Finding A Stable Solution of A System of Nonlinear Equations *

C.T. Kelley† Liqun Qi‡ Xiaojiao Tong§ Hongxia Yin¶

July 30, 2006

Abstract

This paper presents new methods for finding dynamically stable solutions of systems of nonlinear equations. The concepts of stability functions and the so-called stable solutions are defined. Based on those new concepts, two models of stable solutions and three stability functions are proposed. These stability functions are semismooth. Smoothing technology is applied to such stability functions. Smoothing Newton methods are proposed to solve the stable solution models. Convergence properties of these methods are studied. We report on numerical examples which illustrate the utility of the new approach.

Key words. System of Nonlinear Equations; Stable Solutions; Saddle-Node Bifurcation; Hopf Bifurcation; Stability Functions; Smoothing Newton Method.

AMS subject classifications(2000): 65H10, 65K10,

1 Introduction

Consider the following system of nonlinear equations:

$$F(x) = 0, \tag{1.1}$$

*This work is supported by US National Science Foundation grant DMS-0404537, US Army Research Office grants W911NF-04-1-0276 and W911NF-05-1-0171, the Hong Kong Research Grant Council and Natural Science Foundation of China (NSF60474070, NSF70472074 and KYQD200502).

†Department of Mathematics, Center for Research in Scientific Computation, North Carolina State University, Raleigh, North Carolina, USA. E-mail: tim_kelley@ncsu.edu.

‡Department of Mathematics, City University of Hong Kong, Kowloon Tong, Kowloon, Hong Kong. E-mail: maqilq@cityu.edu.hk.

§Institute of Mathematics, Changsha University of Science and Technology, Changsha, Hunan, 410076, China. E-mail: tongxj@csust.edu.cn.

¶Chinese Academy of Sciences Research Center on Data Technology and Knowledge Economy, Department of Mathematics, Graduate University of the Chinese Academy of Sciences, Beijing, 100049, P.R. China. Email: hxyin@gucas.ac.cn.

where $F : \Re^n \to \Re^n$ is a twice continuously differentiable function. Such a system arises from various applications. There are many methods for solving (1.1) [22, 7, 14, 13], most of which aim to find one solution of (1.1). However, such a solution may not be usable for applications. In recent years, some methods have been developed for finding all solutions of (1.1), in particular, in the case that (1.1) is a system of polynomial equations. In general, such methods can only work for small n. When n is not too small, say, bigger than 20, such methods cannot work as in general the number of solutions of (1.1) increases exponentially as n increases. It will be beyond the current computational capacity to find all solutions of (1.1) in such a size in a short time in general. The purpose of this paper is to propose and analyze methods which exclude dynamically unstable solutions, but which do not attempt to find all solutions, or to perform a potentially expensive continuation.

A typical example is the electric power system. The operations of power systems are described by power flow equations in the form of (1.1). The number of variables, n, is large in general. A very important topic in power engineering is to avoid saddle node bifurcation solutions or Hopf bifurcation solutions of (1.1) [1, 2, 6, 8, 9, 11, 16, 20], and solutions near such bifurcation points. If a power system is operating at or near such a bifurcation point, then catastrophic events such as power system blackouts may occur. Such an issue also arises from electronic systems in the design of power supplies [19, 32].

We now introduce a function called **stability function** $h : \Re^n \to \Re^m_+$. In the simple case, $m = 1$. When $h_i(x)$ is equal to or near 0 for one i, we reject x, even if $F(x) = 0$. In this way we can steer the iteration away from unstable solutions, and even include criteria which are specific to the appplication. In the later sections, we will discuss such stability functions. We will see that h is in general a composition function of a minimum or a maximum operation with the eigenvalue or single value functions of matrix functions associated with the Jacobian of F at x. Such functions are in general nonsmooth, possibly semismooth.

Then, we may formulate such a problem (called the stable solution problem) as

$$\begin{cases} F(x) = 0, \\ h(x) \geq \delta, \end{cases} \tag{1.2}$$

where $\delta > 0$ is the vector of tolerance in $\Re^m$.

On the other hand, the system of unstable solutions can be described by

$$\begin{cases} F(x) = 0, \\ h(x) = 0. \end{cases} \tag{1.3}$$

We use one typical example to understand the effect of (1.2). Denote the eigenvalues of $\nabla F(x)$ by $\lambda_i(\nabla F(x))$, with $|\lambda_i(\nabla F(x))| \geq |\lambda_j(\nabla F(x))|$ for $1 \leq i < j \leq n$. If there exists an i such that

$|\lambda_i(\nabla F(x))| = 0$, then x is a saddle-node bifurcation point of the nonlinear system of equations (1.1). So we can define

$$h(x) = \min_{1 \le i \le n} |\lambda_i(\nabla F(x))|$$

and use the system (1.2) to avoid such a saddle-node bifurcation point.

Next we will estimate the distance between the solution satisfying (1.2) and the unstable solution set.

Proposition 1.1 *Suppose that $h : R^n \to R^m$ is Lipschitz continuous with a Lipschitz constant $L > 0$. Let x^* be a stable solution of (1.1) in the sense of (1.2). Denote the unstable solution set as*

$$\Omega = \{x \mid F(x) = 0, h(x) = 0\}.$$

Then it holds

$$\mathrm{dist(x^*, \Omega)} \ge \frac{\|\delta\|}{\mathrm{L}}. \tag{1.4}$$

Proof. For any $x \in \Omega$, from the condition of Lipschitz continuity and the solution of (1.2) x^* it holds

$$\|\delta\| \le \|h(x^*)\| = \|h(x) - h(x^*)\| \le L\|x - x^*\|.$$

This follows that

$$\|x - x^*\| \ge \frac{\|\delta\|}{L}.$$

Then from the definition of the distance of a point to a closed set, we have

$$\mathrm{dist}(x^*, \Omega) = \min_{x \in \Omega} \|x - x^*\| \ge \frac{\|\delta\|}{L}.$$

We complete the proof. □

In order to present some effective approaches for solving (1.2), we can reformulate the system (1.2) as the following equivalent system with variable $z = (x, y),\ x \in R^n,\ y \in R^m$:

Model I: Unconstrained equations

$$\begin{cases} F(x) = 0, \\ h_i(x) - |y_i| = \delta_i, \quad (i = 1, \cdots, m) \end{cases} \tag{1.5}$$

In fact, we can construct other equivalent unconstrained equations such as

$$\begin{cases} F(x) = 0, \\ h_i(x) - \delta_i(1 + y_i^2) = 0, \quad (i = 1, \cdots, m) \end{cases}$$

Another possible model is as follows.

Model II: Bounded constrained equations

$$\begin{cases} F(x) = 0, \\ h(x) - y = \delta, \\ y \geq 0. \end{cases} \tag{1.6}$$

For the above two systems, later we can see that h is in general nonsmooth, as it needs to involve eigenvalue and maximum operations. If h is semismooth or even strongly semismooth, we may use some semismooth Newton methods or smoothing Newton methods to solve (1.5) and (1.6). There are plentiful papers related to the approach of semismooth equations (see [5, 10, 12, 28, 31, 30]). Most of them enjoy nice global and local convergence.

The remaining sections are distributed as follows. In Section 2, we propose three stability functions. In Section 3, we discuss the semismooth properties of these three stability functions. We investigate smoothing functions of such stability functions and their properties in Section 4. In Section 5, we present a smoothing Newton method to find a stable solution for the model of unconstrained equations. We display numerical examples to illustrate our models in Section 6 and make some final comments in Section 7.

2 Stability Functions

In this section, we propose some stability functions for (1.1). We only consider stability functions in the form $h : \Re^n \to \Re_+$, i.e., $m = 1$ in the context of the introduction.

According to the requirements of practical applications and in order to establish effective numerical methods, our requirements for stability functions are as follows:

- the solution x is not a saddle node bifurcation point or close to such a point. If $\nabla_x F(x)$ is singular, then x is called a saddle node bifurcation point [3, 8, 9]. If $\nabla_x F(x)$ is not near singular, then x is not close to a saddle node bifurcation point.

