SIAM J. SCI. COMPUT.
Vol. 17, No. 2, pp. 538-559, March 1996


014



# AN ANALYSIS OF APPROXIMATE NONLINEAR ELIMINATION*

PAUL J. LANZKRON†, DONALD J. ROSE‡, AND JAMES T. WILKES§

**Abstract.** We present a method for solving systems of nonlinear equations suitable for problems where convergence of an approximate Newton method is initially slow. The method, nonlinear elimination (NlEm), eliminates the nonlinear equations and appropriate variables deemed to be causing the problem. An analysis of the method is given and leads to a detailed algorithm that reduces automatically to an approximate Newton method near the root of the system of nonlinear equations. Several examples are given that demonstrate the efficacy of the method.

**Key words.** nonlinear equations, global methods, approximate Newton method

**AMS subject classifications.** 65H10, 65H20

## 1. Introduction.

We consider the numerical solution of a system of nonlinear equations

$$g(w) = 0, \tag{1.1}$$

where $g = (g_1, g_2, \ldots, g_n)^T$ and $w = (w_1, w_2, \ldots, w_n)^T$. Generically we consider the methodology of [1] and the analytic framework presented there. This setting ensures that for any $w_0$ in some set $S$, a sequence of iterates $w_k$ will converge to $w^* \in S$ and $g(w^*) = 0$. The iterates are defined as

$$w_{k+1} = w_k + t_k x_k, \tag{1.2}$$

where $x_k$ *approximates* $z_k$ in

$$g_k' z_k = -g_k, \tag{1.3}$$

$g_k' \equiv g'(w_k)$, and $g_k \equiv g(w_k)$, and $t_k \in (0, 1]$ is chosen to force

$$\|g_{k+1}\| < \theta \|g_k\| \tag{1.4}$$

for some $0 < \theta < 1$. The sense in which $x_k$ approximates $z_k$ of (1.3) is measured by

$$\alpha_k = \|g_k' x_k + g_k\| / \|g_k\| \tag{1.5}$$

and all $\alpha_k \leq \alpha_0 < 1$ suffices for convergence (for some sequence of $t_k$). We call this algorithmic approach GAN, for global approximate Newton, "global" referring to the set S which can be $\mathbb{R}^n$ under appropriate conditions.

Such variations of Newton's method have been used successfully in significant technology areas including circuit and device simulation [2, 5]. The trick in any of these applications is in picking an "inner" solver for (1.3) to make $\alpha_k \to 0$ and $t_k \to 1$ to ensure (1.4) and superlinear convergence. This can be particularly challenging in practice since we often cannot be sure a priori that the sufficient convergence conditions are satisfied, nor can we wait for $k \to \infty$.

The use of nonlinear elimination (NlEm, en-lem) is motivated by problems in which convergence seems to be interminably tedious and yet there is no evidence of ill conditioning (or singularity). This leads us to believe that the nonlinearities in $g$ are unbalanced, by which

*Received by the editors July 6, 1995; accepted for publication (in revised form) October 22, 1994.

†Raytheon Company, 1001 Boston Post Road, Marlborough, MA 01752 (pjl@tif396.ed.ray.com). The work of this author was supported in part by the National Science Foundation under grant NSF-CCR-92-09444.

‡Department of Computer Science, Duke University, Durham, NC 27708 (djr@cs.duke.edu). The work of this author was supported in part by the Office of Naval Research under grant N00014-89-J-1644.

§Department of Mathematical Sciences, Appalachian State University, Boone, NC 28608 (jtw@cs.appstate.edu).

we mean some "subfunction" $g_1(u, v)$ regarded as a function of $u$ given $v$ causes $t_k$ to be small. We propose to eliminate $g_1$ as an inner iteration.

Consider $g(w)$ to be partitioned as

$$g(w) = (g_1(u, v), g_2(u, v))^T,$$
$$w = (u, v)^T. \tag{1.6}$$

This leads to a block Jacobian

$$g'(u, v) = \begin{bmatrix} g_{11}(u, v) & g_{12}(u, v) \\ g_{21}(u, v) & g_{22}(u, v) \end{bmatrix}, \tag{1.7}$$

where $g_{ij}(u, v) = \frac{\partial g_i}{\partial g_j}(u, v)$. For conciseness we write $g_{ij} \equiv g_{ij}(u, v)$.

For smooth $g_1(u, v)$ the solvability of $g_1(u, v) = 0$ for $v$ in an appropriate set leads to an implicit function $h(v)$ such that

$$g_1(h(v), v) = 0. \tag{1.8}$$

Differentiating (1.8) as a function of $v$ leads to

$$g_{11}h'(v) + g_{12} = 0, \tag{1.9}$$

and if $g_{11}(h(v), v)$ is nonsingular, then

$$h'(v) = -g_{11}^{-1}g_{12}. \tag{1.10}$$

Indeed the operational equation (1.10) and the existence of the unique mapping $h(v)$ constitute the essence of the implicit function theorem, see [12, Thm. 5.2.4], which requires the nonsingularity of $g_{11}$.

Assuming that $u = h(v)$ exists on an appropriate set, we attempt to solve equation (1.1), which can be rewritten as

$$f(v) = 0, \tag{1.11}$$

where $f(v) \equiv g_2(h(v), v)$ for $v$ using GAN. To use GAN we must compute the Jacobian $f'(v)$; differentiating (1.11) yields

$$f'(v) = g_{21}h'(v) + g_{22}, \tag{1.12}$$

where $g_{21}$, $g_{22}$ are evaluated at $(h(v), v)$. Making the substitution (1.10), we obtain

$$f'(v) = g_{22} - g_{21}g_{11}^{-1}g_{12} \tag{1.13}$$

and we recognize the right-hand side of (1.13) as a Schur complement. More precisely we see that the Newton direction equation associated with (1.11), namely,

$$f'(v)\Delta v = -f(v), \tag{1.14}$$

can be embedded into the larger matrix equation

$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} -g_1 \\ -g_2 \end{bmatrix}, \tag{1.15}$$

where all functions are evaluated at $(h(v), v)$ and $g_1(h(v), v) = 0$. Equation (1.14) arises from block Gaussian elimination on (1.15) as indicated by (1.13). This is computationally

attractive since nonlinear elimination leads to the same algebra as GAN on the whole system, but now applied at the point $(h(v), v)$. To summarize there are two nonlinear solve processes: an "inner" equation solve to evaluate $h(v)$ from (1.8) and an "outer" Newton iteration to solve (1.11) via (1.15). When $h(v)$ is evaluated "exactly," i.e., $g_1(h(v), v) = 0$, then we show (Theorem 4.2) that the outer GAN (1.15) converges quadratically to the solution $g_2(h(v), v) = 0$ as expected.

Note that when a sparse matrix package can be used for GAN, it is also applicable for NlEm. That is, starting at the point $(h(v), v)$, both NlEm and GAN would need to solve the system (1.15). It is unimportant that the variables be ordered so that the $\Delta u$ variables are first. In fact it is easy to imagine cases where such an ordering would lead to significant fill. After the linear system is solved the $\Delta v$ variables can be gathered easily.

Many investigators have studied nonlinear elimination for particular applications. We discuss two of these applications here. The first is macromodeling circuits. A circuit can be thought of as a $k$-terminal device. In modeling the circuit the user wishes to know how changing the voltage at some of the input terminals affects the voltage at other terminals. The number of unknowns for the system is then $k$ plus the number of internal unknowns in the device. In [6] the authors show how to eliminate the internal unknowns from the system of nonlinear equations that must be solved at each time step. More recent work on applying NlEm to circuit simulation is reported in [13].

A second widely used, although only tangentially connected, application of nonlinear elimination is in nonlinear least squares. Consider the problem of finding $(u, v)$ such that

$$\|y - g(u, v)\|_2 \tag{1.16}$$

is minimized, where $y \in \mathbb{R}^{n\times 1}$ is a set of observations and $g : \mathbb{R}^p \to \mathbb{R}^n$ is nonlinear. Consider the case where

$$g(u, v) \equiv l(v)u = \begin{bmatrix} g_{11}(v) \\ g_{21}(v) \end{bmatrix} u. \tag{1.17}$$

When $v$ is fixed in equation (1.17), equation (1.16) turns into a linear least squares problem. The problem given in (1.16) may then be rewritten as

$$\min_v \|y - g(h(v), v)\|_2, \tag{1.18}$$

where

$$h(v) = \left(l(v)^T l(v)\right)^{-1} l(v)^T y. \tag{1.19}$$

We have assumed, as in [10], that $l(v)$ has full column rank. The case when $l(v)$ does not have full column rank is handled in [7, 8]. The difference between this approach and our approach is that we eliminate variables *and* equations so that $f(v)$ is still a system of $p$ nonlinear equations in $p$ unknowns.

Nonlinear elimination arises naturally in constrained optimization and can be viewed in the context of Lagrange multipliers. Most authors do not choose such an exposition; see [3, p. 141] for a related discussion. We remark also that Brown's method [4] can be regarded as a specialization of nonlinear elimination as discussed in [12, NR 7.14–15, pp. 227–229].

This paper is organized as follows. In §2 the NlEm algorithm is presented along with a simple example. Sections 3–5 contain an analysis of the whole procedure, presented as constructively as possible. Since we suggest that NlEm can be part of a general algorithmic strategy for solving nonlinear systems, this analysis presents conditions that ensure convergence in a way that is compatible with the theory presented in [1]. Section 4 discusses the

Input: $w_0 = (u_0, v_0)$ and $g$.

