the partitioned circuit to a certain extent. We then show that the G-S-N and the N-G-S algorithms converge under certain mild assumptions if there is a path consisting of capacitors from each feedback node to ground, and if the time-step of the (implicit) integration formula is small enough. We also provide an example of a linear circuit on which it is also necessary that the sufficient condition derived herein be satisfied to obtain convergence of the G-S-N and N-G-S algorithms. As corollaries to our main result, we obtain improved sufficient conditions for the Gauss-Jacobi-Newton and the Newton-Gauss-Jacobi algorithms and for circuits partitioned in the Bordered Block Diagonal Form.

1. Introduction

A circuit simulator is used to analyze the behavior of circuits over a time period, given a set of input signals. The circuit being analyzed is described by a set of algebraic differential equations:

$$G\,(\,\dot{y}(t),\,y(t),\,u(t)\,)=0 \qquad t\in[0,T] \tag{1.1a}$$

$$y(0)=y_0 \tag{1.1b}$$

In (1.1a), $u(t)\in R^p$ is the vector of input signals (currents or voltages), $y(t)\in R^n$ is the vector of circuit variables, and $\dot{y}(t)$ is the time derivative of $y(t)$. $G: D_G\subseteq R^n\times R^n\times R^p\rightarrow R^n$ is the function that describes the behavior of the circuit. The details of the formulation of G for a circuit will be explained in Section 2. In most conventional circuit simulators such as SPICE [1], or SLATE [2], the set of algebraic differential equations (1.1a) is discretized by means of an integration formula [3], and an approximation of (1.1a) at a set of discrete time-points $0=t_1<t_2<\cdots<t_N=T$ is obtained. If, for example, the Backward Euler Formula is used to discretize (1.1a), the transformed equations at time-point t_q are

$$G\,(\,\frac{y(t_q)-y(t_{q-1})}{h_q}\,,\,y(t_q)\,,\,u(t_q)\,)=0\,, \tag{1.2}$$

where $h_q=t_q-t_{q-1}$ is the time-step of the integration formula, chosen so that the local truncation error is small [3].

At time point t_q, we can write (1.2) in the form

$$F\,(x)=0. \tag{1.3}$$

where $F: D_F\subset R^n\rightarrow R^n$ is a nonlinear algebraic function of $x=y(\,t_q\,)$. It can be noted here that F at time t_q is defined by the choice of h_q and the input $u\,(t_q)$. This implicit dependence of F on h_q and $u(t_q)$ will be made clear when we discuss the details of the equation formulation in Section 2. If the Newton-Raphson (N-R) algorithm is used to solve (1.3), then at each Newton-Raphson iteration, a system of linear equations of the form

$$A\,z = b \tag{1.4}$$

is obtained. The $n \times n$ matrix A is the jacobian of F evaluated at the previous iterate in the N-R algorithm. The solution z of (1.4) gives the next iterate in the N-R algorithm.

The main result in this paper is a new sufficient condition for the convergence of certain relaxation algorithms used to solve (1.3). We consider two relaxation algorithms, the Gauss-Seidel-Newton (G-S-N) algorithm and the Newton-Gauss-Seidel (N-G-S) algorithm. The G-S-N algorithm uses relaxation at the nonlinear level to solve (1.3). That is, the system of nonlinear equations in (1.3) is partitioned into subsystems, and the solution of the entire system is obtained by solving the subsystems in a certain sequence. The solution of each subsystem is obtained by using a N-R method, and during the solution of each subsystem, the other subsystems are relaxed. The G-S-N method has been used in circuit simulators such as SPLICE [4]. The N-G-S algorithm uses relaxation to solve the linear systems in (1.4) arising out of a N-R algorithm used to solve (1.3). Thus, N-G-S uses a relaxation method at the linear level. The details of the G-S-N and the N-G-S algorithms will be described in Section 2.

We consider block partitioning, where the circuit being analyzed is partitioned into subcircuits consisting of several nodes each, which then serve as the subsystems in the relaxation method. In Section 3, we introduce the concept of a feedback node or variable for a given partitioning of a circuit (or system). The set of feedback nodes captures some of the properties of the interconnection between the partitioned subcircuits (or subsystems). Our main result, which is stated and proved in Section 4, shows that under certain mild assumptions, the G-S-N and the N-G-S methods converge for a given partitioning of the circuit if there is a path consisting entirely of capacitors from each feedback node to ground, and if the time-step is sufficiently small. Existing convergence results [5] provide sufficient conditions that do not take into account the effect of circuit partitioning on the convergence of the relaxation methods, and require a capacitor to ground at *every* node in the circuit and a sufficiently small time-step. The sufficient conditions that we derive are less restrictive, because the set of feedback nodes for a given partitioning may be much smaller than the

set of all nodes in the circuit. Indeed, for unidirectional circuits, it is possible to partition the circuit such that there are no feedback nodes at all in the circuit, and the G-S-N and the N-G-S methods always converge (in one iteration) without the requirement of having a capacitor from any node to ground, as can be inferred from our sufficient condition. Related work on block relaxation algorithms to solve load flow problems can be found in [6]. However, the sufficient condition derived in [6] requires a certain matrix to be *block diagonally dominant*, [7, 8] and it is not clear how this can be achieved in the case of circuits. Our work provides a simple sufficient condition for convergence that takes into account the effect of circuit partitioning.

2. Mathematical Formulation of the Circuit Equations

Consider a circuit described by the nodal analysis formulation [9].

$$\dot{q}(t) + J(x(t), u(t)) = 0, \quad q, x \in R^n \tag{2.1a}$$

$$q(t) = f(x(t)). \quad f : D_f \subset R^n \to R^n. \tag{2.1b}$$

Here, x represents the vector of node voltages in the circuit, and q is the vector of charges at the nodes of the circuit. $J : D_J \subset R^n \times R^p \to R^n$ is the function describing the currents entering each node. After applying the backward Euler formula to discretize $\dot{q}(t)$ at time t_q with time-step $h = t_q - t_{q-1}$, we get

$$\frac{q(t_q) - q(t_{q-1})}{h} = \frac{f(x(t_q)) - f(x(t_{q-1}))}{h} = -J(x(t_q), u(t_q)) \tag{2.2}$$

which can be written as

$$F(x) = \frac{f(x)}{h} + J(x, u(t_q)) + d = 0 \tag{2.3}$$

where $x = x(t_q)$, and d is a constant vector dependent on $x(t_{q-1})$. The jacobian of F can then be written as

$$\frac{\partial F}{\partial x} = \frac{1}{h}\frac{\partial f}{\partial x} + \frac{\partial J}{\partial x}. \tag{2.4}$$

The first term on the right hand side of (2.4) can be expressed as C/h, where C is an $n \times n$ *incremental* capacitance matrix. The entries of C are

$$C_{ij} = -(\textit{incremental capacitance between node } i \text{ and } \textit{node } j) \leq 0. \tag{2.5}$$

$$C_{ii} = (\textit{Sum of incremental capacitors at node } i) \geq 0.$$

We shall assume that there are no negative incremental capacitances. We do not require the capacitance matrix to be symmetric, that is, C_{ij} and C_{ji} need not be equal. However, the capacitance matrix satisfies the following property: For each $i \in 1, \cdots, n$,

$$C_{ii} + \sum_{j \neq i} C_{ij} = C_{i0} \geq 0. \tag{2.6}$$

where C_{i0} is the incremental capacitance from node i to ground. It is possible that $C_{i0} = 0$ if there

is no capacitor to ground at node i.

