Recasting modified nodal analysis to improve reliability in numerical circuit Simulation

Article *in* IEEE Transactions on Circuits and Systems I Regular Papers · April 2005

DOI: 10.1109/TCSI.2004.842869 · Source: IEEE Xplore

Angelo Brambilla
Politecnico di Milano
179 PUBLICATIONS **1,725** CITATIONS

Giancarlo Storti Gajani
Politecnico di Milano
86 PUBLICATIONS **919** CITATIONS

Recasting Modified Nodal Analysis to Improve Reliability in Numerical Circuit Simulation

Angelo Brambilla, Amedeo Premoli, and Giancarlo Storti-Gajani

***Abstract*—Modified nodal analysis (MNA) can be considered as the most adopted method in circuit simulation programs. Once node equations were written and complemented with those from the current controlled branches, a nonlinear system of algebraic and/or differential equations is obtained. In general, by using linear multistep integration methods, differential equations are recast as nonlinear algebraic ones, which are solved through the Newton method at each integration time step. While computer precision is not an issue in most situations, some specific but frequent cases yield ill-conditioned or singular matrices even in originally well posed circuits. This drawback can occur in different situations namely in presence of strongly nonlinear elements and/or when very small integration time steps are used. In the second case, very large conductances in the companion model of capacitors can introduce roundoff errors. In this paper, a transformation of the MNA that overcomes these problems is proposed. It is based on a suitable recombination of equations and electrical variables of the conventional MNA and it can be easily implemented in a circuit simulator.**

***Index Terms*—Modified nodal analysis (MNA), numerical circuit simulation, roundoff errors.**

I. Introduction

Most circuit simulators are based on modified nodal analysis (MNA). In general, for time-domain analysis, MNA leads to a nonlinear ordinary differential equation (ODE) or differential algebraic equations (DAE) system that, in most cases, is transformed in a nonlinear algebraic (resistive) system by means of linear multistep integration methods [1], [2] and, at each integration step, a Newton-like method is used to solve this nonlinear algebraic system. Therefore, from a numerical point of view, the equations modeling a dynamic circuit are transformed to equivalent linear equations at each iteration of the Newton method and at each time instant of the time-domain analysis. Thus, we can say that the time-domain analysis of a nonlinear dynamic circuit consists of the successive solutions of many linear resistive circuits approximating the original (nonlinear and dynamic) circuit at specific operating points.

Even while writing the equations needed at each Newton iteration seems to require the application of a sequence of steps to the nonlinear ODE or DAE system, MNA is structured so that these equations are straightforward formulated from circuit description. Indeed, the entries of the MNA matrix are directly built on a circuit element basis, each giving some contributions that are summed in suitable entries of the MNA matrix in the "loading" phase [1], [3], [4]. Thus, the MNA implementation directly affects the Newton method and, since the numerical problems can be relevant for each Newton iteration, to assume the circuit linear and resistive is not a limit.

The MNA matrix, often sparse, is then the LU factorized [5]; in this phase, attention is payed to the growth of the roundoff errors and of the number of the generated fill-ins [6], [7], that is, extra entries introduced during factorization and row elimination. It is well known that roundoff errors can worsen the conditioning number of the MNA matrix even leading to singularity if they are not suitably handled. Pivoting techniques are employed to limit the growth of roundoff errors during LU factorization. However, it can happen that, during the loading phase of the MNA matrix, very large and small conductances are summed in the same matrix entries leading to roundoff errors that are not recoverable in any way by the LU factorization algorithms. The situations that trigger this kind of numerical problems can be very different; they can be due to nonlinear elements such as diodes or MOSFET transistors, that, according to their current operating points, are characterized by very large or very small differential conductances [8]. Another possible cause is due to capacitors; the above mentioned algebraic approximation of differential equations has an equivalent electrical interpretation for each involved circuit element. For example, the application of the backward Euler (BE) method to the constitutive relation of a capacitor leads to a companion model that admits both the Thévenin and Norton representations. In conventional MNA based simulators, the Norton representation is in general, preferred, since it does not add the current flowing through the capacitor as a new unknown. This representation is composed of a resistor with equivalent conductance $g_{\rm eq} = C/h$ and of an independent current source $\imath_{\rm eq} = g_{\rm eq}\, v(t_n)$, where C is the capacitance value, $v(t_n)$ is the capacitor branch voltage at the t_n time instant and h is the integration time step. It is straightforward to see that $\lim_{h\to 0} |g_{\rm eq}| = \infty$ and this implies that, when the resistive model of a capacitor is loaded into the MNA matrix, cancellations can occur that in some cases make the matrix singular. These cancellations are due to the finite relative precision of the arithmetic logic unit (ALU) of the computer, in particular to bits that are truncated when summing numbers whose ratio is larger than a machine dependent upper bound. As said above these cancellations can not at all be recovered during the subsequent LU factorization, since they are "structural" of the numerical MNA implementation on a computer. These drawbacks are not so unusual in circuit simulation: in stiff dynamic circuits, capacitors or conductances with very different values can coexist; if these circuits are simulated with a short and global

Manuscript received February 12, 2004; revised July 21, 2004. This work was supported by the Italian MIUR under Grant FIRB RBNE012NSW_004. This paper was recommended by Associate Editor D. P. Mehta.

The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, I-20133 Milan, Italy (e-mail: brambill@elet.polimi.it).

Digital Object Identifier 10.1109/TCSI.2004.842869

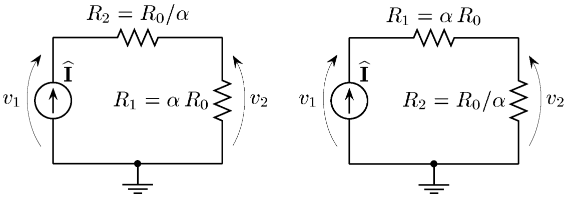


Fig. 1. Simple circuit with two resistors and one independent current source.

integration time step to obtain a suitable accuracy in analyzing fast dynamics, the large capacitances can introduce roundoff errors.

On the other hand, inductors and, more generally, current controlled elements do not pose any problem: these elements are described by independent equations that contribute entries to the MNA matrix that are not summed up to the previously loaded ones. Therefore they do not introduce any roundoff error during their loading and, for this reason, they are not fully discussed.

In this paper, we consider the roundoff problems described above and recast the conventional MNA formulation to avoid them or to limit their effects during matrix loading. The proposed approach was implemented in our own simulator and results were compared to those of well known ones such as Spice2g6, Spice3F2, Aplac, Eldo, and Spectre. We will see that all these fail or give results affected by artifacts. This approach belongs to the "battery" of techniques used to improve the robustness of circuit simulators. Among these techniques, we mention for example, $g_{\min}$ *stepping* that is used only when conventional simulation fails. Basically, it increases the value of small conductances ($g_{\min} = 1$ pS), artificially introduced in nonlinear device models, till an allowed maximum value, expecting that during this process convergence is reached. After convergence has been obtained, starting from the current "approximate" solution the $g_{\min}$ value is lowered by using continuation methods to meet its original value. Upon success of $g_{\min}$ reduction a valid solution of the circuit is obtained. Another method is *source stepping* that is based on the same mechanism of the gmin stepping but in this case it acts on the values of independent sources. The main use of these two methods, of some others not mentioned here and of that proposed in this paper is as "fall-back" option turned on exclusively to circumvent outright failures of conventional simulations.

In the following, we first present a simple example showing how roundoff errors can lead to singular MNA matrices; a recast MNA is then proposed, showing how it overcomes computer precision problems; refinements of the method and examples follow.

II. Very Simple But Critical Circuit

Consider the two essentially identical linear circuits shown in Fig. 1: the loop current and the branch voltages across the two resistors are equal, but the relative position of the two resistors has been swapped by shifting the datum node and changing the orientation of the $\widehat{\mathbf{I}}$ source current. In the following R_0 is a "reference" resistance and α is a dimensionless parameter (R_0, $\alpha \in \mathbb{R}^+$).

Adoption of MNA for the left and right circuits in Fig. 1, leads to

$$\overbrace{\frac{1}{R_0}\begin{bmatrix}\alpha & -\alpha\\ -\alpha & \alpha+\dfrac{1}{\alpha}\end{bmatrix}}^{\mathbf{M}_1}\begin{bmatrix}v_1\\ v_2\end{bmatrix}=\begin{bmatrix}\widehat{\mathbf{I}}\\ 0\end{bmatrix} \tag{1}$$

$$\overbrace{\frac{1}{R_0}\begin{bmatrix}\dfrac{1}{\alpha} & -\dfrac{1}{\alpha}\\ -\dfrac{1}{\alpha} & \alpha+\dfrac{1}{\alpha}\end{bmatrix}}^{\mathbf{M}_2}\begin{bmatrix}v_1\\ v_2\end{bmatrix}=\begin{bmatrix}\widehat{\mathbf{I}}\\ 0\end{bmatrix}. \tag{2}$$

Since $\det(\mathbf{M}_1) = 1/R_0^2$, $\mathbf{M}_1$ is nonsingular and the circuit admits a unique solution. However, if we consider a computer with a finite precision ALU, when the ratio between the values of R_1 and R_2 is increased above a certain value, it happens that the sum $\alpha/R_0 + 1/(R_0\ \alpha)$ is affected by roundoff error and, in particular, the m_{22} entry of $\mathbf{M}_1$ becomes equal to α/R_0. In this case, $\det(\mathbf{M}_1) = 0$ and the simulator gets in trouble to solve the circuit even if it is supposed that the LU factorization of $\mathbf{M}_1$ is not affected by any roundoff error. This happens since $\mathbf{M}_1$ was affected by roundoff errors during its loading, that is, when α/R_0 and $1/(\alpha\ R_0)$ were summed, and this is a "structural" drawback of MNA. The $\mathbf{M}_2$ matrix in (2) never becomes singular for any large α; indeed, when the α/R_0 entry cancels the $1/(\alpha\ R_0)$ one, we have $\det(\mathbf{M}_2) = 1/R_0^2\left(1 - 1/\alpha^2\right) \neq 0$. The $\mathbf{M}_2$ matrix is still affected by roundoff errors but in this case the simulator can successfully end its job.

Now, (2) can be recast by suitably summing rows and columns of $\mathbf{M}_2$. For example, introduce the new unknown $v_1^{\clubsuit} = v_1 - v_2$, it yields

$$\frac{1}{R_0}\begin{bmatrix}\dfrac{1}{\alpha} & -\dfrac{1}{\alpha}\\ -\dfrac{1}{\alpha} & \alpha+\dfrac{1}{\alpha}\end{bmatrix}\begin{bmatrix}v_1^{\clubsuit}+v_2\\ v_2\end{bmatrix}=\begin{bmatrix}\widehat{\mathbf{I}}\\ 0\end{bmatrix}$$
$$\Longrightarrow \frac{1}{R_0}\begin{bmatrix}\dfrac{1}{\alpha} & 0\\ -\dfrac{1}{\alpha} & \alpha\end{bmatrix}\begin{bmatrix}v_1^{\clubsuit}\\ v_2\end{bmatrix}=\begin{bmatrix}\widehat{\mathbf{I}}\\ 0\end{bmatrix}$$

and by summing the two rows of the system we have

$$\overbrace{\frac{1}{R_0}\begin{bmatrix}\dfrac{1}{\alpha} & 0\\ 0 & \alpha\end{bmatrix}}^{\mathbf{M}_2^{\clubsuit}}\begin{bmatrix}v_1^{\clubsuit}\\ v_2\end{bmatrix}=\begin{bmatrix}\widehat{\mathbf{I}}\\ \widehat{\mathbf{I}}\end{bmatrix}$$

and $\mathbf{M}_2^{\clubsuit}$ is no longer affected by roundoff errors during its loading, because the addition of α/R_0 and $1/(\alpha\ R_0)$ does not occur.

