Fast and Robust Inexact Newton Approach to the Harmonic-Balance Analysis of Nonlinear Microwave Circuits

Vittorio Rizzoli, Franco Mastri, Claudio Cecchetti, and Fiorella Sgallari

***Abstract*— The letter discusses a novel approach to nonlinear microwave circuit simulation by the harmonic-balance (HB) technique. The nonlinear system is solved by an inexact Newton method, and the GMRES iteration is used at each step to find a suitable inexact Newton update. The peculiar structure of the Jacobian matrix allows the basis vectors of the Krylov subspace to be computed mostly by the FFT. The resulting simulation tool is fast and robust, and outperforms conventional HB techniques when applied to large-size nonlinear analysis problems.**

***Index Terms*—Harmonic balance, Krylov subspace methods.**

I. Introduction

THE Newton-iteration-based harmonic-balance (HB) technique is generally acknowledged as the principal method for the analysis of nonlinear microwave circuits working in steady-state regime under multitone excitation. Harmonic balance simulators relying upon this method are robust and well behaved, and can normally reach convergence starting from zero harmonics even at very high drive levels [1]. The only outstanding drawback of this simulation approach lies in the huge demand of computer resources when the problem size becomes large. This is due to the fact that the storage of the Jacobian matrix requires N^2 words, and its factorization time is $O(N^3)$, where N is the number of scalar unknowns. These difficulties may be partly overcome making use of an artificially sparse Jacobian matrix coupled with sparse-matrix solvers [1], [2], but this unavoidably cripples the power-handling capabilities of the analysis algorithm. Thus, the need exists for an HB technique that can tackle large-size problems (say, $N \geq 10\,000$) on ordinary workstations, while fully retaining the convergence properties of traditional HB. An algorithm of this kind, named the inexact Newton harmonic balance (INHB) is outlined in this letter. The INHB provides the same robustness and accuracy as ordinary HB techniques with a dramatic reduction of memory storage and CPU time, and thus opens the way to the HB treatment of very large-sized simulation problems.

Manuscript received June 6, 1997. This work was supported in part by the Italian Ministry of University and Scientific Research (MURST) and by the Istituto Superiore delle Poste e delle Telecom-unicazioni (ISPT).

V. Rizzoli is with the Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, 40044 Pontecchio Marconi, Bologna, Italy.

F. Mastri is with the Dipartimento di Ingegneria Elettrica, University of Bologna, 40136 Bologna, Italy.

C. Cecchetti is with Fondazione Ugo Bordoni, 40044 Pontecchio Marconi, Bologna, Italy.

F. Sgallari is with the Dipartimento di Matematica, University of Bologna, 40123 Bologna, Italy.

Publisher Item Identifier S 1051-8207(97)07709-X.

II. Inexact Newton Harmonic-Balance Analysis

Let the HB solving system be formulated as a nonlinear system of N equations in N unknowns of the form $\mathbf{E}(\mathbf{X}) = \mathbf{0}$, where $\mathbf{E} : R^N \rightarrow R^N$ is continuously differentiable. Given an approximation $\mathbf{X}_i$ to the exact solution $\mathbf{X}$, the corresponding Newton update $\mathbf{n}_i$ is defined as the solution of Newton's equation $\mathbf{J}(\mathbf{X}_i)\mathbf{n}_i = -\mathbf{E}(\mathbf{X}_i)$, where $\mathbf{J}(\mathbf{X})$ is the Jacobian matrix of $\mathbf{E}$ with respect to $\mathbf{X}$. The Newton iteration is then defined by $\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{n}_i$. This solution technique requires the Jacobian matrix to be stored and factorized at each step, and is thus not well suited for large values of N because of memory and CPU time requirements. Both difficulties can be simultaneously overcome by resorting to an *inexact* Newton method [3]. An *inexact Newton update* is defined as a vector $\mathbf{s}_i$ satisfying the condition

$$\|\mathbf{E}(\mathbf{X}_i) + \mathbf{J}(\mathbf{X}_i)\mathbf{s}_i\| \leq f_i\|\mathbf{E}(\mathbf{X}_i)\| \qquad (0 \leq f_i < 1) \quad (1)$$

where f_i is named the *forcing term* [4]. The inexact Newton iteration is then defined by $\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{s}_i$. Note that for $f_i = 0$, $\mathbf{s}_i$ reduces to $\mathbf{n}_i$, so that the forcing term is in some way a measure of the allowed deviation of the inexact update from the Newton update. The iteration is terminated when the relative error on each element of $\mathbf{E}(\mathbf{X}_i)$ drops below a prescribed threshold.