From this brief discussion of the formulation of the circuit equations, it is evident that the function F in (2.3) and its jacobian depend heavily on the time-step h and on the capacitances in the circuit. In fact as h is reduced, the capacitances play an increasing role in determining the properties of F. Our convergence results in Section 4 will show that a certain configuration of capacitances in the circuit, together with a small enough time-step, will be sufficient to make the G-S-N and the N-G-S algorithms converge for a given partitioning of the circuit. For the purpose of our analysis, we shall assume that the backward Euler formula is used to discretize the circuit. The use of any other implicit integration formula will not change the results obtained. The essential property we use is that the effect of the capacitances increases as the time-step is reduced.

We introduce some notations. Assign each variable in x to an equation in F. Now consider any partitioning of x into:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_m \end{bmatrix}, \tag{2.7a}$$

where

$$x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \cdot \\ \cdot \\ x_{i,n_i} \end{bmatrix} \qquad i = 1, \cdots, m \tag{2.7b}$$

This partitioning of x induces a corresponding partitioning of F as follows:

$$\begin{aligned} F_1(x_1, x_2, \cdots, x_m) &= 0 \\ F_2(x_1, x_2, \cdots, x_m) &= 0 \\ &\cdot \\ &\cdot \\ F_m(x_1, x_2, \cdots, x_m) &= 0 \end{aligned} \tag{2.8a}$$

$$F_i = \begin{bmatrix} F_{i,1} \\ F_{i,2} \\ \cdot \\ \cdot \\ F_{i,n_i} \end{bmatrix} \tag{2.8b}$$

In (2.8), $F_i : R^{n_i} \rightarrow R^n$ consists of the equations assigned to the variables in x_i.

For this partitioning of the equations in F, we describe the G-S-N and the N-G-S algorithms.

2.1: The Gauss-Seidel-Newton Algorithm

The G-S-N algorithm uses relaxation at the nonlinear level to solve (2.3) partitioned as in (2.8).

Algorithm (G-S-N):

1) Set $k = 0$. Guess $x_i^{(k)}$, $i = 1, \cdots, m$. Typically, this initial guess is the solution at the previous time-point.

2) For $i = 1, \cdots, m$, perform a single N-R iteration in the solution of

$$F_i(x_1^{(k+1)}, x_2^{(k+1)}, \cdots, x_{i-1}^{(k+1)}, \hat{x}_i, x_{i+1}^{(k)}, \cdots, x_m^{(k)}) = 0 \tag{2.9}$$

to obtain $\hat{x}_i$. Set $x_i^{(k+1)} = \hat{x}_i$. (Note that the initial guess in the solution of (2.9) for $\hat{x}_i$ is $x_i^{(k)}$.)

3) Set $k = k + 1$. If the error $\| x^{(k)} - x^{(k-1)} \|$ is small enough, then stop. Otherwise, go to step 2.

Step 2 may be modified such that the subsystems F_i as shown in (2.9) are solved exactly, or up to a fixed number (≥ 1) of N-R iterations. This modification does not alter the sufficient conditions for convergence that we derive later.

2.2: The Newton-Gauss-Seidel Algorithm

The N-G-S algorithm has two iteration loops. The outer iteration loop is similar to the outer loop for the N-R method used to solve the nonlinear system in (2.3). The inner loop is the

relaxation loop, wherein a relaxation procedure is used to solve the linear systems resulting from linearization of F in the outer N-R loop.

Algorithm (N-G-S):

1) Set $k = 0$; Guess $x_i^{(k)}$ for $i = 1, \cdots, m$. As in the G-S-N case, this initial guess is the solution at the previous time-point.

2) Linearize F at $x^{(k)}$ to get the system of linear equations,

$$\begin{bmatrix} A_{11} & A_{12} & . & . & A_{1m} \\ A_{21} & A_{22} & . & . & A_{2m} \\ . & . & . & . & . \\ . & . & . & . & . \\ A_{m1} & A_{m2} & . & . & A_{mm} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ z_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ b_m \end{bmatrix}, \tag{2.10}$$

where $A_{ij} = \dfrac{\partial F_i}{\partial x_j}$, evaluated at $x^{(k)}$.

3) Set $l = 0$. Guess $z_i^{(l)}$, $i = 1, \cdots, m$. The initial guess is $z_i^{(0)} = x_i^{(k)}$.

4) For $i = 1, 2, \cdots, m$, solve

$$A_{ii}\, \hat{z}_i = b_i - \sum_{j<i} A_{ij}\, z_j^{(l+1)} - \sum_{j>i} A_{ij}\, z_j^{(l)} \tag{2.11}$$

Set $z_i^{(l+1)} = \hat{z}_i$.

5) Set $l = l + 1$. If the iterates $z^{(l)}$ have converged or if $l \geq MAXIT$, then go to step 6. Otherwise, go to step 4.

6) For $i = 1, \cdots, m$, set $x_i^{(k+1)} = z_i^{(l)}$.

7) Set $k = k + 1$. If the iterates $x^{(k)}$ have converged, stop. Otherwise, go to step 2.

The positive integer $MAXIT$ determines the maximum number of times the inner loop (steps 4,5) is executed. The conditions for convergence we derive in Section 4 are independent of $MAXIT$. However, experimental evidence shows that the *speed* of convergence is dependent on $MAXIT$ [10].

A detailed description of the G-S-N and the N-G-S algorithms is given in [11]. The G-S-N algorithm is essentially an SOR-Newton algorithm with overrelaxation parameter set to 1 and the N-G-S algorithm is a Newton-SOR algorithm with overrelaxation parameter set to 1.

3. Feedback Variables

We introduce the concept of a *feedback* variable or node in a partitioned circuit. Consider the partitioning of the circuit variables as in (2.6), (2.7), and the corresponding partitioning of the circuit equations as in (2.8). Then, we have the following definition:

Definition 1:

A circuit variable $x_{k,l}$ with $1 \leqslant k \leqslant m$ and $1 \leqslant l \leqslant n_k$ is called a feedback variable if there exist $i, j, k < i \leqslant m, 1 \leqslant j \leqslant n_i$, and $z \in D_F$, such that

$$\frac{\partial F_{k,l}}{\partial x_{i,j}}(z) \neq 0.$$

The corresponding circuit node is then called a feedback node.

In other words, given a partitioning of a circuit into blocks ordered in a certain way, a node k,l inside a block k is a feedback node if the equation k,l is a function of some node i,j in block i with $i > k$. That is, node k,l receives *feedback* from some node in a higher numbered block. An analogous concept is that of a *feed-forward* node, which is defined below.

Definition 2:

A circuit variable $x_{k,l}$ with $1 \leqslant k \leqslant m$ and $1 \leqslant l \leqslant n_k$ is called a feed-forward variable if there exist $i, j, 1 \leqslant i < k, 1 \leqslant j \leqslant n_i$, and $z \in D_F$, such that

$$\frac{\partial F_{k,l}}{\partial x_{i,j}} \neq 0.$$

The corresponding circuit node is then called a feed-forward node.

The circuit equation at a feed-forward node is thus a function of some node in a lower numbered block; that is, there is feed-forward from a lower numbered block to a feed-forward node.

A couple of examples will help in explaining these concepts. Consider Figures 1 and 2. In Figure 1, node 2 is a feedback node for the partitioning shown, and node 3 is a feed-forward node.