1. Choose $g_1$ equations splitting $g(w)$ into $g_1(u, v)$ and $g_2(u, v)$.
   By $g_1(u, v)$ we mean that if $g_1 : \mathbb{R}^n \to \mathbb{R}^j$, then $u \in \mathbb{R}^j$.
2. Given the initial $v_0$, solve $g_1(h(v_0), v_0) = 0$ for $h(v_0)$.
3. $k \leftarrow 0$
4. repeat until convergence
5. Solve
$$\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta v_k \end{bmatrix} = \begin{bmatrix} -g_1 \\ -g_2 \end{bmatrix},$$
with $g_{11}, g_{12}, g_{21}, g_{22}$, and $g_2$ evaluated at $(h(v_k), v_k)$.
6. Find $t_k \in (0, 1]$ such that
$$\|g(h(v_k + t_k \Delta v_k), v_k + t_k \Delta v_k)\| < \theta \|g(h(v_k), v_k)\|$$
Note: a solve for $u = h(v_k + t_k \Delta v_k)$ of
$g_1(u, v_k + t_k \Delta v_k) = 0$
is required for every $t_k$.
7. $v_{k+1} = v_k + t_k \Delta v_k$
Note: $h(v_{k+1})$ is implicitly determined in step 6.

FIG. 1. *Outline of the NlEm algorithm.*

case when (1.8) is solved exactly and §5 discusses how accurately (1.8) must be solved to retain the higher-order convergence of Newton's method. The analysis in §5 leads to a refined algorithm in §6 that unifies NlEm and GAN into a single methodology. Section 7 contains several computed examples.

**2. Implementation overview.** In this section we give a brief overview of how to implement the NlEm algorithm along with an illustrative example. In §6, a detailed algorithm is presented that unifies GAN and NlEm.

**2.1. NlEm algorithm.** The outline of the algorithm is given in Figure 1.

Often the hardest part of the implementation is in step 1, since determining the set of equations that belong in $g_1$ is usually problem dependent. Once the set is chosen the solve in step 2 moves the initial point $(u_0, v_0)$ to a new point in the domain $(h(v_0), v_0)$. There may be an increase in the overall norm of $g$ at this new point; i.e.,

$$\|g(h(v_0), v_0)\| > \|g(u_0, v_0)\|. \tag{2.1}$$

However, the norm of $g$ decreases monotonically thereafter under appropriate conditions. There are times when the set of $g_1$ equations may be changed dynamically after the algorithm has been started. However, arbitrarily changing the set of equations represented by $g_1$ could lead to thrashing (i.e., the norm of $g$ goes down for a while, but the $g_1$ equations are changed causing $\|g\|$ to be as large or larger than previously).

The computation in step 6 implicitly requires a solve of the $g_1$ equation. That is, to compute $h(v_k + t_k \Delta v_k)$ the $g_1$ equation must be solved. Step 6 is potentially expensive, since every new choice of $t_k$ requires a solve of the $g_1$ equations. Note that in step 6 we are trying to make the overall norm of $g$ go down. This is necessary since $g_1 \neq 0$ in practice. When $g_1 \neq 0$ we only have an approximation to $h(v)$, say $\tilde{h}(v)$. Thus, $g_2(h(v), v)$ is approximated

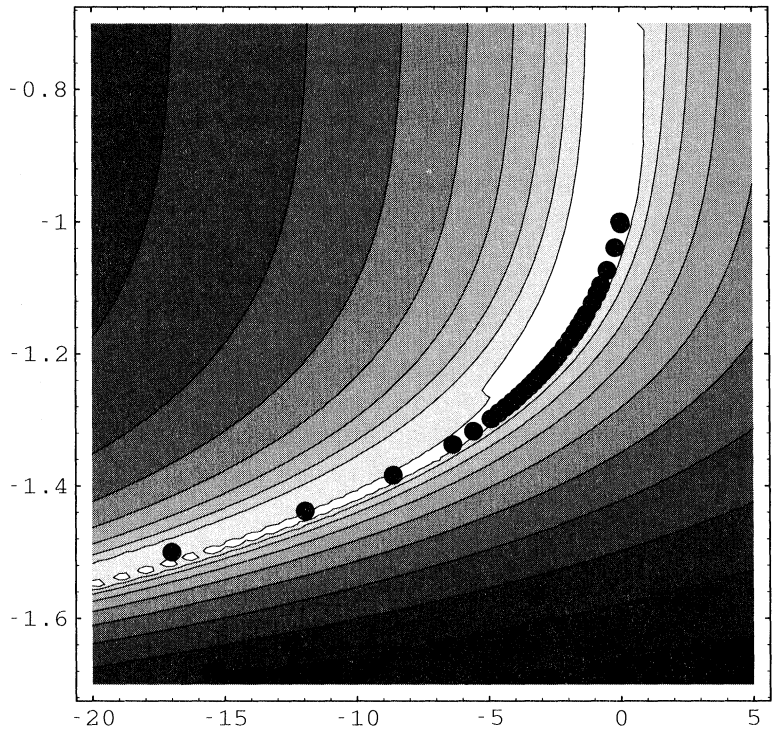


FIG. 2. *Convergence of GAN overlaid on* $\log(\|f\| + 1)$.

by $g_2(\tilde{h}(v), v)$. A decrease in $\|g_2(\tilde{h}(v), v)\|$ does not imply that $\|g_2(h(v), v)\|$ has decreased; see §5. Therefore, we enforce norm reduction on all of $g$.

**2.2. A simple illustrative example.** We believe NlEm will be helpful when the system of equations has some badly scaled components. The poor scaling is not in terms of normal linear misscaling, rather it is due to the nonlinearity of the problem. In this section, we present a simple example demonstrating how NlEm helps.

Consider the system $f$ of two equations given by

$$f_1(u, v) = (u - v^7)^3 + v^3,$$
$$f_2(u, v) = (u + v)^{\frac{1}{5}} + 1. \tag{2.2}$$

When viewed in terms of norms the curve $u = v^7 - v$ is close to the bottom of a very steep valley. A GAN iterate in the valley causes the Newton direction to point tangent to the valley. Since the valley curves and the Newton direction points in a direction nearly tangent to the bottom of the valley, only small values of the damping parameter can be chosen; larger values give an increase in $\|g\|$. In Figure 2 the starting point $(-17.085938, -1.5)$ leads the GAN iterates into the steep curved valley. GAN takes 32 steps to converge to the solution $(0, -1)$.

To use NlEm we must first decide on the equation to eliminate. For this problem it is clear that the first equation is causing the difficulties; that is, movement away from the curve $u = v^7$ causes a large increase in the norm of the system. This is not true for the second equation. Having decided on the equation to eliminate, we must then decide on the variable to be eliminated with it; in this case we choose to eliminate the $u$ variable. In realistic problems the variable eliminated will be more obvious. For a given $v$ we will solve for a $u$ such that

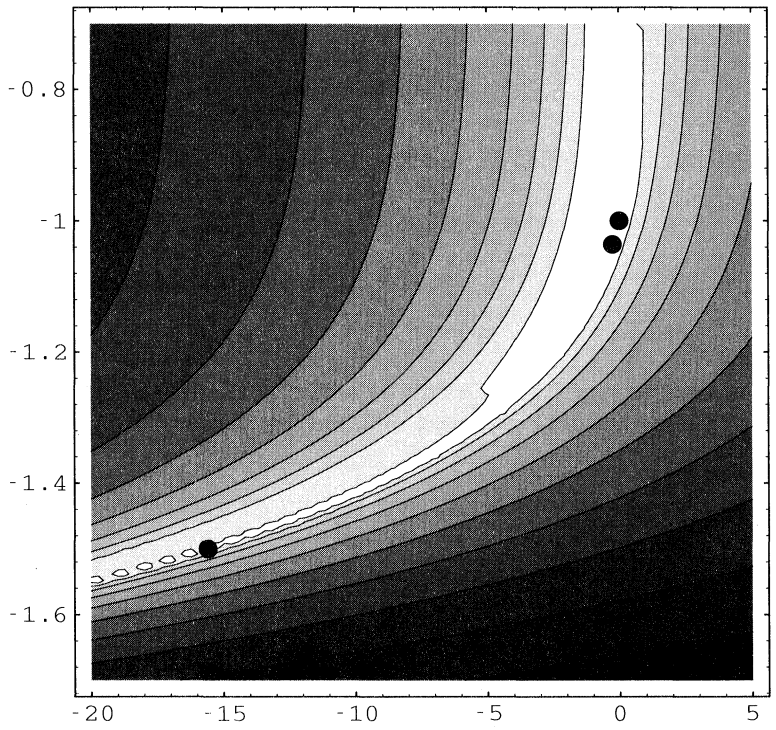


FIG. 3. *Convergence of NlEm overlaid on* $\log(\|f\| + 1)$.

$g_1(u, v) = 0$. Solving for $u$ in the first equation means that $u = h(v) = v^7 - v$. The eliminated system, $g_2(h(v), v) = v^{7/5} + 1$, is a much simpler problem to solve. In Figure 3 we see that the number of NlEm iterations from the same starting point drops to three.

**3. Preliminaries.** Recall that our objective is to solve (1.11) using Newton's method. This section gives a review of the GAN convergence theory and provides results used in later sections.

**3.1. Analysis of GAN.** The following analysis of GAN is derived from [1]. Let $g_k \equiv g(w_k)$ with $w_0$ some initial guess for the solution. There are three basic assumptions used to prove convergence of GAN.

*Assumption* A1. The closed level set

$$S_0 = \{w \mid \|g(w)\| \leq \|g_0\|\} \tag{3.1}$$

is bounded.

The following assumption is equivalent to Assumption A2 presented in [1].