Inexact Newton methods have a number of interesting features that make them an ideal choice for solving large-sized nonlinear systems. They can be globalized by suitable techniques [5], and their efficiency can be greatly improved by appropriately updating the forcing term at each step [4]. Above all, they do not require a large linear system to be exactly solved at each step, since $\mathbf{s}_i$ must satisfy (1), but is otherwise arbitrary. Instead, for a given f_i the inexact update can be iteratively refined starting from an arbitrary initial guess (zeroth-order approximation), until (1) is met. For application to HB analysis, the best results have been obtained making use of the GMRES iterative solver [6]. Let us introduce a suitable approximation $\mathbf{P}_i$ of $\mathbf{J}(\mathbf{X}_i)$. The initial guess is defined by

$$\mathbf{s}_i^{(0)} = -\mathbf{P}_i^{-1}\mathbf{E}(\mathbf{X}_i) \quad (2)$$

where $\mathbf{P}_i$ is named the *preconditioner* [7]. A set of real N-vectors defined by the following recursive relation is then

computed and stored:

$$\mathbf{K}_i^{(1)} = \left[\mathbf{1}_N - \mathbf{J}(\mathbf{X}_i)\mathbf{P}_i^{-1}\right]\mathbf{E}(\mathbf{X}_i)$$
$$\mathbf{K}_i^{(q)} = \mathbf{J}(\mathbf{X}_i)\mathbf{P}_i^{-1}\mathbf{K}_i^{(q-1)} \qquad (q>1) \tag{3}$$

where $\mathbf{1}_N$ is the identity matrix of order N. The vector space spanned by the vectors $\mathbf{K}_i^{(q)}$ with $1 \le q \le Q$ is called a *Krylov subspace of dimension* Q [6]. The Qth-order approximation is then expressed in the form [6]

$$\mathbf{s}_i^{(Q)} = \mathbf{s}_i^{(0)} + \mathbf{P}_i^{-1}\sum_{q=1}^{Q}\alpha_q\mathbf{K}_i^{(q)} \tag{4}$$

so that the corresponding *residual* is

$$\mathbf{r}_i^{(Q)} = \mathbf{E}(\mathbf{X}_i) + \mathbf{J}(\mathbf{X}_i)\mathbf{s}_i^{(Q)} = \mathbf{K}_i^{(1)} + \sum_{q=1}^{Q}\alpha_q\mathbf{K}_i^{(q+1)}. \tag{5}$$

The coefficients α_q are found by a least squares method in such a way that $||\mathbf{r}_i^{(Q)}||$ is minimized. $\mathbf{s}_i^{(Q)}$ is then taken as the inexact Newton update if (1) is satisfied. This result is guaranteed to be achieved for sufficiently large Q, because $\lim_{Q\to\infty}\mathbf{s}_i^{(Q)} = \mathbf{n}_i$ [6]. In practice it has been found that $Q \le 50$ is normally sufficient to obtain the desired accuracy in HB applications, even at high drive levels.

III. Generation of the Basis Vectors of the Krylov Subspace

With the INHB the CPU time required to exactly solve Newton's equation is essentially replaced by the time required to compute the basis vectors (3). As a matter of fact, the subsequent minimization process has negligible cost for large values of N, since the number of coefficients α_q is comparatively small. Most of the CPU time is thus spent in the multiplication of the right-preconditioned Jacobian matrix $\mathbf{J}(\mathbf{X}_i)\mathbf{P}_i^{-1}$ by a sequence of vectors. This basic operation can be performed very efficiently by exploiting the peculiar structure of the Jacobian matrix, as will be shown in this section. Note that the subscript "i" of the current inexact Newton step will be understood in the following for the sake of formal simplicity.

