An approximative Newton method for coupled nonlinear systems

T.F. Chan

▶ **To cite this version:**

T.F. Chan. An approximative Newton method for coupled nonlinear systems. RR-0294, INRIA. 1984. inria-00076263

HAL Id: inria-00076263
https://inria.hal.science/inria-00076263

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.: (3) 954 90 20

Rapports de Recherche

N° 294

AN APPROXIMATE NEWTON METHOD FOR COUPLED NONLINEAR SYSTEMS

Tony F. CHAN

Mai 1984

UNE METHODE DE NEWTON APPROCHEE POUR DES SYSTEMES NON LINEAIRES COUPLES

AN APPROXIMATE NEWTON METHOD FOR COUPLED NONLINEAR SYSTEMS

Tony F. CHAN

Résumé :

On propose une méthode de Newton approchée pour résoudre le système non linéaire $G(u,t) = 0$ et $N(u,t) = 0$ où $u \in R^n$, $t \in R^m$, $G : R^n \times R^m \to R^n$ et $N : R^n \times R^m \to R^m$. La méthode consiste à appliquer l'itération de base S d'un solveur général pour l'équation $G(u,t) = 0$ *pour t fixé*. Elle est en conséquence bien adaptée aux problèmes pour lesquels un tel solveur existe ou peut être implémenté plus efficacement qu'un solveur pour le système couplé. On obtient des conditions sur S pour lesquelles la méthode est localement convergente. Schématiquement, si S est suffisament contractante pour G, alors la convergence pour le système complet est garantie . Sinon, on montre comment construire, à partir de S, $\hat{S}$ pour laquelle la convergence est assurée. Les résultats sont appliqués à des méthodes de continuation où N représente la forme discrète d'une condition de continuation. On montre que sous certaines conditions, l'algorithme converge si S converge pour G.

Mots-clés : Systèmes linéaires couplés, méthode de Newton, méthodes de continuation, optimisation avec contraintes, système d'équations aux dérivées partielles, non linéaires couplées.

Abstract :

We propose an approximate Newton method for solving the coupled nonlinear system $G(u,t) = 0$ and $N(u,t) = 0$ where $u \in R^n$, $t \in R^m$, $G : R^n \times R^m \to R^n$ and $N : R^n \times R^m \to R^m$. The method involves applying the basic iteration S of a general solver for the equation $G(u,t) = 0$ *with t fixed*. It is therefore well-suited for problems for which such a solver already exists or can be implemented

PAPIER RÉCUPÉRÉ ET RECYCLÉ

more efficiently than a solver for the coupled system. We derive conditions for S under which the method is locally convergent. Basically, if S is sufficiently contractive for G, then convergence for the coupled system is guaranteed. Otherwise, we show how to construct a $\hat{S}$ from S for which convergence is assured. These results are applied to continuation methods where N represents a pseudo-arclength condition. We show that under certain conditions the algorithm converges if S is convergent for G.

Keywords : Coupled nonlinear systems, Newton's method, continuation methods, constrained optimization, coupled nonlinear partial differential equations.

========

1. Introduction

In this paper, we are concerned with computational algorithms for solving coupled nonlinear systems of the form:

$$C(z) \equiv \begin{pmatrix} G(u,t) \\ N(u,t) \end{pmatrix} = 0$$

where $z \equiv (u,t), u \in R^n, t \in R^m, G : R^n \times R^m \mapsto R^n$ and $N : R^n \times R^m \mapsto R^m$. More general coupled systems can always be casted into this form. We shall assume that a solution z^* exists and that it is regular, i.e. the Jacobian

$$J = \begin{pmatrix} G_u & G_t \\ N_u & N_t \end{pmatrix}$$

is nonsingular at z^*. †

Since z^* is a regular solution of $C(z) = 0$, many conventional iterative algorithms can be applied to solve for z^*. However, this approach may fail to exploit certain structures which are inherent in the operators G and N but which do not exist in C. Such structures could be symmetry, positive definiteness, separability, sparsity and bandedness. Exploiting these structures may be crucial for the overall efficiency of the computational algorithm, especially for large problems.