Note that other circuit formulations can structurally overcome this problem. Even though sparse tableau analysis (STA) is characterized by a large-dimensional sparse system, each circuit element contributes only one entry to the matrix, and this entry is not summed to other ones. On the other hand, if NA is considered, when two or more element values are very different, also a trivial addition can lead to severe simulation errors due to roundoff. From this standpoint, NA is more critical than STA.

III. Transformed Nodal Analysis

In this section, the matrix loading mechanism of MNA is analyzed in a general framework and it is shown that errors due to machine additions arise only in the NA submatrix of the MNA matrix. We show that it is possible to define topological transformations that avoid the sum of circuit parameters of very different value in the same entry.

Before proceeding, we introduce the nonlinear *machine addition operator*, represented by the $\widetilde{+}$ symbol or $\sum^{\widetilde{+}}$, that is responsible of the roundoff errors discussed in Section II. In particular, in (2) we have $\alpha\widetilde{+}1/\alpha = \alpha$ for values of $\alpha \geq 1/\epsilon^2$ where ϵ is relative precision[1] of the ALU.

A. MNA Matrix

We briefly recall the definition of NA; to this end, consider a resistive linear circuit with $N+1$ nodes and B voltage-controlled branches (two-terminal resistors, independent current sources, and voltage-controlled n-ports), the latter grouped in set $\mathcal{B}$. Choose the datum node and write a linear system of N algebraic equations formulated by zeroing the sum of the currents, expressed in terms of node voltages, entering each corresponding node.

We introduce the source current vector $\widehat{\mathbf{i}} \in \mathbb{R}^B$ and the branch conductance matrix $\mathbf{G} \in \mathbb{R}^{B\times B}$. By assuming that the branches (one for each port) are ordered element by element, the $\mathbf{G}$ matrix is block diagonal : each 1×1 block corresponds to the conductance of a one-port and in any case is nonzero, while $n \times n$ blocks correspond to the conductance matrices of voltage-controlled n-ports. More in detail, the diagonal entries of the $n \times n$ blocks can be zero and, in this case, the nonzero off-diagonal entries, on the same row or column, correspond to voltage-controlled current sources (VCCSs). Obviously, when only one-port resistors are embedded in the circuit, matrix $\mathbf{G}$ is diagonal. By introducing the $N \times B$ incidence matrix $\mathbf{A}$, the system of N equations is written in compact form as

$$\underbrace{\mathbf{A}\;\mathbf{G}\;\mathbf{A}^T}_{\mathbf{G}^\bullet}\;\mathbf{v}^\bullet = \underbrace{\mathbf{A}\;\widehat{\mathbf{i}}}_{\widehat{\mathbf{i}}^\bullet} \Longrightarrow \mathbf{G}^\bullet\;\mathbf{v}^\bullet = \widehat{\mathbf{i}}^\bullet \qquad (3)$$

where $\mathbf{G}^\bullet \in \mathbb{R}^{N\times N}$ is the node conductance matrix and the $\widehat{\mathbf{i}}^\bullet = \left[\widehat{i}_1^\bullet\;\widehat{i}_2^\bullet\;\ldots\;\widehat{i}_N^\bullet\right]^T$ vector groups the node source currents. Equation (3) admits the unique solution $\mathbf{v}^\bullet = [\mathbf{G}^\bullet]^{-1}\widehat{\mathbf{i}}^\bullet$, if and only if the inverse matrix $[\mathbf{G}^\bullet]^{-1}$ exists.[2]

Consider the column vectors $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(B)}$ extracted from $\mathbf{A}$: $\mathbf{a}^{(\mu)}$ corresponds to the μth branch of the circuit and its nonzero entries denote the pair of nodes at which the μth branch is incident. By using the above notation, matrix $\mathbf{G}^\bullet$ can be expanded as

$$\mathbf{G}^\bullet = \sum_{(\mu,\nu)\in\mathcal{G}} \Delta\mathbf{G}^\bullet_{(\mu,\nu)} = \sum_{(\mu,\nu)\in\mathcal{G}} g_{\mu,\nu}\;\mathbf{a}^{(\mu)}\;[\mathbf{a}^{(\nu)}]^T \qquad (4)$$

where set $\mathcal{G} \equiv \left\{(\mu,\nu) \in \mathbb{N}^{B\times B} \mid g_{\mu,\nu} \neq 0\right\}$ include the ordered pairs of row and column indices of the nonzero entries of $\mathbf{G}$ and matrices $\Delta\mathbf{G}^\bullet_{(\mu,\nu)}$, referred by some authors [1], [3] as *stamps*, have a one-to-one correspondence to voltage-controlled elements (i.e., resistors and VCCSs) embedded in the circuit. Furthermore, they are structurally rank-one and sparse : in general only a quartet of entries are nonzero. The location of this quartet of entries strictly depends on the nodes at which the element branches are connected. For both one-port resistors and VCCSs, the entries of the quartet are equal to $\pm g$, where g is the conductance and the transconductance, respectively. Unfortunately, the entries of different stamps can be summed in the same entry of $\mathbf{G}^\bullet$.

Now, consider MNA and circuits embedding, besides voltage-controlled elements, independent voltage sources and the remaining three types of controlled sources. We split the set of branches $\mathcal{B}$ in two complementary subsets: $\mathcal{B}_v$ of voltage-controlled branches (v-branches) and $\mathcal{B}_c$ of current-controlled branches (c-branches); the cardinality of $\mathcal{B}_v$ and $\mathcal{B}_c$ is denoted as B_v and B_c, respectively, with $B = B_v + B_c$. Conventional NA is extended to MNA [2] as follows: currents of c-branches are added as further unknowns and the corresponding branch equations are appended to the NA system. The $N \times B$ incidence matrix $\mathbf{A}$ can be partitioned as $\mathbf{A} = [\mathbf{A}_v\;\mathbf{A}_c]$, with $\mathbf{A}_v \in \mathbb{R}^{N\times B_v}$ and $\mathbf{A}_c \in \mathbb{R}^{N\times B_c}$, the source current vector $\widehat{\mathbf{i}}^T = \left[\widehat{\mathbf{i}}_v^T\;\widehat{\mathbf{i}}_c^T\right]$, with $\mathbf{i}_v \in \mathbb{R}^{B_v}$, $\mathbf{i}_c \in \mathbb{R}^{B_c}$, and the branch voltage vector $\mathbf{v}^T = \left[\mathbf{v}_v^T\;\mathbf{v}_c^T\right]$, with $\mathbf{v}_v \in \mathbb{R}^{B_v}$ and $\mathbf{v}_c \in \mathbb{R}^{B_c}$. By adopting the above partitioning, the KCL and KVL equations are recast as $\mathbf{A}_v\;\mathbf{i}_v + \mathbf{A}_c\;\mathbf{i}_c = \widehat{\mathbf{i}}^\bullet$, $\mathbf{v}_v = \mathbf{A}_v^T\mathbf{v}^\bullet$ and $\mathbf{v}_c = \mathbf{A}_c^T\mathbf{v}^\bullet$, respectively.

As in conventional NA, constitutive relations of v-branches are written, using the branch conductance submatrix $\mathbf{G}^\bullet \in \mathbb{R}^{B_v\times B_v}$ in the form $\mathbf{i}_v = \mathbf{G}^\bullet\;\mathbf{v}_v$, while the characteristics of the c-branches, including independent voltage sources and controlled sources except VCCSs, are represented by the implicit equation

$$\mathbf{B}_c\,\mathbf{v}_c - \mathbf{R}_c\mathbf{i}_c - \widehat{\mathbf{v}}_c = \mathbf{0} \qquad (5)$$

where $\mathbf{B}_c, \mathbf{R}_c \in \mathbb{R}^{B_c\times B_c}$ and $\widehat{\mathbf{v}}_c \in \mathbb{R}^{B_c}$ [1]. Each row of $\mathbf{B}_c$ and $\mathbf{R}_c$ represents a single c-branch of the circuit. The nonzero entries of $\mathbf{B}_c$ and $\mathbf{R}_c$ can be once more described by stamp matrices [1].

By using the above notations, we obtain the system

$$\overbrace{\begin{bmatrix}\mathbf{G}^\bullet & \mathbf{A}_c\\ \mathbf{B}_c\mathbf{A}_c^T & -\mathbf{R}_c\end{bmatrix}}^{\text{MNA matrix}\longrightarrow\mathbf{M}}\begin{bmatrix}\mathbf{v}^\bullet\\ \mathbf{i}_c\end{bmatrix} = \begin{bmatrix}\widehat{\mathbf{i}}^\bullet\\ \widehat{\mathbf{v}}_c\end{bmatrix} \qquad (6)$$

in which the unknowns reduce to subvectors $\mathbf{v}^\bullet$ and $\mathbf{i}_c$. If the circuit is well posed the MNA matrix $\mathbf{M}$ is nonsingular and a solution in terms of $\mathbf{v}^\bullet$ and $\mathbf{i}_c$ is easily found.

The next section shows that suitable transformations can be applied to the NA submatrix so that the errors due to machine

[1] On 64-bit IEEE floating point compliant computers, as most personal computers, we have $\epsilon \simeq 2.22 \times 10^{-16}$.

[2] In practice $[\mathbf{G}^\bullet]^{-1}$ is rarely computed, indeed it is preferred to decompose $\mathbf{G}^\bullet$ in the product of a lower triangular matrix $\mathbf{L}$ and an upper triangular one $\mathbf{U}$.

addition are eliminated or substantially reduced. Note that, since nonzero entries in submatrices $\mathbf{B}_c\mathbf{A}_c^T$ and $\mathbf{R}_c$ are due to a single element, machine addition errors do not occur in loading phase. Therefore, they are not further commented.

B. Transformed MNA

We replace the unknown node voltages by $\mathbf{v}^{\bullet} = \mathbf{Q}\ \mathbf{v}^{\clubsuit}$, where $\mathbf{v}^{\clubsuit} \in \mathbb{R}^B$ is the new unknown vector and $\mathbf{Q} \in \mathbb{R}^{N\times N}$ is nonsingular. This corresponds to a basis change for the unknown vector. Similarly, we replace the known source current vector by $\widehat{\mathbf{i}}^{\diamondsuit} = \mathbf{P}\ \widehat{\mathbf{i}}^{\bullet}$. This corresponds to N linear combinations of the equations, weighted by the rows of $\mathbf{P}$. The above substitutions lead to a recasting of the NA formulation as follows:

$$\widehat{\mathbf{i}}^{\bullet} = \mathbf{G}^{\bullet}\ \mathbf{v}^{\bullet} \Longrightarrow \widehat{\mathbf{i}}^{\diamondsuit} = \mathbf{P}\ \mathbf{G}^{\bullet}\ \mathbf{Q}\ \mathbf{v}^{\clubsuit}. \tag{7}$$

The MNA system in (6) is transformed in the corresponding way

$$\begin{bmatrix} \mathbf{P}\mathbf{G}^{\bullet}\mathbf{Q} & \mathbf{P}\mathbf{A}_c \\ \mathbf{B}_c\mathbf{A}_c^T\mathbf{Q} & -\mathbf{R}_c \end{bmatrix} \begin{bmatrix} \mathbf{v}^{\clubsuit} \\ \mathbf{i}_c \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{i}}^{\diamondsuit} \\ \widehat{\mathbf{v}}_c \end{bmatrix}. \tag{8}$$

Among the possible choices for $\mathbf{P}$ and $\mathbf{Q}$ those having a simple interpretation on the corresponding circuit graph are interesting. The conventional NA is based on node equations, but equations related to cut-sets enclosing more than one node could be adopted as well. Similarly, node voltages are used as unknowns, but their combinations, corresponding to a new set of unknowns relative to one or more datum nodes, could be used. These alternative representations of NA are obtained from the conventional NA submatrix by adopting $\mathbf{P}$ and $\mathbf{Q}$ with entries exclusively equal to 0 or to 1: $\mathbf{P}$ is thus responsible of choosing a new set of cut-sets, while $\mathbf{Q}$ chooses a new set of subtrees and associated datum nodes upon which the unknown voltage vector $\mathbf{v}^{\clubsuit}$ is defined.