*Assumption* A2. $g$ is differentiable, the Jacobian $g'(w)$ is continuous and nonsingular on $S_0$, and the sequence $\|x_k\|$ is uniformly bounded, i.e.,

$$\|x_k\| \leq k_1 \|g_k\| \tag{3.2}$$

for $k_1 \geq 0$ and $w_k \in S_0$.

*Assumption* A3. The Jacobian $g'$ is Lipschitz; i.e.,

$$\|g'(y) - g'(z)\| \leq k_2 \|y - z\| \tag{3.3}$$

for

$$y, z \in S_1 = \{y \mid \|y\| \leq \sup_{z \in S_0} \|z\| + k_1 \|g_0\|\}. \tag{3.4}$$

To continue the analysis of GAN we define the quantities

$$\begin{aligned} A_k &= (g_k + g_k' x_k)/\|g_k\|, \\ B_k &= \left\{ \int_0^1 \left[ g'(w_k + s t_k x_k) - g_k' \right] t_k x_k ds \right\} / \left( t_k \|g_k\| \right)^2, \\ \alpha_k &= \|A_k\|, \qquad \beta_k = \|B_k\|, \end{aligned} \tag{3.5}$$

where we have assumed $g_k \neq 0$. The quantity $\alpha_k$ defines the relative size of the linear residual.

Following the analysis of [1], the mean value theorem yields

$$g_{k+1} = g_k + \int_0^1 g'(w_k + s t_k x_k)(w_{k+1} - w_k) ds. \tag{3.6}$$

Adding and subtracting $g_k'(w_{k+1} - w_k)$ on the right-hand side yields

$$g_{k+1} = g_k + g_k'(w_{k+1} - w_k) + \int_0^1 \left[ g'(w_k + s t_k x_k) - g_k' \right] (w_{k+1} - w_k) ds. \tag{3.7}$$

Recalling that $w_{k+1} = w_k + t_k x_k$ we have

$$g_{k+1} = (1 - t_k) g_k + t_k A_k \|g_k\| + t_k^2 B_k \|g_k\|^2. \tag{3.8}$$

And finally taking norms we are left with

$$\|g_{k+1}\| \leq \|g_k\| \left( (1 - t_k) + t_k \alpha_k + t_k^2 \beta_k \|g_k\| \right). \tag{3.9}$$

Under Assumptions A1–A3, $\beta_k$ can be shown to be bounded, $\beta_k \leq k_1^2 k_2 / 2$. It is usually possible to control the size of $\alpha_k$ by computing $x_k$ more accurately. This is the case when an iterative method is being used to solve the Newton equations. It is clear from (3.9) that if $\alpha_k < 1$ then there exists a $t_k$ such that $\|g_{k+1}\| < \|g_k\|$.

Convergence of GAN is given by the following proposition and theorem.

PROPOSITION 3.1 (see [1]). *Assume that* A1–A3 *hold and that all* $\alpha_k \leq \alpha_0$. *If* $t_k$ *is chosen appropriately, then*

1. *all* $w_k \in S_0$, *the sequence* $\|g_k\|$ *is strictly decreasing, and* $\|g_k\| \to 0$; *furthermore,*
2. $\|g_{k+1}\| / \|g_k\| \to 0$ *iff* $\alpha_k \to 0$ *and for any fixed* $p \in (0, 1]$,

$$\|g_{k+1}\| \leq c_1 \|g_k\|^{1+p}, \tag{3.10}$$

*iff*

$$\alpha_k \leq c_2 \|g_k\|^p \tag{3.11}$$

*for positive constants* $c_1$ *and* $c_2$.

THEOREM 3.2 (see [1]). *Under the conditions of Proposition* 3.1,

1. *there exists a* $w^* \in S_0$ *with* $w^* = \lim w_k$ *and* $g(w^*) = 0$;
2. *on* $S_0$ *the convergence of* $\{w_k\}$ *to* $w^*$ *is superlinear or Q-order* $(p + 1)$ *if* $\alpha_k \to 0$ *or* $\alpha_k \leq c_2 \|g_k\|^p$, *respectively.*

**3.2. Additional results.** We conclude this section with results that will be used in subsequent sections.

PROPOSITION 3.3. *Assume that GAN starting from $w_0$ is convergent to $w^*$ with $g(w^*) = 0$. Let $\|g'^{-1}(w)\| \le M$, for $u \in S_0$, and $\|g_{i+1}\| \le \theta \|g_i\|$ for every $i$, where $g_i \equiv g(w_i)$ and $\theta < 1$. Then*

$$\|w_0 - w^*\| \le \frac{1}{1-\theta} M \|g(w_0)\|. \tag{3.12}$$

*Proof.* Since GAN is assumed to be convergent, we know that $w_i \in S_0$ for all $i$. Thus.

$$\begin{aligned}\|w_0 - w^*\| &= \left\| w_0 - \left( w_0 + \sum_0^\infty t_i \Delta w_i \right) \right\| = \left\| \sum_0^\infty t_i \Delta w_i \right\| \\ &\le \sum_0^\infty \|\Delta w_i\| = \sum_0^\infty \| - g_i'^{-1} g_i \| \\ &\le M \sum_0^\infty \|g_i\| \le M \|g_0\| \sum_0^\infty \theta^i = \frac{1}{1-\theta} M \|g(w_0)\|. \end{aligned}$$

□

The proofs in this paper are complicated somewhat by the changes in sizes of norms. We will address this problem by restricting the matrix norms we consider to $p$-norms. These norms have the following properties.

*Property* P1. Let $u \in \mathbb{R}^m$, and $v = [0, 0, \ldots, 0, u]^T \in \mathbb{R}^n$; then

$$\|u\|_p = \|v\|_p. \tag{3.13}$$

For example,

$$\|u\|_p = \left( \sum_{i=1}^m w_i^p \right)^{\frac{1}{p}} = \left( \sum_{i=1}^m w_i^p + \sum_{i=m+1}^n 0^p \right)^{\frac{1}{p}} = \|v\|_p. \tag{3.14}$$

Note that (3.13) holds however the elements of $u$ are distributed in $v$.

*Property* P2. Let $u \in \mathbb{R}^m$, $v = [u, 0, 0, \ldots, 0]^T \in \mathbb{R}^n$ and $w = [u, y]^T \in \mathbb{R}^n$; then

$$\|v\|_p \le \|w\|_p. \tag{3.15}$$

The following lemma can easily be verified to show that Properties P1 and P2 carry through to matrices.

LEMMA 3.4. *Let $A \in \mathbb{R}^{j \times k}$, $j, k \le n$; then*

$$\|A\|_p = \left\| \begin{bmatrix} 0 & 0 \\ 0 & A \end{bmatrix} \right\|_p \le \left\| \begin{bmatrix} B & C \\ D & A \end{bmatrix} \right\|,$$

*where* $\begin{bmatrix} B & C \\ D & A \end{bmatrix} \in \mathbb{R}^{n \times n}$.

We will refer to (3.13) as Property P1, and (3.15) as Property P2 without distinguishing between matrices and vectors.

LEMMA 3.5. *Let the nonsingular $A \in \mathbb{R}^{n \times n}$ be partitioned as*

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

*with $A_{11} \in \mathbb{R}^{j\times j}$ and nonsingular. If $\|A^{-1}\| < k$ and properties* P1 *and* P2 *hold for $\|\cdot\|$, then*

$$\|(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}\| < k. \tag{3.16}$$

*Proof.*

$$\begin{aligned}\|(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}\| &= \left\| \begin{bmatrix} 0 & 0 \\ 0 & (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \end{bmatrix} \right\| \\ &\le \|A^{-1}\| \\ &< k. \qquad \square \end{aligned} \tag{3.17}$$

PROPOSITION 3.6. *If $g$ is Lipschitz on a bounded set $S$, then $\|g\|$ is bounded.*

*Proof.* Since $S$ is bounded, there exists $k \in \mathbb{R}^+$ such that $\|w - y\| \le k$ for all $w, y \in S$. Let $w_0$ be a particular member of $S$ with $\|g(w_0)\| = C$. Then, for any $w \in S$, we have

$$\|g(w)\| \le \|g(w) - g(w_0)\| + \|g(w_0)\| = L\|w - w_0\| + C = Lk + C, \tag{3.18}$$

where $L$ is the Lipschitz constant. $\square$

PROPOSITION 3.7. *Let $M(w) \in \mathbb{R}^{n\times n}$ with $M(w)$ Lipschitz on some convex set with Lipschitz constant $k_0$. If $M(w)$ is partitioned as*

$$\begin{bmatrix} M_{11}(w) & M_{12}(w) \\ M_{21}(w) & M_{22}(w) \end{bmatrix} \tag{3.19}$$

*with $M_{11}(w) \in \mathbb{R}^{j\times j}$ and $\|M_{11}^{-1}(w)\| \le k_1$, then the Schur complement of $M(w)$ is also Lipschitz with Lipschitz constant*

$$k_0(1 + k_1k_2)^2, \tag{3.20}$$

*where $\|M(w)\| \le k_2$.*

*Proof.* Applying Proposition 3.6 gives us $\|M(w)\| \le k_2$. Using Properties P1 and P2, we know that $\|M_{ij}\| \le k_2$ for $i, j = 1, 2$. Let $A \equiv M(w_1)$ and $B \equiv M(w_2)$.