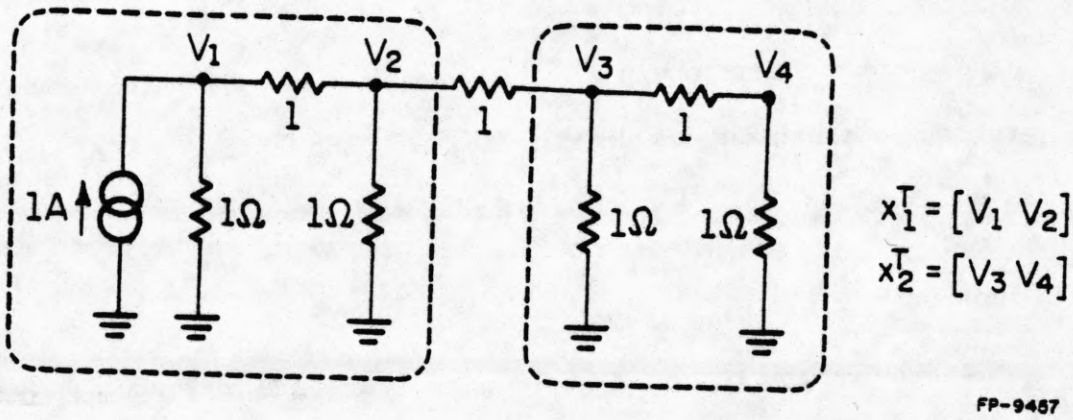

Figure 1

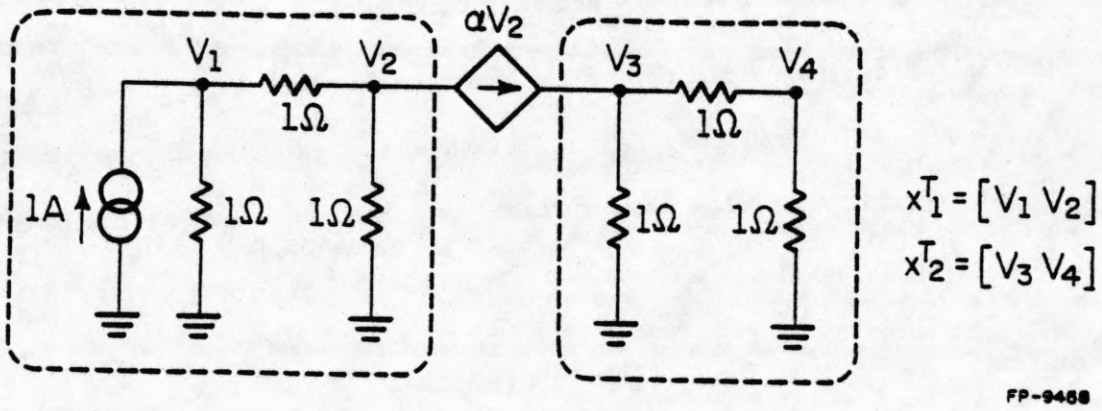

Figure 2

Now in Figure 2, for the partitioning shown, there are no feedback nodes at all, whereas nodes 3 and 5 are feed-forward nodes.

There is a simple technique to detect all the feedback and feedforward nodes in a circuit for a given partitioning of the circuit into blocks, and an ordering of the blocks. First, we construct a directed graph $G = (V, E)$ with vertex set V being the set of nodes in the circuit, and arc set defined by

$$E = \{ (u, v) : \text{node } u \text{ is "affected" by node } v \}.$$

By a node u being affected by node v, we mean that the circuit equation at node u is a function of the node voltage at node v. For the partitioning and ordering of the blocks of the circuit, define a labeling function $L: V \rightarrow 1, \cdots, m$, where $L(u)$ is the number of the block to which u belongs. Then u is a feedback node if there is some node v such that (v,u) is an arc in E, and $L(v) > L(u)$. u is a feed-forward node if there is some node v with $L(v) > L(u)$ such that (v,u) is an arc in E. If the blocks of the partitioned circuit are chosen so that they are the strongly connected components of G, then it is possible to order them so that there are no feedback nodes at all [12].

4. Convergence Properties

An extensive study of iterative algorithms such as the G-S-N and the N-G-S algorithms can be found in [11]. The general sufficient conditions for the convergence of these methods require the spectral radius of a certain matrix to be less than 1. For a nonlinear system such as a circuit, it is difficult to predict if the spectral radius of a certain matrix associated with the circuit will be less than one. However, it is possible to show, as we do in this section, that if the topology of the circuit satisfies certain properties then the spectral radius of the required matrix will be less than one if the time-step is sufficiently small. Specifically, we require the existence of capacitors to ground at certain nodes in the circuit, and not necessarily at every node.

Let

$$H(z) = H_L^{-1}(z)\, H_U(z), \tag{4.1}$$

where

$$H_L(z) = \begin{bmatrix} A_{11}(z) & 0 & \cdot & \cdot & 0 \\ A_{21}(z) & A_{22}(z) & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{m1}(z) & A_{m2}(z) & \cdot & \cdot & A_{mm}(z) \end{bmatrix} \tag{4.2a}$$

$$H_U(z) = \begin{bmatrix} 0 & A_{12}(z) & \cdot & \cdot & A_{1m}(z) \\ 0 & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & A_{(m-1)m}(z) \\ 0 & 0 & \cdot & \cdot & 0 \end{bmatrix} \tag{4.2b}$$

and A_{ij} is defined as in Section 2.

We shall state the theorems proven in [11]. These theorems establish sufficient conditions for the convergence of the G-S-N and N-G-S algorithms in terms of the spectral radius of the matrix H as defined by (4.1), (4.2).

Theorem 4.1 (Gauss-Seidel-Newton Theorem):

Given $F:D_F \subset R^n \rightarrow R^n$, assume that $F(x^*)=0$ for some interior point of D_F and that F is continuously differentiable on an open neighborhood $S \subset D_F$ of x^*. Consider a partition of x as given in (4.2.5). Assume that $A_{ii}(x^*): i=1,...,m$ are nonsingular and that $\rho(H(x^*))<1$ where $H(x^*)$ is defined as in (4.3.1) and $\rho(H)$ is the spectral radius of H. Let $x^{k+1}=g(x^k)$ be the mapping defined by the Gauss-Seidel-Newton iteration. Then, the mapping $x^{k+1}=g(x^k)$ is well defined on an open ball $S \subset S_0$, and x^* is a point of attraction of the iteration $x^{k+1}=g(x^k)$.

Proof: The proof is discussed in detail in [11], pages 323-325.

Theorem 4.2 (Newton-Gauss-Seidel Theorem): Let $F:D_F \subset R^n \rightarrow R^n$ be G-differentiable in an open neighborhood $S_0 \subset D_F$ of a point $x^* \epsilon D_F$ at which the jacobian of F is continuous and $F(x^*) = 0$. Suppose $H_L(x^*)$ as defined by (4.3.1) is nonsingular and that $\rho(H(x^*))<1$. Then, for any $K \geq 1$, x^* is a point of attraction of the Newton-Gauss-Seidel iteration.

Proof: The proof is discussed in detail in [11], pages 326-328.

We next state and prove the main result (Theorem 4.3) in this paper.

Main Result

Theorem 4.3: Let F be a system of nonlinear equations in (2.3) partitioned as in (2.8). Let x^* be the solution of (2.3). Let F be continuously differentiable at x^*, and let $A(x^*) = \frac{\partial F}{\partial x}(x^*)$. Assume the following:

(1) All principal square submatrices of $A(x^*)$ are invertible, and their inverses have bounded norms.

(2) The jacobian $\partial J / \partial x$ is independent of the time-step and has bounded entries.

(3) There is a path consisting entirely of capacitors from each feedback node to ground.

Then,

i) The iterates in the $G-S-N$ algorithm converge to x^* if the time-step h is small enough, and if the initial guess is sufficiently close to x^*.

ii) If F is G-differentiable at x^*, then the N-G-S method converges if the time-step h is small enough, and if the initial guess is sufficiently close to the solution x^*.

Proof:

We introduce the concept of *capacitor connected components* of a circuit.