In our case, to avoid critical machine addition errors, the topological transformations must avoid situations such as those outlined in the simple example of Section II. This is accomplished by first defining a measure giving the (possible) roundoff error introduced by each circuit element, i.e., how "large" is the contribution of the element in relation to other elements, and then by choosing a transformation matrix that, applied to the incidence matrix $\mathbf{A}_v$, leaves only one nonzero entry in columns corresponding to branches recognized as being "large". The same type of transformation has to be applied to the rows of $\mathbf{A}_v^T$. In general, due to the presence of nonreciprocal elements, such as VCCSs, the two transformations are not the same, but the algorithm, introduced in the following, is simplified if we choose to label as large all ports of a multiport element, if *any one* of them introduces large entries in the NA submatrix. This choice leads to $\mathbf{Q} = \mathbf{P}^T$.

The measure, later introduced to find how "critical" is a circuit element in relation to roundoff errors, partitions the set of all branches in the two disjoint subsets of "small" and "large" branches. While the former can be safely loaded in $\mathbf{M}$ before applying any transformation, the latter, due to nonlinearity intrinsic to machine addition, must be loaded one element at a time by applying the transformation before loading the corresponding *single* entry in $\mathbf{M}$. It is thus of main importance to remark that, since $\mathbf{G}^{\bullet}$ is decomposable in the addition of a number of terms, we formally have

$$\mathbf{P}\ \mathbf{A}_v\mathbf{G}\mathbf{A}_v^T\mathbf{Q}\ \mathbf{v}^{\clubsuit} = \left[\sum_{(\mu,\nu)\in\mathcal{G}} \underbrace{g_{\mu,\nu}\ \mathbf{P}\ \mathbf{a}_v^{(\mu)}\ [\mathbf{a}_v^{(\nu)}]^T\mathbf{Q}}_{\text{single stamp}}\right]\ \mathbf{v}^{\clubsuit}. \tag{9}$$

In the actual implementation, it is possible to apply the transformation separately to any subset of circuit branches and then add the result. In particular, if we denote the set of "large" conductances as $\mathcal{G}_\sigma$ and the complementary set of "small" conductances as $\mathcal{G}_\varrho$, the transformations must be applied independently to elements in the two subsets $\mathcal{G}_\sigma$ and $\mathcal{G}_\varrho$. A computationally effective approach is to build $\mathbf{G}^{\bullet}$ by using the decomposition

$$\begin{aligned} \mathbf{P}\ \mathbf{G}^{\bullet}\ \mathbf{Q}\ \mathbf{v}^{\clubsuit} = \Bigg[& \underbrace{\mathbf{P}\left(\overset{\widetilde{+}}{\sum_{\mu,\nu\in\mathcal{G}_\varrho}} g_{\mu,\nu}\ \mathbf{a}_v^{(\mu)}\ [\mathbf{a}_v^{(\nu)}]^T\right)\mathbf{Q}}_{\text{stamps}\in\mathcal{G}_\varrho} \\ & \widetilde{+} \overset{\widetilde{+}}{\sum_{\mu\nu\in\mathcal{G}_\sigma}} \underbrace{g_{\mu,\nu}\ \mathbf{P}\ \mathbf{a}_v^{(\mu)}\ [\mathbf{a}_v^{(\nu)}]^T\mathbf{Q}}_{\text{modified stamps}\in\mathcal{G}_\sigma}\Bigg]\mathbf{v}^{\clubsuit}. \end{aligned} \tag{10}$$

The reading of the above equations from left to right sketches the proposed algorithm.

1. Load all elements in $\mathcal{G}_\varrho$ by adopting the conventional procedure;
2. apply the transformation to the $\mathbf{M}$ matrix, partially formed in the previous step;
3. add contributions from elements in $\mathcal{G}_\sigma$ applying the transformation *before* loading each single element.

The transformation so introduced obviously affects also branches in $\mathcal{G}_\varrho$, but, since operations involving the corresponding matrix entries can be assumed to be perfectly linear, this is not a problem. The same can be said of the transformations that are introduced on blocks of $\mathbf{M}$ other than $\mathbf{G}^{\bullet}$; recall, in fact, that equations in these other blocks are either independent or contain only unit entries. The proposed transformation is linear from an algebraic standpoint, but nonlinear if machine addition is considered. Its role is to reduce as much as possible nonlinearities (i.e., roundoff errors) introduced by machine additions in MNA.

C. Algorithm for the Definition of the Transformation Matrices

The transformation matrix $\mathbf{P}$ is completely defined by the branches in $\mathcal{G}_\sigma$ and their interconnections; obviously, the transformation so obtained acts also on branches in $\mathcal{G}_\varrho$. The definition of these two complementary sets is thus crucial to obtain a correct transformation and is the first preliminary step of the proposed algorithm.

The level of "criticalness" of a branch depends on the result of the addition of the entries of the corresponding stamp matrix to entries of the NA submatrix due to other stamps: if any two terms being added have a ratio smaller than machine precision ϵ (or greater than $1/\epsilon$), the smaller term is annihilated by the larger one; this corresponds to the maximum possible error and the resulting matrix can be singular. If the ratio of the two terms

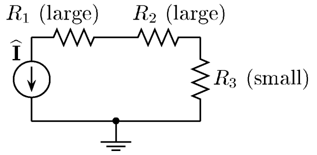


Fig. 2. Significant case in the determination of large and small conductances: R_1 satisfies only the global condition; R_2 satisfies both global and local conditions.

is only slightly larger than ϵ (or slightly smaller than $1/\epsilon$), some of the low-order bits of the smaller term will be lost; in this case, errors are less severe and in most cases the resulting matrix is non singular, but the accuracy of the solution so obtained can be rather low. To this end, we first define e_r as a measure of the maximum acceptable machine addition error for stamps, i.e., the maximum error allowed in the addition of a stamp matrix in the construction of the NA submatrix.

Noting that addition of conductances occurs when two branches are directly connected (i.e., having a node in common), one may be tempted to define $\mathcal{G}_\sigma$ and $\mathcal{G}_\varrho$ on a strictly local basis, i.e., comparing values of branches that are directly connected. Unfortunately, since the transformation obtained acts on all branches in the circuit, it is possible that values corresponding to branches that are not connected directly end up being added in the final MNA matrix. On the other hand, comparing each branch to *all* other branches in the circuit often leads to an over-dimensioned $\mathcal{G}_\sigma$. More formally, we can say the following.

- A sufficient condition for a branch to be in $\mathcal{G}_\sigma$ is to have a conductance g_i such that $g_i/g_{\min} > e_r/\epsilon$, where $g_{\min}$ is the smallest conductance over all branches *directly* connected to one of the nodes connected to the branch.
- A necessary condition for the same branch to be in $\mathcal{G}_\sigma$ is to have a conductance g_i that satisfies a similar condition, where $g_{\min}$ is the minimum conductance in the whole circuit.

Due to their nature the necessary condition will be called *global* while the sufficient condition *local*.

As stated above, if $\mathcal{G}_\sigma$ is formed by using only the local condition, it is possible that other conductances, that satisfy the global condition but not the local one, cause relevant errors because of the partial transformation that would be induced by this incomplete $\mathcal{G}_\sigma$ set. Consider the simple example in Fig. 2, if the local condition is used $\mathcal{G}_\sigma$ contains only R_2, while, with the global condition, also R_1 must be put in $\mathcal{G}_\sigma$. In this same case the transformation induced by the local condition leads to a singular matrix because of the entries due to R_1.

Set $\mathcal{G}_\sigma$ can thus be defined by first considering the set of all branches satisfying the global condition and then by removing all connected subgraphs in which no branch satisfies the local condition.

Once $\mathcal{G}_\sigma$ is found, the algorithm that follows defines a transformation matrix that reduces contribution of all branches in the set to one single entry in matrix $\mathbf{G}^\bullet$. Set $\mathcal{G}_\sigma$ will, in general, contain separate subsets of connected branches, if we assume that no loops are present (this restriction will be removed in the following), each connected subset is a (sub)tree; $\mathbf{P}$ is built independently, but simultaneously, for each of these subtrees.

Transformed MNA (TMNA) Algorithm

0. Define $\mathcal{N}_\sigma$ as the set of nodes to which branches in $\mathcal{G}_\sigma$ are connected and associate to each node $j \in \mathcal{N}_\sigma$ a number $\nu(j)$ denoting the number of branches in $\mathcal{G}_\sigma$ that are incident to this node.
1. Define the set of leaf nodes $\mathcal{N}_{\sigma 1}$ as the subset of nodes in $\mathcal{N}_\sigma$ such that $\nu(j) = 1 \, \forall j \in \mathcal{N}_{\sigma 1}$, i.e., nodes having only one branch in $\mathcal{G}_\sigma$ connected to them.
2. Build the partial $\mathbf{P}$ matrix so that the cutset equation of each leaf node, possibly modified in previous iterations, is added to the one corresponding to its father node.
3. Mark all leaf nodes as "done," decrement $\nu(j)$ for all father nodes and iterate from step 1 until only one or two nodes are left undone. If two nodes are left, choose one as leaf, the other as father and act accordingly.

In the sequel, we refer to this algorithm as TMNA, two iterations of the algorithm (on a generic graph) are shown in Fig. 3. Branches representing "large" conductances are evidenced by thick lines, $\mathcal{G}_\sigma$ is composed, in our case, by two subtrees. Numbers labeling each node are the $\nu(j)$ values at each iteration, nodes labeled with "1" are leaf nodes, while those labeled with "0" are "done".

At each step, *all* contributions that were added in a leaf node (i.e., also those inherited by nodes that had been leaves at previous steps) must be added to the corresponding father, in this way one of the central nodes in each subtree acts as datum node for all the remaining ones; this is evidenced in the last step in Fig. 3. Note that, in the rightmost subtree of the example, the choice of the final datum node among the two central nodes, i.e., those marked with "1" in the second step, is arbitrary. If any subtree is connected to ground the algorithm must be slightly modified and the node that acts as father for the ground node must be chosen as the new relative ground node.

If a loop is present in the $\mathcal{G}_\sigma$ set, it is not possible to form the corresponding transformation matrix, as at least one component appears in more than one entry of the final NA matrix. A simple solution is to open the loop either by introducing a "virtual" short circuit element or, if possible, by inverting the constitutive relation of one of the v-branches (Thévenin representation of companion model of capacitors). In either way the added or transformed branch ends up in the c-branch portion of the MNA matrix.