In addition to these general properties, one may have special knowledge of G and N, perhaps already implemented in easily available efficient solvers, whereas such solvers may not exist for the coupled system. Situations like this occur quite often in applications to continuation methods, optimization problems and coupled partial differential equations. In continuation methods, G may represent a nonlinear system in u with dependence on some parameters t and N may represent an arclength condition constructed to follow the solution manifolds. If G represents a discretization of a well studied mathematical model (e.g. the Navier-Stokes equations with t being the Reynold's number), one may have special solvers for G (for fixed t) whereas these special techniques cannot be easily adapted to solve the coupled system [4]. Similar situations occur in constrained optimization problems, where t may represent the Lagrange multipliers and N the constraints. In coupled partial differential equations, for example those that arise in semiconductor modelling [8], G (with t fixed) may be some standard differential operator for which special efficient solvers exist whereas no such efficient solvers exist for the coupled system.

For the above reasons, in this paper we shall consider a special class of algorithms for solving the coupled system which makes use of a general solver (presumably efficient) for G, *for fixed* t. We shall assume that this solver is available in the form of a fixed point iteration operator S, which takes an approximate solution u_i and produce the next iterate $u_{i+1} = S(u_i, t)$. Since most solvers for nonlinear systems are iterative in nature, it should be relatively easy to extract S from them.

We emphaize that it may not be straightforward to incorporate such a solver S in most conventional methods for solving the coupled system. The most obvious approach is to use S in conjunction with a block relaxation method in which S is used to solve for u as a solution to the equation $G(u,t) = 0$ with t fixed. However, such an approach will most likely fail if the Jacobian J of the coupled system is not positive definite or diagonally dominant near the solution, which is usually the case if the coupling between the two equations is strong. Another classical method which has much better local convergence properties is Newton's method. However, a linear system with J must be solved at each step. This requires some approximations to G_u and G_t which may not be readily available (in S or otherwise). Moreover, while it is possible to exploit a solver for G_u when solving for the linear system with J [5, 12], it is not obvious how to exploit the special solver S if it does not use the Jacobian G_u explicitly.

†Throughout this paper, subscripts in u, t and z denote partial differentiation.

Thus it seems desirable to have a class of algorithms for solving the coupled system which can be proven to be convergent, at least locally, under rather general conditions but which can also effectively exploit a special solver S for G. Such an algorithm would, for example, allow continuation techniques to be easily applied to an application area in a modular fashion and with the efficiency built into special solvers specifically designed for the application. It would also allow constraints to be added to a special solver for a class of unconstrained optimization problems without a sacrifice in computational efficiency.

In Section 2, we present such an algorithm which is based on Newton's method for solving the coupled system $C(z) = 0$. The basic idea is to use S to *approximately* solve the linear systems involving J at each step of Newton's method. If S represents one step of Newton's method for the equation $G(u,t) = 0$ with t fixed, then the algorithm reduces exactly to Newton's method for the coupled system, with a block elimination algorithm [5, 12] applied to solve the systems with J. If S implements any other convergent method for $G(u,t) = 0$, then the linear systems for J are solved only approximately. In this way, the algorithm can be viewed as an *inexact Newton method* [7], except that the size of the residual is not directly controlled. In Section 3, we analyze the local convergence properties of this algorithm. Basically, we prove that if $\rho(S_u)$ or $||S_u||$ is sufficiently smaller than 1, then the algorithm is locally convergent. In other words, if S implements a reasonably fast convergent method for G, then the algorithm will converge locally for C. If $\rho(S_u)$ or $||S_u||$ is not small enough, we show how to construct a $\hat{S}$ from S for which convergence is assured. In Section 4, applications to arclength continuation methods are discussed. We show that under rather mild conditions the algorithm is locally convergent if S is convergent for G. Some concluding remarks are given in Section 5.

2. Algorithm

At each step of Newton's method applied to $C(z) = 0$, the following linear system

$$\begin{pmatrix} G_u & G_t \\ N_u & N_t \end{pmatrix} \begin{pmatrix} \delta u \\ \delta t \end{pmatrix} = -\begin{pmatrix} G \\ N \end{pmatrix}$$

has to be solved for the changes $(\delta u, \delta t)$ in the Newton iterates. In order to exploit a solver for G_u (assuming it is available), the above system is often solved by the following

Block Elimination Algorithm: [12]

1. Solve $G_u w = -G$ for w, where $w \in R^n$.
2. Solve $G_u v = G_t$ for v, where $v \in R^n \times R^m$.
3. Solve $(N_t - N_u v)\delta t = -(N + N_u w)$ for δt.
4. Compute $\delta u = w - v\delta t$.