Definition 4.1:

Let $G_C = (V, E_C)$ be a graph with V being the set of nodes in the circuit, and with nodes $(u, v) \in E_C$ if there is a capacitor between them in the circuit. Then the set of nodes corresponding to a connected component of G is a capacitor connected component of the circuit. The graph G_C is called the graph induced by the capacitors of the circuit.

The following example will clarify the above Definition. Consider the circuit shown in Figure 3a. The corresponding graph G_C is shown in Figure 3b. The capacitor connected components are then {1,2} and {3,4}.

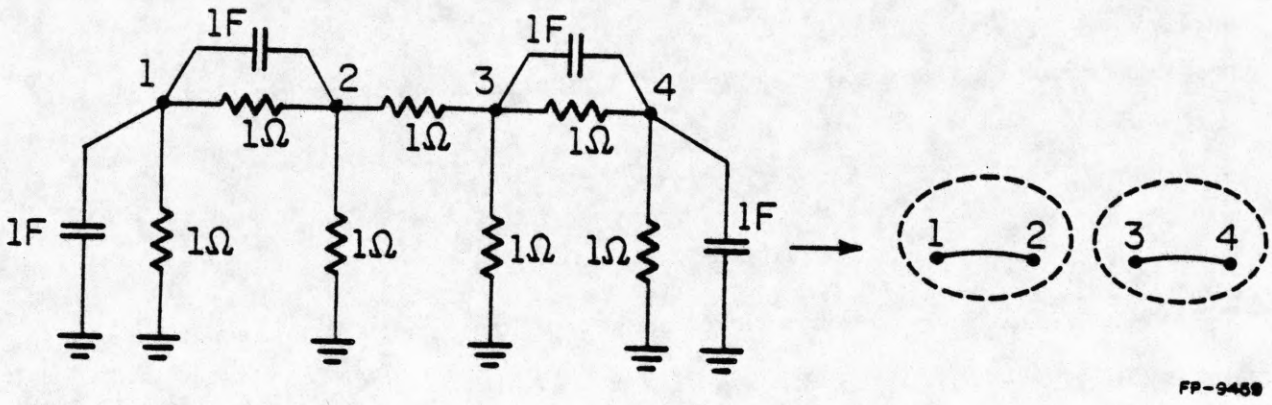

Figure 3a

Figure 3b

Let $x_{F,1}, x_{F,2}, \cdots, x_{F,K}$ be the vectors of circuit variables with the following properties:

(1) For each $i = 1, \cdots, K$, $x_{F,i}$ contains variables whose corresponding nodes form capacitor connected components of the circuit.

(2) Each $x_{F,i}$, $i = 1, \cdots, K$, contains at least one feedback variable defined by the partition of the circuit.

(3) Each feedback variable belongs to exactly one of $x_{F,1}, \cdots, x_{F,K}$.

Let $x_F^T = [\, x_{F,1}^T \ \cdots \ x_{F,P}^T \,]$. Let the vector x_N contain all variables in x that are not in x_F. Choose a permutation matrix P, such that

$$P^T x = \begin{bmatrix} x_N \\ x_F \end{bmatrix} \tag{4.3}$$

Then it is easy to see that,

$$P^T H_L P = \begin{bmatrix} H_{NN,L} & H_{NH,L} \\ H_{FN,L} & H_{FF,L} \end{bmatrix} \tag{4.4a}$$

$$P^T H_U P = \begin{bmatrix} 0 & 0 \\ H_{FN,U} & H_{FF,U} \end{bmatrix} \tag{4.4b}$$

The submatrices on the right hand side of (4.4a), (4.4b) have certain properties. First of all,

$$H_{FF,L} = \frac{1}{h} C_{FF,L} + D_{FF,L} \tag{4.5a}$$

$$H_{FF,U} = \frac{1}{h} C_{FF,U} + D_{FF,U} \tag{4.5b}$$

where $C_{FF,L} + C_{FF,U} = C_{FF}$ is the restriction of the capacitance matrix C to the nodes in x_F. The matrices $D_{FF,L}$, $D_{FF,U}$, $H_{NF,L}$, $H_{FN,L}$ and $H_{FN,U}$ do not contain any entries that are contributed by capacitors, and hence their entries are bounded, irrespective of the time-step. The matrices $H_{NN,L}$ and $H_{FF,L}$ have inverses which have bounded norms.

Since P is a permutation matrix, $P^{-1} = P^T$. Hence

$$\rho((P^T H_L P)^{-1} (P^T H_U P)) = \rho(H_L^{-1} H_U) \tag{4.6}$$

Note that for any $h > 0$,

$$(P^T H_L P)^{-1} (P^T H_U P) = (hP^T H_L P)^{-1} (hP^T H_U P) \tag{4.7}$$

Using (4.4a), (4.4b) and (4.7) together with extensive algebraic manipulation, we get

$$(P^T H_L P)^{-1} (P^T H_U P) = \begin{bmatrix} L & M \\ S & T \end{bmatrix} \tag{4.8}$$

where,

$$L = h (H_{NN,L} - H_{NF,L} H_{FF,L}^{-1} H_{FN,L})^{-1} H_{NF,L} H_{FF,L}^{-1} H_{FN,U} \tag{4.9}$$

$$M = h (H_{NN,L} - H_{NF,L} H_{FF,L}^{-1} H_{FN,L})^{-1} H_{NF,L} H_{FF,L}^{-1} H_{FF,U} \tag{4.10}$$

$$S = h\,(h\,H_{FF,L} - h\,H_{FN,L}H_{NN,L}^{-1}H_{NF,L})^{-1}H_{FN,U} \tag{4.11}$$

$$T = (hH_{FF,L} - hH_{FN,L}H_{NN,L}^{-1}H_{NF,L})^{-1}\,h\,H_{FF,U} \tag{4.12}$$

We show in Claim 1. that as $h \to 0$. L. M. and S tend to 0. and T tends to a certain matrix. We will need the following Lemma to prove Claim 1.

Lemma 4.1

Let $B = [b_{ij}]$ be an $N \times N$ matrix with the following properties:

$$|b_{ii}| \geq \sum_{j \neq i} |b_{ij}| > 0; \quad i = 1, \cdots, n \tag{4.13}$$

such that strict inequality holds in (4.13) for at least one row of B. Let W be the set of rows where equality holds in (4.13), and let Y be the set of rows where strict inequality holds in (4.13). Let $G = (V, E)$ be the directed graph of B. that is $V = \{1, \cdots, N\}$. and for $i \neq j$. $(i, j) \in E$ if and only if $b_{ij} \neq 0$. Assume that for each $i \in W$. there exists a $k \in Y$ such that there is a path from i to k in G. Let D be the diagonal matrix whose entries are the diagonal elements of B. Then

$$\rho(I - D^{-1}B) < 1. \tag{4.14}$$

Further, if $b_{ii} > 0$. for all i. and $b_{ij} \leq 0$ for all $i \neq j$ then $B^{-1} \geq 0$. (That is, each entry of B^{-1} is non-negative.)

Proof of Lemma 4.1: The proof is given in the Appendix.

We use Lemma 4.1 to prove the following claim.