IV. Application

In this section, we detail an application of the approach described in Section III to the circuit shown in Fig. 4, which is characterized by the $g_1 = 1/(\alpha \; R_1)$, $g_2 = 1/(\alpha \; R_2)$, $g_3 = 1/(\alpha \; R_3)$, $g_6 = 1/(\alpha \; R_6)$, $g_7 = 1/(\alpha \; R_7)$, $g_8 = 1/(\alpha \; R_8)$ and $g_4 = \alpha/R_4$, $g_5 = \alpha/R_5$ parametric conductances, where $\alpha \in \mathbb{R}^+$, and assuming that all resistances have the same order of magnitude.

According to the conventional nodal analysis, node voltages $v_1^\bullet$, $v_2^\bullet$, $v_3^\bullet$, $v_4^\bullet$ and $v_5^\bullet$ represent the unknowns and the $\hat{\imath}_1^\bullet$, $\hat{\imath}_2^\bullet$, $\hat{\imath}_3^\bullet$, $\hat{\imath}_4^\bullet$ and $\hat{\imath}_5^\bullet$ node source currents constitute the known terms. For $\alpha \simeq 1$ all branches are in $\mathcal{G}_\rho$ and $\mathcal{G}_\sigma$ is empty; when α is increased g_4 and g_5 enter in $\mathcal{G}_\sigma$, while other conductances still remain in $\mathcal{G}_\rho$.

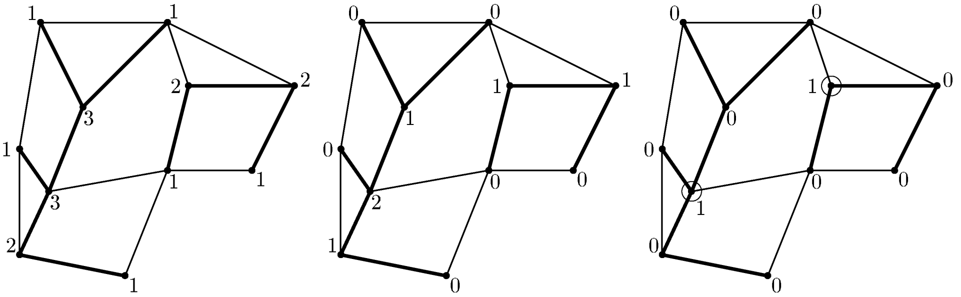


Fig. 3. Two consecutive iterations of the algorithm on a generic graph; the labels represent the $\nu(j)$ value for each node at each iteration.

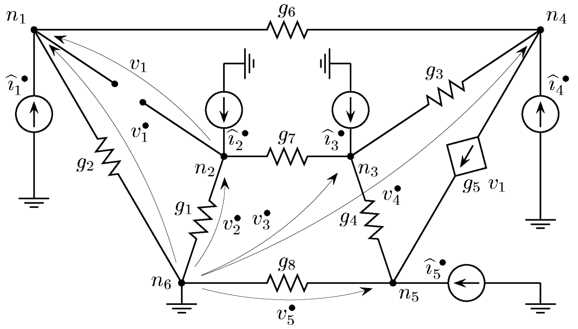


Fig. 4. Circuit with six nodes and nine branches.

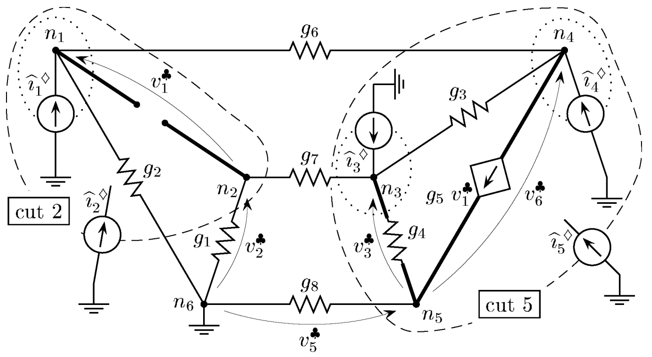


Fig. 5. Tree voltages and cut-set currents. Cutset labeled by 2 and 5, circumscribed by dashed close curves, substitute node 2 and 5. The other cutsets, circumscribed by dotted close curves, coincide with the original nodal cutsets.

Fig. 5 shows with thick lines, branches in $\mathcal{G}_\sigma$; note that it is composed of two disjoint subgraphs. Since no loops are present, no preprocessing is needed. By applying the proposed algorithm, we obtain the transformation matrix

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

that recombines node voltages and nodal cut-set currents so that the following new voltage vector and source current vector are obtained

$$\begin{bmatrix} v_1^\clubsuit \\ v_2^\clubsuit \\ v_3^\clubsuit \\ v_6^\clubsuit \\ v_5^\clubsuit \end{bmatrix} = \mathbf{Q}^{-1} \begin{bmatrix} v_1^\bullet \\ v_2^\bullet \\ v_3^\bullet \\ v_4^\bullet \\ v_5^\bullet \end{bmatrix} \text{ and } \begin{bmatrix} \widehat{\imath}_1^\diamondsuit \\ \widehat{\imath}_2^\diamondsuit \\ \widehat{\imath}_3^\diamondsuit \\ \widehat{\imath}_4^\diamondsuit \\ \widehat{\imath}_5^\diamondsuit \end{bmatrix} = \mathbf{P} \begin{bmatrix} \widehat{\imath}_1^\bullet \\ \widehat{\imath}_2^\bullet \\ \widehat{\imath}_3^\bullet \\ \widehat{\imath}_4^\bullet \\ \widehat{\imath}_5^\bullet \end{bmatrix}.$$

Combinations of nodal cut-sets lead to the new larger cut-sets shown in Fig. 5; combinations of node voltages lead to the set of tree branch voltages show in the same figure. By considering n_6 as the datum node and adopting conventional NA, we have (11) shown at the bottom of the page, where (trans)conductances in $\mathcal{G}_\sigma$ are tagged with a box. It can be easily seen that $\mathbf{G}^\bullet$ can become singular in presence of the roundoff errors introduced when $\alpha \rightarrow 1/\epsilon$. In fact, row summations yields a null row vector.

The adoption of the above defined transformation leads to (12), shown at the bottom of the next page. To have better insight into what happens when α is increased, we define the following measure of the accuracy in the computation of the inverse of a generic square matrix $\mathbf{Z}$:

$$\rho(\mathbf{Z}) = \|\mathbf{VZ} - \mathbf{1}_n\|_F \qquad (13)$$

where $\mathbf{1}_n$ is the $n \times n$ identity matrix, $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{V} = \mathbf{Z}^{-1} + \mathbf{E}$; $\mathbf{E}$ is a matrix that accounts inversion

$$\begin{bmatrix} \widehat{\imath}_1^\bullet \\ \widehat{\imath}_2^\bullet \\ \widehat{\imath}_3^\bullet \\ \widehat{\imath}_4^\bullet \\ \widehat{\imath}_5^\bullet \end{bmatrix} = \overbrace{\begin{bmatrix} g_2 + g_6 & 0 & 0 & -g_6 & 0 \\ 0 & g_1 + g_7 & -g_7 & 0 & 0 \\ 0 & -g_7 & g_3 + \boxed{g_4} + g_7 & -g_3 & -\boxed{g_4} \\ -g_6 + \boxed{g_5} & -\boxed{g_5} & -g_3 & g_3 + g_6 & 0 \\ -\boxed{g_5} & \boxed{g_5} & -\boxed{g_4} & 0 & \boxed{g_4} + g_8 \end{bmatrix}}^{\mathbf{G}^\bullet} \begin{bmatrix} v_1^\bullet \\ v_2^\bullet \\ v_3^\bullet \\ v_4^\bullet \\ v_5^\bullet \end{bmatrix} \qquad (11)$$

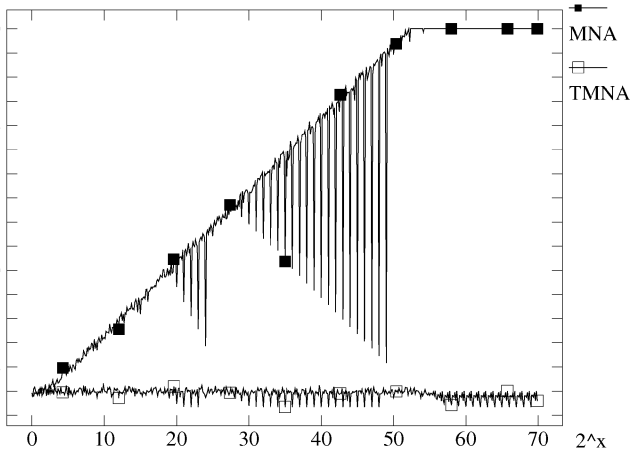


Fig. 6. Frobenius norms $\rho(\mathbf{G}^\bullet)$ and $\rho(\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q})$ labeled as MNA and TMNA, respectively; x-axis: logarithmic, radix 2; y-axis: logarithmic, radix 10.

errors. In Fig. 6 the value of $\rho(\mathbf{Z})$ is plotted for both $\mathbf{G}^\bullet$ and $\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}$.

The R_0 value is set equal to 1 Ω and α varies in the interval $[1, 2^{70}]$ that covers the relative precision of the computer ALU. The two curves refer to $\log_{10}(\rho(\mathbf{G}^\bullet))$ (labeled as MNA) and $\log_{10}(\rho(\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}))$ (labeled as TMNA). We see that the TMNA curve is always close to the $\log_{10}(\epsilon) = -16$ value meaning that $\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}$ is LU factorized[3] with the highest possible accuracy even when $\alpha = 2^{70} > 10^{16}$. On the other hand the MNA curve has a linear behavior till $\alpha = 2^{55} \simeq 10^{16}$ but some sawtooth oscillations; above this value the MNA matrix become singular.

V. Numerical Aspects as Further Improvements

In this section, we consider the effects of roundoff errors which are still present even though the proposed approach is adopted to build the transformed TMNA matrix. In fact, even if conductances belonging to the $\mathcal{G}_\sigma$ set are loaded only once, they are in general added to others in $\mathcal{G}_\varrho$, still leading to some roundoff errors.

Further, we show that, by exploiting the inherent circuit partition introduced by the $\mathcal{G}_\sigma$ and $\mathcal{G}_\varrho$ subsets, the Gauss–Jacobi iterative method can be employed to determine branch voltages referring to the elements in $\mathcal{G}_\sigma$ [9], [10]. The adoption of the Gauss–Jacobi is absolutely not mandatory but it can speed up the determination of the circuit solution with respect to the direct method based on the LU factorization of the TMNA matrix.

A. Iterative Improvement of the Accuracy

Since the TMNA matrix may be affected by roundoff errors, even if singularity has been ruled out by the transformations shown in the previous section, we once more adopt (13) to measure the accuracy in the computation of the inverse of a generic square matrix $\mathbf{Z}$. In our case, since in general $\mathbf{E} \neq \mathbf{0}$, we have that $\mathbf{VZ} \neq \mathbf{ZV}$ in most situations so that $\|\mathbf{VZ} - \mathbf{1}_n\|_F \neq \|\mathbf{ZV} - \mathbf{1}_n\|_F$; obviously we wish that both these norms converge to zero, corresponding to maximum accuracy.

When TMNA is applied, the system assumes the form

$$\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}\ \mathbf{v}^\clubsuit = \mathbf{P}\ \widehat{\mathbf{i}}^\diamondsuit. \tag{14}$$

In the following we show that even when $\rho(\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}) > 0$ an accurate solution can be found if the Newton iterative procedure is applied to the linear problem (14). While it may seem a paradox to apply the Newton method to a linear problem, recall that roundoff errors come from the discrete "nonlinear" behavior of the computer arithmetic unit [11]. The application of a similar method to the computation of least square solutions is discussed in [12], [13].