$$\begin{aligned} &\| \left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right) - \left(B_{22} - B_{21}B_{11}^{-1}B_{12}\right) \| \\ &\le \|A_{22} - B_{22}\| \\ &\quad + \|B_{21}B_{11}^{-1}B_{12} - B_{21}B_{11}^{-1}A_{12} + B_{21}B_{11}^{-1}A_{12} - A_{21}A_{11}^{-1}A_{12}\| \\ &\le k_0\|x - y\| + \|B_{21}B_{11}^{-1}\left(B_{12} - A_{12}\right)\| + \| \left(B_{21}B_{11}^{-1} - A_{21}A_{11}^{-1}\right) A_{12}\| \\ &\le k_0\|x - y\| + k_2k_1k_0\|x - y\| \\ &\quad + \| \left(B_{21}B_{11}^{-1} - A_{21}B_{11}^{-1} + A_{21}B_{11}^{-1} - A_{21}A_{11}^{-1}\right) A_{12}\| \\ &\le k_0\|x - y\| + k_2k_1k_0\|x - y\| \\ &\quad + k_2\| \left(B_{21} - A_{21}\right) B_{11}^{-1}\| + k_2\|A_{21}\left(B_{11}^{-1} - A_{11}^{-1}\right)\| \\ &\le k_0\|x - y\| + k_2k_1k_0\|x - y\| + k_2k_1k_0\|x - y\| \\ &\quad + k_2^2\|B_{11}^{-1}\|\|A_{11}^{-1}\|\| \left(B_{11} - A_{11}\right)\| \\ &\le k_0\|x - y\| + 2k_2k_1k_0\|x - y\| + k_2^2k_1^2k_0\|x - y\| \\ &= k_0(1 + k_1k_2)^2\|x - y\|. \qquad \square \end{aligned} \tag{3.21}$$

**4. Exact NlEm.** The exact NlEm algorithm is Newton's method applied to the equation

$$f(v) \equiv g_2(h(v), v) = 0. \tag{4.1}$$

To show that the algorithm converges, we need to show that, under appropriate conditions, $h(v)$ exists and that the basic Assumptions A1–A3 hold on the function $f(v)$. These assumptions and making $\alpha_k \to 0$ ensure that Proposition 3.1 and Theorem 3.2 hold.

First, assume that GAN is convergent on $g$ and that from any point $(u, v)$ in a given set, GAN converges on $g_1$ giving $g_1(\hat{u}, v) = 0$ where $\hat{u} = h(v)$. For clarity sake, let $S_0(g)$ be the set $S_0$ given in (3.1) and let $S_1(g)$ be the set $S_1$ given in (3.4). Because of the partitioning of $g$, projections of sets are needed. In particular, let

$$S_1^2(g) = \{v \mid (u, v) \in S_1(g) \quad v \in \mathbb{R}^{n-j}\}. \tag{4.2}$$

The following assumptions are required for proof of the convergence of GAN applied to (4.1).

Assumption A1 remains unchanged and for Assumption A2 we make the stronger restriction that $\|g'^{-1}\| \leq k_1$ over $S_0(g)$. Assumption A3 is modified below.

*Assumption* A4. Let

$$\|\hat{g}_1\| = \max_{w \in S_1(g)} \|g_1(w)\|; \tag{4.3}$$

then the set

$$S_0(g_1) = \{(u, v) \mid v \in S_1^2(g) \quad \text{and} \quad \|g_1(u, v)\| \leq \|\hat{g}_1\|\} \tag{4.4}$$

is bounded.

*Assumption* A5. The function $g_1$ is differentiable and $g_{11}$ is continuous and nonsingular on $S_0(g_1)$, and

$$\|g_{11}^{-1}(w)\| \leq k_5, \qquad w \in S_0(g_1). \tag{4.5}$$

Since convergence of the $g_1$ system is required only on the projection of $S_0(g_1)$, i.e., $g_1(u, v) = 0$ is solved with $v$ fixed; the projection sets of $S_0(g_1)$ are defined as

$$S_0^1(g_1) = \{u \mid (u, v) \in S_0(g_1), \qquad u \in \mathbb{R}^j\}, \tag{4.6}$$

$$S_0^2(g_1) = \{v \mid (u, v) \in S_0(g_1), \qquad v \in \mathbb{R}^{n-j}\}. \tag{4.7}$$

This leads to a modification of Assumption A3 given previously.

*Assumption* A3. The Jacobian $g'$ is Lipschitz with Lipschitz constant $k_2$ on

$$S_1(g_1) = \{(u, v) \mid v \in S_0^2(g_1) \quad \text{and} \quad \|u\| \leq \max_{w \in S_0^1(g_1)} \|w\| + k_5\|\hat{g}_1\|\}, \tag{4.8}$$

where $\|\hat{g}_1\|$ is defined as before.

The sets $S_0(g)$, $S_1(g)$, $S_0(g_1)$, and $S_1(g_1)$ are illustrated in two dimensions in Figure 4; the $u$-axis is horizontal and the $v$-axis is vertical. Note that $S_0(g)$ need not be connected. Under Assumptions A1–A5, there is a root of $g$ in both of the disconnected regions. The sets $S_1(g)$ and $S_1(g_1)$ are both convex, but the set $S_1(g_1)$ has a capsule shape because it is stretched in the direction of the eliminated variable.

These sets come into play during various phases of the solution algorithm. Initially $w_k$ is in $S_0(g)$. The approximate Newton direction is computed and a potential $w_{k+1}$ produced. The potential $w_{k+1} = w_k + t_k x_k$ may be outside $S_0(g)$, but is within $S_1(g)$ (of course, when the final $w_{k+1}$ is determined it must be in $S_0(g)$). At this point in the algorithm, a root $u$ is sought for $g_1(u, v)$ from a starting point in $S_1(g)$. This root will be somewhere in $S_0(g_1)$. The Newton method used to find that root may generate points in $S_1(g_1)$ (in the same way as $w_k + t_k x_k$ may be outside of $S_0(g)$).

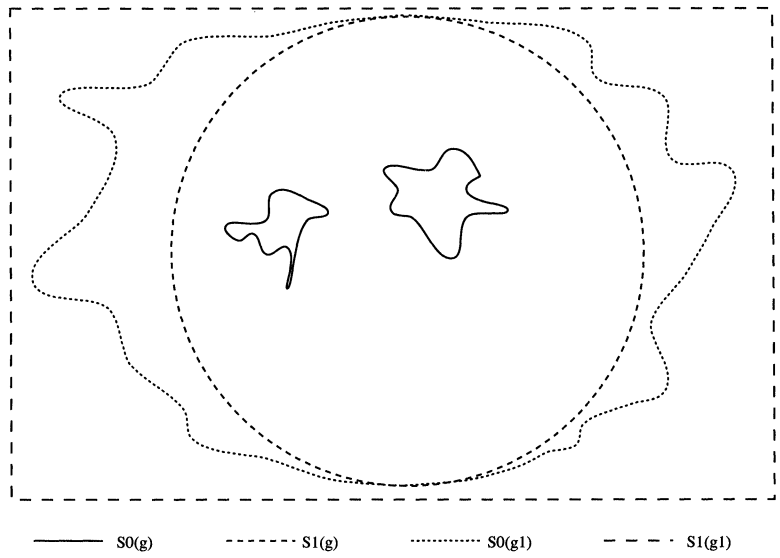


FIG. 4. *Illustration of the sets* $S_0(g)$, $S_1(g)$, $S_0(g_1)$, *and* $S_1(g_1)$ *in two dimensions.*

LEMMA 4.1. *Consider Assumptions* A3–A5. *For* $v \in S_1^2(g)$, *there is a* $u \in S_0^1(g_1)$ *such that*

$$g_1(u, v) = 0. \tag{4.9}$$

*Thus,* $h(v)$ *exists for* $v \in S_1^2(g)$ *and*

$$h'(v) = -g_{11}^{-1}(h(v), v)g_{12}(h(v), v). \tag{4.10}$$

*Proof.* Let

$$\|g'(w_1) - g'(w_2)\| \leq \hat{k}_2\|w_1 - w_2\| \tag{4.11}$$

hold for $\|\cdot\|$ satisfying Properties P1 and P2. Let $w_1 = (u_1, v)$ and $w_2 = (u_2, v)$. It follows that

$$\|g_{11}(w_1) - g_{11}(w_2)\| \leq k_2\|u_1 - u_2\|. \tag{4.12}$$

Assumptions A4 and A5 and (4.12) represent the conditions for Proposition 3.1 and Theorem 3.2 to hold. Thus $g_1(u, v)$ has a root for fixed $v \in S_0^1(g_1)$. The remainder of the lemma follows from the implicit function theorem. □

THEOREM 4.2. *Consider Assumptions* A1–A5. *If* $(u_0, v_0) \in S_0(g)$ *with* $g_1(u_0, v_0) = 0$, *and at every iteration* (1.15) *is solved exactly, then GAN converges to the solution of* $f(v) = 0$ *and, furthermore, convergence is quadratic* (*Q-order* 2).

*Proof.* The proof will show that Assumptions A1–A3 hold for $f(v)$ and that $\alpha_k = 0$ ($\alpha_k$ as defined in (3.5)).

(A1) Let $z \in S_0(f) = \{v | \|f(v)\| \leq \|f(v_0)\|\}$, then by Property P1,

$$\|g(h(z), z)\| = \|f(z)\| \leq \|f(v_0)\| = \|g(h(v_0), v_0)\|. \tag{4.13}$$

Thus, $(h(z), z) \in S_0(g)$ and by Property P2, $S_0(f)$ is bounded.

(A2) Since $(h(z), z) \in S_0(g)$ for $z \in S_0(f)$, we know that

$$\|\Delta v_k\| \leq \left\| \begin{bmatrix} \Delta u_k \\ \Delta v_k \end{bmatrix} \right\| = \|x_k\| \leq k_1 \|g(h(v_k), v_k)\| = k_1 \|f(v_k)\|, \tag{4.14}$$

using Properties P1 and P2 where appropriate.