Claim 1:

$$\lim_{h \to 0} L = 0 \tag{4.13}$$

$$\lim_{h \to 0} M = 0 \tag{4.14}$$

$$\lim_{h \to 0} S = 0 \tag{4.15}$$

$$\lim_{h \to 0} T = C_{FF,L}^{-1} C_{FF,U} \tag{4.16}$$

Proof of Claim 1: It can be easily seen from (4.9) and (4.10) that (4.13) and (4.14) are true. It remains to show that (4.15), (4.16) are true. From (4.5a), (4.5b), and (4.11), we can see that if $C_{FF,L}$ is invertible, then

$$\lim_{h \to 0} S = \lim_{h \to 0} h\, C_{FF,L}^{-1} H_{FN,U} = 0. \tag{4.17}$$

To see that $C_{FF,L}$ is invertible, note that $C_{FF,L} + C_{FF,U}$ is a splitting of C_{FF} such that the diagonal elements of C_{FF} are the diagonal elements of $C_{FF,L}$. In addition, C_{FF} is a block diagonal matrix with the diagonal blocks corresponding to the variables $x_{F,1}, \cdots, x_{F,P}$. Since each $x_{F,i}$ contains at least one feedback node, and each feedback node is connected to ground by a path of capacitors, each diagonal block is irreducibly diagonally dominant [13]. The only nonzero rows in the matrix $C_{FF,U}$ correspond to the feedback nodes. Each block of $C_{FF,L}$ is diagonally dominant. Let i be a row in C_{FF} such that the diagonal dominance in row i is not strict. Then, there is a path $ii_1i_2 \cdots j$ in the graph of C_{FF} with strict diagonal dominance in row j. let k be the smallest index such that the $i_k i_{k+1}$ entry of $C_{FF,L}$ is zero. Then clearly there is a path $ii_1 \cdots i_k$ in $C_{FF,L}$, and strict diagonal dominance exists in row i_k. Thus $C_{FF,L}$ satisfies the assumptions of Lemma 4.1. From Lemma 4.1, each diagonal block of $C_{FF,L}$ is invertible. Thus $C_{FF,L}$ is invertible. This proves (4.15) and (4.16) and consequently, Claim 1. □

To prove Theorem 4.3, we need the following claim:

Claim 2:

$$\rho(\, C_{FF,L}^{-1} C_{FF,U}\,) < 1. \tag{4.18}$$

Proof of Claim 2: From the proof of Claim 1, we know that $C_{FF} = C_{FF,L} + C_{FF,U}$ is block diagonal, with each diagonal block being irreducibly diagonally dominant. Also, since the matrix C_{FF} is the restriction of a capacitance matrix to the set of nodes x_F, it has positive diagonal entries and nonpositive off-diagonal entries. Thus, C_{FF} is a block diagonal matrix, with each diagonal block

being an $M-$ matrix [11]. In the proof of Claim 1, we have seen that $C_{FF\,L}$ satisfies the assumptions of Lemma 4.1, and thus $C_{FF\,L}$ is invertible, and $C_{FF\,L}^{-1} \geqslant 0$. Thus the splitting $C_{FF\,L} + C_{FF\,U}$ is a regular splitting of the $M-$matrix C_{FF} [11]. Thus, from [11] page 56, we have

$$\rho(C_{FF\,L}^{-1} C_{FF\,U}) < 1. \tag{4.19}$$

This proves Claim 2. □

The assertions of Claim 1 and Claim 2 together with (4.6) and (4.8) imply that

$$\lim_{h \to 0} \rho (H_L^{-1} H_U) < 1. \tag{4.19}$$

Since the spectral radius of a matrix is a continuous function of its entries, (4.19) implies that there exists a time-step $h > 0$ such that

$$\rho (H_L^{-1} H_U) < 1. \tag{4.20}$$

Theorem 4.3 is now essentially proven. Part a) is immediate from (4.20) and Theorem 4.1. Part b) is immediate given the assumption of G-differentiability together with (4.20) and Theorem 4.2. □

Results similar to Theorem 4.3 can be obtained for the Gauss-Jacobi-Newton (G-J-N) algorithm and the Newton-Gauss-Jacobi (N-G-J) algorithm in which the relaxation method used is the Gauss-Jacobi method rather than the Gauss-Seidel method. As a corollary to Theorem 4.3, we have

Corollary 4.1: Let F, x^* and $A(x^*)$ be as defined in Theorem 4.3. In addition to the assumptions of Theorem 4.3, assume that there is a path consisting of capacitors from each *feed-forward* node to ground.

Then,

i) The iterates in the G-J-N algorithm converge to x^* if the time-step h is small enough, and if the initial guess is sufficiently close to x^*.

ii) If F is G-differentiable at x^*, then the N-G-J method converges if the time-step h is small enough, and if the initial guess is sufficiently close to the solution x^*.

Proof: The proof of Corollary 1 is identical to that of Theorem 4.3 with the feedback variable/node being replaced by " feedback or feedforward variable/node ".

A special partition of the circuit is the *Bordered Block Diagonal Form (BBDF)*. The circuit is partitioned into two subsystems: one subsystem corresponds to a set of tearing nodes whose removal causes the circuit to be decomposed into independent components which constitute the other subsystem [14]. The circuit variables are thus partitioned into

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4.21}$$

where x_2 corresponds to the set of tearing nodes, and x_1 corresponds to the rest of the circuit. This is essentially a partition of the circuit variables as in (2.6) with m =2. The jacobian A then has the form

$$\frac{\partial F}{\partial x} = A = \begin{bmatrix} A_{11}^1 & & & A_{12} \\ & A_{11}^2 & & \\ & & \ddots & \\ & & A_{11}^\ell & \\ A_{21} & & & A_{22} \end{bmatrix} \tag{4.22}$$

where the $A_{11}^1, \ldots, A_{11}^\ell$ correspond to the independent components of the circuit left after the tearing nodes are removed from the circuit. The matrix A is thus block diagonal with the borders A_{12} and A_{21} tacked on. This leads to the name Bordered-Block Diagonal Form for the partitioning scheme. Details of the BBDF form and means to achieve such partitions can be found in [14, 15]. The feedback nodes defined by this partition would lie entirely within the subsystem x_1, and the feed-forward nodes would lie in x_2. However Corollary 4.2 shows that in the case of a circuit partitioned as in (4.21), capacitances to ground at tearing nodes, together with other assumptions as in

Theorem 4.3, suffice for the convergence of the G-S-N and the N-G-S algorithms on circuits partitioned as in the BBDF. This is surprising, since the tearing nodes contain the feed-forward nodes defined by the partition, and not the feedback nodes. Intuitively, the explanation for this is that when x is partitioned into two systems, alternately solving for x_1 and x_2 may be viewed as solving for x_1 first, then x_2 and so on, or solving for x_2, then x_1 and so on.

Corollary 4.2

Let F, x^*, and $A(x^*)$ be as defined in the statement of Theorem 4.3. Consider a BBDF partitioning of F and x. In addition to the assumptions of Theorem 4.3, assume that there is a path consisting of capacitors from each *tearing* node to ground.

Then,

i) The iterates in the G-S-N algorithm converge to x^* if the time-step h is small enough, and if the initial guess is sufficiently close to x^*.

ii) If F is G-differentiable at x^*, then the N-G-S method converges if the time-step h is small enough, and if the initial guess is sufficiently close to the solution x^*.

Proof: The proof of Corollary 2 is similar to that of Theorem 4.3.

It is to be expected that the partitioning of a circuit can play an important part in the convergence properties of the G-S-N and the N-G-S algorithms applied to solving the circuit. The results in this Section justify this intuition. Every partitioning of a circuit determines a set of *feedback* nodes of the circuit. If paths to ground consisting *entirely of capacitors* exist at these nodes, convergence of the G-S-N and the N-G-S algorithms is guaranteed if the time-step is sufficiently small. In case of the BBDF partitioning scheme, the *tearing* nodes play the part of the feedback nodes. In the next Section we give an example of a circuit for which the condition of Theorem 4.3 is also *necessary* for the G-S-N method to converge for some time-step. Thus, the sufficient conditions

derived here are the best possible in the sense that they are *topological*, that is they require the topology of the circuit to satisfy certain constraints.

