We present an efficient modular algorithm for solving the coupled nonlinear system $G(u, t) = 0$ and $N(u, t) = 0$ where $u \in R^n$, $t \in R^m$, $G : R^n \times R^m \mapsto R^n$ and $N : R^n \times R^m \mapsto R^m$. The algorithm is modular in the sense that it only makes use of 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 more efficiently than a solver for the coupled system. Local convergence results are given. Basically, if S is sufficiently contractive for G, then convergence for the coupled system is guaranteed. The algorithm is applied to two applications: (1) numerical continuation methods and (2) constrained optimization. Numerical results are given for the case where G represents a nonlinear elliptic operator. Three choices of S are considered: Newton's method, a two-level nonlinear multi-grid solver and a supported Picard iteration.

An Efficient Modular Algorithm for Coupled Nonlinear Systems

Tony F. Chan

Research Report YALEU/DCS/RR-328
September 1984

This work was supported in part by the Department of Energy under contract DE-AC02-81ER10996, and by the Army Research Office under contract DAAG-83-0177.

Keywords: *Coupled nonlinear systems, Newton's method, continuation methods, constrained optimization, multi-grid methods, supported Picard iteration.*

This paper was presented as an invited talk at the Fourth IIMAS Workshop on Numerical Analysis held on July 23-27, 1984 at Guanajuato, Mexico.

1. Introduction

This paper is concerned with numerical algorithms for solving coupled nonlinear systems of the form:

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

where $u \in R^n$, $t \in R^m$, $G : R^n \times R^m \mapsto R^n$ and $N : R^n \times R^m \mapsto R^m$. We assume that a solution z^* exists and that the Jacobian

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

is nonsingular at z^*. We consider two applications in this paper. One is to continuation methods where G represents a nonlinear system in u with dependence on some parameters t and N represents an arclength condition constructed to follow the solution manifolds. Another application is to constrained optimization problems, where G represents the Lagrangian function, t the Lagrange multipliers and N the constraints.

In this paper, we are primarily interested in the case where G is large, sparse and structured, such as discretizations of partial differential equations. Very often, efficient techniques exist for exploiting such structures when one solves $G(u,t) = 0$ for u with t fixed. If $m \ll n$, it is desirable to be able to solve $C(z) = 0$ with about the same efficiency. However, while many conventional iterative algorithms can be applied directly to $C(z) = 0$ to solve for z^*, they often fail to exploit structures in G. Consider the use of Newton's method, for example. The Jacobian J of $C(z)$ does not necessarily inherit the following properties which the Jacobian G_u of G may possess: symmetry, positive definiteness, separability and bandedness [4].

In this paper, we present an algorithm for solving the coupled system which makes use of a general solver for $G = 0$, *for fixed t*. We 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 $u_{i+1} = S(u_i, t)$. For example, for Newton's method we have $S^{Newton} = u - G_u^{-1}(u,t)G(u,t)$. The algorithm is not restricted to Newton's method, however, and in general any sufficiently convergent solver for G can be used. Such an algorithm can therefore exploit special efficient solvers specifically designed for solving $G = 0$.

In Section 2, we present the algorithm and some convergence results. In Section 3, we discuss applications to numerical continuation methods and constrained optimization. In Section 4, numerical results are presented for the case where G is an elliptic operator and for three choices of S: Newton's method, a nonlinear multi-grid method and a supported Picard iteration.

2. The Algorithm

Our algorithm is motivated by Newton's method applied to $C(z) = 0$. At each iteration, 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, often by the following block Gaussian elimination algorithm.

Algorithm BE:

```
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 $\delta t$.
4. Compute $\delta u = w - v\delta t$.
```

The idea in the new algorithm is to use S to approximately solve for w and v in Steps 1 and 2 in Algorithm BE. 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 $S(u_i, t_i) - u_i$, 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 $G(u,t) = 0$. In [3], it is argued that it is reasonable to approximate v by $-S_t$. In particular, if $S = S^{Newton}$ then this approximation is exact. In practice, we can approximate S_t by a difference approximation. We therefore arrive at 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 $u_{i+1} = u_i + w - vd$.

In [3], it is shown that if S is sufficient contractive, then the matrix in Step 3 of Algorithm ANM is nonsingular and thus the algorithm is well-defined.