Note that $m + 1$ linear systems involving G_u have to be solved. Now assume that we have a solver for $G(u,t) = 0$ in the form of a fixed point iteration $u \leftarrow S(u,t)$ with t fixed. For example, Newton's method for G would correspond to

$$S^{Newton} = u - G_u^{-1}(u,t)G(u,t).$$

The idea in the new algorithm is to use S to approximately solve fo w and v in Steps (1) and (2) in the Block Elimination Algorithm. Since the vector w is precisely the change in the iterate u in one step of Newton's method applied to $G(u,t) = 0$, it seems natural to approximate w by

$$w = S(u_i, t_i) - u_i,$$

where u_i and t_i are the current iterates. The situation for approximating v is slightly more complicated since it does not directly correspond to an iteration based on $G(u,t) = 0$. However, note that by differentiating the equation $G(u,t) = 0$ with respect to t we obtain

$$G_u u_t + G_t = 0$$

and thus at convergence

$$v = -u_t.$$

Since at convergence $u = S(u,t)$, it follows by differentiation that

$$u_t = S_u u_t + S_t$$

at z^*. Thus if S is sufficiently contractive for G (for example if $||S_u||$ is sufficiently small), then it seems reasonable to approximate v by $-S_t$. In particular, if $S = S^{Newton}$ then $S_u = 0$ and this approximation is exact. If S can easily be differentiated with respect to t (for example if S is linear in t), then S_t can be computed without too much difficulty. In general, S_t can be approximated by finite differencing S with respect to t. We summarize the above in the following :

Algorithm ANM (*Approximate Newton Method*) : Given an initial guess (u_0, t_0), iterate the following steps until convergence:

1. Compute $w = S(u_i, t_i) - u_i$.
2. For $j = 1, m$ compute
$$v_j = -\frac{S(u_i, t_i + \epsilon_j e_j) - S(u_i, t_i)}{\epsilon_j},$$
where v_j denotes the j-th column of v, ϵ_j denotes a small finite difference interval and e_j denotes the j-th unit vector.
3. Solve the following m by m system for d:
$$(N_t(u_i, t_i) - N_u(u_i, t_i)v)d = -(N(u_i, t_i) + N_u(u_i, t_i)w)$$
4. Compute $t_{i+1} = t_i + d$.
5. Compute $u_{i+1} = u_i + w - vd$.

Note that similar to the Block Elimination Algorithm, $m+1$ calls of S are needed per iteration. Moreover, for this algorithm to be well-defined, $(N_t - N_u v)^{-1}$ must exist at all the iterates so that the linear system for d can be solved. For $S = S^{Newton}$, we shall show that $(N_t - N_u v)^{-1}$ does exist at z^* if G_u is nonsingular there. For it follows from a LU-factorization of J with G_u as pivot that

$$det(J) = det(G_u) det(N_t - N_u G_u^{-1} G_t);$$

and since at z^* $det(J) \neq 0$ and $v = -u_t = G_u^{-1} G_t$, it follows that $det((N_t - N_u v)) \neq 0$ at z^*. Therefore, for a general, sufficiently contractive solver S (so that $v \approx -u_t$), it is reasonable to assume that $(N_t - N_u v)^{-1}$ exists locally around the solution z^*.

3. Convergence

In this section, we analyze the local convergence of Algorithm ANM. For this purpose, we view the algorithm as a fixed point iteration F:

$$z_{i+1} \leftarrow F(z_i).$$

We note that the necessary and sufficient condition for convergence is $\rho(F_z) < 1$ at the solution z^* and a sufficient condition is $||F_z|| < 1$ in some norm. We shall denote the first n components of F by F^1 and the last m components by F^2. Since all relevant quantities in this local analysis are to be evaluated at the solution z^*, from now on we shall drop all the arguments. We also note that at z^*, we have $w = 0$ and $d = 0$.

To simplify the analysis, we shall write

$$v = -S_t + \epsilon$$

where ϵ, with

$$||\epsilon|| = O(\max_{1 \le j \le m} |\epsilon_j|),$$

represents the truncation error in the finite difference approximation of v to $-S_t$.