5. Discussion and Examples

The main result in this paper (Theorem 4.3) basically says that all nodes in a circuit are not equal in their effect on the convergence properties of the G-S-N and the N-G-S methods used to solve the circuit. Earlier convergence results on block relaxation methods of the type we have discussed did not bring out this fact. In the work of [5], the sufficient condition that was derived did not take into account the effect of the partitioning of the circuit on the convergence properties of the G-S-N and N-G-S methods. Conditions on the convergence of block relaxation methods for the solution of load flow problems in power systems analysis have been studied in [6]. However, this work defines a concept of *block diagonal dominance*, and it is not clear what properties the circuit needs to satisfy for the matrix A in (2.10) to be block diagonally dominant.

Our work essentially contributes an *easily testable* sufficient condition for convergence, that also takes into account the partitioning of the circuit. We have introduced the concept of feedback and feedforward nodes which capture, to a certain extent, the topological properties of the partitioned circuit. We have then shown that these sets of nodes play an important part in the convergence of the G-S-N and the N-G-S methods (Theorem 4.3, Corollary 4.1). As a corollary to our main result (Corollary 4.2), we also obtain a sufficient condition for the convergence of the G-S-N and N-G-S algorithms on a circuit partitioned in the BBDF form.

We can show that in some sense, our sufficient condition is the best possible. Consider the partitioned circuit of Figure 4. Nodes 2 and 4 are the feedback nodes. We wish to use the backward Euler formula with time-step h to discretize this circuit in time, and then use the G-S-N algorithm to solve the circuit at each time-point. Theorem 4.3 states that the G-S-N method applied to solve this partitioned circuit will converge for some choice of time-step if there are paths consisting of capacitors from the feedback nodes to ground. To establish a capacitive path to ground from each feedback node, we need a capacitor to ground from at least one of 2, 3 and one of

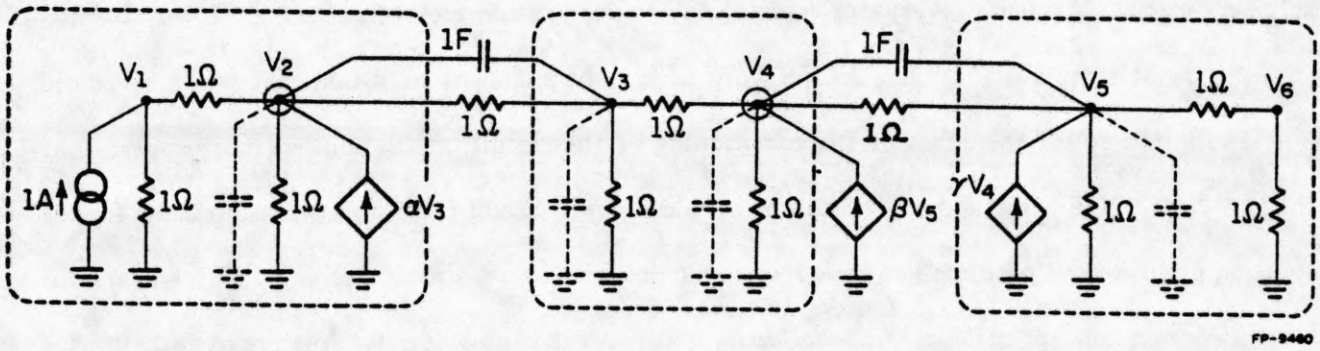

Figure 4

4. 5. Let $\alpha = \gamma = \beta = 100$. Fix the time-step h to be 0.001 seconds. We checked for the convergence of the G-S-N algorithm with different configurations of capacitances to ground in the circuit. The observations are summarized below:

1) No Capacitors to ground : If there are no capacitors to ground at any node in the circuit, the spectral radius of the iteration matrix is 1.1036, and the G-S-N algorithm does not converge. In fact, it can be checked that the spectral radius of the iteration matrix will be strictly larger than 1 for any choice of the time-step h .

2) Capacitor (1 Farad) to ground at node 2 : If a capacitor to ground of 1 Farad is placed at node 2, then there is a path consisting of capacitors from the feedback node 2 to ground, but there is no such path between node 4 (the other feedback node) and ground. The spectral radius of the iteration matrix is 1.0956, and the G-S-N algorithm does not converge. This is true for any choice of

the time-step h.

3) Capacitor (1 Farad) to ground at node 3 : There is a path to ground of capacitors from node 2 but not from node 4 to ground. The spectral radius of the iteration matrix is 1.0956, and the G-S-N algorithm does not converge.

4) Capacitor (1 Farad) to ground at node 4 : There is a path consisting of capacitors from node 4 to ground, but not from node 2 to ground. The spectral radius of the iteration matrix is 1.0961, and the G-S-N algorithm does not converge.

5) Capacitors (1 Farad each) to ground at nodes 2, 4 : There are paths consisting of capacitors from each feedback node to ground. Theorem 4.3 implies that the G-S-N algorithm converges for some time-step. We observe that for h =0.001 seconds, the spectral radius of the iteration matrix is 0.5525, and the G-S-N method converges.

6) Capacitors (1 Farad each) to ground at nodes 3, 5 : There are paths consisting of capacitors from each feedback node to ground. Theorem 4.3 implies that the G-S-N algorithm converges for some time-step. We observe that for h =0.001 seconds, the spectral radius of the iteration matrix is 0.5526, and the G-S-N method converges.

Thus, we see that for the G-S-N algorithm to converge for some time-step, it is necessary (Observations 5, 6) that the feedback nodes be connected to ground by paths consisting entirely of capacitors. This example thus demonstrates that the sufficient condition of Theorem 4.3 is also necessary for convergence in certain examples, and is thus the best possible (as mentioned at the end of Section 4).

This example also illustrates the distinctness of the sufficient condition of Theorem 4.3 from sufficient conditions which require the jacobian matrix A (Equation (2.10)) to be point-wise [5] or block-wise [6] diagonally dominant. Consider the circuit of Figure 1 with capacitors to ground as

in 5) above. The matrix A for this circuit (with capacitors replaced by companion models) is shown below.

$$\begin{vmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2003 & -1101 & 0 & 0 & 0 \\ 0 & -1001 & 1003 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2003 & -1101 & 0 \\ -100 & 0 & 0 & -1001 & 1003 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 \end{vmatrix}$$

Figure 5

A close inspection of the matrix in Figure 5 shows that it is not point-wise diagonally dominant in the fifth row, and it is not block-wise diagonally dominant. However, the G-S-N algorithm converges as predicted by Theorem 4.3.

Similar results were observed for a BBDF partitioning of the circuit in Figure 4, with nodes 2 and 4 chosen as the tearing nodes. For the convergence of the G-S-N method (for some time-step), it was necessary to have paths consisting of capacitors to ground from each tearing node (Corollary 4.2).

The G-S-N and the N-G-S methods were implemented in a simulator written in C, running on a VAX-11/780 computer. We tested some digital MOS circuits. Two of the circuits tested were a ring of inverters (Figure 6) and a one bit full adder (Figure 7). We summarize the results for the G-S-N algorithm in Table 1. The measured parameter NRI is the average number of iterations of the outer relaxation loop at every time-step (averaged over the simulation interval), and is a measure of the speed of convergence of the algorithm. The parameter K is the maximum number of N-R iterations used to estimate the solution of each subsystem in the inner loop. The term NLBT stands for the *Nearly Lower Block Triangular Form* in which the circuit is partitioned into more than two subcircuits, and the partitioning is such that the adjacency matrix of the circuits (when linearized) are "nearly" lower block triangular. The term BBDF refers to a partition of the circuit by removal

of a set of tearing nodes. The NLBT partition and the tearing nodes for the BBDF partitioning are marked in Figures 6 and 7.