The convergence of Algorithm ANM is analyzed in [3]. If we ignore the truncation error in the finite difference approximation to v in Step 2 of the algorithm, we have the following local convergence result:

Theorem 2.1. *Algorithm ANM converges (locally) iff $\rho(PS_u) < 1$, where $P \equiv I + v(N_t - N_u v)^{-1} N_u$.*

The following sufficient conditions for convergence follows immediately:

Theorem 2.2. *Algorithm ANM converges if $||S_u|| < \frac{1}{||P||}$, in any vector induced norm.*

Theorem 2.3. *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 $\rho(S_u) < \frac{1}{\rho(P)}$.*

If for a particular S, S_u does not satisfy any of these bounds, then we can define a modified iteration operator :

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

i.e. iterate S k times. Since $\hat{S}_u = S^k_u$,, we can choose k large enough to make $\rho(\hat{S})$ or $||\hat{S}_u||$ small enough for convergence.

3. Applications

3.1. Continuation Methods

Here G represents a system of parameterized nonlinear equations, with u playing the role of the main variable and t the parameters. The goal is to trace solution manifolds of $G = 0$. This is often done by freezing all parameters except one (and therefore we restrict our attention to the case $m = 1$) and curve-following continuation methods [1, 5, 7, 8] are used to trace the branches corresponding to this one parameter. Usually, a predictor-corrector method is used. A predicted value is generated from a known solution (u, t) and the local unit tangent $(\dot{u}, \dot{t})$ defined by:

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

The corrector is defined to be the solution of a coupled nonlinear system:

$$\begin{aligned} G(u(\delta s), \lambda(\delta s)) &= 0 \\ N(u(\delta s), \lambda(\delta s)) &= 0, \end{aligned}$$

where N defines a local parameterization of the solution curve with parameter δs. Two typical N's that are widely used in the literature are:

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

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 [7] for N^1 and [8] for N^2.

Algorithm ANM is especially well-suited for this application. First, for large problems, the solution of the coupled nonlinear system often constitutes the most costly part of the overall continuation process and Algorithm ANM does this efficiently by making it possible to exploit structures in G. A second advantage is that Algorithm ANM allows the continuation procedure itself (such as the step length control, the predictor, the tangent computation) to be separate from the specific solver for G, making it much easier for general purpose continuation codes [9] to be applied efficiently to application areas with specialized solvers (e.g. the Navier-Stokes equations).

Since the functions N^1 and N^2 are related to G itself, the convergence results of the last section can be refined further [3].

Theorem 3.1. *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.*

Theorem 3.2. *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.*

These sufficient conditions are very conservative in general and it is argued in [3] that in practice Algorithm ANM is convergent *whenever S is convergent for G.*

3.2. Constrained Optimatization

Another application of Algorithm ANM can be found in equality constrained optimization. Consider the problem:

$$\min F(u)$$
$$\text{subject to} \quad N_i(u) = 0 \qquad i = 1, ..., m.$$

Define the Lagrangian $L(u,t) \equiv F(u) + \sum_{i=1}^m t_i N_i(u)$. The first order condition for a minimum is:

$$G(u,t) \equiv \nabla L(u,t) = \nabla F(u) + \sum_{i=1}^{m} t_i \nabla N_i(u) = 0$$

and

$$N(u,t) \equiv \left\{ \begin{matrix} N_1(u) \\ \vdots \\ N_m(u) \end{matrix} \right\} = 0,$$

which is in the form of a coupled nonlinear system. Algorithm ANM is well-suited for this problem if $m \ll n$ (i.e. relatively few constraints) and an efficient method is available for solving the unconstrained problem $\nabla F(u) = 0$. For linear constraints, the second term in G is constant and such a solver can easily be adapted to define an efficient S for solving $G(u,t) = 0$ for fixed t. Nonlinear constraints are more difficult to handle and we shall not dwell on that here. Thus, Algorithm ANM allows the addition of constraints to an unconstrained problem to be treated in a very efficient manner.

4. Numerical Results

4.1. Continuation Methods

We have applied Algorithm ANM to solve the following parameterized nonlinear equation:

$$u_{xx} + te^u = 0, \qquad 0 \leq x \leq 1 \tag{4.1}$$

with the boundary conditions

$$u(0) = u(1) = 0,$$

by the pseudo arclength continuation method using the parameterization N^1. This problem has one simple turning point at $(||u||_\infty \approx 1.3, t \approx 3.5)$, separating two branches of solution. Problem (4.1) is discretized on a uniform mesh with n interior grid points by a standard second order centered difference approximation. The resulting discrete system of nonlinear equations $G(u,t) = 0$ has dimension n.