In order to evaluate F_z, we need to compute F_u^1, F_t^1, F_u^2 and F_t^2. From the definition of Algorithm ANM, it follows that, at z^*,

$$\begin{aligned} F_u^1 &= I + w_u - (vd)_u = I + w_u - vd_u, \\ F_t^1 &= w_t - (vd)_t = w_t - vd_t, \\ F_u^2 &= d_u, \\ F_t^2 &= I + d_t. \end{aligned}$$

Therefore, we need to evaluate w_u, w_t, d_u and d_t at z^*. From the definition of w in Step (1) of Algorithm ANM, it is easily seen that

$$\begin{aligned} w_u &= S_u - I, \\ w_t &= S_t. \end{aligned}$$

After some manipulations, it can also be verified from the definition of d in Step (3) of Algorithm ANM that, at the solution z^*,

$$\begin{aligned} d_u &= -(N_t - N_u v)^{-1} N_u S_u, \\ d_t &= -I - (N_t - N_u v)^{-1} N_u \epsilon. \end{aligned}$$

Combining these results gives the following :

Lemma 3.1. *At the solution z^*,*

$$F_z = \begin{pmatrix} PS_u & P\epsilon \\ -(N_t - N_u v)^{-1} N_u S_u & -(N_t - N_u v)^{-1} N_u \epsilon \end{pmatrix},$$

where $P \equiv I + v(N_t - N_u v)^{-1} N_u$.

From Lemma 3.1 it follows directly that if $||S_u||$ and $||\epsilon||$ are sufficiently small in some norm, then $||F_z|| < 1$ and Algorithm ANM converges locally. If S_t is directly computable, then $\epsilon = 0$. If finite differencing is used and if the variables are scaled appropriately, then ϵ can usually be chosen to be of the order of the square root of the machine precision [10] which in most cases is much less than 1. Therefore, to simplify the analysis, we shall take $\epsilon = 0$ from now on. This assumption should have a relatively minor effect on the local convergence.

With this assumption, we have

$$F_z = \begin{pmatrix} PS_u & 0 \\ -(N_t - N_u v)^{-1} N_u S_u & 0 \end{pmatrix}.$$

Since $\rho(F_z) = \rho(PS_u)$, it immediately follows that

Theorem 3.1. *Algorithm ANM converges iff* $\rho(PS_u) < 1$.

Specifically, if $S = S^{Newton}$, we have $S_u = 0$ and therefore the quadratic convergence of Newton's method for the coupled system is recovered. On the other hand, it would be nice to determine sufficient conditions on the contractivity of S for Algorithm ANM to be convergent.

First of all, we have the following general sufficient condition:

Theorem 3.2. *Algorithm ANM converges if* $||S_u|| < \frac{1}{||P||}$, *in any vector induced norm.*

Proof. Follows from $\rho(PS_u) \leq ||PS_u|| \leq ||P||\,||S_u||$. ∎

Since in general $\rho(S_u) \leq ||S_u||$, it is desirable to have less stringent conditions on $\rho(S_u)$ instead. Unfortunately, this is possible only if P and S_u belong to special classes of matrices.

Lemma 3.2. *If* S_u *is normal, then* $\rho(PS_u) \leq ||P||_2\,\rho(S_u)$. *If in addition,* P *is normal, then* $\rho(PS_u) \leq \rho(P)\rho(S_u)$.

Proof. Follows easily from the fact for any matrix A, $\rho(A) \leq ||A||_2$ with equality if A is normal. ∎

Lemma 3.3. *If* P *and* S_u *are simultaneously diagonalizable, and the corresponding eigenvalues of* P *and* S_u *are* π_j *and* σ_j, *then* $\rho(PS_u) = \max_{1\leq j\leq n} |\sigma_j\pi_j|$.

These two lemmas give the following sufficient conditions on $\rho(S_u)$ for the convergence of Algorithm ANM:

Theorem 3.3.

If S_u *is normal, then Algorithm ANM converges if* $\rho(S_u) < \frac{1}{||P||_2}$. *If* P *and* S_u *are both normal, or are simultaneously diagonalizable, then Algorithm ANM converges if* $\rho(S_u) < \frac{1}{\rho(P)}$.