Define

$$S_1(f) \equiv \{v \mid \|v\| \leq \sup_{w \in S_0(f)} \|w\| + k_1 \|f(v_0)\|\}. \tag{4.15}$$

(A3) From Assumption A5 $g_{11}^{-1}(h(v), v)$ exists and is bounded. Note that $S_1(f) \subset S_0^2(g_1)$ and $g_1(h(v), v) = 0$ for $v \in S_0^2(g_1)$; thus

$$(h(v), v) \in S_0(g_1) \subset S_1(g_1). \tag{4.16}$$

These observations and Proposition 3.7 gives

$$\|f'(v_1) - f'(v_2)\| \leq k_2(1 + k_5 k_3)^2 \|(h(v_1), v_1)^T - (h(v_2), v_2)^T\| \tag{4.17}$$

for $v_1, v_2 \in S_1(f)$.

By the triangle inequality and Property P1 it follows that

$$\left\| \begin{bmatrix} h(v_1) - h(v_2) \\ v_1 - v_2 \end{bmatrix} \right\| \leq \|v_1 - v_2\| + \|h(v_1) - h(v_2)\|. \tag{4.18}$$

Since bounded derivative implies Lipschitz continuous (see Theorem 3.2.4 [12]) and

$$\|h'(v)\| = \| - g_{11}^{-1}(h(v), v) g_{12}(h(v), v)\| \leq k_5 k_3, \tag{4.19}$$

we know that

$$\|h(v_1) - h(v_2)\| \leq k_7 \|v_1 - v_2\|. \tag{4.20}$$

for some $k_7$.

Putting these results together gives

$$\|f'(v_1) - f'(v_2)\| \leq \left(k_2(1 + k_5 k_3)^2(1 + k_7)\right) \|v_1 - v_2\| = k_8 \|v_1 - v_2\|. \tag{4.21}$$

Finally, since in the exact case $\Delta v_k = -f_k'^{-1} f_k$, it follows that $\alpha_k = 0$. Therefore, Proposition 3.1 and Theorem 3.2 hold for $f(v)$. □

**5. Approximate NlEm.** In an approximate Newton method, the Newton correction is computed by

$$M_k x_k = -g_k, \tag{5.1}$$

where $M_k$ is an approximation of $g_k'$. Convergence of this method is given by showing that $\alpha_k < 1$ in (3.5).

Recall that NlEm can be viewed as GAN applied to (1.11). The $\alpha_k$ for this nonlinear system is

$$\alpha_k = \frac{\|f(v_k) + f'(v_k)\Delta v_k\|}{\|f(v_k)\|}. \tag{5.2}$$

In approximate NlEm, (1.15) and $g_1(u, v_k) = 0$ are not solved exactly. Since the $g_1$ system is not solved exactly, the actual $h(v_k)$ is not computed at iteration $k$. Thus, verifying $\alpha_k < 1$ is impossible because the actual $f$ and $f'$ are not computed. Denote by $\tilde{h}(v)$ the computed approximation to $h(v)$. Although $h(v)$ was continuous and differentiable, $\tilde{h}(v)$ may be neither. In the remainder of this section we will use $\tilde{g}_k$, $\tilde{g}'_k$, $\tilde{g}_{1k}$, $\tilde{g}_{2k}$, $\tilde{g}_{11k}$, $\tilde{g}_{12k}$, $\tilde{g}_{21k}$, $\tilde{g}_{22k}$ to denote $g$, $g'$, $g_1$, $g_2$, $g_{11}$, $g_{12}$, $g_{21}$, $g_{22}$, respectively, evaluated at $(\tilde{h}(v_k), v_k)$. Recall that at each step of NlEm, a linear system is solved. Define the residual of this solve in the approximate case to be

$$r_k \equiv \tilde{g}'_k \begin{bmatrix} \Delta u_k \\ \Delta v_k \end{bmatrix} + \tilde{g}_k. \tag{5.3}$$

In this section, a bound is given for the $\alpha_k$ in (5.2). This bound will allow us to show that in the limit quadratic convergence is attainable if the $g_1$ system and (1.15) are solved accurately enough. The analysis of the convergence of approximate NlEm proceeds in two stages. In the first stage, an expression is derived that bounds the $\alpha_k$ in (5.2) in terms of the accuracy of solving the $g_1$ system of equations, i.e., $\|g_{1k}\|$. In the second stage, the terms of that expression are bounded in two ways with two theorems. The first theorem is existential, giving the conditions under which approximate NlEm attains higher-order convergence. These conditions are not verifiable. The second theorem gives computable conditions for approximately determining $\alpha_k$.

### 5.1. Bounding $\alpha_k$.

LEMMA 5.1. *Under the assumptions of Theorem* 4.2, *the* $\alpha_k$ *of* (5.2) *can be bounded by*

$$\begin{aligned} \alpha_k \|f_k\| = \alpha_k \|g_{2k}\| \le & \left\| \tilde{g}_{21k} \tilde{g}_{11k}^{-1} \int_0^1 \left[ g_{11}(\tilde{h} + s(h - \tilde{h}), v_k) - \tilde{g}_{11k} \right] (h - \tilde{h}) ds \right\| \\ & + \left\| \int_0^1 \left[ g_{21}(\tilde{h} + s(h - \tilde{h}), v_k) - \tilde{g}_{21k} \right] (h - \tilde{h}) ds \right\| \\ & + \| (g_{22k} - g_{21k} g_{11k}^{-1} g_{12k}) - (\tilde{g}_{22k} - \tilde{g}_{21k} \tilde{g}_{11k}^{-1} \tilde{g}_{12k}) \| \, \|\Delta v_k\| \\ & + \| r_{2k} - \tilde{g}_{21k} \tilde{g}_{11k}^{-1} r_{1k} \|, \end{aligned} \tag{5.4}$$

*where we have partitioned* $r_k$ *into* $r_{1k}$, $r_{2k}$ *corresponding to the blocking of* $g$.

*Proof.* Recall that

$$\begin{aligned} \alpha_k \|f_k\| = \alpha_k \|g_{2k}\| &= \|(g_{21k} h' + g_{22k}) \Delta v_k + g_{2k}\| \\ &= \|(g_{22k} - g_{21k} g_{11k}^{-1} g_{12k}) \Delta v_k + g_{2k}\| \end{aligned} \tag{5.5}$$

where, from (5.3), $\Delta v_k$ satisfies

$$\begin{bmatrix} \tilde{g}_{11k} & \tilde{g}_{12k} \\ \tilde{g}_{21k} & \tilde{g}_{22k} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta v_k \end{bmatrix} + \begin{bmatrix} \tilde{g}_{1k} \\ \tilde{g}_{2k} \end{bmatrix} = \begin{bmatrix} r_{1k} \\ r_{2k} \end{bmatrix}. \tag{5.6}$$

That the $\tilde{g}_{ijk}$ exist, for $i, j = 1, 2$, and $\tilde{g}_{11k}$ nonsingular follows from

$$(\tilde{h}(v_k), v_k) \in S_0(g) \subset S_0(g_1). \tag{5.7}$$

From (5.6) it follows that

$$\begin{aligned} \Delta v_k = & (\tilde{g}_{22k} - \tilde{g}_{21k} \tilde{g}_{11k}^{-1} \tilde{g}_{12k})^{-1} (\tilde{g}_{2k} - \tilde{g}_{21k} \tilde{g}_{11k}^{-1} \tilde{g}_{1k}) \\ & + (\tilde{g}_{22k} - \tilde{g}_{21k} \tilde{g}_{11k}^{-1} \tilde{g}_{12k})^{-1} (r_{2k} - \tilde{g}_{21k} \tilde{g}_{11k}^{-1} r_{1k}). \end{aligned} \tag{5.8}$$

We begin by deriving an inequality relating $g_{2k}$ in terms of $\tilde{g}_{2k}$. Applying the mean value theorem gives

$$g_{2k} = \tilde{g}_{2k} + \int_0^1 g_2'(\tilde{h} + s(h - \tilde{h}), v_k) \begin{bmatrix} h - \tilde{h} \\ 0 \end{bmatrix} ds$$

$$= \tilde{g}_{2k} + \tilde{g}_{21k}(h - \tilde{h}) + \int_0^1 \left[ g_{21}(\tilde{h} + s(h - \tilde{h}), v_k) - \tilde{g}_{21k} \right] (h - \tilde{h}) ds. \tag{5.9}$$

Similarly, for $g_1$,

$$0 = g_{1k} = \tilde{g}_{1k} + \tilde{g}_{11k}(h - \tilde{h}) + \int_0^1 \left[ g_{11}(\tilde{h} + s(h - \tilde{h}), v_k) - \tilde{g}_{11k} \right] (h - \tilde{h}) ds. \tag{5.10}$$

We derive an expression for $g_{2k}$ by solving for $h - \tilde{h}$ in the second term of (5.10) and substituting into (5.9). The expression is given by

$$g_{2k} = \tilde{g}_{2k} - \tilde{g}_{21k}\tilde{g}_{11k}^{-1}\tilde{g}_{1k} - \tilde{g}_{21k}\tilde{g}_{11k}^{-1} \int_0^1 \left[ g_{11}(\tilde{h} + s(h - \tilde{h}), v_k) - \tilde{g}_{11k} \right] (h - \tilde{h}) ds + \int_0^1 \left[ g_{21}(\tilde{h} + s(h - \tilde{h}), v_k) - \tilde{g}_{21k} \right] (h - \tilde{h}) ds. \tag{5.11}$$

Inequality (5.4) is determined by substituting (5.8) and (5.11) into (5.5) and applying the triangle inequality. □

PROPOSITION 5.2. *Under the assumptions of Theorem* 4.2, *the* $\alpha_k$ *of Lemma* 5.1 *satisfies*

$$\alpha_k \|g_{2k}\| \leq k_2(1 + k_5k_3)^2 c \|\tilde{g}_{1k}\| \|\Delta v_k\| + \frac{k_3 + k_5k_3^2}{2} c^2 \|\tilde{g}_{1k}\|^2 + (1 + k_3k_5)\|r_k\|, \tag{5.12}$$

*where* $c = \frac{M}{1-\theta}$.