We present numerical results for three solvers S. The first is simply S^{Newton}. The second is a two level full-approximation-scheme multi-grid method for solving (4.1) [2]. Briefly, two nonlinear Gauss-Seidel smoothing sweeps are used before the correction problem is injected onto and solved on a coarser grid (one with $\frac{n-1}{2}$ grid points, n odd), after which the correction is interpolated (linearly) and added to the solution on the original grid which is then smoothed again with two more smoothing sweeps. In our implementation, the coarse grid problem is solved by 4 iterations of Newton's method. The two-level multi-grid algorithm is representative of many multi-grid solvers in terms of convergence properties and extends to higher dimensional problems in a straightforward way. The third solver implements a supported Picard iteration [6]. The system $G = 0$ for (4.1) can be written as $Au = F(u,t)$, which naturally suggests the following Picard iteration:

$$u^{i+1} \leftarrow A^{-1}F(u^i,t).$$

Structures in A can be exploited, for example, a fast elliptic solver can be used for A^{-1}. Unfortunately, this iteration is convergent if and only if $\rho(A^{-1}F_u) < 1$ and this does not necessarily hold for a given problem. In many applications, however, there are only a few divergent eigenvalues of $A^{-1}F_u$. The main idea in the supported Picard iteration is to apply Newton's method in the subspace spanned by the eigenvectors of $A^{-1}F_u$ corresponding to eigenvalues with magnitude exceeding 1. This can be implemented in an efficient manner because the dimension of this subspace is small. This method is ideally suited for our problem because there is only one divergent eigenvalue on the upper solution branch and therefore the Newton's method is effectively applied to a scalar problem. For more details on the implementation, the reader is referred to [6].

To illustrate the convergence behavior of Algorithm ANM on this problem, we have chosen to show the iterations for two points on the solution branch, one on the lower branch and the other on the upper branch, both with $t_0 = 3$. An initial guess to the solution is obtained by an Euler predictor based on the unit tangent $(\dot{u}_0, \dot{t}_0)$ and the finite difference interval ϵ is chosen to be 0.0001. The iteration is stopped when $\max\{||\delta z||_\infty, ||G||_\infty, |N|\} \leq 10^{-5}$. The results for the case $n = 31$ are tabulated in Tables 1 and 2. It can be seen that the convergence is quite satisfactory for all the solvers.

4.2. Constrained Optimization

We have also applied Algorithm ANM to the following minimization problem:

$$\textit{minimize} \qquad F(u) \equiv u^T Au + \lambda \sum_{i=1}^{n} e^{u_i}$$

$$\textit{subject to} \qquad \sum_{i=1}^{n} l_i u_i = 0$$

where A is the discrete Laplacian in one dimension with zero Dirichlet boundary condition.

Note that the first order condition for the unconstrained problem is exactly the discretization of (4.1) and since the constraint is linear, any one of the three solvers considered so far can easily be adapted to construct a solver for $\nabla L(u,t) = 0$ for fixed multipliers t.

We shall present numerical results for the case $\lambda = 3$ and $n = 31$ which corresponds to the results given for the continuation problem. Only the results for S^{Newton} and S^{Picard} will be shown. The unconstrained solution is as shown in Figure 1. The iterations for three different constraints are shown in Tables 3 - 5 and the corresponding solutions are shown in Figures 2-4. The convergence is rapid for all cases.

iter	$\|\|u\|\|_\infty$	t	$\|\|\delta z\|\|_\infty$	$\|\|G\|\|_\infty$	$\|N\|$
Newton					
0	0.7283397E+00	0.3190261E+01	0.0E+00	0.5E−04	0.7E−07
1	0.7308150E+00	0.3173259E+01	0.2E−01	0.2E−04	0.3E−07
2	0.7308277E+00	0.3173151E+01	0.1E−03	0.2E−06	0.5E−07
3	0.7308277E+00	0.3173151E+01	0.1E−06	0.1E−06	0.4E−07
Multi-Grid					
0	0.7283397E+00	0.3190261E+01	0.0E+00	0.5E−04	0.7E−07
1	0.7308369E+00	0.3173014E+01	0.2E−01	0.2E−04	0.2E−07
2	0.7307944E+00	0.3173375E+01	0.4E−03	0.2E−05	0.7E−07
3	0.7308255E+00	0.3173168E+01	0.2E−03	0.3E−06	0.1E−07
4	0.7308269E+00	0.3173157E+01	0.1E−04	0.1E−06	0.3E−07
5	0.7308279E+00	0.3173148E+01	0.8E−05	0.9E−07	0.3E−07
Picard					
0	0.7283341E+00	0.3190261E+01	0.0E+00	0.5E−04	0.6E−07
1	0.7308021E+00	0.3173283E+01	0.2E−01	0.6E−05	0.7E−07
2	0.7308219E+00	0.3173152E+01	0.1E−03	0.2E−06	0.7E−07
3	0.7308219E+00	0.3173152E+01	0.3E−06	0.2E−06	0.3E−07