- the solution x is not a Hopf bifurcation point or near such a point. If $\nabla_x F(x)$ has a pair of conjugate eigenvalues passing the imaginary axis while the other eigenvalues have negative real parts, then x is called a Hopf bifurcation point [8, 11, 20].

- the solution is stable in the sense that the eigenvalues of $\nabla F(x)$ has negative real part. This is the classically stable concept in nonlinear equations;

- the stable function proposed will be local Lipschitz at least so that we can construct some effective approach to solve the system (1.2).

Recall that we have assumed that F is twice continuously differentiable. We can satisfy the first and second requirements above if we set in (1.2) that

$$h_0(x) = \min_{1 \le i \le n} |Re \lambda_i(\nabla F(x))|. \tag{2.1}$$

Here $\lambda_i(A)$ is the ith eigenvalue of A, repeated for algebraic multiplicity. The difficulty with such a stability function is that h_0 is not Lipschitz continuous, then methods based on semismoothness or Lipschitz continuity cannot be applied.

2.1 Special Case I

If we only wish to impose a nonsingularity constraint on ∇F, we can formulate the problem differently. Let σ_i denote the ith singular value. Define

$$h_1(x) = \min_{1 \le i \le n} \sigma_i^2(\nabla F(x)) = \min_{1 \le i \le n} \lambda_i(\nabla F(x)^T \nabla F(x)). \tag{2.2}$$

We may reformulate (1.2) as

$$\begin{cases} F(x) = 0, \\ h_1(x) \ge \delta. \end{cases} \tag{2.3}$$

Note that (2.3) misses the Hopf bifurcation constraints.

2.2 Special Case II

Here wish to find a stable solution in the classical sense that all the eigenvalues of $\nabla F(x)$ are in the left half plane. To make this possible and consider the requirements for stability functions, we use the Cayley transform to construct a stability function.

Let $\sigma > 0$ and let A be an $n \times n$ matrix. Define

$$C(A) = (A - \sigma I)^{-1}(A + \sigma I). \tag{2.4}$$

C is called the Cayley transform of A. If n is not too large, it is possible to actually compute C. If n is large, the action of C on a vector can be computed with a matrix-vector product and a linear solvent.

Let $\{\lambda\}$ be the eigenvalues of A. The eigenvalues of C are $\{\mu\}$, where

$$\mu = \frac{\lambda + \sigma}{\lambda - \sigma}. \tag{2.5}$$

Now, let $\lambda = x + jy$, where $x < 0$. Then

$$\mu = \frac{x + \sigma + jy}{x - \sigma + jy}$$

implies that $|\mu| < 1$. If, on the other hand $x > 0$, then $|\mu| > 1$.

So we have a stable solution if the spectral radius of $C(\nabla F)$, $\rho(C(\nabla F)) < 1$. However, we note that the spectral radius of $C(\nabla F)$ is non-Lipschitz, hence we work with the norm.

To do this we first pick $\sigma > 0$ so that $\nabla F - \sigma I$ is nonsingular at the initial iterate x_0. Then define

$$h_2(x) = 1 - \|C(\nabla F(x))\|_1 \tag{2.6}$$

where $\|\cdot\|_1$ is the ℓ^1 matrix norm.

So, one formulation for (1.2) is

$$\begin{cases} F(x) = 0, \\ h_2(x) \geq \delta. \end{cases} \tag{2.7}$$

We can replace $C(\nabla F)$ with $C(\nabla F)^p$, for any integer p, which may improve performance.

2.3 Special Case III

In this subsection, rather than using the Cayley transform and the norm, we will use symmetric functions of the eigenvalues of symmetric matrices to construct a stability function and to set the system of stable solutions.

The following inequality comes from [18] (Prop. 16.2, P 182)

$$\max \lambda(A + A^T)/2 \geq \max Re(\lambda(A)). \tag{2.8}$$

So the negative real value of eigenvalues for A can be replaced by $\max \lambda(A + A^T) < 0$.

Define

$$h_3(x) = -\max_{1 \leq i \leq n} \lambda_i(\nabla F(x) + \nabla F(x)^T) \tag{2.9}$$

The stable solution of (1.2) can be set by

$$\begin{cases} F(x) = 0, \\ h_3(x) \geq \delta. \end{cases} \tag{2.10}$$

3 Semismooth Property of Stability Functions

To analyze the semismooth property of the stability functions introduced in Section 2, we first recall the definition and properties of spectral functions.

Denote the set of all $n \times n$ real symmetric matrices by $\mathcal{S}_n$, and the set of all $n \times n$ real orthogonal matrices by $\mathcal{O}_n$.

Definition 3.1 *A spectral function is one that it can be written as a composition of a symmetric function* $f : \Re^n \to \Re$ *and the eigenvalue function* $\lambda(\cdot) : \mathcal{S}_n \to \Re^n$*, often denoted by* $(f \circ \lambda)$*, and the symmetric function* f *means that* $f(x) = f(Px)$ *with any permutation matrix* P*.*

For each $X \in \mathcal{S}_n$, the spectral functions are denoted as $(f \circ \lambda)(X)$. Throughout this paper, the set of orthogonal matrices corresponding to X is defined by

$$O(X) := \{P \in \mathcal{O}_n \mid PXP^T = \text{diag}(\lambda(X))\}.$$

According to the concept above, the following functions belong to the class of spectral functions:

$$\underline{H}(X) \equiv (f \circ \lambda)(X) = \min_{1 \le i \le n} \lambda_i(X), \quad \overline{H}(X) \equiv (f \circ \lambda)(X) = \max_{1 \le i \le n} \lambda_i(X). \tag{3.1}$$

with $X \in \mathcal{S}_n$.

Recall that a differentiable function is called an LC^1 (SC^1) function if its derivative function is locally Lipschitz (semismooth). We say a function F is semidifferentiable at $x \in R^n$ if the limit

$$\lim_{\substack{\tau \searrow 0 \\ \hat{h} \to h}} \frac{F(x + \tau \hat{h}) - F(x)}{\tau}$$

exists for every direction $h \in R^n$ (see [24]). There are some interesting properties for a spectral function as follows (Lemma 3.2-Lemma 3.3 and Proposition 3.5 in [24]).

Lemma 3.1 *A spectral function* $(f \circ \lambda)$ *satisfies the following properties:*
(i) it is directional differentiable if f *is semidifferential;*
(ii) it is LC^1 *if and only if* f *is* LC^1*;*
(iii) it is SC^1 *if and only if* f *is* SC^1*.*
(iv) $(f \circ \lambda)$ *is differentiable at point* X *if and only if* f *is differentiable at point* $\lambda(X)$*. And the gradient of* $(f \circ \lambda)$ *at* X *is given by*

$$\nabla (f \circ \lambda)(X) = U^T \Big(\text{diag}(\nabla f(\lambda(X))) \Big) U, \ \forall U \in O(X). \tag{3.2}$$

More generally, the gradient of $(f \circ \lambda)$ *has the following formula*

$$\nabla (f \circ \lambda)(X) = V^T \Big(\text{diag}(\nabla f(\mu)) \Big) V, \tag{3.3}$$

for any orthogonal matrix $V \in \mathcal{O}_n$ *and* $\mu \in \Re^n$ *satisfying* $X = V^T(\text{diag } \mu)V$*.*

Lemma 3.2 *For any symmetric function* $f : R^n \to R$*, the spectral function* $(f \circ \lambda)$ *is semismooth if and only if* f *is semismooth. If* f *is* ρ*-order semismooth* $(0 < \rho < \infty)$*, the spectral function* $(f \circ \lambda)$ *is* $\min\{1, \rho\}$*-order semismooth.*

According to above lemmas, we have the following conclusion.