We preface the proof with a few remarks. Previously we noted the existence of $\tilde{g}_{ijk}$ and the nonsingularity of $\tilde{g}_{11k}$. The assumptions allow us to actually bound these quantities. That is, $\|\tilde{g}_{ijk}\| \leq k_3$, $i, j = 1, 2$, and $\|\tilde{g}_{11k}^{-1}\| \leq k_5$. Note also that

$$(\tilde{h}(v_k) + t_k \Delta u_k, v_k + t_k \Delta v_k) \in S_1(g) \subset S_0(g_1). \tag{5.13}$$

Thus we may use $\Delta u_k$ to generate a starting guess for the solution of $g_1(u, v_k + t_k \Delta v_k)$. The algorithmic implication of this point is that in the regime of quadratic convergence of (the whole) $g$, NlEm need not do a solve of the $g_1$ equations; they will already be small enough. Finally, note that

$$(h(v_k), v_k), (\tilde{h}(v_k), v_k) \in S_0(g_1); \tag{5.14}$$

thus

$$(\tilde{h}(v_k) + s(h(v_k) - \tilde{h}(v_k)), v_k) \in S_1(g_1) \tag{5.15}$$

for $0 \leq s \leq 1$ and therefore the Lipschitz condition holds along the line segment.

*Proof.* The proof proceeds by bounding each of the terms of (5.4). Expanding norms and applying the Lipschitz condition to the first term in (5.4) yields

$$\|\tilde{g}_{21k}\tilde{g}_{11k}^{-1}\int_0^1 \left[g_{11}(\tilde{h}+s(h-\tilde{h}),v_k)-\tilde{g}_{11k}\right](h-\tilde{h})ds\| \le k_5\frac{k_3^2}{2}\|h-\tilde{h}\|^2. \tag{5.16}$$

Similarly, the second term can also be bounded yielding

$$\left\|\int_0^1 \left[g_{21}(\tilde{h}+s(h-\tilde{h}),v_k)-\tilde{g}_{21k}\right](h-\tilde{h})ds\right\| \le \frac{k_3}{2}\|h-\tilde{h}\|^2. \tag{5.17}$$

Applying Proposition 3.7 and Property P1 to the third term of (5.4), we get

$$\|(g_{22k}-g_{21k}g_{11k}^{-1}g_{12k})-(\tilde{g}_{22k}-\tilde{g}_{21k}\tilde{g}_{11k}^{-1}\tilde{g}_{12k})\| \le k_2(1+k_5k_3)^2\|h-\tilde{h}\|. \tag{5.18}$$

The fourth term is bounded by applying Properties P1 and P2 to note that

$$\|r_{2k}-\tilde{g}_{21k}\tilde{g}_{11k}^{-1}r_{1k}\| \le \|L^{-1}\|\|r_k\|, \tag{5.19}$$

where

$$L=\begin{bmatrix} I & 0 \\ \tilde{g}_{21k}\tilde{g}_{11k}^{-1} & I \end{bmatrix}. \tag{5.20}$$

Thus

$$\|r_{2k}-\tilde{g}_{21k}\tilde{g}_{11k}^{-1}r_{1k}\| \le (1+k_3k_5)\|r_k\|. \tag{5.21}$$

Collecting (5.16)–(5.21) gives

$$\begin{aligned}\alpha_k\|g_{2k}\| &\le k_2(1+k_5k_3)^2\|h-\tilde{h}\|\|\Delta v_k\| \\ &\quad +\frac{k_3+k_5k_3^2}{2}\|h-\tilde{h}\|^2+(1+k_3k_5)\|r_k\|.\end{aligned} \tag{5.22}$$

Since $h(v_k)$ is the root of the equation $g_1(u,v_k)$ with $v_k$ fixed and the assumptions assure that GAN converges from $\tilde{h}(v_k)$, we may apply Proposition 3.3 to conclude that

$$\|h-\tilde{h}\| \le c\|\tilde{g}_{1k}\|. \tag{5.23}$$

Substituting (5.23) into (5.22) gives the desired result. □

**5.2. Convergence theorems.** We continue by giving conditions for bounding the terms of $\alpha_k$ in (5.12). Two theorems are presented. The first theorem gives the conditions under which higher-order convergence can be expected; however, the result is existential. It is not possible to verify the conditions computationally. The second theorem gives a method for testing the size of $\alpha_k$ computationally.

THEOREM 5.3. *Under the assumptions of Proposition* 5.2, *if conditions*

C1: $\|g_1(\tilde{h}(v_k),v_k)\| \le \|g_2(h(v_k),v_k)\|$, *and*

C2: $\|r_k\| \le \|g_2(h(v_k),v_k)\|^{(1+p)}$

*hold, then*

$$\begin{aligned}\alpha_k\|g_{2k}\| &\le (1+k_3k_5)\|g_{2k}\|^{1+p} \\ &\quad +\left(2k_1k_2(1+k_5k_3)^2c_1+\frac{k_3+k_5k_3^2}{2}c_1\right)\|\tilde{g}_{2k}\|^2,\end{aligned} \tag{5.24}$$

*where*

$$\|\tilde{g}_{2k}\| \leq (1 + k_5 c_1)\|g_{2k}\| + \frac{k_3}{2} c_1^2 \|g_{2k}\|^2. \tag{5.25}$$

*Proof.* The theorem follows from Proposition 5.2 by bounding the quantities $\|\tilde{g}_{1k}\|$, $\|\Delta v_k\|$, and $\|\tilde{g}_{2k}\|$. The bound for $\|\tilde{g}_{1k}\|$ is given by condition C1.

By Assumption A2 we know that

$$\left\| \begin{bmatrix} \Delta u_k \\ \Delta v_k \end{bmatrix} \right\| \leq k_1 \|\tilde{g}_k\|. \tag{5.26}$$

Applying Properties P1 and P2 and condition C1 it follows that

$$\|\Delta v_k\| \leq k_1(\|\tilde{g}_{1k}\| + \|\tilde{g}_{2k}\|) \leq 2k_1\|\tilde{g}_{2k}\|. \tag{5.27}$$

All that remains is to bound $\|\tilde{g}_{2k}\|$. We apply the mean value theorem to give

$$\tilde{g}_{2k} = g_{2k} + \int_0^1 g_2'(h + s(\tilde{h} - h), v_k) \begin{bmatrix} \tilde{h} - h \\ 0 \end{bmatrix} ds. \tag{5.28}$$

Following the methodology that led to inequality (5.17) and applying the triangle inequality and (5.23) gives

$$\|\tilde{g}_{2k}\| \leq \|g_{2k}\| + k_5 c_1 \|\tilde{g}_{1k}\| + \frac{k_3}{2} c_1^2 \|\tilde{g}_{1k}\|. \tag{5.29}$$

Finally, applying condition C1 yields the inequality (5.25). □

Neither condition C1 nor C2 are verifiable. The following theorem gives conditions that are verifiable.

THEOREM 5.4. *Under the assumptions of Proposition* 5.2, *if conditions*

C1a: *the* $g_1$ *equation is solved for* $\tilde{h}(v_k)$ *so that* $\|g_1(\tilde{h}(v_k), v_k)\| \leq \|g_2(\tilde{h}(v_k), v_k)\|$, *and*

C2a: $\|r_k\| \leq \|g_2(\tilde{h}(v_k), v_k)\|^{(1+p)}$

*hold, then*

$$\begin{aligned} \alpha_k \|g_{2k}\| &\leq (1 + k_3 k_5)\|\tilde{g}_{2k}\|^{1+p} \\ &\quad + \left(2k_1 k_2 (1 + k_5 k_3)^2 c_1 + \frac{k_3 + k_5 k_3^2}{2} c_1\right) \|\tilde{g}_{2k}\|^2. \end{aligned} \tag{5.30}$$

The proof of Theorem 5.4 follows directly from the proof of Theorem 5.3. The intention of Theorem 5.4 is twofold. First, if $\|\tilde{g}_{2k}\|$ is small enough, then at the very least $\alpha_k < 1$ and, therefore, the method is converging (cf. Proposition 3.1). Second, in the limit when $\|r_k\| \leq \|\tilde{g}_{2k}\|^2$, quadratic convergence is attained.

Note that in the statement of Theorem 5.4 we write that $g_1$ should be solved so that $\|g_1(\tilde{h}(v_k), v_k)\| < \|g_2(\tilde{h}(v_k), v_k)\|$. Unfortunately, this inequality can only be checked after a function evaluation is done to determine $g_2(\tilde{h}(v_k), v_k)$. To avoid having to recompute $g_2$ every time a new $\tilde{h}(v_k)$ is generated, i.e., after every iteration of GAN on the $g_1$ system, we suggest that $g_1$ be solved so that $\|g_1(\tilde{h}(v_k), v_k)\| < \|g_2(\tilde{h}(v_{k-1}), v_{k-1})\|^2$.