Table 1: $||u_0||_\infty = 0.641, t_0 = 3.0, \delta s = 0.4$

iter	$\|\|u\|\|_\infty$	t	$\|\|\delta z\|\|_\infty$	$\|\|G\|\|_\infty$	$\|N\|$
Newton					
0	0.2075063E+01	0.2895127E+01	0.0E+00	0.3E−04	0.7E−07
1	0.2075096E+01	0.2893029E+01	0.2E−02	0.4E−05	0.1E−06
2	0.2075096E+01	0.2893032E+01	0.3E−05	0.4E−06	0.1E−06
Multi-Grid					
0	0.2075063E+01	0.2895127E+01	0.0E+00	0.3E−04	0.7E−07
1	0.2075097E+01	0.2893288E+01	0.2E−02	0.5E−05	0.4E−07
2	0.2075093E+01	0.2892977E+01	0.3E−03	0.1E−05	0.2E−07
3	0.2075096E+01	0.2893027E+01	0.5E−04	0.3E−06	0.4E−07
4	0.2075096E+01	0.2893031E+01	0.4E−05	0.4E−06	0.2E−07
Picard					
0	0.2075073E+01	0.2895126E+01	0.0E+00	0.3E−04	0.5E−07
1	0.2075105E+01	0.2893017E+01	0.2E−02	0.4E−05	0.6E−07
2	0.2075107E+01	0.2893031E+01	0.1E−04	0.4E−06	0.5E−07
3	0.2075107E+01	0.2893031E+01	0.3E−06	0.6E−06	0.1E−06

Table 2: $||u_0||_\infty = 1.973, t_0 = 3.0, \delta s = 0.4$

iter	$\|\|u\|\|_\infty$	t	$\|\|\delta z\|\|_\infty$	$\|\|G\|\|_\infty$	$\|N\|$
Newton					
0	$0.9987573E-01$	$0.0000000E+00$	$0.0E+00$	$0.1E+00$	$0.7E-01$
1	$0.1170606E+00$	$-0.5526059E+02$	$0.6E+02$	$0.3E-01$	$0.7E-08$
2	$0.1020966E+00$	$-0.5137571E+02$	$0.4E+01$	$0.9E-05$	$0.0E+00$
3	$0.1021055E+00$	$-0.5138050E+02$	$0.5E-02$	$0.1E-07$	$0.0E+00$
4	$0.1021055E+00$	$-0.5138050E+02$	$0.9E-07$	$0.1E-07$	$0.0E+00$
Picard					
0	$0.9987573E-01$	$0.0000000E+00$	$0.0E+00$	$0.1E+00$	$0.7E-01$
1	$0.1368882E+00$	$-0.5531568E+02$	$0.6E+02$	$0.7E-01$	$0.7E-08$
2	$0.1020990E+00$	$-0.5137750E+02$	$0.4E+01$	$0.1E-04$	$0.0E+00$
3	$0.1021054E+00$	$-0.5138047E+02$	$0.3E-02$	$0.2E-07$	$0.0E+00$
4	$0.1021054E+00$	$-0.5138047E+02$	$0.4E-06$	$0.2E-07$	$0.0E+00$

Table 3: Constraint: $u(16) = 0.0$

iter	$\|\|u\|\|_\infty$	t	$\|\|\delta z\|\|_\infty$	$\|\|G\|\|_\infty$	$\|N\|$
Newton					
0	$0.9987573E-01$	$0.0000000E+00$	$0.0E+00$	$0.1E+00$	$0.2E-01$
1	$0.1418061E+00$	$-0.5032611E+02$	$0.5E+02$	$0.3E-01$	$0.4E-08$
2	$0.1315020E+00$	$-0.4681818E+02$	$0.4E+01$	$0.1E-04$	$0.0E+00$
3	$0.1315103E+00$	$-0.4682352E+02$	$0.5E-02$	$0.4E-07$	$0.0E+00$
4	$0.1315104E+00$	$-0.4682352E+02$	$0.1E-05$	$0.2E-07$	$0.0E+00$
Picard					
0	$0.9987573E-01$	$0.0000000E+00$	$0.0E+00$	$0.1E+00$	$0.2E-01$
1	$0.1501235E+00$	$-0.5038120E+02$	$0.5E+02$	$0.6E-01$	$0.4E-08$
2	$0.1315220E+00$	$-0.4682264E+02$	$0.4E+01$	$0.2E-04$	$0.0E+00$
3	$0.1315103E+00$	$-0.4682348E+02$	$0.8E-03$	$0.2E-07$	$0.0E+00$
4	$0.1315103E+00$	$-0.4682349E+02$	$0.7E-05$	$0.2E-07$	$0.0E+00$