Theorem 3.1 *The stability functions $h_1(x)$, $h_2(x)$ and $h_3(x)$ introduced in Section 2 are semismooth.*

Proof. Denote

$$Y(x) = \nabla F(x)^T \nabla F(x), \quad Z(x) = \nabla F(x) + \nabla F(x)^T.$$

$$f_1(y) = \min_{1\le i\le n} \{y_i\}, \quad f_2(y) = \max_{1\le i\le n} \{y_i\}.$$

The twice continuously differentiable assumption of $F(x)$ implies that $Y(x)$ and $Z(x)$ are continuously differential. On the other hand, $f_1(y)$ and $f_2(y)$ are symmetric and semismooth.

From the notation above, $h_1(x)$ and $h_3(x)$ can be rewritten as

$$h_1(x) = (f_1 \circ \lambda)(Y(x)), \quad h_3(x) = (f_2 \circ \lambda)(Z(x)).$$

Then the semismooth property of $h_1(x)$ and $h_3(x)$ follows from the property of semismooth composition functions.

Denote

$$W(x) = C(\nabla F(x)) = (\nabla F(x) - \sigma I)^{-1}(\nabla F(x) + \sigma I), \quad R^n \to R^{n\times n}.$$

$W(x)$ is continuously differentiable. In addition, l^1 norm is semismooth. Then from the property of semismooth composition functions again we have the semismoothness of $h_2(x)$. We complete the proof. □

4 Smoothing Functions of The Stability Functions

We will consider smoothing Newton methods for solving the nonsmooth systems (1.2), (1.5) or (1.6). To this end, we study smooth approximation for the three stability functions proposed in Section 2. On the other hand, we note that the three stability functions are related to the so-called maximum nonsmooth function, i.e., $f(x) = \max\{x_1, x_2, \cdots, x_n\}$. So we study the smoothing function of the maximum function and its properties in the first.

For any $\epsilon > 0$, define a smoothing function of $f(x)$ by

$$f_s(\epsilon, x) = \begin{cases} \epsilon \;\ln\left(\sum_{i=1}^n e^{x_i/\epsilon}\right), & \text{if } \epsilon \neq 0 \\ \max_{1\le i\le n}\{x_i\}, & \text{if } \epsilon = 0, \end{cases} \tag{4.1}$$

where ϵ is called *smoothing parameter*. This smoothing function is also called exponential penalty function and has the following characteristics (see [4, 30]).

Lemma 4.1 *$f_s(\epsilon, x)$ has the following properties for any $\epsilon > 0$.*
(i) $f_s(\epsilon, x)$ is increasing with respect to ϵ, i.e., for any $\epsilon_1 > \epsilon_2 > 0$, it holds $f_s(\epsilon_1, x) \geq f_s(\epsilon_2, x)$. Further, we have

$$0 \leq f_s(\epsilon, x) - f(x) \leq \epsilon \ln n. \tag{4.2}$$

(ii) $f_s(\epsilon, x)$ is continuously differentiable and

$$\partial_x f_s(\epsilon, x) = \sum_{i=1}^{n} a_i(\epsilon, x) e^i, \tag{4.3}$$

$$\begin{cases} \frac{df_s(\epsilon,x)}{d\epsilon} = \ln \sum_{i=1}^{n} e^{x_i/\epsilon} - \frac{1}{\epsilon} \sum_{i=1}^{m} a_i(\epsilon, x) x_i, \\ \lim_{\epsilon \to 0+} \frac{df_s(\epsilon,x)}{d\epsilon} = \ln |I(x)|, \\ \ln |I(x)| \leq \frac{df_s(\epsilon,x)}{d\epsilon} \leq \ln n \end{cases} \tag{4.4}$$

with fixed x, where

$$a_i(\epsilon, x) = \frac{e^{x_i/\epsilon}}{\sum_{i=1}^{m} e^{x_i/\epsilon}}, \quad I(x) = \{i \mid f(x) = x_i\}. \tag{4.5}$$

(iii) For any fixed $x \in R^n$,

$$\text{dist}(\partial_x f_s(\epsilon, x), \partial f(x)) = o(\epsilon), \tag{4.6}$$

where $dist(x, S)$ represents the distance from point x to set S.

Based upon the smoothing function f_s for the maximum function f, we discuss some properties of the smoothing function for the special spectral function $(f \circ \lambda)(X)$ with $X \in \mathcal{S}_n$. Define

$$G(X) \equiv (f \circ \lambda)(X) = \max_{1 \leq i \leq n} \lambda_i(X), \qquad R^{n \times n} \to R. \tag{4.7}$$

We construct the smoothing function of $G(X)$ by the above technology and denote it as $G_s(\epsilon, X)$, which has the following version for $\epsilon > 0$

$$G_s(\epsilon, X) \equiv (f_\epsilon \circ \lambda)(X) = \epsilon \ln \Big(\sum_{i=1}^{n} e^{\lambda_i(X)/\epsilon} \Big). \tag{4.8}$$

Then combining the properties of the spectral function, we have the following conclusions (see Proposition 2.2 in [4]).

Proposition 4.1 *For any $X \in \mathcal{S}_n$, there exist an orthogonal matrix $Q \in O(X)$ satisfying*

$$X = Q^T(diag(\lambda(X))Q.$$

Furthermore, let λ_1 be the largest eigenvalue of $\lambda = \lambda(X)$. Then the following two conclusions hold.
(i)

$$\nabla(f_\epsilon \circ \lambda)(X) = Q^T(diag(\nabla f_\epsilon(\lambda)))Q = Q^T(diag(\mu(\epsilon, \lambda)))Q \in R^{n \times n}, \tag{4.9}$$

with

$$\mu_i(\epsilon,\lambda) = \frac{e^{\lambda_i/\epsilon}}{\sum_{j=1}^n e^{\lambda_j/\epsilon}} = \frac{e^{(\lambda_i-\lambda_1)/\epsilon}}{\sum_{j=1}^n e^{(\lambda_j-\lambda_1)/\epsilon}}. \tag{4.10}$$

(ii) If X has multiplicity r for the largest eigenvalue λ_1, then the first r rows of Q must form an orthogonal basis of the eigenspace associated with λ_1, these rows form a matrix $Q_1 \in \mathcal{R}^{n\times r}$, and it holds

$$\lim_{\epsilon\to 0} \mu_i(\epsilon,\lambda) = \begin{cases} \frac{1}{r}, & i = 1,\cdots,r, \\ \\ 0, & i = r+1,\cdots,n. \end{cases} \tag{4.11}$$

Therefore,

$$\begin{aligned} \lim_{\epsilon\to 0} \nabla(f_\epsilon\circ\lambda)(X) &= Q^T(diag(\frac{1}{r},\cdots,\frac{1}{r},0,\cdots,0))Q \\ &= \frac{1}{r}Q_1^T I_r Q_1. \end{aligned} \tag{4.12}$$

From [23],

$$\partial\lambda_1(X) = \{Q_1^T Y Q_1 : Y \in C_r\}$$

with

$$C_r = \{v \in \mathcal{S}_r : V \text{ is positive semidefinite, } tr(V) = 1\}.$$

Hence we have

$$\lim_{\epsilon\to 0} \nabla(f_\epsilon\circ\lambda)(X) \in \partial\lambda_1(X). \tag{4.13}$$

In the reminder of this section, we will discuss the smoothing functions of the three stability functions introduced in Section 2.

4.1 Smoothing Function for h_1

The function $h_1(x)$ can be written as equivalently

$$h_1(x) = \min_{1\le i\le n} \lambda_i(\nabla F(x)^T\nabla F(x)) = -\max_{1\le i\le n} \lambda_i(-\nabla F(x)^T\nabla F(x)). \tag{4.14}$$

Denote $Y(x) = -\nabla_x F(x)^T\nabla_x F(x)$. It is easy to see that Y is a symmetric real matrix. By using the smoothing technology above we have that, for $Y \in \mathcal{R}^{n\times n}$,

$$(f_\epsilon\circ\lambda)(Y) = \epsilon\ln(\sum_{i=1}^n e^{\lambda_i(Y)/\epsilon}). \tag{4.15}$$

Furthermore, for $\epsilon > 0$, we define $\theta_\epsilon^1 : \mathcal{R}^n \to \mathcal{R}$ to be the smoothing function of $h_1(x)$ as

$$\theta_\epsilon^1(x) := -(f_\epsilon\circ\lambda\circ Y)(x) = -\epsilon\,\ln\Big(\sum_{i=1}^n e^{\lambda_i(-\nabla F(x)^T\nabla F(x))/\epsilon}\Big), \quad \forall x\in\mathcal{R}^n. \tag{4.16}$$

which is a smooth approximation of the function h_1. Therefore, we can approximate the problem (2.3) by the following smooth system of equations:

$$\begin{cases} F(x) = 0, \\ \theta_\epsilon^1(x) \geq \delta. \end{cases} \tag{4.17}$$

Moreover, from Lemma 4.1, Proposition 4.1 and the chain rule, we have the following results.