Due to roundoff errors, the solution of (14) is

$$\mathbf{v}^\clubsuit = \Big([\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}]^{-1} + \mathbf{E}\Big)\mathbf{P}\ \widehat{\mathbf{i}}^\diamondsuit.$$

We introduce the residue $\mathbf{r} = \mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}\ \mathbf{v}^\clubsuit - \mathbf{P}\ \widehat{\mathbf{i}}^\diamondsuit$ and the recursion $\mathbf{v}^\clubsuit_{k+1} = \mathbf{v}^\clubsuit_k - \Big([\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}]^{-1} + \mathbf{E}\Big)\mathbf{r}_k$ which represents the Newton method; some algebraic handling leads to

$$\mathbf{v}^\clubsuit_k = \left\{[\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}]^{-1} + \mathbf{E}(-\overbrace{\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}\ \mathbf{E}}^{\mathbf{H}})^k\right\}\mathbf{P}\ \widehat{\mathbf{i}}^\diamondsuit. \tag{15}$$

Note that the $\mathbf{H}$ recursion matrix is equivalent to the $\mathbf{VZ} - 1_m$ term in (13). The $\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}$ matrix in (14) can be LU decomposed, that is, $\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q} = \mathbf{L}\ \mathbf{U}$ and the solution of (14) is computed with a two step approach: first $\mathbf{x} = \mathbf{L}^{-1}\ \mathbf{P}\ \widehat{\mathbf{i}}^\diamondsuit$, then $\mathbf{v}^\clubsuit = \mathbf{U}^{-1}\mathbf{x}$, where both $\mathbf{L}$ and $\mathbf{U}$ can be affected by roundoff errors. Consider the $\mathbf{L}$ and $\mathbf{L}^{-1}$ matrices and denote as $\boldsymbol{\xi}_i$ the ith row of $\mathbf{L}$ and as $\boldsymbol{\zeta}_j$ the jth column of $\mathbf{L}^{-1}$. The $\boldsymbol{\zeta}_j$ vectors are computed through backward substitutions thus, despite any

$$\begin{bmatrix} \widehat{\imath}^\diamondsuit_1 \\ \widehat{\imath}^\diamondsuit_2 \\ \widehat{\imath}^\diamondsuit_3 \\ \widehat{\imath}^\diamondsuit_4 \\ \widehat{\imath}^\diamondsuit_5 \end{bmatrix} = \overbrace{\begin{bmatrix} g_2+g_6 & g_2+g_6 & 0 & -g_6 & -g_6 \\ g_2+g_6 & g_1+g_2+g_6+g_7 & -g_7 & -g_6 & -g_6-g_7 \\ 0 & -g_7 & g_3+\boxed{g_4}+g_7 & -g_3 & g_7 \\ -g_6+\boxed{g_5} & -g_6 & -g_3 & g_3+g_6 & g_6 \\ -g_6 & -g_6-g_7 & g_7 & g_6 & g_6+g_7+g_8 \end{bmatrix}}^{\mathbf{P}\ \mathbf{G}^\bullet\ \mathbf{Q}} \begin{bmatrix} v^\clubsuit_1 \\ v^\clubsuit_2 \\ v^\clubsuit_3 \\ v^\clubsuit_6 \\ v^\clubsuit_5 \end{bmatrix} \tag{12}$$

[3] The routines employed for LU factorization are those from the LAPACK library.

possible roundoff error, the $\boldsymbol{\xi}_i$ and $\boldsymbol{\zeta}_j$ vectors satisfy, by construction, the following properties.

1) The scalar product $\langle \boldsymbol{\xi}_i, \boldsymbol{\zeta}_j \rangle = 1$ if $i = j$.
2) $\langle \boldsymbol{\xi}_i, \boldsymbol{\zeta}_j \rangle = 0$ if $i < j$.

Therefore, the recursion matrix $\mathbf{H_L} \equiv \mathbf{L}\ \mathbf{L}^{-1} - \mathbf{1}_n$ (referring to $\mathbf{x} = \mathbf{L}^{-1}\ \mathbf{P}\ \widehat{\mathbf{i}}^{\diamondsuit}$) is lower triangular with zero entries on its main diagonal and it is immediate to prove that $\mathbf{H}_\mathbf{L}^k = \mathbf{0}$ for some $k \leq L$ where L is the dimension of $\mathbf{L}$. The same reasoning can be applied to $\mathbf{U}$, so that we can conclude that (14) is solved exactly in no more than $2L$ steps. In practice, the relative magnitude of roundoff errors is related to the precision of the arithmetic unit of the computer, so that, in most cases, only few iterations of the Newton algorithm are needed to have an accurate solution despite the L value.

B. Gauss–Jacobi Implementation

The application of TMNA has lead to formulation (10) where conductances in $\mathcal{G}_\sigma$ contribute a single entry as shown by the $\mathbf{P}\ \mathbf{G}^{\bullet}\ \mathbf{Q}$ matrix in (12). This formulation suggests a straightforward partition of the circuit and of the Jacobian matrix. In the following, this property is exploited and a version of the Gauss–Jacobi iterative method that solves (10) is introduced. The adoption of the Gauss–Jacobi method is absolutely not mandatory; we mention it since in some cases it can speed up the solution of the circuit. We recall that the aim of the approach proposed in this paper is as a "fall-back" method to improve robustness of simulators in extreme cases where otherwise they could fail. In this situations we believe a possible loss of efficiency is largely tolerated.

Before describing our version of the Gauss–Jacobi method a drawback of TMNA must first be underlined: linear combinations of rows and columns of the matrix in (6) partially destroy the sparsity typical of the MNA formulation. This increases the effort to LU factorize the matrix in (10). The employment of the Gauss–Jacobi method partially solves this drawback.

To simplify notation, we recast matrices introduced before as

$$\begin{aligned}\widetilde{\mathbf{G}} &= \mathbf{P}\mathbf{G}^{\bullet}\mathbf{Q}\\ \widetilde{\mathbf{A}} &= \mathbf{P}\mathbf{A}_c\\ \widetilde{\mathbf{B}} &= \mathbf{B}_c\mathbf{A}_c^T\mathbf{Q}.\end{aligned}$$

Moreover, variables are arranged so that conductances in $\mathcal{G}_\sigma$ contribute exclusively to the main diagonal of $\mathbf{G}$ and are limited to the $\mathbf{G}_{\sigma\sigma}$ submatrix defined in (16); this allow us to partition and rewrite (10) as

$$\begin{bmatrix} \mathbf{G}_{\sigma\sigma} & \mathbf{G}_{\sigma\varrho} & \mathbf{A}_\sigma \\ \mathbf{G}_{\varrho\sigma} & \mathbf{G}_{\varrho\varrho} & \mathbf{A}_\varrho \\ \mathbf{B}_\sigma & \mathbf{B}_\varrho & -\mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{v}_\sigma^{\clubsuit} \\ \mathbf{v}_\varrho^{\clubsuit} \\ \mathbf{i}_c \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{i}}_\sigma^{\diamondsuit} \\ \widehat{\mathbf{i}}_\varrho^{\diamondsuit} \\ \widehat{\mathbf{v}}_c \end{bmatrix} \tag{16}$$

where

$$\widetilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G}_{\sigma\sigma} & \mathbf{G}_{\sigma\varrho} \\ \mathbf{G}_{\varrho\sigma} & \mathbf{G}_{\varrho\varrho} \end{bmatrix} \quad \widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_\sigma \\ \mathbf{A}_\varrho \end{bmatrix} \quad \widetilde{\mathbf{B}} = [\mathbf{B}_\sigma\ \ \mathbf{B}_\varrho]$$

and source variables and unknowns are partitioned accordingly. The $v_\sigma^{\clubsuit}$ vector represents branch voltage of elements in $\mathcal{G}_\sigma$, $v_\varrho^{\clubsuit}$ those in $\mathcal{G}_\varrho$ and $\mathbf{i}_c$ represents currents flowing through c-branches. In the sequel, to ease notation, the $\clubsuit$ and $\diamondsuit$ superscripts are omitted in vectors $v^{\clubsuit}$ and $\widehat{\mathbf{i}}^{\diamondsuit}$.

In turn, we decompose $\mathbf{G}_{\sigma\sigma}$ as the sum of three matrices: $\mathbf{G}_{\sigma\sigma} = \mathbf{D}_\sigma + \mathbf{L}_\sigma + \mathbf{U}_\sigma$ where $\mathbf{D}_\sigma$ is composed of only the diagonal elements of $\mathbf{G}_{\sigma\sigma}$, $\mathbf{L}_\sigma$ is composed of the elements strictly below the diagonal of $\mathbf{G}_{\sigma\sigma}$ and $\mathbf{U}_\sigma$ is composed of the remaining upper elements of $\mathbf{G}_{\sigma\sigma}$. Now, we can introduce the recursive equation (17), shown at the bottom of the page, where, in general, $\mathbf{M}_\varrho$ is sparse. Substitution of the first equation in (17) into the second, yields

$$\begin{aligned}\mathbf{v}_\sigma^{k+1} =& \mathbf{D}_\sigma^{-1}\Bigg\{ - (\mathbf{L}_\sigma + \mathbf{U}_\sigma)\\ &+ [\mathbf{G}_{\sigma\varrho}\ \ \mathbf{A}_\sigma] \begin{bmatrix} \mathbf{G}_{\varrho\varrho} & \mathbf{A}_\varrho \\ \mathbf{B}_\varrho & \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{G}_{\varrho\sigma} \\ \mathbf{B}_\sigma \end{bmatrix} \Bigg\} \mathbf{v}_\sigma^k\\ &+ \mathbf{D}_\sigma^{-1}\Bigg\{\widehat{\mathbf{i}}_\sigma - [\mathbf{G}_{\sigma\varrho}\ \ \mathbf{A}_\sigma]\begin{bmatrix} \mathbf{G}_{\varrho\varrho} & \mathbf{A}_\varrho \\ \mathbf{B}_\varrho & \mathbf{R} \end{bmatrix}^{-1}\begin{bmatrix} \widehat{\mathbf{i}}_\varrho \\ \widehat{\mathbf{v}}_c \end{bmatrix}\Bigg\}\end{aligned} \tag{18}$$

which represents the Gauss–Jacobi iteration: the $\mathbf{v}_\sigma^{k+1}$ vector is computed starting from the known terms and from the previous $\mathbf{v}_\sigma^k$ estimation of the solution [10]. From (18) we see that only the $\mathbf{M}_\varrho$ matrix is LU factorized; we thus expect that, when suitable conditions about convergence of (18) hold, (17) can be solved in a more efficient way compared to the direct solution of (16) mainly for large $\mathbf{G}_{\sigma\sigma}$ submatrices. To introduce conditions about convergence, from (18), we derive

$$\begin{aligned}&\mathbf{v}_\sigma^{k+1} - \mathbf{v}_\sigma^k\\ &\quad = \mathbf{D}_\sigma^{-1}\Bigg\{ -(\mathbf{L}_\sigma + \mathbf{U}_\sigma) + [\mathbf{G}_{\sigma\varrho}\ \ \mathbf{A}_\sigma]\begin{bmatrix} \mathbf{G}_{\varrho\varrho} & \mathbf{A}_\varrho \\ \mathbf{B}_\varrho & \mathbf{R} \end{bmatrix}^{-1}\begin{bmatrix} \mathbf{G}_{\varrho\sigma} \\ \mathbf{B}_\sigma \end{bmatrix}\Bigg\}\\ &\quad\times \left(\mathbf{v}_\sigma^k - \mathbf{v}_\sigma^{k-1}\right)\end{aligned} \tag{19}$$