We note that in general $\rho(S_u) < 1$ is neither a necessary nor a sufficient condition for the convergence of Algorithm ANM. In other words, it could happen (and we have carried out numerical experiments confirming it) that a non-contractive S for G can lead to a convergent Algorithm ANM. This can happen, for example, if P and S_u are simultaneously diagonalizable and P has a small eigenvalue corresponding to a large eigenvalue of S_u (or vice versa), so that the product is smaller than 1. In this way, P can be thought of as a projection (an oblique one in general) operator. In practice, however, it would only be prudent to employ a contractive S with $\rho(S_u) < 1$.

Theorems 3.2 and 3.3 give upper bounds on $\rho(S_u)$ and $||S_u||$ for the convergence of Algorithm ANM. Based on the assumption that $(N_t - N_u v)$ and G_u are nonsingular at all the iterates, it follows that $\rho(P)$ and $||P||$ are bounded. Therefore, the upper bounds for $\rho(S_u)$ and $||S_u||$ in Theorems 3.2 and 3.3 are bounded away from zero. The size of $\rho(P)$ and $||P||$ depends on both N and S and must be estimated for the particular application.

If for a particular S, S_u does not satisfy any of these bounds, then convergence is not guaranteed by the above theorems. However, the following general technique can be systematically used to overcome the problem, provided $||S_u|| < 1$ in some norm. Define a modified iteration operator $\hat{S}$ by

$$\hat{S}(u,t) = \overbrace{S(S\cdots S(S}^{k \text{ times}}(u,t),t),\cdots,t),t).$$

In other words, $\hat{S}$ is obtained by iterating S k times with t fixed. It follows that

$$\hat{S}_u = S_u^k.$$

Therefore, we have

$$\rho(\hat{S}) \leq \|\hat{S}_u\| \leq \|S_u^k\| \leq \|S_u\|^k.$$

If $\|S_u\| < 1$, then a large enough value of k can always be chosen so that $\rho(\hat{S})$ or $\|\hat{S}_u\|$ satisfies one of the bounds in Theorems 3.3 and 3.2. For efficiency reasons, k should be chosen to be the smallest integer such that the largest applicable bound is satisfied.

4. Arclength Continuation

In this section, we apply the results of the last section to an important application area. In arclength continuation methods [2, 9, 12, 14], G represents a system of parameterized nonlinear equations, with u playing the role of the main variable, t the parameters and N represents certain auxilliary conditions. We shall restrict our attention to path-following continuation where $m = 1$, although the algorithm and theory developed in Sections 2 and 3 apply to other related problems as well (e.g. augmented systems defining singular points [1, 3, 11, 13, 15]).

The function N is usually defined in terms of the unit tangent $(\dot{u}, \dot{t})$ at a solution (u, t) which is the solution of:

$$G_u \dot{u} + G_t \dot{t} = 0$$
$$\|\dot{u}\|_2^2 + \dot{t}^2 = 1.$$

We shall concentrate on two typical N's that are widely used in the literature:

$$N^1 = \dot{u}_0^T(u - u_0) + \dot{t}_0(t - t_0) - \delta s,$$
$$N^2 = e_j^T \begin{pmatrix} u - u_0 \\ t - t_0 \end{pmatrix} - \delta s, \qquad 1 \leq j \leq n+1,$$

where (u_0, t_0) is a known solution on the solution curve, δs is a continuation step and e_j is the j-th unit vector. For more details the reader is referred to [12] for N^1 and [14] for N^2.

We shall first estimate $\rho(P)$ and $\|P\|$ for these two N's and then apply the results of the last section. In particular, we would like to determine the conditions under which Algorithm ANM converges if S is convergent for G, i.e if $\rho(S_u) < 1$.

First, we need the following elementary result.

Lemma 4.1.

$$\rho(P) = \max(1, |\frac{N_t}{N_t - N_u v}|).$$

Proof. Since $m = 1$, P is a rank one perturbation of I and thus P has $n - 1$ eigenvalues equal to 1 and one eigenvalue equal to $1 + \frac{N_u v}{N_t - N_u v}$.