Proposition 4.2 *Let θ_ϵ^1, defined in (4.16), be the smoothing function of h_1. Then the following two conclusions hold.*
(i) For $\epsilon > 0$,

$$-\epsilon \ \ln n \leq \theta_\epsilon^1(x) - h_1(x) \leq 0; \tag{4.18}$$

(ii) Denote the derivate of $\nabla(f_\epsilon \circ \lambda)(Y)$ by (see (4.9))

$$D^\epsilon(Y) = \nabla(f_\epsilon \circ \lambda)(Y).$$

Then it holds that

$$\nabla\theta_\epsilon^1(x) = \Big\{\beta^\epsilon(x) \in \Re^n \ \Big| \ (\beta(x))_k = \sum_{i,j=1}^n (D^\epsilon(Y))_{ij}\Big[\sum_{l=1}^n (\frac{\partial^2 f_l}{\partial x_i \partial x_k} \cdot \frac{\partial f_l}{\partial x_j} + \frac{\partial^2 f_l}{\partial x_j \partial x_k} \cdot \frac{\partial f_l}{\partial x_i})\Big]\Big\}. \tag{4.19}$$

Proof. (i) Denote

$$\bar{h}_1(x) = \max_{1\leq i\leq n} \lambda_i(-\nabla F(x)^T \nabla F(x)).$$

$$\bar{\theta}_\epsilon^1(x) = \epsilon \ \ln\Big(\sum_{i=1}^n e^{\lambda_i(-\nabla F(x)^T \nabla F(x))/\epsilon}\Big).$$

Then from Lemma 4.1 it holds

$$0 \leq \bar{\theta}_\epsilon^1(x) - \bar{h}_1(x) \leq \epsilon \ \ln n.$$

(4.18) follows directly.

(ii) The conclusion of (4.19) is obtained directly from the chain rule and Proposition 4.1. □

4.2 Smoothing Functions for h_2

We consider the model (2.7):

$$h_2(x) = 1 - \|C(\nabla F(x)\|_1. \tag{4.20}$$

where $F(x)$ is a smooth function, and for $X \in \mathcal{R}^{n\times n}, C(X) = (X - \sigma I)^{-1}(X + \sigma I)$ with $\sigma > 0$.

For convenience of expression, we denote

$$\hat{h}_2(x) = \|C(\nabla_x F(x)\|_1 \equiv \|M(x)\|_1 = \max_{1\leq i\leq n} \sum_{j=1}^n |M_{ij}(x)|, \tag{4.21}$$

where

$$M(x) = (M_{ij}(x)) \equiv \Big(\nabla F(x) - \sigma I\Big)^{-1}\Big(\nabla F(x) + \sigma I\Big) \in R^{n\times n}.$$

Notice that the nonsmoothness of $\hat{h}_2(x)$ comes from the absolute value function and the maximum function. Then we consider a double smoothing technology with smoothing parameter $\epsilon > 0$.

The first level smoothing function for $|M_{ij}(x)|$ is defined by

$$\begin{aligned} M^{\epsilon}_{ij}(x) &\equiv \epsilon \ln\Big(e^{M_{ij}(x)/\epsilon} + e^{-M_{ij}(x)/\epsilon}\Big) \\ &= |M_{ij}(x)| + \epsilon \ln\Big(e^{(M_{ij}(x)-|M_{ij}(x)|)/\epsilon} + e^{(-M_{ij}(x)-|M_{ij}(x)|)/\epsilon}\Big), \end{aligned} \tag{4.22}$$

where the aim of the second equality is to control the computing overflow.

The second level smoothing function is for the maximum function $\max_{1\le i\le n}\{f_i\}$. Denote

$$\Phi^{\epsilon}_i(x) \equiv \sum_{j=1}^{n} M^{\epsilon}_{ij}(x). \tag{4.23}$$

From the smoothing technology introduced in Subsection 4.1, we construct a smoothing function for $\hat{h}_2(x)$ by

$$\begin{aligned} \bar{\theta}^2_{\epsilon}(x) &= \epsilon \ln\Big(\sum_{i=1}^{n} e^{\Phi^{\epsilon}_i(x)/\epsilon}\Big) \\ &= \bar{\Phi}_{\epsilon}(x) + \epsilon \ln\Big(\sum_{i=1}^{n} e^{(\Phi^{\epsilon}_i(x)-\bar{\Phi}_{\epsilon}(x))/\epsilon}\Big), \end{aligned} \tag{4.24}$$

where $\bar{\Phi}_{\epsilon}(x) = \max_{1\le i\le n}\Phi^{\epsilon}_i(x)$. The second equality in (4.24) is designed for avoiding the overflow phenomenon in computation. Finally, we construct the smoothing function of $h_2(x)$, denoted by $\theta^2_{\epsilon}(x)$ by

$$\theta^2_{\epsilon}(x) = 1 - \Big[\bar{\Phi}_{\epsilon}(x) + \epsilon \ln\Big(\sum_{i=1}^{n} e^{(\Phi^{\epsilon}_i(x)-\bar{\Phi}_{\epsilon}(x))/\epsilon}\Big)\Big]. \tag{4.25}$$

The system of (2.7) can be replaced by the following smoothing system

$$\begin{cases} F(x) = 0, \\ \theta^2_{\epsilon}(x) \ge \delta. \end{cases} \tag{4.26}$$

We also have the following proposition similar to the smoothing function of $h_1(x)$.

Proposition 4.3 *Let θ^2_{ϵ}, defined in (4.25), be the smoothing function of h_2. We have for $\epsilon > 0$*
(i)

$$-\epsilon\ (\ln n + n\ \ln 2) \le \theta^2_{\epsilon}(x) - h_2(x) \le 0; \tag{4.27}$$

(ii)

$$\nabla\theta_\epsilon^2(x) = -\sum_{i=1}^{n}\sum_{j=1}^{n} a_i(\epsilon,x)\frac{e^{M_{ij}(x)/\epsilon} - e^{-M_{ij}(x)/\epsilon}}{e^{M_{ij}(x)/\epsilon} + e^{-M_{ij}(x)/\epsilon}}\nabla M_{ij}(x), \tag{4.28}$$

where

$$a_i(\epsilon,x) = \frac{e^{\Phi_i^\epsilon(x)}}{\sum_{j=1}^{n} e^{\Phi_i^\epsilon(x)}},$$
$$\nabla M_{ij}(x) = \gamma_{ij}(x)$$

with $\gamma_{ij}(x)$ *defined*

$$\begin{aligned}\gamma_{ij}(x) &= \nabla\Big(G_{i\cdot}^1(x) \bullet G_{\cdot j}^2(x)\Big) \\ &= \nabla\Big(G_{i\cdot}^1(x)\Big)G_{\cdot j}^2(x) + \nabla\Big(G_{\cdot j}^2(x)^T\Big)G_{i\cdot}^1(x)^T,\end{aligned} \tag{4.29}$$

where the product included in the second equation above is the usual product of a matrix and a vector.