Making use of the piecewise formulation of the HB technique, the circuit is partitioned into a linear and a nonlinear subnetwork connected through n_D ports (*device ports*). The nonlinear subnetwork may be described by the parametric equations [1]

$$\mathbf{v}(t) = \mathbf{u}\left[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \ldots, \frac{d^n\mathbf{x}}{dt^n}, \mathbf{x_d}(t)\right]$$
$$\mathbf{i}(t) = \mathbf{w}\left[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \ldots, \frac{d^n\mathbf{x}}{dt^n}, \mathbf{x_d}(t)\right] \tag{6}$$

where $\mathbf{v}(t)$ and $\mathbf{i}(t)$ are vectors of voltages and currents at the common ports, $\mathbf{x}(t)$ is a vector of state variables (SV), and $\mathbf{x_d}(t)$ is a vector of time-delayed state variables, i.e., $x_{di}(t) = x_i(t-\tau_i)$. The dimension of all vectors in (6) is equal to n_D. The complex HB error at a generic intermodulation (IM) product $\Omega_\mathbf{k}$ of the forcing tones is given by [1]

$$\mathbf{E_k} = \mathbf{Y}(\Omega_\mathbf{k})\mathbf{U_k} + \mathbf{W_k} + \mathbf{N_k} \tag{7}$$

where $\mathbf{U_k}$, $\mathbf{W_k}$ are the $\mathbf{k}$th harmonics of (6), $\mathbf{Y}(\omega)$ is the linear subnetwork admittance matrix, and $\mathbf{N_k}$ is a vector of Norton equivalent current sources of the free generators of frequency $\Omega_\mathbf{k}$. Similarly, the $\mathbf{k}$th harmonic of $\mathbf{x}(t)$ will be denoted by $\mathbf{X_k}$.

In order to obtain a real solving system, the circuit equations are formulated in terms of the real and imaginary parts of both the HB errors and the SV harmonics. Accordingly, the real Jacobian matrix is partitioned frequency-wise into submatrices $\mathbf{J_{k,s}}$ of the form

$$\mathbf{J_{k,s}} = \begin{bmatrix} \dfrac{\partial \mathrm{Re}[\mathbf{E_k}]}{\partial \mathrm{Re}[\mathbf{X_s}]} & \dfrac{\partial \mathrm{Re}[\mathbf{E_k}]}{\partial \mathrm{Im}[\mathbf{X_s}]} \\ \dfrac{\partial \mathrm{Im}[\mathbf{E_k}]}{\partial \mathrm{Re}[\mathbf{X_s}]} & \dfrac{\partial \mathrm{Im}[\mathbf{E_k}]}{\partial \mathrm{Im}[\mathbf{X_s}]} \end{bmatrix}. \tag{8}$$

The basic operation to be performed in the construction of the Krylov subspace (3) is the multiplication of the Jacobian matrix by some real vector, say $\mathbf{g}$, to find another real vector, say $\mathbf{f}$. We may partition $\mathbf{f}, \mathbf{g}$ into subvectors frequency-wise in a way consistent with (8), so that the $\mathbf{k}$th subvector of $\mathbf{f}$ takes on the form

$$\mathbf{f_k} \triangleq \begin{bmatrix}\mathbf{c_k}\\ \mathbf{d_k}\end{bmatrix} = \sum_\mathbf{s}\mathbf{J_{k,s}}\ \mathbf{g_s} \triangleq \sum_\mathbf{s}\mathbf{J_{k,s}}\begin{bmatrix}\mathbf{a_s}\\ \mathbf{b_s}\end{bmatrix}. \tag{9}$$