which shows that $\lim\limits_{k\to\infty}\left(\mathbf{v}_\sigma^{k+1} - \mathbf{v}_\sigma^k\right) = \mathbf{0}$ if the λ_i eigenvalues of the recursion matrix

$$\mathbf{R}_\sigma = \mathbf{D}_\sigma^{-1}\Bigg\{ -(\mathbf{L}_\sigma + \mathbf{U}_\sigma) + [\mathbf{G}_{\sigma\varrho}\ \ \mathbf{A}_\sigma]\begin{bmatrix} \mathbf{G}_{\varrho\varrho} & \mathbf{A}_\varrho \\ \mathbf{B}_\varrho & \mathbf{R} \end{bmatrix}^{-1}\begin{bmatrix} \mathbf{G}_{\varrho\sigma} \\ \mathbf{B}_\sigma \end{bmatrix}\Bigg\} \tag{20}$$

are inside the unit circle in the complex plane. Some bounds on these eigenvalues can be obtained by means of the Geršgorin theorem [10], however we believe that it is very difficult to give

$$\begin{cases} \begin{bmatrix} \mathbf{v}_\varrho^{k+1} \\ \mathbf{i}_c^{k+1} \end{bmatrix} = \overbrace{\begin{bmatrix} \mathbf{G}_{\varrho\varrho} & \mathbf{A}_\varrho \\ \mathbf{B}_\varrho & -\mathbf{R} \end{bmatrix}^{-1}}^{\mathbf{M}_\varrho} \left\{ \begin{bmatrix} \widehat{\mathbf{i}}_\varrho \\ \widehat{\mathbf{v}}_c \end{bmatrix} - \begin{bmatrix} \mathbf{G}_{\varrho\sigma} \\ \mathbf{B}_\sigma \end{bmatrix} \mathbf{v}_\sigma^k \right\} \\ \mathbf{v}_\sigma^{k+1} = \mathbf{D}_\sigma^{-1}\left\{ \widehat{\mathbf{i}}_\sigma - (\mathbf{L}_\sigma + \mathbf{U}_\sigma)\mathbf{v}_\sigma^k - [\mathbf{G}_{\sigma\varrho}\ \ \mathbf{A}_\sigma] \begin{bmatrix} \mathbf{v}_\varrho^{k+1} \\ \mathbf{i}_c^{k+1} \end{bmatrix} \right\} \end{cases} \tag{17}$$

bounds to the entries of the submatrices in (20) such that convergence of the Gauss–Jacobi method is ensured. If (20) were composed of only the $\mathbf{L}_\sigma + \mathbf{U}_\sigma$ matrices, we would be able to say that the diagonally dominance of $\mathbf{G}_{\sigma\sigma}$ ensures convergence. In fact eigenvalues would be inside circles in the complex plane centred in 0 with radius less than 1. Unfortunately, the right-most term contributes entries to the recursion matrix. For example, if we think of an almost singular $\mathbf{M}_\varrho$ matrix, we expect very large entries in $\mathbf{M}_\varrho^{-1}$ and thus also in the recursion matrix, possibly leading to eigenvalues outside the unit circle.

By considering the numerical implementation, we decided, for this reason, to monitor the $\left\|\mathbf{v}_\sigma^{k+1} - \mathbf{v}_\sigma^k\right\|_2$ norm and to see if it gives rise to a decreasing succession : in case the $\left\|\mathbf{v}_\sigma^{k+1} - \mathbf{v}_\sigma^k\right\|_2 < \left\|\mathbf{v}_\sigma^k - \mathbf{v}_\sigma^{k-1}\right\|_2$ inequality is not satisfied, (10) is solved with a direct method [1], [4].

In the following, as an example, we report the application of the Gauss–Jacobi method to problem (12) referring to the circuit shown in Fig. 5, already partitioned in the $\mathcal{G}_\sigma$ and $\mathcal{G}_\varrho$ subsets. In this case the $\mathbf{R}$, $\mathbf{A}_\sigma$, $\mathbf{A}_\varrho$, $\mathbf{B}_\sigma$ and $\mathbf{B}_\varrho$ matrices are not present and

$$\mathbf{G}_{\sigma\sigma} = \begin{bmatrix} -g_7 + \boxed{g_6} & -g_4 \\ 0 & g_4 + \boxed{g_5} + g_8 \end{bmatrix}$$

$$\mathbf{G}_{\sigma\varrho} = \begin{bmatrix} -g_7 & g_4 + g_7 & g_7 \\ -g_8 & -g_4 & g_8 \end{bmatrix}$$

$$\mathbf{G}_{\varrho\sigma} = \begin{bmatrix} g_3 + g_7 & 0 \\ g_3 + g_7 & -g_8 \\ -g_7 & g_8 \end{bmatrix}$$

$$\mathbf{G}_{\varrho\varrho} = \begin{bmatrix} g_3 + g_7 & -g_7 & -g_7 \\ g_2 + g_3 + g_7 + g_8 & -g_7 & -g_7 - g_8 \\ -g_7 - g_8 & g_7 & g_7 + g_8 + g_9 \end{bmatrix}.$$

If we chose, for example, $g_1 = g_2 = g_3 = g_4 = g_7 = g_8 = g_9 = 1$ mS and $g_5 = g_6 = 1$ kS, the eigenvalues of $\mathbf{R}_\sigma$ are $-3.6458\,10^{-9}$ and $1.6458\,10^{-9}$ and the Gauss–Jacobi method converges to the solution with a relative accuracy equal to that of the ALU in four iterations. Convergence speed further improves if the values of $g_5 = g_6$ are increased. If we chose $g_8 = -1.5$ mS $+$ 5 nS, $\mathbf{G}_{\varrho\varrho}$ is almost singular and the two eigenvalues of $\mathbf{R}_\sigma$ are 0.325 and $-1.5385\,10^{-10}$, respectively; the Gauss–Jacobi method converges to a solution with the above mentioned accuracy in 34 iteration. If the value of g_8 is further lowered from -1.5 mS$+5$ nS to -1.5 mS, at which $\mathbf{G}_{\varrho\varrho}$ is singular, below a certain value the Gauss–Jacobi method does no longer converge, since at least one of the two eigenvalues goes outside the unit circle.

VI. Simulation Results

In this section, we report some results about practical circuits that can cause failures of simulators due to roundoff errors inherent to the conventional MNA formulation.

The circuit shown in Fig. 7 implements a charge pump that can revert the polarity of the C1 capacitor voltage $v_c(t)$.

The M1, M3, and M2, M4 pairs of MOSFET transistors are alternatively turned-on and -off, never simultaneously closed to avoid cross-conduction. Their switching regulates, through current injection, the charge stored into the C1 capacitor and the corresponding $v_c(t)$ voltage. The operational amplifiers have high input impedance, sense $v_c(t)$ and refer the branch voltage to ground at the output. We shall evidence that all the transistors can be left open (turned-off), so that C1 is floating.

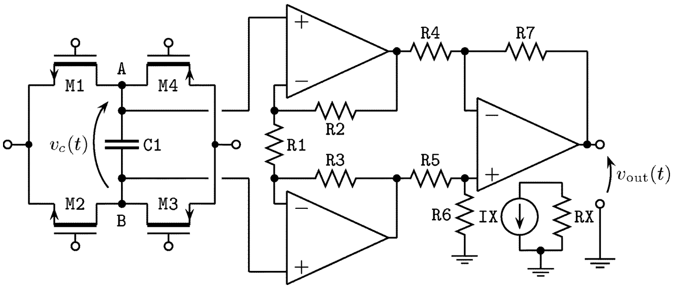


Fig. 7. Schematic of the charge pump.

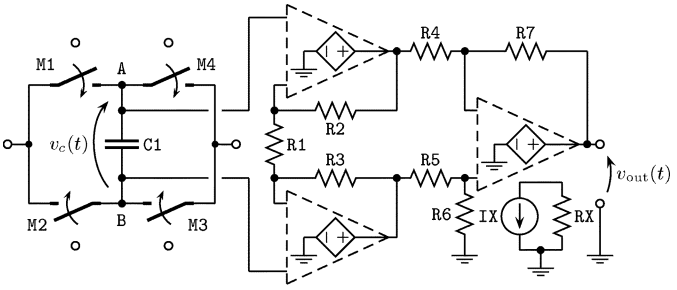


Fig. 8. Simplified model of the charge pump shown in Fig. 7.

Suppose that the charge pump is embedded in a larger circuit, for example in a phase locked loop, and that the designer decides to simplify the model to speed up simulation, while keeping its main behavior. To this end, the MOSFETs are substituted by voltage-controlled switches and the operational amplifiers by large gain voltage-controlled voltage sources. The simplified model is shown in Fig. 8.

Assume that the voltage-controlled switches are characterized by the $r_{\rm on} = 100\ \Omega$ resistance when closed and by the $r_{\rm off} = 0.1$ TΩ resistance when open. During the time-domain analysis, at each time instant at which the circuit solution is determined, the capacitor is represented by its "companion model" (see the introduction) composed of a resistor with resistance $r_{\rm eq} = h/C_1$ connected in parallel to an independent current source. The $r_{\rm eq}$ value depends on h, therefore, if $h \to 0$ and the four switches are open, the circuit of Fig. 8 can suffer of roundoff errors since the very small resistance $r_{\rm eq}$ is connected to the very large ones (the $r_{\rm off}$ resistances of the switches). In our case, being $\texttt{C1} = 10$ nF, an integration step $h \leq 1$ ps yields $r_{\rm eq} = 1$ mΩ, enough to introduce significant roundoff errors and cause convergence failures of conventional simulators.

In general, circuit simulators employ a unique integration time step h for the entire circuit; roughly speaking the minimum value of h depends on the "fastest" circuit dynamics. If the circuit is "stiff" or there is a fast variation of the driving waveforms, h is automatically shortened to meet the requested accuracy. Consider now the portion of the circuit composed of the IX independent current source and the RX linear resistor: it is completely decoupled meaning that the electrical solutions are completely independent. At the time instant t_x, the IX source generates a current step of 1 mA with a 1 pS rising front. If, as

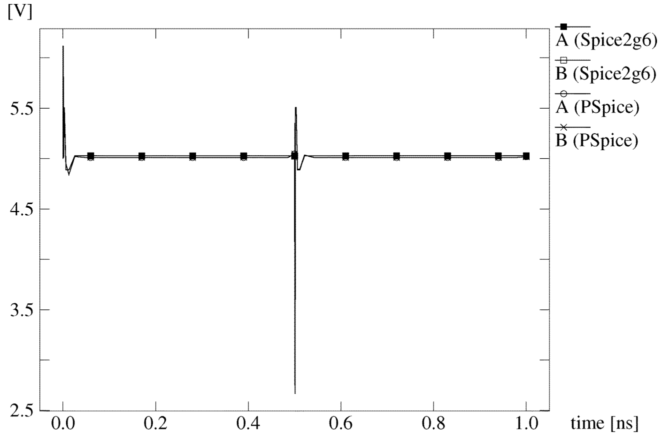
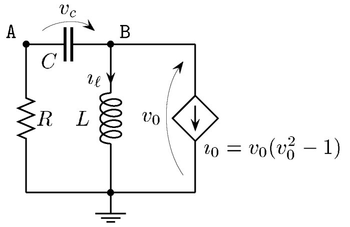