Proof. (i) From Lemma 4.1 we have

$$0 \le M_{ij}^\epsilon(x) - |M_{ij}(x)| \le \epsilon\ \ln 2,$$

this follows

$$0 \le \max_{1\le i\le n}\sum_{j=1}^{n} M_{ij}^\epsilon(x) - \max_{1\le i\le n}\sum_{i=1}^{n}|M_{ij}(x)| \le n\epsilon\ \ln 2. \tag{4.30}$$

In addition, from Lemma 4.1 again, it holds

$$0 \le \bar\theta_\epsilon^2(x) - \max_{1\le i\le n}\Phi_i^\epsilon(x) \le \epsilon\ \ln n. \tag{4.31}$$

Combining with (4.30)-(4.31) and the notation above, we have the following derivation

$$\begin{aligned}&\bar\theta_\epsilon^2(x) - \hat h_2(x) \\ &\quad= \bar\theta_\epsilon^2(x) - \max_{1\le i\le n}\Phi_i^\epsilon(x) + \max_{1\le i\le n}\sum_{i=1}^{n} M_{ij}^\epsilon(x) - \max_{1\le i\le n}\sum_{j=1}^{n}|M_{ij}(x)| \\ &\quad\le \epsilon\ \ln n + n\epsilon\ \ln 2 = \epsilon(\ln n + n\ \ln 2).\end{aligned}$$

Then the conclusion (4.27) is proved.

(ii) From Lemma 4.1 and the chain rule we have

$$\nabla\theta_\epsilon^2(x) = -\sum_{i=1}^{n} a_i(\epsilon,x)\nabla\Phi_i^\epsilon(x) \tag{4.32}$$

with

$$a_i(\epsilon, x) = \frac{e^{\Phi_i^\epsilon(x)/\epsilon}}{\sum_{j=1}^n e^{\Phi_j^\epsilon(x)/\epsilon}}.$$

On the other hand, we obtain by direct computation

$$\nabla \Phi_i^\epsilon(x) = \sum_{j=1}^n \frac{e^{M_{ij}(x)/\epsilon} - e^{-M_{ij}(x)/\epsilon}}{e^{M_{ij}(x)/\epsilon} + e^{-M_{ij}(x)/\epsilon}} \nabla M_{ij}(x). \tag{4.33}$$

With a similar way we can prove that $\nabla M_{ij}(x) = \gamma_{ij}(x)$. From (4.32) and (4.33) we have (4.28). The proposition is proved. □

In fact, we can compute $\gamma_{ij}(x)$ in (4.29) for $\nabla\Big(G^2_{\cdot j}(x)^T\Big)$ and $\nabla\Big(G^1_{i\cdot}(x)\Big)$ as follows.

(i) The derivative of $\nabla\Big(G^2_{\cdot j}(x)^T\Big)$ can be computed directly.

(ii) Consider the derivative of $\nabla\Big(G^1_{i\cdot}(x)\Big)$. Denote

$$A(x) \equiv (\nabla F(x) - \sigma I) = (G^1(x))^{-1}.$$

Then it holds for each fixed i and $k = 1, \cdots, n$,

$$A^{-1}_{i\cdot}(x) A_{\cdot k}(x) = \begin{cases} 1, & \text{if } k = i, \\ 0, & \text{if } k \neq i. \end{cases}$$

It follows for $k = 1, \cdots, n$,

$$\nabla_x\Big(A^{-1}_{i\cdot}(x)\Big)\Big(A_{\cdot k}(x)\Big) + \nabla_x\Big(A^T_{\cdot k}(x)\Big)\Big(A^{-1}_{i\cdot}(x)\Big)^T = 0.$$

Above formula can be written equivalently as

$$\nabla_x\Big(A^{-1}_{i\cdot}(x)\Big)A(x) + \nabla_x\Big(A(x)^T\Big)\Big(A^{-1}_{i\cdot}(x)\Big)^T = 0. \tag{4.34}$$

Note that the second term above is a product of a third order tensor with a vector (or called a first order tensor). The result of product is

$$\nabla_x\Big(A(x)^T\Big)\Big(A^{-1}_{i\cdot}(x)\Big)^T \in R^{n\times n},$$

where each element of the product is (for each fixed i)

$$\left(\nabla_x\Big(A(x)^T\Big)\Big(A^{-1}_{i\cdot}(x)\Big)^T\right)_{lk} = \sum_{j=1}^n \left(\nabla_x\Big(A(x)^T\Big)\right)_{lkj} A^{-1}_{ij}(x).$$

From (4.34), we deduce that

$$\nabla_x\Big(A^{-1}_{i\cdot}(x)\Big) = -\left(\nabla_x\Big(A^T(x)\Big)\Big(A^{-1}_{i\cdot}(x)\Big)^T\right)A^{-1}(x). \tag{4.35}$$

So we have

$$\nabla_x\Big(G^1_{i\cdot}(x)\Big) = -\Bigg(\nabla_x\Big(((G^1(x))^{-1})^T\Big)\Big(G^1_{i\cdot}(x)\Big)^T\Bigg)G^1(x) \in \Re^{n\times n}. \tag{4.36}$$

Here $(G^1(x))^{-1} = \nabla f(x) - \sigma I$. Its derivative can be computed directly.

4.3 Smoothing Function for h_3

We now consider the model (2.10). Similar to the function $h_1(x)$, we can construct the smoothing function of $h_3(x)$ as follows.

Denote $Y(x) = \nabla_x F(x) + \nabla_x F(x)^T$. Then Y is a symmetric real matrix. For $\epsilon > 0$, define $\theta^3_\epsilon(x) : \mathcal{R}^n \to \mathcal{R}$ by

$$\theta^3_\epsilon(x) := -(f_\epsilon \circ \lambda \circ Z)(x) = -\epsilon \,\ln\Big(\sum_{i=1}^n e^{\lambda_i(\nabla F(x)+\nabla F(x)^T)/\epsilon}\Big), \quad \forall x \in \mathcal{R}^n. \tag{4.37}$$

Finally, we can obtain the smoothing system of (2.10) as

$$\begin{cases} F(x) = 0, \\ \theta^3_\epsilon(x) \geq \delta. \end{cases} \tag{4.38}$$

Similar to the cases of h_1 and h_2, we have the following proposition.

Proposition 4.4 *Let θ^3_θ, defined in (4.37), be the smoothing function of h_3. Then we have the following two conclusions.*

(i) For $\epsilon > 0$

$$-\epsilon\, \ln n \leq \theta^3_\epsilon(x) - h_3(x) \leq 0. \tag{4.39}$$

(ii) Denote (see (4.9)) the derivate of $\nabla(f_\epsilon \circ \lambda)(Y)$ by

$$D^\epsilon(Y) = \nabla(f_\epsilon \circ \lambda)(Y).$$

Then it holds that

$$\nabla\theta^3_\epsilon(x) = \Big\{\beta^\epsilon(x) \in \Re^n \;\Big|\; (\beta(x))_k = \sum_{i,j=1}^n (D^\epsilon(Y))_{ij}\Big[\frac{\partial^2 f_i(x)}{\partial x_j \partial x_k} + \frac{\partial^2 f_j(x)}{\partial x_i \partial x_k}\Big]\Big\}. \tag{4.40}$$

Proof. The proof is similar to the one of Proposition 4.2. □

5 Smoothing Newton Methods

We generalize the three systems of equations (2.3), (2.7), (2.10) as

$$\begin{cases} F(x) = 0, \\ h(x) \geq \delta, \end{cases} \tag{5.1}$$

which are equivalent to the following unconstrained system of semismooth equations

$$\begin{cases} F(x) = 0, \\ -h(x) + |y| + \delta = 0 \end{cases} \tag{5.2}$$

with $y \in R$. Moreover, the absolute function $|y|$ can be smoothed by

$$\phi(\epsilon, y) = \epsilon \ln(e^{y/\epsilon} + e^{-y/\epsilon}). \tag{5.3}$$

By using a uniform notation, we denote the corresponding smoothing functions of $h_1(x), h_2(x), h_3(x)$ as $\theta(\epsilon, x)$. Based on the model of unconstrained equations (5.2), the systems of (4.17),(4.26), (4.38) can be written equivalently as

$$\begin{cases} F(x) = 0, \\ G(\epsilon, x, y) \equiv -\theta(\epsilon, x) + \phi(\epsilon, y) + \delta = 0. \end{cases} \tag{5.4}$$