∎

For N^1, we have $N_u = \dot{u}_0^T$ and $N_t = \dot{t}_0$. As discussed before, if S_u is sufficiently contractive, then †

$$v \approx -u_t = -\frac{\dot{u}}{\dot{t}}.$$

It follows that

$$\rho(P) = \max(1, |\frac{\dot{t}_0 \dot{t}}{\dot{t}_0 \dot{t} + \dot{u}_0^T \dot{u}}|).$$

If $\dot{t}_0 \dot{t}$ has the same sign as $\dot{u}_0^T \dot{u}$, which would be the case if (u_0, t_0) and (u, t) are not on opposite sides of a turning point, then we have $\rho(P) \leq 1$. If $\dot{t}_0 \dot{t}$ and $\dot{u}_0^T \dot{u}$ have opposite signs, then $\rho(P) > 1$.

†The assumption that G_u is nonsingular ensures that $\dot{t} \neq 0$.

But note that the term $\dot{t}_0\dot{t} + \dot{u}_0^T\dot{u}$ is the cosine of the angle θ between the unit tangents at (u,t) and at (u_0,t_0), which is usually kept appreciably above zero by the continuation method. Since $|\dot{t}_0| \le 1$ and $|\dot{t}| \le 1$, we have in this case $\rho(P) \le \frac{1}{\cos\theta}$. In particular, as $\delta s \to 0$, $\theta \to 0$ and hence $\rho(P) \le 1$. Moreover, if S is sufficiently contractive, then v tends to a scalar multiple of N_u and this implies that P tends to being a normal matrix. As for $||P||$, we have

$$P = I - \frac{\dot{u}\dot{u}_0^T}{\dot{t}_0\dot{t} + \dot{u}_0^T\dot{u}}$$

and therefore

$$||P||_p \le 1 + \frac{1}{\cos\theta}, \qquad p = 2, \infty.$$

As $\delta s \to 0$, $||P||_p \le 2$ for $p = 2, \infty$. If P is normal, we have the tighter bound $||P||_2 = \rho(P) \le 1$. Combining the above estimates of $\rho(P)$ and $||P||$ with the results of Section 3, we obtain the following sufficient conditions for the convergence of Algorithm ANM:

Theorem 4.1. *For N^1, as $\delta s \to 0$, Algorithm ANM converges locally if any one of the following conditions holds:*

1. $||S_u||_p < \frac{1}{2}$, *for* $p = 2$ *or* ∞.
2. $||S_u||_2 < 1$, *if P is normal.*
3. $\rho(S_u) < \frac{1}{2}$, *if S_u is normal.*
4. $\rho(S_u) < 1$, *if P and S_u are either both normal or simultaneously diagonalizable.*

For N^2 we have $(N_u, N_t) = e_j^T$. If $j \le n$ then $N_t = 0$ and we have $\rho(P) = 1$. If $j = n+1$ then $N_u = 0$ and we also have $\rho(P) = 1$. Therefore in any case, $\rho(P) = 1$. However, P is not normal unless v is a multiple of e_j. Next we shall estimate $||P||$. First note that if $j = n+1$ then $P = I$ and hence $||P|| = 1$ in any vector induced norm. If $1 \le j \le n$, then

$$P = I - \frac{ve_j^T}{(v)_j},$$

where $(v)_j$ denotes the j-th component of the vector v. In practice, the index j is usually chosen so that $|(v)_j| = \max_{1\le i\le n} |(v)_i|$ and hence $||P||_\infty \le 2$ and $||P||_2 \le 1 + \sqrt{n}$. Combining these estimates of $\rho(P)$ and $||P||$ with the results of Section 3 gives the following sufficient conditions for the convergence of Algorithm ANM:

Theorem 4.2. *For N^2, assuming that the index j is chosen such that $|(v)_j| = \max_{1\le i\le n} |(v)_i|$, Algorithm ANM converges if any one of the following conditions holds:*

1. $||S_u||_\infty < \frac{1}{2}$.
2. $||S_u||_2 < \frac{1}{1+\sqrt{n}}$.
3. $||S_u||_2 < 1$, *if P is normal.*
4. $\rho(S_u) < \frac{1}{1+\sqrt{n}}$, *if S_u is normal.*
5. $\rho(S_u) < 1$, *if P and S_u are either both normal or simultaneously diagonalizable.*