Fig. 9. Output from Spice2g6 and PSpice, the waveforms are affected by artifacts due to roundoff errors.

said above, the four switches are open, conventional simulators fail warning the user that the MNA matrix is singular.[4]

Apparently, this seems unbelievable, since it is impossible that the generation of a sharp current step in a portion of the circuit gives problems in another quiescent and decoupled portion of the same circuit. It is obvious that, if the value of the capacitor is lowered, the matrix is no longer singular; in particular, the value of C1 can be patiently chosen so that the matrix is ill conditioned but not singular. This allows some simulators, for example the old Spice2g6 and the more recent PSpice, to complete the analysis, even though "artifacts" are introduced in the solution. Fig. 9 shows the voltages computed by Spice2g6 and PSpice at the A and B nodes of the circuit of Fig. 8.[5]

These waveforms must be coincident and constant, since they refer to a quiescent portion of the circuit and C1 is at 0 V; instead we clearly see some artifacts at the beginning of the analysis and at $t_x = 5$ nS. The presence of artifacts at the beginning of the analysis can be explained because Spice2g6 and PSpice try to increase the integration time step h starting from a conservative fraction of the simulation interval that in our case is 1 ns. We did the same analysis with either some other "University simulators" such as Spice3, Gnucap, NgSpice or commercial ones such as Eldo, Spectre and Aplac; all these failed or introduced artifacts.[6] Note that roundoff errors that affect the circuit of Fig. 8 depend on the ratio $r_{\text{eq}}/r_{\text{off}}$. Even if r_{off} is decreased, roundoff errors can still be present if the value of C1 is increased of the same measure. Addition of small parasitic capacitors in parallel to the four switches to take into account those of MOSFETs can limit effects of roundoff errors.



```
simulator lang=spectre

global 0

Options options rforce=1u

TranB tran stop=10 ic=all maxstep=1m \
           errpreset=conservative

p  (B  0)  B  0  pvccs coeffs=[0 -1 0 1]
l  (B  0)  inductor  l=1 ic=199u
c  (B  A)  capacitor c=1 ic=200u
r  (A  0)  resistor  r=1
```

Fig. 10. Schematic of the Van der Pol oscillator and the related netlist for the Spectre simulator $R = 1\ \Omega,\ C = 1$ F, $L = 1$ H, $i_0 = v_0(v_0^2 - 1)$.

When the Newton method does not converge at a time instant during a time-domain analysis, it is common practice to reduce the integration time step, since it is expected that, thanks to the "continuity property" of the solution, time step shortening should help convergence. This action, in presence of the mentioned roundoff errors, does exactly the opposite as we have shown in the paper. Further, the usage of a common integration step h for all circuit equations can yield roundoff problems in stiff circuits, since capacitors whose values can differ of some magnitude orders are present.

If TMNA is adopted, the Newton algorithm converges in 2 iterations at the critical t_x time instant; the simulation successfully terminates and the obtained results have the required accuracy.[7] In this case, the model of the linear capacitor is "monitored" and the proposed transformation is applied if it leads to a "critical" equivalent conductance.

The second circuit considered is a version of the well known and largely studied Van der Pol oscillator. Its schematic is shown in Fig. 10 together with the Spectre simulator netlist; with respect to the conventional version it has the peculiarity of having the *R* resistor connected in series to the *C* capacitor.

The initial condition for the *C* capacitor voltage v_c is 200 μV and that for the L inductor current i_ℓ is 199 μA; the parameters of the oscillator are so that the $v_0(t)$ voltage is damped. When $v_0(t) = 0$ and $i_\ell(t) \neq 0$ the VCCS modeling the nonlinear resistor is working in the origin of its trans-characteristic and has a differential conductance $di_0/dv_0|_{v_0=0} = -1$ S and, in this case, the *C* capacitor is short circuited by the series of the *R* resistor and of the VCCS. In turn this means that the dv_c/dt "speed" at which the *C* capacitor charges/discharges diverges in a neighborhood of $v_0(t) = 0$. This effect is depicted

[4] Some simulators indicate nodes A and B as possible location of singularity.

[5] The circuit parameters that introduce artifacts in the solution strongly depends on the ALU and the operating system running on the computer. Therefore some tuning may be needed to reproduce these results.

[6] Spectre (version 4.4.6–012 103) fails giving the message that the MNA matrix is singular. Eldo (version V5.6_1_2) probably adopts a circuit partitioner and splits the circuit in Fig. 8 in two decoupled subcircuits. Therefore the application of the current step does not give any problem since only the subcircuit composed of IX and RX is solved. However, it fails to solve the simple circuit of Fig. 2 computing a largely wrong solution. If the circuit of Fig. 8 is modified to force the solution of the entire circuit, it once more fails. Spectre and Eldo run on sun4u sparc Ultra-60 SunOS 5.8. Aplac (version 7.90) gives erroneous results affected by artifacts.

[7] The LINUX executable of our simulator, named *pan*, implementing TMNA and the netlists of the reported circuits are available at the URL http://brambilla.elet.polimi.it. It was tested on the LINUX Debian 3.0 release and we expect that it also works on other standard LINUX releases.

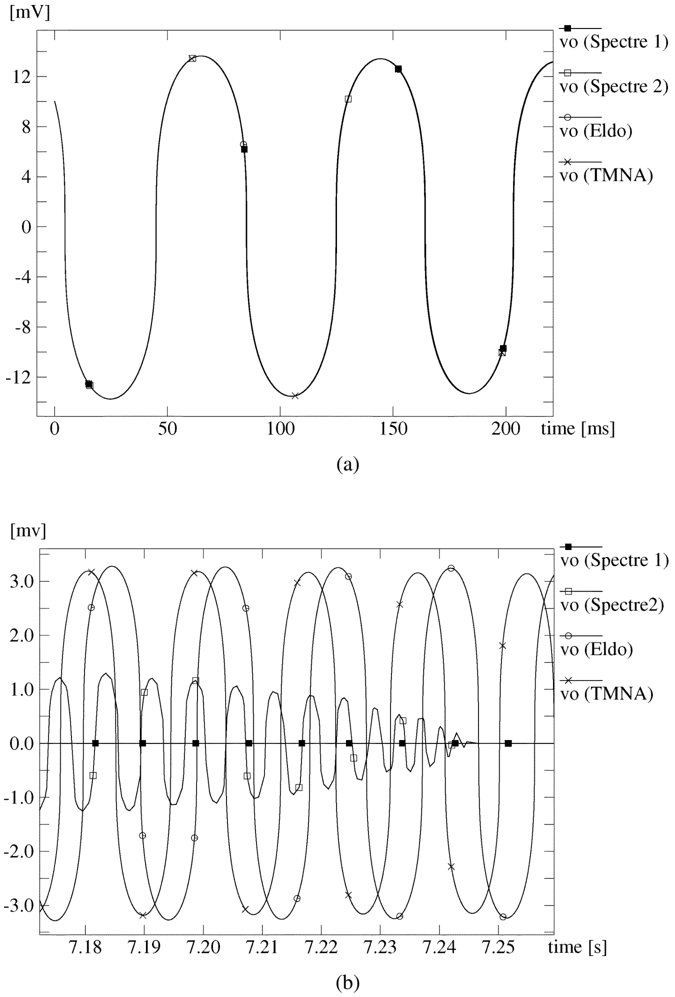
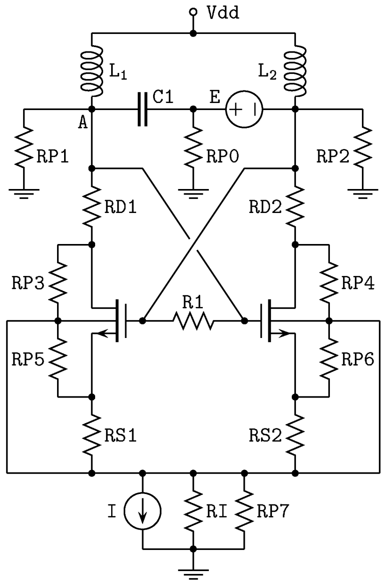


Fig. 11. (a) The $v_0(t)$ voltage computed by Spectre, Eldo and TMNA. In particular (Spectre 1) refers to the circuit netlist reported in Fig. 10 while (Spectre 2) refers to the same circuit but *C* and *R* swapped. (b) A portion of the $v_0(t)$ voltage waveforms computed by Spectre, Eldo, and TMNA after about 7.18 s from the beginning of the transient analysis.



```
* differential oscillator
* node 10 --> A

.tran 1p 200p 0 0.1p uic
.tran 1p 30m uic

.model nm nmos level=3 kp=1m vto=3
+              cgso=0 rd=10 rs=10

l1 100 10 1m ic= 646m
l2 100 20 1m ic=-636m
c1  90 20 2m ic=1.55
r1  10 20 650
vl  60 20 dc 0
m1  10 60 30 30 nm w=1u l=1u
vm1 30 50 dc 0
m2  20 10 40 40 nm w=1u l=1u
vm2 40 50 dc 0
i   50  0 dc 10m
ri  50  0 1000Meg
vdd 100 0 dc 5

ve  10 90 pulse(0 10 5p 1p 1p 1m 1)
.end
```

Fig. 12. Schematic of the differential oscillator and the related Spice netlist.

in Fig. 11(a) by the almost vertical portions of the waveforms. The infinite speed in charging/discharging of the capacitor constraints simulators to shrink the h integration time step; the entries on the MNA matrix related to the A and B nodes can thus be affected by roundoff errors. On the other hand, if *C* and *R* are swapped when h is shortened the *C* capacitor, being grounded, contributes only one entry in the conventional MNA matrix and therefore roundoff errors, even though present, are not critical.

This simple circuit was simulated with Spectre, Eldo, Spice2g6 and with the our approach. Fig. 11(a) shows the corresponding $v_0(t)$ waveforms. In particular the waveform labeled as (Spectre 1) refers to the circuit whose netlist is reported in Fig. 10, while that labeled as (Spectre 2) refers to the same netlist but *C* and *R* swapped. We see that the waveforms agree very well. Fig. 11(b) shows the same waveforms after about 7.18 s of transient analysis. The two results computed by Spectre are unexpectedly very different even though element swapping preserves the electrical characteristics of the circuit. In particular we see that the (Spectre 1) waveform is constant at 0 V while (Spectre 2) quickly dampens to 0 V. Furthermore for what concerns the first waveform, Spectre simulator displayed messages similar to the following one:

```
Warning from spectre at time = 357.681 ms during
transient analysis "tranb."
Convergence difficulties resulted in
error requirements being unsatisfied.
```

and “skipped” forward the simulation time in order to complete in some way its job. The old Spice2g6 simulated the Van der Pol oscillator till about 4 s then aborted with the often seen message “time step too small.” An accurate inspection of the output waveform revealed that there are no time points that meet the condition $v_0(t) = 0$. Therefore, probably Spice2g6 failed just when it met one of those points. The waveforms computed by Eldo and TMNA have the same shape but they are time shifted; this effect is due to "warping" introduced by integration methods [15].