Table 1: *NRI* for the G-S-N Algorithm

K	Ring of inverters		Full Adder	
	BBDF	NLBT	BBDF	NLBT
1	4.97	4.97	4.05	4.46
2	4.95	4.96	3.52	4.03
3	4.95	4.96	3.50	3.95

Digital MOS circuits usually have low gain and seem to be robust with regard to convergence. We found that the G-S-N and the N-G-S methods converged for most of the partitions of the circuits we tested. (However, the speeds of convergence varied widely for different partitionings of the same circuit [10].) This is not true in high gain analog MOS circuits and for bipolar circuits, where partitioning needs to be done carefully to ensure convergence. We believe that our results will be especially beneficial in choosing a partitioning for a circuit that has high gain, such as the example in Figure 4.

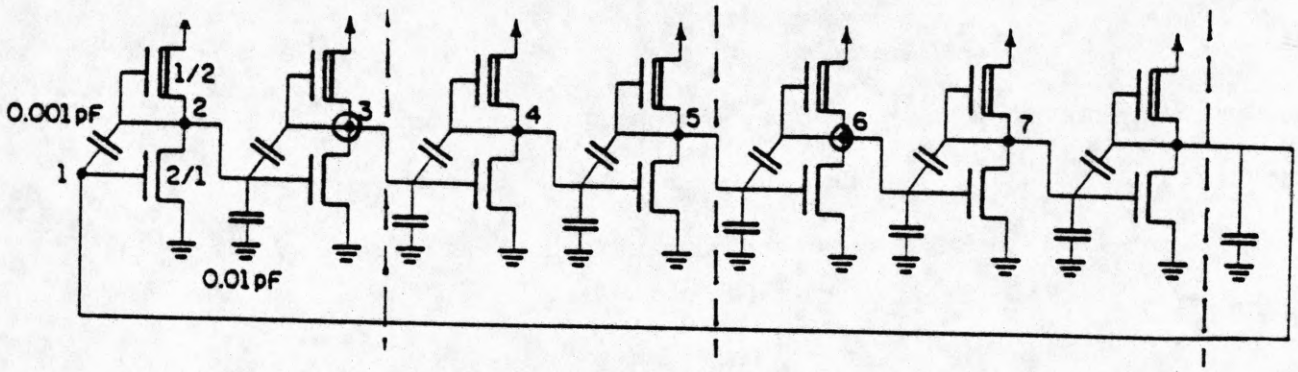

Figure 6: Ring of Inverters.

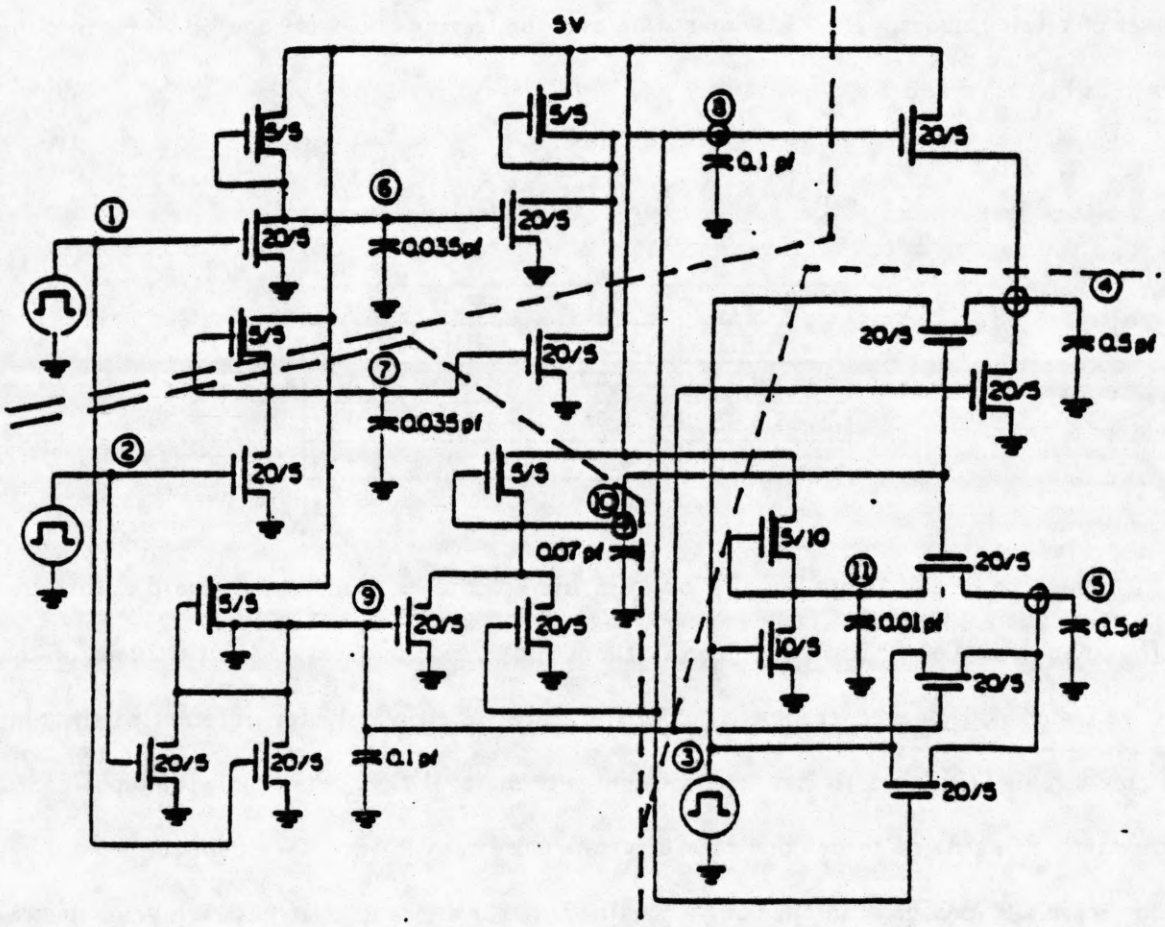

Figure 7: One-Bit Full Adder.

The convergence results of Section 4 may be applied to other systems of algebraic differential equations of the form (2.1). The main assumption that the system of equations must satisfy is that of (2.5) and (2.6). Our results can be applied to get similar convergence results for systems of the form

$$Q\,\dot{x}(t) = F\,(x(t), u(t)) \tag{5.1}$$

where x is the vector of unknown variables, and u is some known stimulus. The matrix Q in (5.1) needs to satisfy the assumptions satisfied by the matrix C in (2.5), (2.6). The need for a capacitor to ground at node i in the circuit is really a way of ensuring that there is strict diagonal

dominance in the i^{th} row of the capacitance matrix C. Similarly, the existence of a path consisting of capacitors from node i to ground is equivalent to saying that in the matrix C, there is a path from row i to some row j where strict diagonal dominance exists. Thus, the results of Theorem 4.3, Corollaries 4.1 and 4.2 can be easily applied to systems of the form (5.1).

In conclusion, this paper provides a *topological* sufficient condition for the convergence of relaxation methods used to solve block-partitioned systems of algebraic equations arising from differential equations (after the application of an implicit integration formula). We have examples of circuits where these conditions are also necessary for convergence, thus showing that the sufficient conditions are the best possible.