COROLLARY 5.5. *Under the conditions of Theorem* 5.4, *with* $p = 1$, *if* C1a *is replaced with*

$$\|g_1(\tilde{h}(v_k), v_k)\| = O(\|g_2(\tilde{h}(v_{k-1}), v_{k-1})\|^2), \tag{5.31}$$

*then when* $\|g_2(\tilde{h}(v_{k-1}), v_{k-1})\|$ *is sufficiently small, i.e., close enough to the root, the convergence of the method is quadratic.*

*Proof.* The proof follows from Theorem 3.2 with the observation that sufficiently close to the root

$$\|g_2(\tilde{h}(v_k), v_k)\| = O(\|g_2(\tilde{h}(v_{k-1}), v_{k-1})\|^2), \tag{5.32}$$

and therefore condition C1 holds. □

**5.3. Accuracy requirement for $g_1$ system.** As pointed out in §2, inaccuracies in computing $f_k$ cause a practical problem in the solution process. Even if the seemingly fortuitous inequality

$$\|g_2(\tilde{h}(v_k), v_k)\| < \|f(v_k)\| \tag{5.33}$$

holds, there might not be any $t_k$ such that

$$\|g_2(\tilde{h}(v_{k+1}), v_{k+1})\| < \|g_2(\tilde{h}(v_k), v_k)\|. \tag{5.34}$$

Algorithmically, failure to satisfy equation (5.34) for a given choice of $t_k$ leads to two options. The first is to keep trying smaller $t_k$'s. The second is to go back to the previous iteration because the $g_1$ system was not solved accurately at that point. This section gives an approximate formula for determining whether $\|\tilde{g}_{1k}\|$ is close enough to zero to make $\|\tilde{g}_{2k+1}\| < \|\tilde{g}_{2k}\|$ a possibility. Satisfying this formula will give the user confidence that a norm reduction will eventually occur for $t_k$ small enough.

PROPOSITION 5.6. *Recall* $f_k \equiv g_2(h(v_k), v_k)$. *Define* $\tilde{f}_k$ *to be an approximation of* $f_k$. *Let*

$$\|\tilde{f}_k - f_k\| \le \delta_k \|f_k\|. \tag{5.35}$$

*If* $0 < \delta_k,\ \delta_{k+1} < 1$, *and* $\|\tilde{f}_{k+1}\| \le \mu_1 \|\tilde{f}_k\|$, $\mu_1 < 1$, *then*

$$\|f_{k+1}\| \le \mu_2 \|f_k\|, \tag{5.36}$$

*where*

$$\mu_2 \|f_k\| \equiv \mu_1 \frac{(1+\delta_k)}{1-\delta_{k+1}} \|f_k\|. \tag{5.37}$$

*Proof.* We have

$$\begin{aligned} \|\tilde{f}_{k+1}\| &\le \mu_1 \|\tilde{f}_k\| \\ &\le \mu_1 \|\tilde{f}_k - f_k\| + \mu_1 \|f_k\| \\ &\le \mu_1 \delta_k \|f_k\| + \mu_1 \|f_k\| = (1+\delta_k)\mu_1 \|f_k\|; \end{aligned} \tag{5.38}$$

adding $\|\tilde{f}_{k+1} - f_{k+1}\|$ to both sides gives

$$\begin{aligned} \|\tilde{f}_{k+1} - f_{k+1}\| + \|\tilde{f}_{k+1}\| &\le (1+\delta_k)\mu_1 \|f_k\| + \|\tilde{f}_{k+1} - f_{k+1}\| \\ \|f_{k+1}\| &\le (1+\delta_k)\mu_1 \|f_k\| + \delta_{k+1} \|f_{k+1}\|; \end{aligned} \tag{5.39}$$

solving for $\|f_{k+1}\|$ gives

$$\|f_{k+1}\| \le \mu_1 \frac{(1+\delta_k)}{1-\delta_{k+1}} \|f_k\|. \tag{5.40}$$

□

**NlEm** $(l, w^l, tol, MAX)$ /* at the outer level, l = 1 */
1. **if** $g^l$ consists of no equations, **return**.
2. **NlEm**$(l+1, P^{l+1}(w^l), tol_{-1}, MAX_{-1})$
   Note that this call to NlEm changes the $w^{l+1}$ unknowns.
3. $k = 0$, $w_0^l = w_l$
4. **while** $((\|g^l(w_k^l)\| > tol)$ **and** $(k < MAX))$
5. compute $x_k^l$
6. **repeat**
7. choose a $t_k$
8. $\hat{w}_{k+1}^l = w_k^l + t_k x_k^l$
9. **NlEm**$(l+1, P^{l+1}(\hat{w}_{k+1}^l), tol_k, MAX_k)$
10. **until** $(\|g^l(\hat{w}_{k+1}^l)\| < \theta\|g^l(w_k^l)\|)$ **or** $t_k$ is too small
11. **if** $t_k$ is too small, **return** with $w^l$ unchanged
12. $k = k+1$, $w_k^l = \hat{w}_k^l$
13. **if** $(k > max)$ **return** with $w^l$ unchanged
14. **return** with $w^l = w_k^l$

FIG. 5. *A detailed view of the NlEm algorithm.*

For our purposes we will assume that there is some $\mu_2$ we wish to achieve. We compute a candidate $\tilde{f}_{k+1}$, yielding a $\mu_1$. We are then able to determine if $\mu_2$ is achievable. Recall from (5.11) and subsequent analysis that

$$\|\tilde{g}_{2k} - g_{2k}\| \leq C\|h - \tilde{h}\|^2 + \|\tilde{g}_{21k}\tilde{g}_{11k}^{-1}\tilde{g}_{1k}\|. \tag{5.41}$$

An estimate for $\delta_k$ can be obtained by dropping the $C\|h - \tilde{h}\|^2$ term. Hence

$$\delta_k \cong \frac{\|\tilde{g}_{21k}\tilde{g}_{11k}^{-1}\tilde{g}_{1k}\|}{\|\tilde{g}_{2k} - \tilde{g}_{21k}\tilde{g}_{11k}^{-1}\tilde{g}_{1k}\|}, \tag{5.42}$$

where the denominator is an $O(\|h - \tilde{h}\|^2)$ approximation for $g_2(h(v_k), v_k)$.

**6. Implementation details.** While the outline of NlEm given in §2 is sufficient for a programmer to generate a piece of code, we have noticed certain improvements that make the transition from NlEm back to GAN transparent. Thus, in this section we present a more detailed algorithm and address certain issues raised in §5.

The algorithm in Figure 5 is a recursive algorithm that allows NlEm to be performed on more than two levels; i.e., NlEm is used to solve the $g_1$ equations. One place we view this as a possibility is on a system of nonlinear PDEs. The $g_1$ equation at one level might be one or more of the PDEs, while the $g_1$ equation at the next level might be the equations associated with some of the grid points. While we believe this to be a case where nested NlEM might be useful, we have not actually implemented NlEm for this problem.

Define by $g^l$ the set of equations associated with level $l$. $g^1$ is the full set of equations. Define $P^{l+1}$ to be a projector function that, when applied to the variables $w^l$ (the variables associated with level $l$), returns $w^{l+1}$.

There are several interesting points about this routine.

1. If $g^2$ consists of no equations then the code above is GAN.
2. Notice that after computing $x_k^l$, all of $\hat{u}_{k+1}^l$ is updated. When the answer is close enough to the root that GAN is in the regime of quadratic convergence, the call

to NlEm at the next step will (if the $tol_k$ is chosen reasonably) return doing no computation. Thus, the transition from NlEm back to GAN is achieved with just the cost of an extra function call.

3. We make $tol_k$ smaller and $MAX_k$ bigger as $t_k$ decreases. As pointed out in §5 there is a problem with determining whether norm reduction fails because $t_k$ has not been made small enough, or if the $g_1$ equations were not solved accurately enough at the previous level. While solving the $g_1$ equations more accurately at this level does not help the problem of failure at the moment, we have more confidence that it will not be a problem on the next iteration.
4. The maximum number of NlEm iterations needs to be set carefully. $MAX_{-1}$ should be set very high so that NlEm does not give up too early trying to solve them. On the other hand $MAX_k$ should be set to a small level (increasing as $t_k$ decreases). Long periods of time should not be spent trying to get solutions for $t_k$'s when some smaller $t_k$ will get an answer faster. We know that the answer can be found for some smaller $t_k$ since the $u$ from the previous iteration can be used if $t_k$ is sufficiently close to 0.
5. It is sometimes the case that $g^2$ is not known before beginning. In that case $g^2$ would be calculated before the initial call to **NlEm**. It is important that $g^2$ not be changed after this. If for some reason the user does wish to dynamically change $g^2$, it is important to check that after the initial call to **NlEm** there has been a norm reduction. If this is not done, thrashing can occur and the code will not terminate.

**7. Examples.** In this section we give examples showing the performance of NlEm. The first two examples demonstrate how NlEm is used for a system of nonlinear equations arising from the discretization of an ODE. The first example shows that NlEm can improve performance on a problem where grid refinement might otherwise have been used. That is, the user would recognize that most of the action is taking place at some set of points and the rest of the domain is quiescent. Instead of gridding the domain differently, a fine uniform grid is used, and more computation is done on the points of interest. The cost of the algorithm is improved by five times. The second example shows how NlEm can be used to solve problems where too few grid points are used. When too few grid points are used, the steep valleys as seen in §2 sometimes occur. NlEm can be used on the "bad" grid points giving results where GAN fails. The third example is a simple 2-D semiconductor device. By eliminating two of the PDEs, we get convergence with an example that would not converge using GAN.

**7.1. Elimination by grid 1.** Consider the nonlinear boundary value problem given by

$$-u'' + u^3 + \left(4\frac{(x-0.5)^2}{10^8} - 2\times 10^{-4}\right)u - 10^9 e^{-3(\frac{x-0.5}{0.01})^2} = 0,$$

$$u(0) = 0, \quad u(1) = 0. \tag{7.1}$$

This has the solution

$$u(x) = 10^3 e^{-(\frac{x-0.5}{0.01})^2}. \tag{7.2}$$

The action in this problem is around $x = 0.5$, where a large spike occurs.