5.1 Smoothing Newton Algorithm

The system of (5.4) is a basis on which we solve the stable solution system (i.e., solving the system (5.2)). We will use the smoothing Newton method proposed in [26] to construct an algorithm, in which we consider ϵ as a variable and set the equivalent system as

$$\Phi(\epsilon, x, y) = \begin{pmatrix} \epsilon \\ F(x) \\ G(\epsilon, x, y) \end{pmatrix} = 0. \tag{5.5}$$

Denote $w = (\epsilon, x, y)$, and define a merit function of (5.5) as

$$\Psi(w) = \|\Phi(\epsilon, x, y)\|^2. \tag{5.6}$$

Choose $\bar{\epsilon} \in R_{++}$ and $\gamma \in (0, 1)$ such that $\gamma\bar{\epsilon} < 1$. Let $\bar{w} := (\bar{\epsilon}, 0) \in R \times R^{n+1}$. Define $\beta : R^{n+2} \to R_+$ by

$$\beta(w) := \gamma \ \min\{1, \Psi(w)\}.$$

Based on the choice of the parameters, we have the following equivalent relationship (see Proposition 4 in [26])

$$\Phi(w) = 0 \Longleftrightarrow \beta(w) = 0 \Longleftrightarrow \Phi(w) = \beta(w)\bar{w}. \tag{5.7}$$

This also yields the type of smoothing Newton methods as follows.

Algorithm 1: Smoothing Newton Algorithm

Step 0. (Initialization)

Choose constant $\eta \in (0,1)$ and $\alpha \in (0,1/2)$. Let $\epsilon^0 := \bar{\epsilon}, x^0 \in R^n, 0 \neq y^0 \in R$ be an arbitrary point. Set $k := 0$.

Step 1. (Stop Test)

If $\|\Phi(w^k)\| = 0$ then stop. Otherwise let $\beta_k := \beta(w^k)$.

Step 2. (Compute Perturbed Newton Direction)

Compute $\triangle w^k := (\triangle\epsilon^k, \triangle x^k, \triangle y^k) \in R \times R^n \times R$ by

$$\Phi(w^k) + \Phi'(w^k)\triangle w^k = \beta_k \bar{w}. \tag{5.8}$$

Step 3. (Line Search)

Let l_k be the smallest nonnegative integer l satisfying

$$\Psi(w^k + \eta^l \triangle w^k) \leq [1 - 2\alpha(1 - \gamma\bar{\epsilon})\eta^l]\Psi(w^k). \tag{5.9}$$

Define $w^{k+1} := w^k + \eta^{l_k}\triangle w^k$.

Step 4. Set $k := k+1$ and go to Step 1.

we define a set to help the proof of the global convergence of Algorithm 1.

$$\Omega := \{w = (\epsilon, x, y) \in R \times R^n \times R |\ \epsilon \geq \beta(w)\bar{\epsilon}\}. \tag{5.10}$$

It is obviously that $w^0 = (\epsilon^0, x^0, y^0) \in \Omega$. The points in Ω ensure the strictly nonnegative requirement of the smoothing variable ϵ at nonsolution points, which is the necessary condition for the smoothing approach.

The remaining part of this section devotes to the convergence analysis of Algorithm 1.

5.2 Global Convergence

Proposition 5.1 *Suppose that for any $k > 0$ and $w^k \in R_{++} \times R^n \times R$, $\Phi(w^k)$ is nonsingular. Then Algorithm 1 is well-defined at the kth iteration. Furthermore, for each fixed $k \geq 0$, if $\epsilon^k \in R_{++},\ w^k \in \Omega$, we have*

$$\epsilon^{k+1} \in R_{++}, \quad w^{k+1} \in \Omega.$$

Proof. From the properties of smoothing functions discussed in Section 4 we know that for any $(\epsilon, x, y) \in R_{++} \times R^n \times R$, the function $\Phi(w)$ is continuously differentiable. Then by using a proof similar to the proofs of Proposition 5 - Proposition 7 in [26], we can prove the conclusion. □

Now we have the global convergence of Algorithm 1.

Theorem 5.1 *Suppose that for every $k \geq 0$ and $\epsilon^k \in R_{++}, w^k \in \Omega$, $\Phi'(w^k)$ is nonsingular.*
(i) Then an infinite sequence $\{w^k\}$ is generated by Algorithm 1.
(ii) Suppose that for any accumulation point $w^* = (\epsilon^*, x^*, y^*)$ of $\{w^k\}$ with $\epsilon^* \in R_{++},\ w^* \in \Omega$, $\Phi'(w^*)$ is nonsingular. Then each accumulation point of $\{w^k\}$, denoted by $\hat{w}$, is a solution of $\Phi(w) = 0$.

Proof. (i) The generation of an infinite sequence $\{w^k\}$ can be proved by Proposition 5.1 and the fact that $w^0 \in \Omega$.

(ii) From the relationship of $\Phi(w), \beta(w)$ in (5.7) and the line search (5.9) we have that $\Psi(w^k)$ and $\beta(w^k)$ are monotonically decreasing. Moreover, for each accumulation point $\hat{w}$, $\Psi(\hat{w}) = 0$ or $\Psi(\hat{w}) > 0$ implies $\beta(\hat{w}) = 0$ or $\beta(\hat{w}) > 0$, respectively. So from $\Psi(w^{k+1}) < \Psi(w^k)$, there exist $\hat{\Psi}, \hat{\beta} \geq 0$ such that $\Psi(w^k) \to \hat{\Psi}$ and $\beta(w^k) \to \hat{\beta}$. If $\hat{\Psi} = 0$, the conclusion is proved. Otherwise, $\hat{\Psi} > 0$ implies that the accumulation point of $\{w^k\}$, denoted by $\hat{w} = (\hat{\epsilon}, \hat{x}, \hat{y}) \in \Omega$, satisfies $\hat{\epsilon} > 0$. This means that the function $\Phi(w)$ is continuously differentiable at all iterative points w^k and the accumulation point $\hat{w}$. Then the conclusion can be proved by the way of the classical Newton algorithm. Here we omit the detailed process of the proof. □

5.3 Local Convergence

Let $w^* = (0, x^*, y^*)$ be an accumulation point of $\{w^k\}$ generated by Algorithm 1 and a solution of $\Phi(w) = 0$, i.e., a stable solution. It is not difficult to derive that for $\epsilon > 0$, the differential of the smoothing function has the following version:

$$\Phi'(w) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & F'(x) & 0 \\ -\frac{d\theta(\epsilon,x)}{d\epsilon} + \frac{d\phi(\epsilon,y)}{d\epsilon} & \nabla_x \theta(\epsilon, x)^T & \frac{d\theta(\epsilon,y)}{dy} \end{pmatrix}. \tag{5.11}$$

We first have the following nonsingular property of the generalized Jacobian $\partial\Phi(w^*)$.

Proposition 5.2 *Let $w^* = (0, x^*, y^*)$ be a stable solution. Then for the case $y^* \neq 0$, every element of the generalized Jacobian, i.e., $V \in \partial\Phi(w^*)$, is nonsingular.*

Proof. It is easy to see that at the solution point $w^* = (0, x^*, y^*)$ with $y^* \neq 0$, for any $V \in \partial\Phi(w^*)$ there exist $M = (M_\epsilon, M_x) \in \partial G(w^*)$ such that

$$V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & F'(x^*) & 0 \\ M_\epsilon & M_x & \alpha_y \end{pmatrix}. \tag{5.12}$$

Here

$$\alpha_y = \begin{cases} 1, & \text{if } y^* > 0 \\ -1, & \text{if } y^* < 0 \end{cases}. \tag{5.13}$$

On the other hand, from the definition of stable solutions we know that $F'(x^*)$ is nonsingular. This combines with (5.13) to obtain the nonsingularity of V. The conclusion is proved. $\square$

By using a similar way of proof in [26] (see Theorem 8 in [26]), we can obtain the local convergence of Algorithm 1.