From (8), (9) making use of (7) we obtain

$$\mathbf{c_k} = \mathrm{Re}\left[\mathbf{Y}(\Omega_\mathbf{k})\sum_\mathbf{s}\left\{\frac{\partial \mathbf{U_k}}{\partial \mathrm{Re}[\mathbf{X_s}]}\mathbf{a_s} + \frac{\partial \mathbf{U_k}}{\partial \mathrm{Im}[\mathbf{X_s}]}\mathbf{b_s}\right\}\right.$$
$$\left. + \sum_\mathbf{s}\left\{\frac{\partial \mathbf{W_k}}{\partial \mathrm{Re}[\mathbf{X_s}]}\mathbf{a_s} + \frac{\partial \mathbf{W_k}}{\partial \mathrm{Im}[\mathbf{X_s}]}\mathbf{b_s}\right\}\right]. \tag{10}$$

Note that $\mathbf{d_k}$ is the imaginary part of the same quantity. In turn, the derivatives of $\mathbf{U_k}$ are given by [1]

$$\frac{\partial \mathbf{U_k}}{\partial \mathrm{Re}[\mathbf{X_s}]} = \sum_{m=0}^{n}(j\Omega_\mathbf{s})^m[\mathbf{F}_{m,\mathbf{k-s}} + (-1)^m\mathbf{F}_{m,\mathbf{k+s}}]$$
$$\frac{\partial \mathbf{U_k}}{\partial \mathrm{Im}[\mathbf{X_s}]} = \sum_{m=0}^{n}j(j\Omega_\mathbf{s})^m[\mathbf{F}_{m,\mathbf{k-s}} - (-1)^m\mathbf{F}_{m,\mathbf{k+s}}] \tag{11}$$

where the complex matrices $\mathbf{F}_{m,\mathbf{p}}$ are linear combinations of the Fourier coefficients of the derivatives of (6) evaluated in steady-state conditions. Similar expressions hold for the derivatives of $\mathbf{W_k}$ [1]. By replacing (11) into (10) we may express the summations in brackets in the form

$$\sum_\mathbf{s}\left\{\frac{\partial \mathbf{U_k}}{\partial \mathrm{Re}[\mathbf{X_s}]}\mathbf{a_s} + \frac{\partial \mathbf{U_k}}{\partial \mathrm{Im}[\mathbf{X_s}]}\mathbf{b_s}\right\} = \sum_{m=0}^{n}\sum_\mathbf{s}\mathbf{F}_{m,\mathbf{k-s}}\mathbf{z}_{m,\mathbf{s}} \tag{12}$$

and the like, where $\mathbf{z}_{m,\mathbf{s}} = (j\Omega_\mathbf{s})^m(\mathbf{a}_s + j\mathbf{b}_s)$ and $\mathbf{z}_{m,-\mathbf{s}} = \mathbf{z}^*_{m,\mathbf{s}}$. The inner summation in (12) has the structure of a discrete convolution, and can thus be computed by the fast Fourier transform (FFT) [8]. The same procedure obviously applies to all subvectors $\mathbf{c_k}$, $\mathbf{d_k}$, so that the basis vectors of the Krylov subspace can be determined mostly by FFT's. The computation is thus very efficient, and the CPU time is a slowly increasing function of the number of spectral lines. The coefficients $\mathbf{F}_{m,\mathbf{p}}$ are computed and stored at the beginning of

each inexact Newton step, so that all the information needed to multiply the Jacobian matrix by a vector is available in the computer memory. Thus the entries of the Jacobian matrix need not be re-computed at each iteration, in spite of the fact that the matrix itself is not stored. In this way the memory savings are dramatic for large N, and a further enhancement of the computational efficiency is obtained.

Finally, it turns out that the block-diagonal Jacobian matrix obtained by letting $\mathbf{J}_{\mathbf{k},\mathbf{s}} = \mathbf{0}$ for $\mathbf{k} \neq \mathbf{s}$ represents an excellent preconditioner for broad classes of microwave circuits. Indeed, this matrix is inexpensive to store and to factorize due to its block-diagonal structure, and at the same time is accurate enough to ensure convergence of the Newton-iteration-based HB analysis at low drive levels [1].

IV. A Performance Benchmark

In order to demonstrate the numerical performance of the INHB, in this section we report the results of several analyses of a nine-FET distributed amplifier $(n_D = 18)$ driven by three tones of equal available power P_{in}. All the calculations are performed on an HP 755 workstation with a central memory of 192 MB, starting from zero harmonics. All IM products are determined to a relative accuracy better than 10^{-5}.