Table 4: Constraint: $u(16) = 0.05$

iter	$\|\|u\|\|_\infty$	t	$\|\|\delta z\|\|_\infty$	$\|\|G\|\|_\infty$	$\|N\|$
Newton					
0	$0.9987573E-01$	$0.0000000E+00$	$0.0E+00$	$0.1E+00$	$0.3E+00$
1	$0.2130776E+00$	$-0.1571455E+02$	$0.2E+02$	$0.2E-01$	$0.0E+00$
2	$0.2104818E+00$	$-0.1573100E+02$	$0.2E-01$	$0.4E-06$	$0.0E+00$
3	$0.2104816E+00$	$-0.1573133E+02$	$0.3E-03$	$0.3E-07$	$0.1E-07$
4	$0.2104816E+00$	$-0.1573133E+02$	$0.2E-05$	$0.3E-07$	$0.1E-07$
Picard					
0	$0.9987573E-01$	$0.0000000E+00$	$0.0E+00$	$0.1E+00$	$0.3E+00$
1	$0.2200581E+00$	$-0.1502713E+02$	$0.2E+02$	$0.2E-01$	$0.1E-06$
2	$0.2104825E+00$	$-0.1573232E+02$	$0.7E+00$	$0.2E-04$	$0.0E+00$
3	$0.2104816E+00$	$-0.1573128E+02$	$0.1E-02$	$0.6E-07$	$0.0E+00$
4	$0.2104815E+00$	$-0.1573128E+02$	$0.2E-05$	$0.3E-07$	$0.0E+00$