We give a few details about this example using 100 discretized equations. An initial guess for $u_1 \ldots u_{100}$ is made. The $g_1$ system is the discretized equations 49-52. Using the initial guess for $u_{48}$ and $u_{53}$ as fixed values, the nonlinear subsystem $g_1$ is solved to a very high accuracy (e.g., $10^{-12}$ for the norm of the residual). Following this initial solve of the $g_1$ system, the main loop begins. The entire system $g$ is evaluated at the new point; that is, $u_1 - u_{48}$ and $u_{53} - u_{100}$ are unchanged and $u_{49} - u_{52}$ have the values just computed. The

TABLE 1

*Comparison of NlEm to GAN for example given in §7.1. "Points" refers to the number of grid points in the domain. "its" refers to the number of iterations required. The grid points between LEFT and RIGHT are in* $g_1$.

| Points | GAN time | GAN its | NlEm time | NlEm its | LEFT | RIGHT |
|---|---|---|---|---|---|---|
| 100 | 0.15 | 10 | 0.03 | 5 | 48 | 53 |
| 500 | 1.47 | 10 | 0.17 | 5 | 240 | 261 |
| 1000 | 2.81 | 10 | 0.32 | 5 | 480 | 521 |
| 5000 | 10.7 | 10 | 1.66 | 5 | 2400 | 2601 |

Jacobian is computed and the Newton equations are solved, giving the update vector $\Delta u$. Then a damping parameter $t$ is chosen, and $u_{48} + t\Delta u_{48}$ and $u_{53} + t\Delta u_{53}$ are computed. These are used as fixed values in the $g_1$ solve, as above, except that the accuracy of the solution is relaxed, to say $t * (\|g\|/\|g_i\|)^2$, where $g_i$ is the residual at the initial guess. Although we used the damping parameter to control the accuracy, any function of $t$ can be used that shows similar behavior; i.e., the accuracy requirement is increased for smaller $t$ and is relaxed when $t$ approaches 1. This ensures that as the regime of quadratic convergence is approached, few (if any) iterations of Newton's method on equations 49–52 are required. The main loop repeats when a damping parameter is chosen so that a norm reduction for the whole $g$ has occurred following the $g_1$ solve.

In Table 1 we see that NlEm runs about five times faster than GAN. It is so much faster because in the first few iterations of GAN, very small values of the damping parameter are required. The very small damping parameters mean that the function must be evaluated many times. NlEm also has to use small values for the damping parameter for the $g_1$ equations, but performing a large number of function evaluations on four percent of the points is not very costly.

**7.2. Elimination by grid 2.** Consider the nonlinear boundary value problem given by

$$
\begin{aligned}
&-u'' + \alpha(x)u' + u^3e^u + k(x) = 0,\\
&u(0) = 0, \quad u(1) = 0,\\
&\alpha(x) = 10^9 e^{-(\frac{x-0.5}{0.01})^2}\sqrt{|x-0.5|}, \quad k(x) = \begin{cases} -10^6 & x < 0.5,\\ 10^6 & x \geq 0.5, \end{cases}
\end{aligned} \tag{7.3}
$$

where $u'$ is discretized using central differences. Fifty grid points were used on this problem; but because central differences were used, this was not enough grid points. Using the same code as in the previous example, the computed solution under these circumstances shows ringing around 0.5. GAN is unable to solve this problem. NlEm solves the problem in 17 iterations. Both methods solve the problem when more grid points are used.

**7.3. Example from semiconductor device simulation.** One of the areas in which we have been directing the application of this method is semiconductor device simulation [9]. NlEm was implemented in the semiconductor device simulator SIMUL [11]. In the example presented here, the equations take the form

$$-\nabla\cdot(\epsilon\nabla u) + n - p + N = 0, \tag{7.4}$$

$$-\nabla\cdot\mathbf{J}_n + R_n(n,p) = 0, \tag{7.5}$$

$$-\nabla\cdot\mathbf{J}_p + R_p(n,p) = 0, \tag{7.6}$$

where $u, n$, and $p$ are functions in space (for the example we present here, $(x, y) \in \mathbb{R}^2$), $\epsilon$ and $N$ are spatially related constants, $J_n$ and $J_p$ are electron and hole current densities, and $R_n$ and $R_p$ are recombination terms.

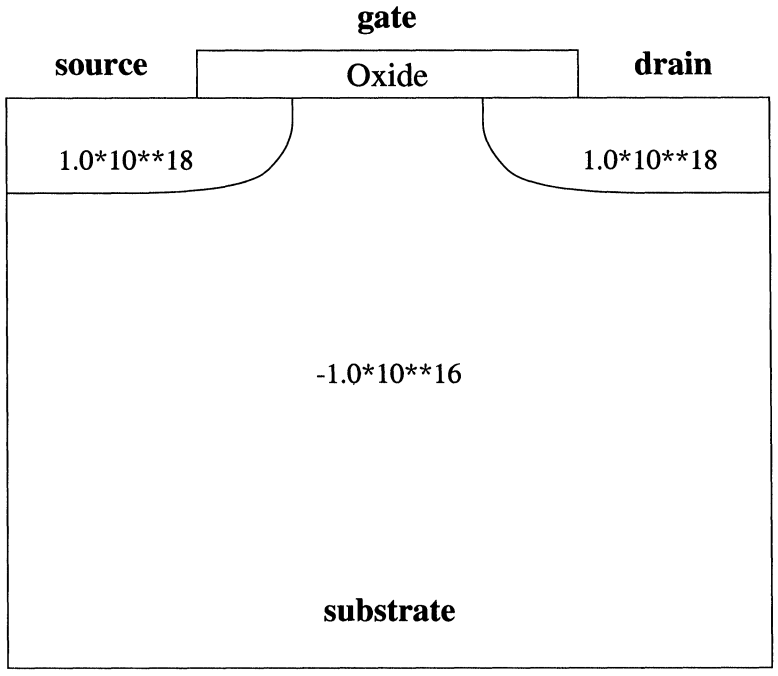


FIG. 6. *NMOS transistor. The numbers represent the background dopings of the N and P regions.*

These steady state equations are usually solved using GAN. Under certain circumstances GAN fails to converge or converges too slowly. In these cases, other methods are employed to generate a good initial guess so that GAN converges well. These methods include nonlinear Gauss-Seidel and continuation in the boundary conditions. In nonlinear Gauss-Seidel each equation is solved in turn using the most recent values for the other variables as constants. Sometimes nonlinear Gauss-Seidel also fails in which case continuation in the voltage at the contacts is employed.

The example given in Figure 6 is an NMOS transistor. The voltages at the various contacts are source=0, gate=1, substrate=0, and drain=1 volts. The problem was originally solved using a combination of nonlinear Gauss-Seidel and continuation in 235 seconds. When the system is solved with NlEm, with the $g_1$ equations being the electron and hole continuity equations, the solution time is 36 seconds. Other choices for the $g_1$ equations did not perform as well. In the experience of the author of SIMUL, choosing electron and hole continuity equations to be $g_1$ works in many cases. We also point out that when the problem is easy, e.g., for drain voltage of 0.1 volts, NlEm takes longer than GAN. We reiterate that NlEm should be used in cases where GAN is having difficulties.

**Acknowledgment.** We are indebted to Stephan Müller for implementing NlEm in the semiconductor device simulation package SIMUL and for patiently helping the authors find an interesting example. We also thank Wolfgang Fichtner for allowing us to spend time with his student.

REFERENCES

[1] R. E. BANK AND D. J. ROSE, *Global approximate Newton methods*, Numer. Math., 37 (1981), pp. 279-295.
[2] R. E. BANK, D. J. ROSE, AND W. FICHTNER, *Numerical methods for semiconductor device simulation*, SIAM J. Sci. Statist. Comput., 4 (1993), pp. 416-435.

[3] D. P. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[4] K. M. BROWN, *A quadratically convergent Newton-like method based upon Gaussian elimination*, SIAM J. Numer. Anal., 6 (1969), pp. 560-569.

[5] W. M. COUGHRAN, E. GROSSE, AND D. J. ROSE, *CAzM: A circuit analyzer with macromodeling*, IEEE Trans. Computer-Aided Design, ED-30, (1983), pp. 1207-1213.

[6] ———, *Aspects of computational circuit analysis*, in VLSI CAD Tools and Applications, W. Fichtner and M. Morf, eds, Kluwer Academic Publishers, Norwell, MA, 1987.

[7] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1983), pp. 413-432.

[8] ———, *Differentiation of pseudo-inverses, separable nonlinear least squares problems and other tales*, in Generalized Inverses and Applications, M. Z. Nashed, ed., Academic Press, New York, 1976, pp. 303-324.

[9] P. J. LANZKRON, D. J. ROSE, J. T. WILKES, S. MÜLLER, AND W. FICHTNER, *The use of nonlinear elimination in steady-state circuit and device simulation*, Proc. of the NUPAD IV, Seattle, WA, IEEE, 1992.

[10] W. H. LAWTON AND E. A. SYLVESTRE, *Elimination of linear parameters in nonlinear regression*, Technometrics, 13 (1971), pp. 461-467.

[11] S. MÜLLER, K. KELLS, J. LITSIOS, U. KRUMBEIN, A. SCHENK, AND W. FICHTNER, *SIMUL* 1.1 *Manual*, Integrated Systems Laboratory, ETH Zurich, Switzerland, 1993.

[12] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[13] J. T. WILKES, *A New Method for Solving Systems of Nonlinear Equations in Circuit Simulation*, Ph.D. thesis, Duke University, Durham, NC, 1994.