Many of the conditions in Theorems 4.1 and 4.2 are very conservative. For N^1, if S_u is normal and reasonably contractive, then it is most likely that the condition $\rho(S_u) < 1$ is sufficient because P should be close to being normal. For N^2, the estimates for $||P||$ are especially conservative if v

has a particularly large component. In fact, if $\frac{v}{(v)_j}$ is equal to e_j, then P is normal and $||P||_p = 1$ for $p = 2$ or ∞, and the bounds in Theorem 4.2 all become 1. In particular, under these conditions, we have that, for both N^1 and N^2, the condition $\rho(S_u) < 1$ is sufficient for the convergence of Algorithm ANM provided S_u is normal. This is a very satisfactory result because it means that in practice Algorithm ANM converges if S is convergent for G. Therefore it can be applied reliably to a large class of problems with most solvers S for these continuation methods, especially in conjunction with the technique for constructing $\hat{S}$. The algorithm has been successfully applied to a limited number of small continuation problems.

5. Concluding Remarks

In this paper, we have proposed a general algorithm for solving a general class of coupled nonlinear systems. It is especially suitable for problems for which efficient solvers exist for part of the system but not for the whole. The algorithm can be applied in a modular fashion with calls to these solvers and fully exploits the efficiency built into them. The local convergence analysis shows that if the solvers are sufficiently contractive, then the algorithm converges locally. A general technique enables one to construct a modified $\hat{S}$ that ensures convergence. For two important arclength continuation methods, we show that under mild conditions, the algorithm converges if the solver is convergent.

It is known that the Block Elimination Algorithm may be unstable near a solution where G_u is singular [5, 4]. If G_u^{-1} is used explicitly in S, then instability is to be expected for Algorithm ANM as well. However, it may be possible to adapt deflation techniques developed in [5, 6] for the Block Elimination Algorithm to Algorithm ANM. In any case, for problems in which S is well-behaved, Algorithm ANM should not encounter any stability problem.

References

[1] J.P. Abbott, *An Efficient Algorithm for the Determination of Certain Bifurcation Points*, Journal of Computational and Applied Mathematics, 4(1978), pp. 19–27.

[2] E. Allgower and K. Georg, *Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations*, SIAM Review, 22/1(1980), pp. 28–85.

[3] W. Beyn, Defining Equations for Singular Solutions and Numerical Applications, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems*,Birkhauser Verlag, Basel, 1984.

[4] T.F. Chan, Techniques for Large Sparse Systems Arising from Continuation Methods, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems*,Birkhauser Verlag, Basel, 1984.

[5] ———, *Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems*, Siam J. Sci. Stat. Comp., 5/1 March(1984),

[6] ———, *Deflated Decomposition of Solutions of Nearly Singular Systems*, 225, Computer Science Department, Yale Univ., 1982. To appear in Siam J. Numer. Anal., 1984.

[7] R.S. Dembo, S. Eisenstat and T. Steihaug, *Inexact Newton Methods*, SIAM J. Numer. Anal., 18/2(1982), pp. 400–408.

[8] W. Fichtner and D.J. Rose eds., Joint SIAM-IEEE Electron Devices Society Conference, *Numerical Simulation of VLSI Devices*, 1982. Siam J. Sci. Stat. Comp., 4/3 Sept. 1983.

[9] C.B. Garcia and W.I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria*, Prentice-Hall, Englewood Cliffs, N.J., 1981.

[10] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, *Computing Forward-Difference Intervals for Numerical Optimization*, Siam J. Sci. Stat. Comp., 4/2 June(1983), pp. 310–321.

[11] A. Jepson and A. Spence, Singular Points and Their Computations, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems*,Birkhauser Verlag, Basel, 1984.

[12] H.B. Keller, Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems, P. Rabinowitz ed., *Applications of Bifurcation Theory*, pages 359–384, Academic Press, New York, 1977.

[13] G. Moore and A. Spence, *The Calculation of Turning Points of Nonlinear Equations*, SIAM J. Numer. Anal., 17(1980), pp. 567–576.

[14] W.C. Rheinboldt, *Solution Fields of Nonlinear Equations and Continuation Methods*, SIAM J. Numer. Anal., 17(1980), pp. 221–237.

[15] R. Seydel, *Numerical Computation of Branch Points in Nonlinear Equations*, . .mer. Math., 33(1979), pp. 339–352.

Imprimé en France

par

l' Institut National de Recherche en Informatique et en Automatique