Theorem 5.2 *Let $w^* = (0, x^*, y^*)$ with $y^* \neq 0$ be an accumulation point of $\{w^k\}$ generated by Algorithm 1 and a solution of $\Phi(w) = 0$. Then the whole sequence $\{w^k\}$ is convergent to w^*. Furthermore, we have*

$$\|w^{k+1} - w^*\| = o(\|w^k - w^*\|), \qquad \epsilon^{k+1} = o(\epsilon^k). \tag{5.14}$$

Proof. From Proposition 5.2 we have that for sufficiently large k it exists a constant C such that

$$\|\Phi'(w^k)^{-1}\| \leq C.$$

On the other hand, the solution point w^* means that for large enough k, it holds

$$\beta(w^k) = \gamma\Psi(w^k).$$

Then by a proof similar to those in [26] (see the proof of Theorem 8 in [26]), we can derive that

$$\|w^{k+1} + \triangle w^k - w^*\| = o(\|w^k - w^*\|)$$

and the step-length of line-search is 1. This means that

$$\|w^{k+1} - w^*\| = o(\|w^k - w^*\|).$$

The convergence of $\{w^k\}$ and the first expression in (5.14) are proved. The second conclusion in (5.14) can be proved by $\beta(w^k) = \gamma\Psi(w^k)$. We omit the details of the proof. $\square$

Remark 5.1 *In the local convergence analysis, we only consider the case that $y^* \neq 0$ at the solution $w^* = (0, x^*, y^*)$. Since $y^* = 0$ implies $h(x^*) = \delta$, this means that we obtain a solution close to an unstable solution, because of the small constant δ. In this case, we can consider to enlarge the constant δ and repeat the computation so that we may find a better stable solution with $y^* \neq 0$.*

6 Numerical Examples

In this section, we describe three examples for testing the effects of the stability functions proposed in Section 2 by using Algorithm 1. One example is called the **Bratu Problem**; the other one is called the **Beam Problem**; and the third one is a **Power System**. We choose the **Special Case III** model (see (2.9)-(2.10)) as an example to do the numerical experiments, i.e., to solve the the system (5.1)-(5.4) with $h(x) = h_3(x)$.

The parameters used in Algorithm 1 and the stopping rule in the algorithm are chosen as

$$\eta = 0.5,\ \ \alpha = 0.00005,\ \ \bar{\epsilon} = 0.2,\ \ \gamma = 0.02,\ \ \|\Phi(w)\| \le 10^{-5}.$$

The control parameter of stable solution models (denoted by δ in (1.2)) is specialized for all tested examples

$$\delta = 0.0001.$$

6.1 Examples: The Bratu Problem and The Beam Problem

Both of the two examples are the two point bounded value problem.

Example 1: The Bratu problem . It is as follows:

$$F(u,\alpha)(x) = -u''(x) - \alpha e^{u(x)} = 0, \qquad \text{for } 0 < x < 1, \tag{6.1}$$

subject to homogeneous Dirichlet boundary conditions

$$u(0) = u(1) = 0.$$

In (6.1), $\alpha \ge 0$ is a parameter. We consider the case that $\alpha = 3.5$ and discretize the problem by a standard second order central difference scheme with n interior nodes. Then we obtain a nonlinear problem

$$F(u,\alpha) = -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - \alpha e^{u_i} = 0, \tag{6.2}$$

in R^n.

It is known that the Bratu problem has two solutions, one is a stable solution and the other is an unstable solution with maximal eigenvalues $max(\lambda_i(-\nabla_u F(u^*,\alpha)) = -0.8695$ and $max(\lambda_i(-\nabla_u F(u^*,\alpha)) = 0.9275$, respectively. See Fig. 1.

Example 2: The Beam Problem. It is described as

$$F(u,\alpha)(x) = -u''(x) - \alpha\ \sin(u(x)) = 0 \qquad \text{for } 0 < x < 1, \tag{6.3}$$

subject to the boundary conditions

$$u(0) = u(1) = 0.$$

FIG. 1.1. *Bratu Problem*

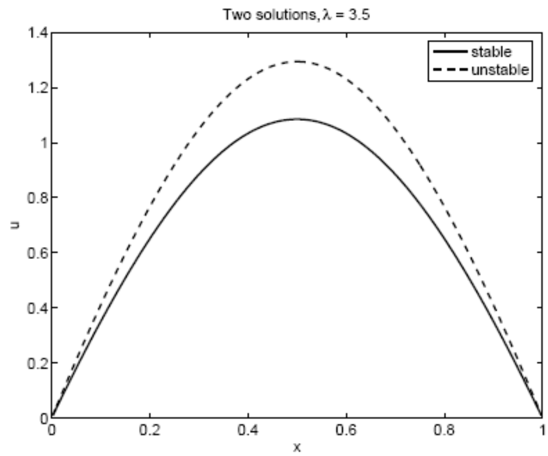

Fig.1 Solution Curves

In (6.3), $\alpha \geq 0$ is a parameter. As the same as the Bratu problem, we discretize the problem with a standard second order central difference scheme with n interior nodes. Then the following nonlinear problem is obtained

$$F(u, \alpha) = -\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - \alpha \ \sin(u_i) = 0, \tag{6.4}$$

in R^n.

It is also known that there are three solutions for the case that $\alpha = 11$. One is $u(x) = 0$, which is unstable.

For these two examples, F' is symmetric, and therefore all the eigenvalues are real. So we just use $\nabla F(x)$ to substitute $\nabla F(x) + \nabla F(x)^T$ in the stable function $h_3(x)$.

By using Algorithm 1 with the chosen algorithm parameters, the numerical results are reported in Table 1 for the two examples.

Table 1. Numerical Results for Bratu Problem and Beam Problem

Ex.	u^0	n	IterN	IterF	IterDF	y^*	$\|F(x^*, \alpha)\|$	$max(\lambda(-\nabla_u F^*))$	CPU(s)
Bratu	$0 \in R^n$	100	10	13	11	0.8694	$3.8511E-12$	-0.8695	0.8212
		200	8	11	9	0.8736	$8.3365E-10$	-0.8737	2.6839
		400	10	13	11	0.8746	$8.8887E-11$	-0.8747	19.5982
Beam	$x(i)$	100	14	15	15	-2.1726	$3.1930E-12$	-2.1728	0.7611
		200	11	13	12	-2.1716	$1.7314E-11$	-2.1717	3.0243
		400	10	11	11	-2.1701	$7.2367E-11$	-2.1714	17.0445

where u^0 is the initial point; $x(i) = i/(n+1)$ $(i = 1, 2, \cdots, n)$; n is the dimension of the solved problem (i.e., the discretization number for the problem); **IterN** is the iterative number

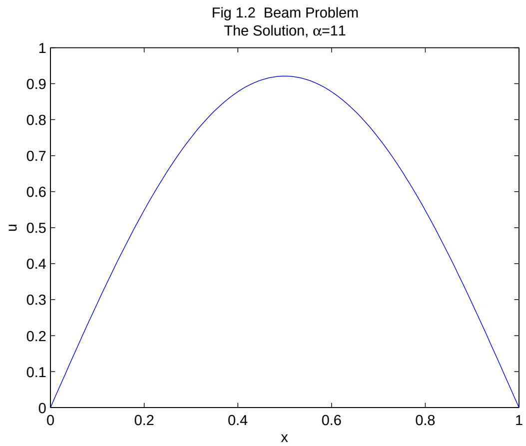

Fig.2 Solution Curves

of Algorithm 1; **IerF** and **IterDF** are the computing number of F and its derivative, respectively; y^* is the slack variable defined in the unconstrained equations (5.2); $\|F(x^*, \alpha))\|$ is the final value of F; $max(\lambda(-\nabla_u F^*))$ is the maximal eigenvalue of $-\nabla_u F(u^*, \alpha)$; **CPU(s)** (second) is the cost time to finish the calculation.

For the Beam Problem, the stopping rule is set $\|\Phi(w)\| \le 10^{-8}$, and the solution curve for $n = 100$ is shown in Fig.2.