Table 5: Constraint: $u(8)+u(16)+u(24)=0.5$

Figure 1: Unconstrained Solution

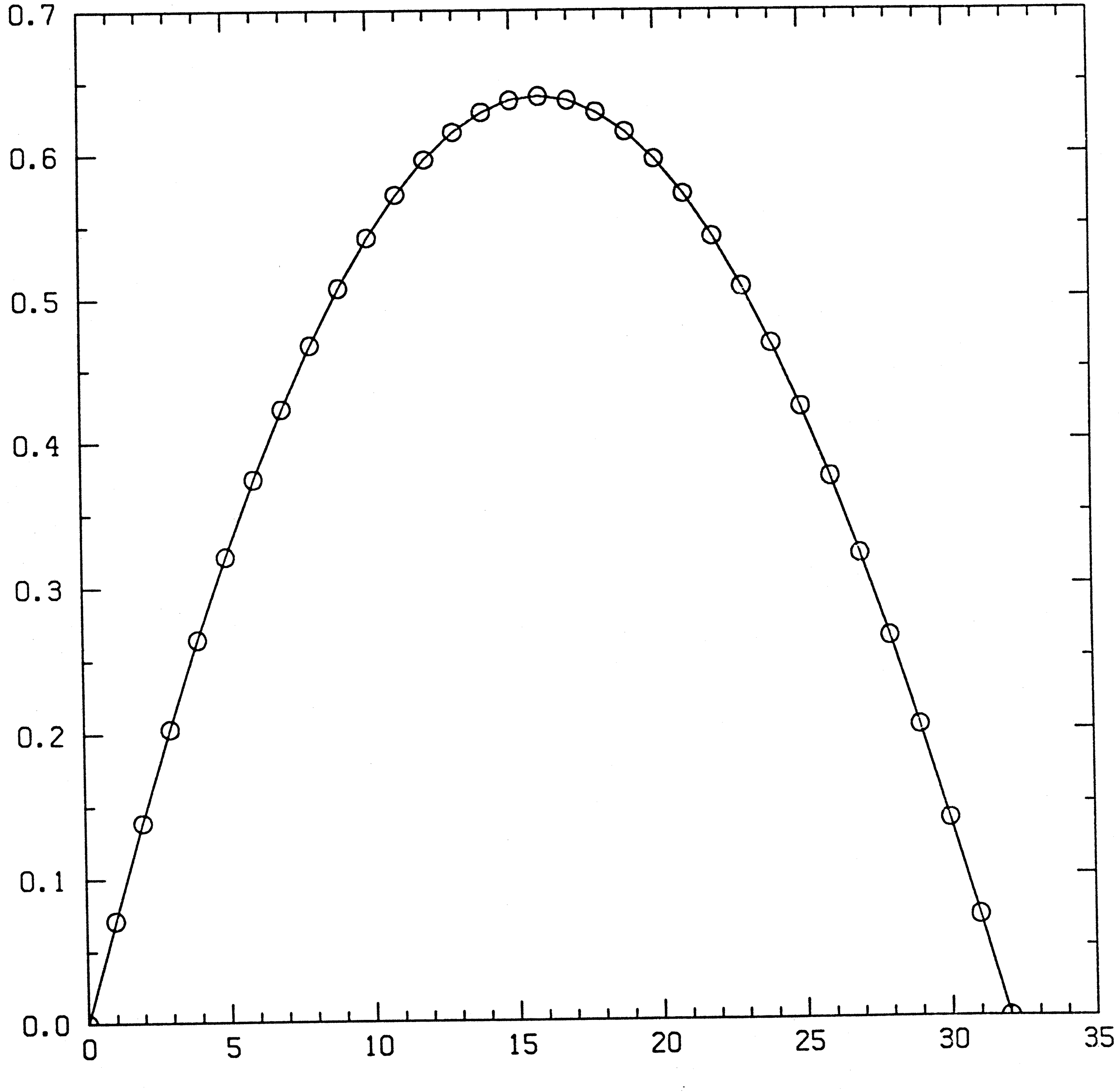


Figure 2: Constraint: $u(16) = 0.0$

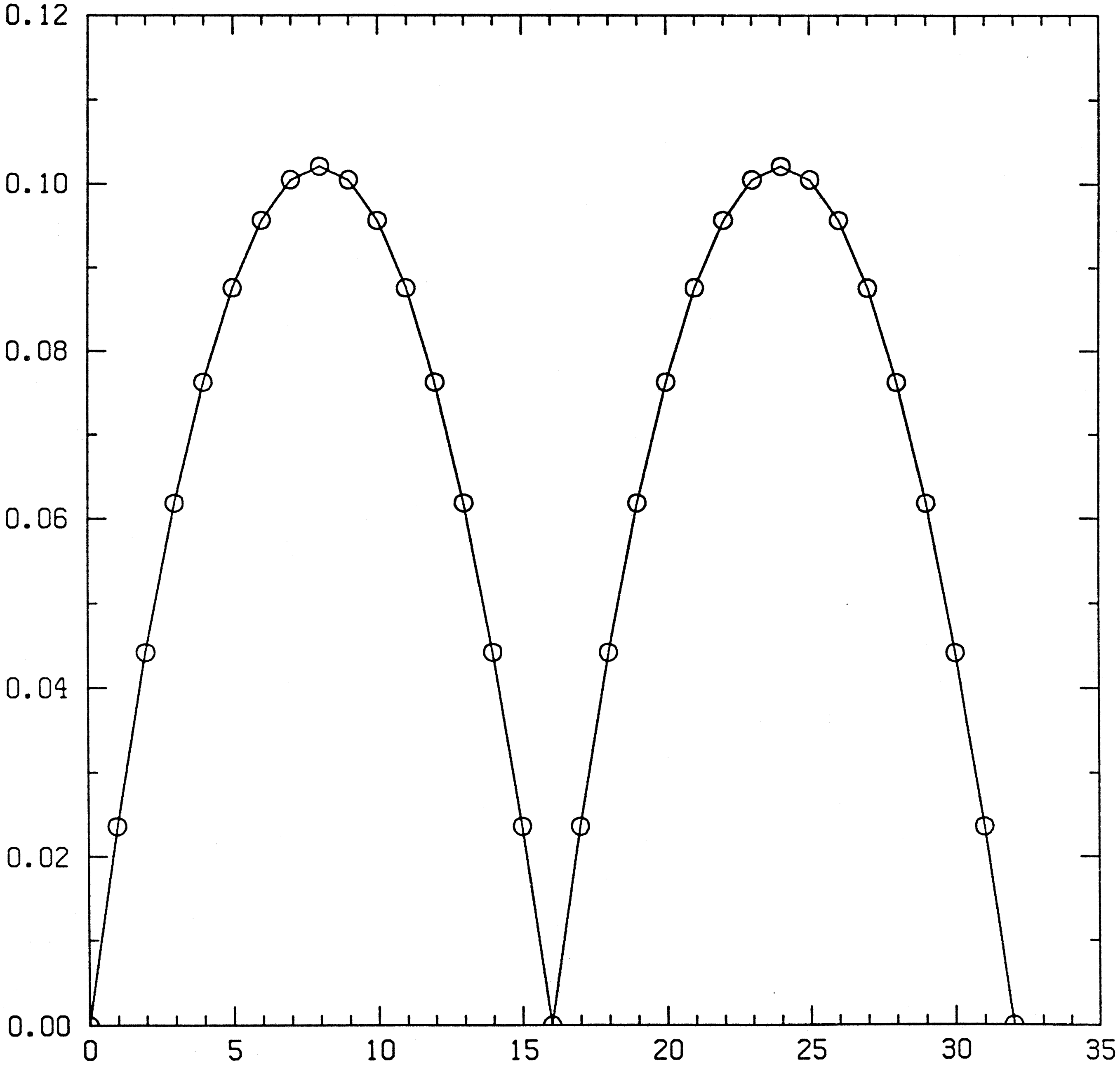


Figure 3: Constraint: $u(16) = 0.05$

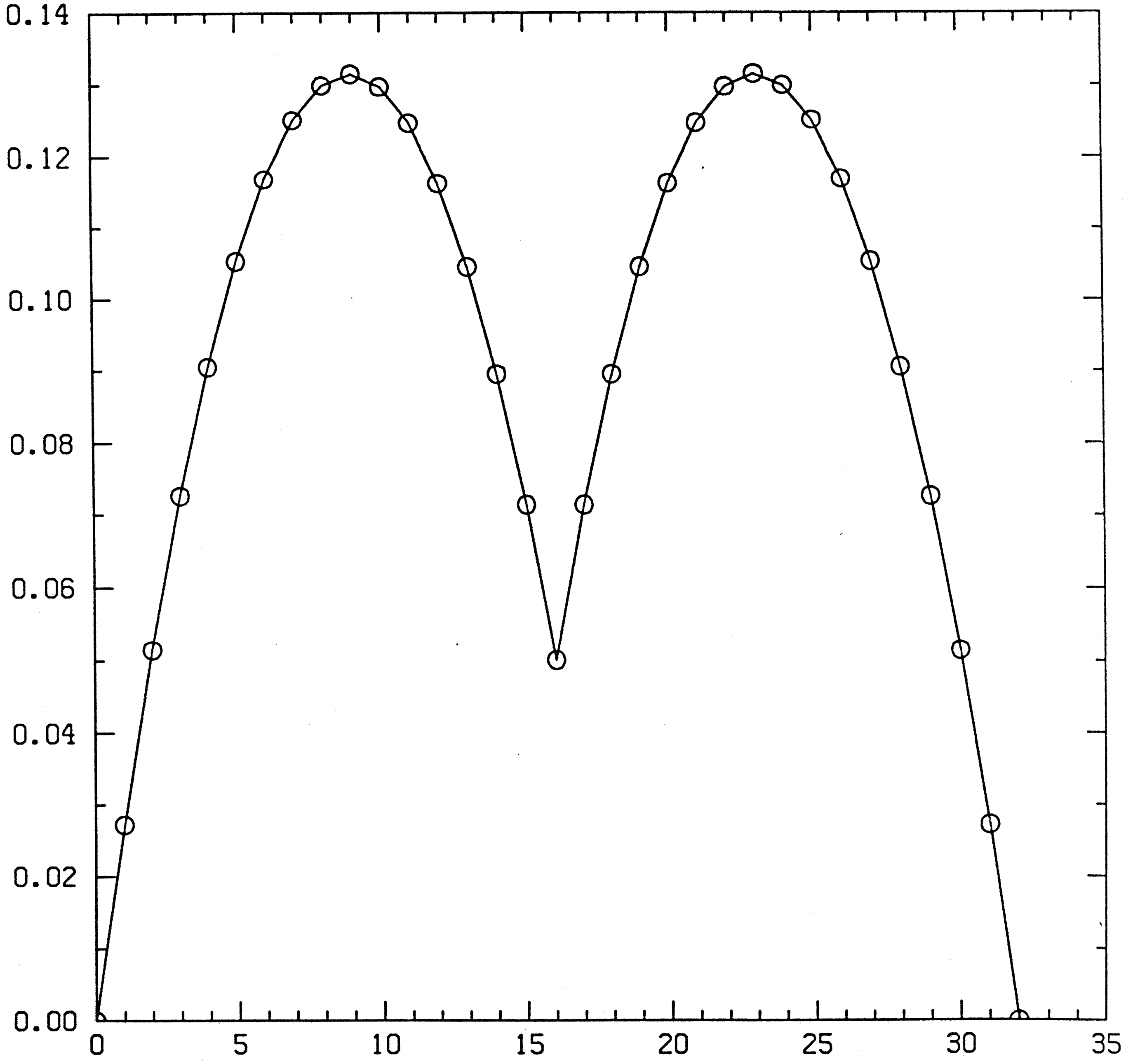
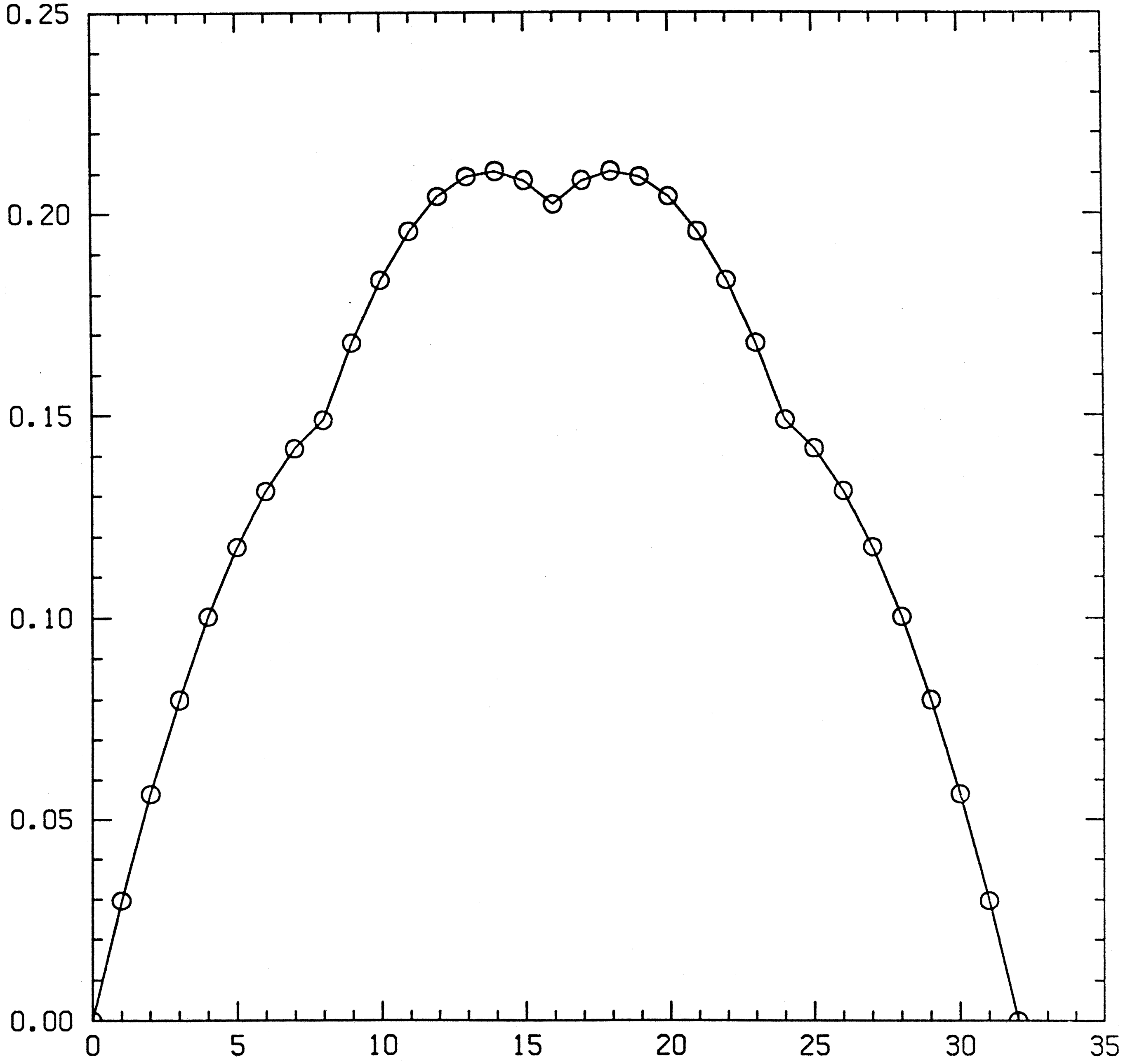


Figure 4: Constraint: $u(8) + u(16) + u(24) = 0.5$



References

[1] 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.

[2] A. Brandt, *Multi-level Adaptive Solution to Boundary Value Problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] T.F. Chan, *An Approximate Newton Method for Coupled Nonlinear Systems*, Technical Report Research Report YALEU/DCS/RR-300, Computer Science Department, Yale Univ., February 1984. Submitted to Siam J. Numer. Anal..

[4] ————, Techniques for Large Sparse Systems Arising from Continuation Methods, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems*, International Series of Numerical Math., Vol. 70, Birkhauser Verlag, Basel, 1984, pp. 116–128.

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

[6] H. Jarausch and W. Mackens, *CNSP A Fast, Globally Convergent Scheme to Compute Stationary Points of Elliptic Variational Problems*, Technical Report 15, Institute for Geometrie und Praktische Mathematik, Rheinisch-Westfalische Technische Hochschule, Aachen, 1982.

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

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

[9] W.C. Rheinboldt and J.V.Burkardt, *A Locally Parameterized Continuation Process*, ACM Trans. Math. Soft., 9/2 June(1983), pp. 215–235.