Appendix

We prove Lemma 4.1 by induction on n, the order of B. For the basis case, take $n=1$, that is B is a real number. It is easy to see that $I - D^{-1}B$ is 0, and thus (4.14) is satisfied. Assume the hypothesis is true for $n < N$. Let B be a matrix as in Lemma 4.1 with $n = N$. Denote by $Q = [q_{ij}]$ the matrix $I - D^{-1}B$. Note that Q has the same directed graph associated with it as does B. Also, each diagonal entry of Q is zero, and for W, Y as defined in the statement of the Lemma,

$$\text{for } \textit{all } i \in W, \ \sum_{j=1}^{N} |q_{ij}| = 1 \tag{A1}$$

$$\text{for } \textit{all } i \in Y, \ \sum_{j=1}^{N} |q_{ij}| < 1 \tag{A2}$$

The statement of the Lemma regarding paths in the graph corresponding to B is equivalent to saying that for every $i \in W$, there is a path in the graph of Q between i and some $j \in Y$. It is easy to see from the Gerschgorin Circle Theorem[11] that $\rho(Q) \leq 1$. Assume that λ is an eigenvalue of Q with $|\lambda| = 1$, and let a be the corresponding eigenvector, chosen so that $\|a\|_\infty = 1$. We differentiate two cases:

Case 1: $|a_i| = 1$ for all $i \in \{1, \cdots, N\}$. For this case, consider $i \in Y$. We have

$$1 = |\lambda a_i| \leq \sum_{j=1}^{N} |q_{ij}| \, |a_j| = \sum_{j=1}^{N} |q_{ij}| < 1, \tag{A3}$$

which is a contradiction.

Case 2: $|a_i| < 1$ for some i. In this case, let

$$\Gamma = \left\{ i : |a_i| = 1 \right\} \tag{A4}$$

$$\Delta = \left\{ i : |a_i| < 1 \right\} \tag{A5}$$

We can assume without loss of generality that $\Gamma = \{1, \cdots, r\}$ and $\Delta = \{r+1, \cdots, N\}$. (If this

were not so, we could choose a permutation matrix U such that $U\ a = \hat{a}$, with $|\hat{a}_i| < 1$ for $i \leqslant r$, and $|\hat{a}_i| = 1$ for $r+1 \leqslant i \leqslant N$. Then consider the matrix $\hat{Q} = U\ Q\ U^T$, and note that $\hat{a}$ is the eigenvector of $\hat{Q}$ corresponding to λ. We could thus work with $\hat{Q}$ instead of Q.)

The matrix Q can be written as

$$Q = \begin{vmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{vmatrix}, \tag{A6}$$

where Q_{11} is $r \times r$, and Q_{22} is $N-r \times N-r$. Consider $k > r$. Assume that $q_{kj} \neq 0$ for some $j \leqslant r$. Then,

$$1 \leqslant |\lambda a_k| \leqslant \sum_{l \neq k} |q_{kl}|\,|a_l| < \sum_{l \neq k} |q_{kl}| \leqslant 1 \tag{A7}$$

which is a contradiction. Hence $q_{kj} = 0$ for all $j \leqslant r$. Thus $Q_{21} = 0$.

Let W_1 be the set of rows in Q_{11} such that if $i \in W_1$, then

$$\sum_{l \leqslant r} |a_{il}| = 1. \tag{A8}$$

Clearly, $W_1 \subseteq W$. Let $Y_1 = \{1, \cdots, r\} - W_1$. For each $i \in W_1$, let S_i be the set of nodes in Y such that there is a path in the graph associated with Q from i to every node in S_i. Assume that $j \in S_i$ implies $j > r$. Then, for $j \in S_i$, let $i\ i_1 i_2 \ldots i_{t-1} i_t = j$ be the shortest path in G from i to j. Let i_d, $2 \leqslant d \leqslant t$ be the first index $> r$ in this path. Then, it is clear that $q_{i_{d-1} i_d} \neq 0$. That is, $i_{d-1} \in Y_1$, and there is a path from i to i_{d-1} in the graph corresponding to Q_{11}. On the other hand, assume there is a $j \in S_i$ with $j \leqslant r$. Consider any path from i to j. This path cannot visit any l with $l > r$, since $Q_{21} = 0$. Thus, there is a path from i to j entirely in the graph corresponding to Q_{11}. Also $j \in Y_1$. Thus Q_{11} satisfies the assumptions of the Lemma. Similarly, we can show that Q_{22} satisfies the assumptions of the Lemma. Also each of Q_{11}, Q_{22} has order at most $N-1$. Thus, by the induction hypothesis,

$$\rho(Q_{11}) < 1, \qquad \rho(Q_{22}) < 1. \tag{A9}$$

Since $Q_{21} = 0$, (A9) implies that $\rho(Q) < 1$. Thus, by induction on the order of the matrix Q, the

the lemma is proved.

To prove the second part of the Lemma, note that if $b_{ii} > 0$, and $b_{ij} \leqslant 0$ for all $i \neq j$, then, using (A9) together with (2.4.8) from [11], we have $B^{-1} \geqslant 0$. □

Acknowledgement

This work was supported by the Semiconductor Research Corporation under Contract 84-06-049.

References

[1] L.N. Nagel, " SPICE 2: A Computer Program to Simulate Semiconductor Circuits," ERL Memo No. ERL-M520, Univ. of California, Berkeley, CA, May 1975.

[2] P. Yang, I.N. Hajj, T.N. Trick, "SLATE: A Circuit Simulation Program with Latency Exploitation and Node Tearing," *Proc. Int. Conf. on Circuits and Computers*, Port Chester, NY, pp.353-355, October 1980.

[3] C.W. Gear, *Numerical Initial Value Problems for Ordinary Differential Equations*, Prentice Hall, Englewood Cliffs NJ, 1971.

[4] R. Saleh, J.E. Kleckner, and A.R. Newton, " Iterated Timing Analysis in SPLICE1," *Proc. Int. Conf. on Computer Aided Design*, Santa Clara, pp.139-140, November 1983.

[5] J. White, R. Saleh, A. Sangovianni-Vincentelli, and A.R. Newton, "Accelarating Relaxation Algorithms for Circuit Simulation using Waveform Newton, Iterated Step-size Refinement, and Parallel Techniques," *Proc. Int. Conf. on Computer Aided Design*, Santa Clara, pp.5-7, November 1985.

[6] M. Ilic-Spong, I.N. Katz, and J. Zaborsky, "Block Diagonal Dominance for Systems of Nonlinear Equations with Applications to Load Flow Calculations in Power Systems," Paper Preprint, 1985.

[7] J. More, "Nonlinear Generalizations of Matrix Diagonal Dominance with Applications to Gauss-Seidel Iterations," *SIAM J. Numer. Anal.*, Vol. 9, No. 2, pp. 357-378, June 1972.

[8] D.G. Feingold and R.S. Varga, " Block Diagonally Dominant Matrices and Generalizations of the Gerschgorin Circle Theorem," *Pacific J. Math*, Vol. 12, pp. 1241-1250, 1962.

[9] L.O. Chua and P.M. Lin, *Computer-aided Analysis of Electronic Circuits*, Prentice-Hall, Englewood Cliffs NJ, 1975.

[10] M.P. Desai, "Parallel Algorithms for Circuit Simulation," M.S. Thesis, University of Illinois, Urbana-Champaign, October 1986.

[11] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Equations in Several Variables*, Academic Press, New York, 1970.

[12] S. Even, *Graph Algorithms*, Computer Science Press, Rockville MD, 1979.

[13] R. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs NJ, 1962.

[14] I.N. Hajj, " Sparsity Considerations in Network Solution by Tearing," *IEEE Trans. Circuits and Systems, CAS-27, No.10, pp.357-366, May 1980.*

[15] A. Sangovianni-Vincentelli, L.K. Chen, L.O. Chua, " An Efficient Heuristic Algorithm for Tearing Large-scale Networks." *IEEE Trans. Circuits and Systems*, CAS-24, No.12, pp.709-717, December 1977.