6.2 Example 3: A Power System

Consider a simple power system discussed in [2] as follows:

$$F(V, \delta) = \begin{pmatrix} -V^2 G + V(G\cos\delta + B\sin\delta) - P_d \\ -V^2(B - B_C) - V(G\sin\delta - B\cos\delta) - Q_d \end{pmatrix} = 0 \tag{6.5}$$

where the variable is $x = (V, \delta)$, others, such as $B_C, X_C B, P_d, Q_d$, are given constants with

$$G = \frac{R}{R^2 + (X_L - X_C)^2}, \quad B = \frac{X_L - X_C}{R^2 + (X_L - X_C)^2}, \quad 0 \le V \le 2, \quad 0 \le \delta \le 2\pi.$$

and

$$R = 0.1, \qquad X_L = 0.5.$$

Paper [2] discussed how to determine P_d, Q_d, B_C, X_C to maximize the distance to saddle-node bifurcation. Here we just consider how to find a stable solution for (6.5).

According to the data proposed in [2], we consider two cases for different constants B_C, X_C, P_d, Q_d. This also means that we consider two systems of equations.

Case I: The parameters in equations are specialized as

$$BC = 0.0, \quad XC = 0.0, \quad Pd = 0.6661, \quad Qd = 0.1665.$$

We choose various initial points for the system (5.1)-(5.4) with $h(x) = h_3(x)$. However, Algorithm 1 fails to find a solution. This shows that the problem may have no stable solution.

Case II: The parameters in equations are specialized as

$$BC = 1.17424, \quad XC = 0.48809, \quad Pd = 2.4, \quad Qd = 0.01$$

By using Algorithm 1 with the stopping rule $\|\Phi(w)\| \leq 10^{-7}$, we find the same solution from two different initial points, which are used in the computation in [2]. The computing results are reported in Table 2.

Table 2. Numerical Results for PS Example with Case-II

x^0	IterN	IterF	IterDF	x^*	y^*	$\|F(x^*)\|$	$\lambda(-\nabla_u F(x^*))$	CPU(s)
$\binom{5}{1}$	12	14	13	$\binom{0.6042}{0.1169}$	1.9858	$7.4593E-017$	$\binom{-1.9859}{-6.0000}$	0.4006
$\binom{4.7830}{0.7358}$	14	17	15	$\binom{0.6042}{0.1169}$	1.9856	$7.8063E-017$	$\binom{-1.9859}{-6.000}$	0.5007

The symbols in Table 2 have the same meaning as Table 1. We also find a stable solution for different initial points. For the two cases of the power system, we find a stable solution for one case; another one may have no stable solution.

The computing results show that the stable solution models and the smoothing Newton approach are effective. Moreover, for the three examples, the solution of y satisfies $y^* \neq 0$. This confirms the local convergence result of Algorithm 1 in Section 5.

7 Final Comments

In this paper, we presented a new mathematical problem based upon engineering applications – Finding a stable solution for a system of nonlinear equations. Two stable solution models and three stability functions are proposed. The semismooth property and smoothing functions of such stability functions are investigated. Smoothing Newton methods for solving such stable solution models are discussed, and the convergence of the algorithm is studied. Three examples are given to show that our models and the three stability functions we proposed, as well as the smoothing Newton algorithm, are practically useable. Further research on larger examples will be the next step of our research.

Acknowledgment. The authors would like to thank Michael Tse for the discussion.

References

[1] F. Alvarado, I. Dobson and Y. Hu, "Computation of closest bifurcations in power systems", *IEEE Trans. Power System* 9 (1994) 918-928.

[2] C.A. Cañizares, "Calculating optimal system parameters to maximize the distance to saddle-node bifurcation points", *IEEE Trans. Circuits and System* 45 (1998) 225-237.

[3] C.A. Cañizares and A. J. Conejo, "Sensitivity-based security-constrained OPF market clearing model", *IEEE Transactions on Power Systems* 20 (2005) 2051-2060.

[4] X. Chen, H.-D. Qi, L. Qi and K.-L. Teo, *Smooth convex approximation to the maximum eigenvalue function*, J. of Global Optimization 30 (2004) 253–270.

[5] X.J. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Mathematics of Computation 67 (1998) 519–540.

[6] H.D. Chiang, I. Dobson and R.J. Thomas, "On voltage in electric power systems", *IEEE Trans. Power Systems* 5 (1990) 601-611.

[7] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[8] I. Dobson, "An iterative method to compute the closest saddle node or Hopf bifurcation in multidimensional parameter space", *Proceedings of the IEEE International Symposium on Circuits and Systems*, San Diego, 1992, 2513-2516.

[9] I. Dobson and L. Lu, "Computing an optimum direction in control space to avoid saddle node bifurcation and voltage collapse in electric power systems", *IEEE Trans. Automatic Control* 37 (1992) 1616-1620.

[10] S.A. Gabriel and J.S. Pang, "A trust region method for constrained nonsmooth equations", in: *Large Scale Optimization-State of the Art, W.W. Hager, D.W. Hearn and P.M. Pardalos, eds.*, Kluwer Academic Publishers, Norwell, USA, pp. 155-181, 1994.

[11] C.P. Gupta, R.K. Varma and S.C. Srivastava, "A method to determine closest Hopf bifurcation in power systems considering exciter and load dynamics", *Proceedings of 'Energy Management and Power Delivery Conference 1998 (EMPD'98)*, Singapore, 1998, 293-297.

[12] C. Kanzow, "Strictly feasible equation-based method for mixed complementarity problems", *Numer. Math.* 89 (2001) 135-160.

[13] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, no. 16 in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[14] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[15] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, no. 1 in Fundamentals of Algorithms, SIAM, Philadelphia, 2003.

[16] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. V. Cutsem and V. Vittal, "Definition and classification of power system stability", *IEEE Transaction on Power Systems* 19 (2004) 1387-1401.

[17] A.S. Lewis, "Derivatives of spectral functions", *Mathematics of Operations Research* 21 (1996) 576-588.

[18] A. Lewis and M. Overton, "Eigenvalue optimization", *Acta Numerica*, 5 (1996) 149-190.

[19] Y. Ma, H. Kawakami and C. K. Tse, "Bifurcation analysis of switched dynamical systems with periodically moving borders", *IEEE Transactions on Circuits and Systems* 51 (2004) 1184–1193.

[20] Y.V. Makarov, Z.Y. Dong and D.J. Hill, "A general method for small signal stability analysis", *IEEE Transaction on Power Systems* 13 (1998) 979-985.

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[22] J.M. Ortega and W.C. Rheinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[23] F. Oustry, "A second-order bundle method to minimize the maximum eigenvalue function", *Math. Program* 89 (2000) 1-33.

[24] H. Qi and X. Yang, "Semismoothness of spectral functions", *SIAM J. Matrix Anal. Appl.* 25 (2004) 766-783.

[25] L. Qi, "Convergence analysis of some algorithms for solving nonsmooth equations", *Math. Oper. Res.* 18 (1993) 227–244.

[26] L. Qi, D. Sun and G. Zhou, "A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities", *Math. Prog.* 87 (2000) 1-35.

[27] L. Qi and J. Sun, "A nonsmooth version of Newton's method", *Mathematical Programming* 58 (1993) 353–367.

[28] L. Qi, X.J. Tong and D.H. Li, "An active-set projected trust region algorithm for box constrained nonsmooth equations", *J. Optim. Theory Appl.* 120 (2004) 601-625.

[29] D. Sun and J. Sun, "Strong semismoothness of eigenvalues of symmetric matrices and its application to inverse eigenvalue problems", *SIAM J. Numer Anal.* 40 (2003) 2352-2367.

[30] X.J. Tong and S. Zhou, "A smoothing projected Newton-type method for semismooth equations with bound constraints", *Journal of Industrial Management Optimization* 1 (2005) 235-250.

[31] M. Ulbrich, "Nonmonotone trust-region method for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problem", *SIAM J. Optim.* 11 (2001) 889-917.

[32] X. Wu, C. K. Tse, O. Dranga and J. Lu, "Fast-scale instability of single-stage power-factor-correction of power supplies", *IEEE Transactions on Circuits and Systems* 53 (2006) 204–213.