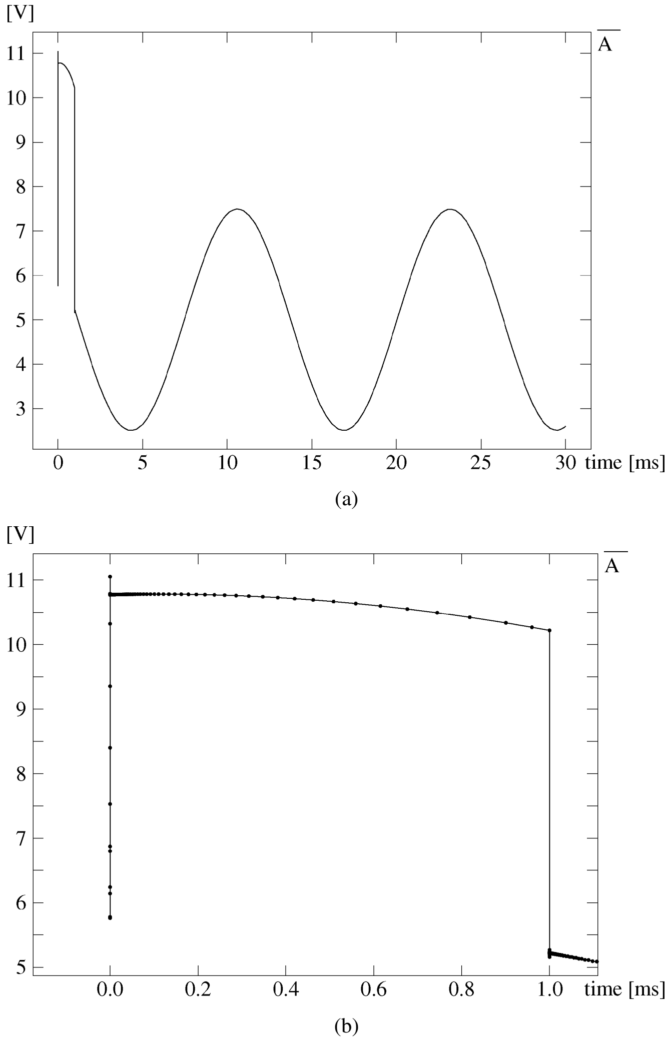
The third circuit considered is shown in Fig. 12; it is a differential oscillator largely studied in literature (see e.g., [14]).

In the schematic, we have reported also "hidden" resistances, labeled with the RP prefix, automatically introduced by most circuit simulators to improve robustness. In general these resistances are set to 1 TΩ $(1/g_{\min})$. In Fig. 12, the parasitic RD drain and RS source resistances of MOSFET transistors are also considered. The model used for the MOSFETs is Spice level 3 and the Spice netlist is reported in Fig. 12.

To quick start the oscillator a voltage pulse is generated by the E independent voltage source. The pulse has rising and falling edges of 1 ps, a duration of 1 ms and an amplitude of 10 V. Once more the application of a pulse causes a drastic reduction of h that, in turn, leads to roundoff errors mainly in the A node, where "large" and "small" conductances sum. Recalling that inductors lead to a Thévenin companion model with resistance L/h, when $h \rightarrow 0$ the two inductors in Fig. 10 are equivalent to "open circuits", while the C1 capacitor contributes a "large" conductance to node A. This circuit causes failures of Eldo and Spectre, while PSpice and Spice2g6 introduce artifacts in the solution. Our circuit simulator, with TMNA enabled, successfully ends the simulation and ensures the required accuracy. Fig. 13 shows the voltage waveform at node A (TMNA enabled); the voltage pulse that triggers the oscillator is clearly visible.

There are other classes of circuits that can lead to convergence problems due to roundoff errors. Among them, we recall those known as "post layout" circuits whose netlists are extracted from layouts and thus include models of interconnects. In very large-scale layouts, delays introduced by interconnects can be comparable to those of gates and must be accounted for in the design of the electronic circuits. In general, they are modeled by a lumped nonhomogeneous capacitor–resistor ladder network. This model is directly derived by extraction tools from the geometrical three-dimensional description of the layout. It can happen that some very small parasitic resistors of a few tenths of microohms are extracted, possibly modeling a short piece of metal between an array of contacts (vias). In general the extraction tool can be driven to ignore or to group very small resistance, they are thus present in rare cases, but, if this happens, these very small resistances can be connected to very large ones modeling turned-off MOSFETs and, once more, the simulator can fail due to roundoff errors.

For example, consider the circuit of the three-state inverter shown in Fig. 14. The inverter input is labeled as A, the output enabling terminal as OE and the output as Y. The RP resistor of 10 $\mu\Omega$ connects the output Y to the gate of the M13 MOSFET. When the output is disabled the M11 and M12 MOSFETs show a very high equivalent drain-source resistance.



(a)

(b)

Fig. 13. (a) Voltage waveform at node A of the oscillator shown in Fig. 10 obtained with TMNA enabled. (b) Enlarged portion of Fig. 11(a) showing the voltage pulse that triggers the oscillators. Dots correspond to the time instants of simulation.

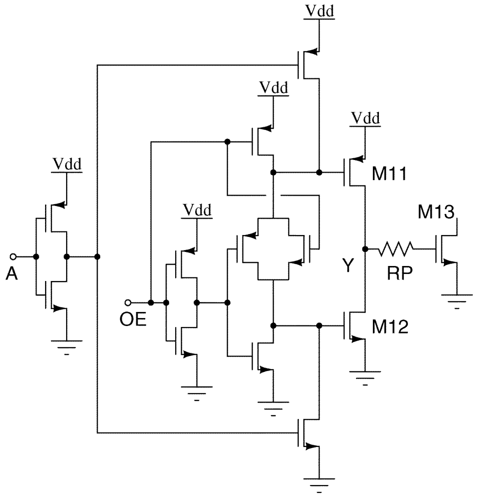


Fig. 14. Schematic of the CMOS three-state inverter.

Suppose that we are using two different MOSFET models, for example, the *BSIM3v3* by the University of California, Berkeley and the *M903* by Philips. The original version of the latter does not have any drain/source resistance that models for example the contacts; this means that the drain/source equivalent resistances of the M11 and M12 MOSFETs are connected directly to RP and roundoff errors can be thus introduced. Indeed our simulator ends with a singular matrix error possibly at node Y when conventional MNA is employed. On the other hand, the first MOSFET model adopts parasitic drain/source resistances (in general of a few ohms) so that the RP resistance is no longer directly connected to the equivalent drain/source resistances of the MOSFETs but to these parassitic ones whose values are adequate to avoid roundoff errors. In this case our simulator does not have any problem when conventional MNA is employed.

A simple proof that confirms what happens is obtained by inserting in series to RP two other resistors that connect it to the Y output of the inverter (in other words the insertion is made on the left terminal of RP). These two new resistors have a value of $1\ \Omega$ and $-1\ \Omega$, respectively. We can immediately state that their insertion *does not modify* the solution of the circuit being equivalent to a short circuit ! On the other hand, the MOSFETs (M903 model) are no longer directly connected to RP but to a resistor of $\pm 1\ \Omega$ and our simulator successfully ends its job without turning the TMNA on, not suffering of any roundoff error.

VII. Conclusion

The paper has revisited the classical MNA used in many of the electronic circuit simulators. In particular, due to the finite precision of computer arithmetics, the MNA can fail when incomparable levels of conductances appear in the circuit. These disagreements among conductances can generate cancellations of too small conductances in the system of equations describing the circuit. In turn, these cancellations can generate ill-conditioning till, possibly, singularity of the system matrix.

The paper has proposed a recasting of the system by transforming the unknown vector and linearly combining the scalar equations, avoiding the effects of critical cancellations of small conductances. The proposed algorithm was implemented in our simulator *pan*: it behaves successfully for critical circuits where the commercial simulators generally fail.

References

[1] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrand Reinhold, 1983.

[2] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*. New York: Mc Graw-Hill, 1987.

[3] A. T. Yang, *Numerical Analysis Methods*, W.-K. Chen, Ed. Boca Raton, FL: CRC, 1995, The Circuits and Filters Handbook, ch. 53, pp. 1412–1427.

[4] L. T. Pillage and R. A. Rohrer, *Electric Circuits and System Simulation Methods*. New York: McGraw-Hill, 1995.

[5] J. K. Reid, *Large Sparse Sets of Linear Equations*. London, U.K.: Academics, 1971.

[6] K. S. Kundert, "Sparse matrix technique," in *Circuit Analysis, Simulation and Design*, A. Rüehli, Ed. Dordrecht, The Netherlands: North Holland, 1986, pt. 1, vol. 3.

[7] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. Oxford, U.K.: Clarendon Press, 1965.

[8] P. Maffezzoni, "Applying iterative methods to the analysis of VLSI circuits," *Int. J. Ciruit Theory Appl.*, vol. 32, pp. 91–96, 2004.

[9] J. K. White and A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*. Boston, MA: Kluwer, 1986.

[10] J. M. Ortega and W. G. Poole, *An Introduction to Numerical Methods for Differential Equations*. New York: Pitman, 1981.

[11] E. P. Rudd, "The solution of Ill-scaled circuit equations," *IEEE Trans Circuits Syst.*, vol. CAS-31, no. 7, pp. 671–672, Jul. 1984.

[12] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses*. New York: Wiley, 1974.

[13] J. Stoer, *Einfuhrung in die Numerische Mathematik*. New York: Springer-Verlag, 1972.

[14] E. Hegazi, H. Sjöland, and A. A. Abidi, "A filtering technique to lower *LC* oscillators phase noise," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1921–1930, Dec. 2001.

[15] A. Brambilla and G. Storti-Gajani, "Frequency warping in time-domain circuit simulation," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 7, pp. 904–913, Jul. 2003.

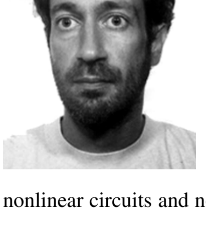
Angelo Brambilla received the Dr. Ing. degree in electronics engineering from the University of Pavia, Pavia, Italy, in 1986.

Currently, he is Full Professor in the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy, where he has been working in the areas of circuit analysis and simulation.

Amedeo Premoli was born in Crema, Italy, in 1942. He received the laurea degree (*summa cum laude*) in electronic engineering from Politecnico di Torino, Turin, Italy, in 1965.

From 1966 to 1986, he was a member of Istituto Elettrotecnico Nazionale Galileo Ferraris, Turin, Italy. From 1986 to 1995, he was a Full Professor of Elettrotecnica (Basic Circuit Theory) in the Dipartimento di Elettrotecnica, Elettronica ed Informatica (Department of Electrical and Computer Sciences), Università di Trieste, Trieste, Italy. He was a Visiting Professor at EPFL, Lausanne, Switzerland, in September 1984, June 1987, and October 1990. Since 1995, he has been a Full Professor of Elettrotecnica (Basic Circuit Theory) in the Dipartimento di Elettronica e Informazione (Department of Electronic and Computer Sciences), Politecnico di Milano, Milan, Italy. His main research interests are focused on circuit and system theory and applications, including, initially, the analysis and design of distributed circuits, approximation of transfer functions, design of electrical active filters and multilayer coating optical filters; more recently, the numerical processing of measurements, applications of optimization techniques, electrical and electrothermal simulation and dynamics of electronic circuits and cellular neural networks. He is author or coauthor of about 140 scientific papers (among which about 80 were published in international journals), and a book on design of electrical filters.

Prof. Premoli was an Associate Editor of IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications from 1993 to 1995.



Giancarlo Storti-Gajani received the Dr. Ing. degree in electronic engineering and the Ph. D. degree in electronic systems from Politecnico di Milano, Milan, Italy, in 1986, and 1991, respectively.

Since 1992, he has been an Assistant Professor, and since 2002, an Associate Professor of Circuit Theory at Politecnico di Milano. His research interests have been initially focused on the development of architectures for signal processing (in particular for audio and music applications) and neural networks. More recently, his research interests include nonlinear circuits and nonlinear dynamics.