1) A small-signal IM test ($P_{\text{in}} = -10$ dBm) is carried out first, taking into account all IM products up to the fifth order. The number of positive IM products is then $P = 115$, so that $N = n_D(2P + 1) = 4158$. For this analysis the conventional HB with full Jacobian matrix requires 157 MB of memory (138 MB for the storage of the Jacobian matrix alone), and the CPU time is 4215 s. These figures drop to 17 MB and 40 s with an HB analysis based on the block-diagonal Jacobian matrix. The INHB analysis requires 22 MB of memory and 74 s, showing that its performance is comparable to that of the sparse-Jacobian HB at those low drive levels ($P_{\text{in}} < -3$ dBm for the present case) for which the sparse-matrix analysis is possible. The numerical results produced by the three analysis techniques are strictly identical.
2) The same IM test as in 1) is repeated with $P_{\text{in}} = +20$ dBm, a power level well within the power saturation region of the amplifier. At this drive level the conventional HB analysis requires 32 674 s, while the INHB analysis time is only 203 s, with a speedup factor of about 161. Once again, the numerical results produced by the two analysis techniques are strictly identical. Note that the CPU time ratio between 2) and 1) is 2.74 with the INHB, and 7.75 with the full-Jacobian HB. This shows that the INHB has considerably better power-handling capabilities than the conventional HB.
3) With the same input power level as in 2), ($P_{\text{in}} = +20$ dBm per tone) the IM analysis is repeated taking into account all IM products up to the ninth order. The number of positive IM products is now $P = 579$, and the number of unknowns is $N = 20\,862$. In this case the storage of the Jacobian matrix alone would require about 3.5 GB, so that the conventional HB analysis becomes impossible. On the contrary, the INHB analysis requires only 68 MB of memory, and converges smoothly in 1982 s. A comparison with 2) shows that the ratio between the numbers of unknowns for the two jobs is about 5, while the CPU time ratio is only 9.76. For a regular HB analysis dominated by the Jacobian matrix factorization time, the CPU time ratio would be of the order of $5^3 = 125$. This gives a clear example of the slow dependence of the INHB computational time on the number of unknowns.

Other examples of application of the INHB technique to large nonlinear simulation problems were discussed in [9]. With respect to the numerical performance reported in [9], the improved algorithms discussed in this letter provided a speed increase of about 10%, and a reduction in the overall memory storage of more than 25%.

References

[1] V. Rizzoli *et al.*, "State-of-the-art harmonic-balance simulation of forced nonlinear microwave circuits by the piecewise technique," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 12–28, Jan. 1992.
[2] K. S. Kundert and A. Sangiovanni-Vincentelli, "Simulation of nonlinear circuits in the frequency domain," *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 521–535, Oct. 1986.
[3] R. S. Dembo, S. C. Eisenstat, and T. Steinhaug, "Inexact Newton methods," *SIAM J. Numerical Analysis*, vol. 19, pp. 400–408, Apr. 1982.
[4] S. C. Eisenstat and H. F. Walker, "Choosing the forcing terms in an inexact Newton method," *SIAM J. Scientific Computing*, vol. 17, pp. 16–32, Jan. 1996.
[5] S. C. Eisenstat and H. F. Walker, "Globally convergent inexact Newton methods," *SIAM J. Optimization*, vol. 4, pp. 393–422, May 1994.
[6] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual method for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 856–869, July 1986.
[7] W. D. McQuain *et al.*, "Preconditioned iterative methods for sparse linear algebra problems arising in circuit simulation," *Computer Math. Applic.*, vol. 27, pp. 25–45, Aug. 1994.
[8] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. Berlin, Germany: Springer-Verlag, 1981.
[9] V. Rizzoli, F. Mastri, F. Sgallari, and G. Spaletta, "Harmonic-balance simulation of strongly nonlinear very large-size microwave circuits by inexact Newton methods," in *1996 IEEE MTT-S Int. Microwave Symp. Dig.*, San Francisco, CA, June 1996, pp. 1357–1360.