MATHEMATICS OF COMPUTATION
Volume 80, Number 275, July 2011, Pages 1703–1723
S 0025-5718(2010)02443-3
Article electronically published on November 30, 2010

CIRCUMSCRIBED ELLIPSOID ALGORITHM FOR FIXED-POINT PROBLEMS

C. BOONYASIRIWAT, K. SIKORSKI, AND C. TSAY

We dedicate this paper to the memory of Leonid Khachiyan, collaborator and friend, who introduced the circumscribed ellipsoid algorithm as the first way of solving linear programming problems in polynomial time.

Abstract. We present a new implementation of the almost optimal Circumscribed Ellipsoid (CE) Algorithm for approximating fixed points of nonexpanding functions, as well as of functions that may be globally expanding, however, are nonexpanding/contracting in the direction of fixed points. Our algorithm is based only on function values, i.e., it does not require computing derivatives of any order. We utilize the absolute and residual termination criteria with respect to the second norm. The numerical results confirm that the CE algorithm is much more efficient than the simple iteration algorithm whenever the Lipschitz constant is close to 1. We also compare it with the Newton-Raphson method. In some tests the Newton-Raphson method is more efficient than the CE method, especially when the problem size is large. However, the CE algorithm is an excellent method for low dimensional functions with discontinuities and/or low regularity. Our implementation can be downloaded from http://www.cs.utah.edu/∼sikorski/cea.

1. Introduction

Given a *domain* $Q \subseteq E$, where E is a complete metric space, and a function $f : Q \to Q$, the fixed-point problem consists in finding, if it exists, a point $\mathbf{x}^* \in Q$ such that $\mathbf{x}^* = f(\mathbf{x}^*)$. Many problems can be formulated as fixed-point problems. For example, a root-finding problem for nonlinear equations $f(\mathbf{x}^*) = 0$ can be rearranged as $g(\mathbf{x}^*) = f(\mathbf{x}^*) + \mathbf{x}^* = \mathbf{x}^*$, which is a fixed-point problem. The applications of fixed-point problems include economic equilibria [9, 35], game theory [9, 18, 34, 35, 55], boundary value problems [2, 4, 24, 54], and chaos and dynamical systems [13, 45]. In our paper we deal with finite-dimensional problems, with domains of the functions being closed Euclidean balls.

A number of fixed-point theorems have been derived in the last century. Each theorem is focused on a specific class of functions defined on a specific domain. Banach's fixed-point theorem [5] states that any contractive function of $\Re^n$ into itself has a unique fixed point. Banach demonstrated that the simple iteration (SI) algorithm, $\mathbf{x}_{i+1} = f(\mathbf{x}_i)$, generates a Cauchy sequence that converges to the fixed point of any such function. More general fixed-point theorems focused on continuous functions only. Brouwer demonstrated in [11] that any continuous function

Received by the editor October 3, 2008 and, in revised form, April 23, 2010.

2010 *Mathematics Subject Classification.* Primary 47H10, 65H10; Secondary 65Y20, 68Q25.

from a nonempty, convex and compact subset of $\Re^n$ into itself has at least one fixed point. However, Brouwer's proof using topological arguments was nonconstructive. In 1967, Scarf [33] developed a simplicial algorithm for approximating fixed points for Brouwer's maps from a simplex into itself. This was the first constructive proof of Brouwer's fixed-point theorem. Since then, fixed-point computation has become an intensive research area and many new algorithms have been proposed including restart methods [27, 28], homotopy methods [3, 16, 17, 52] and ellipsoid algorithms [25, 26, 23, 42, 47]. As we mentioned earlier, a fixed-point problem $\mathbf{x}^* = f(\mathbf{x}^*)$ can be transformed into a root-finding problem $g(\mathbf{x}^*) = f(\mathbf{x}^*) - \mathbf{x}^* = 0$. Therefore, effective root-finding algorithms such as Newton type methods can also be used to solve fixed-point problems (see Allgower and Georg [3]).

Algorithms dealing with nonlinear fixed-point computation are iterative methods. We consider algorithms that are based on function evaluations, and assume that at least one function evaluation is required at each iteration. Most CPU time is usually consumed by function evaluations. Therefore, we can define the worst case *cost* of an algorithm, for a class of functions $\mathcal{F}$, as the maximum number of function evaluations required to achieve prescribed precision for all functions in $\mathcal{F}$. The problem *complexity* is defined as the minimal cost among all possible algorithms for the class $\mathcal{F}$. An algorithm is almost *optimal* if and only if its cost is almost minimal. The complexity depends on the class of functions and the selected error criterion, as shown in [8, 21, 40, 41, 43]. We consider contractive functions with contraction factor ρ close to 1, nonexpanding functions ($\rho = 1$), and then a larger class of functions that are contractive/nonexpanding only in the direction of fixed points (but may be globally expanding); see Boonyasiriwat et al. [8] and Vassin and Eremin [50]. The contraction/nonexpansion property is defined with respect to the *Euclidean* norm. The ε-approximations of fixed points are obtained by using the *absolute* or *residual* error criteria.

In this paper we present a new implementation of the Circumscribed Ellipsoid (CE) algorithm that does not require the dimensional deflation procedure of Tsay [47], and as a consequence has the cost lower by a factor of n. Our CE algorithm enjoys almost optimal cost $O(2n^2 \ln \frac{1}{\varepsilon})$ for obtaining residual solutions $\mathbf{x} : ||f(\mathbf{x}) - \mathbf{x}||_2 \leq \varepsilon$ of nonexpanding and directionally nonexpanding functions, and exhibits the cost $O(2n^2(\ln \frac{1}{\varepsilon} + \ln \frac{1}{1-\rho}))$ for obtaining absolute solutions $\mathbf{x} : ||\mathbf{x} - \mathbf{x}^*||_2 \leq \varepsilon$ for contractive and directionally contractive functions with factor $\rho < 1$ [8, 23]. We outline several numerical tests and compare the CE algorithm with simple iteration and Newton-type methods. The implementation of Newton-type methods called TENSOLVE [10] was utilized in this work. We do not directly compare our software with HOMPACK [52] or MINPACK [29] since they require differentiable functions and analytic formulas for Jacobians. Moreover, the test functions specified in those collections do not belong to our class, since in their fixed-point reformulations they are expanding in the direction of fixed points.

We remark that the ellipsoid algorithms are not applicable for the infinity-norm case. In [36] we developed a Bisection Envelope Fixed-point algorithm (BEFix), and in [37] we developed a Bisection Envelope Deep-cut Fixed-point algorithm (BEDFix) for computing fixed points of two-dimensional nonexpanding functions. Those algorithms exhibit the optimal complexity of $2\lceil \ln_2(1/\varepsilon) \rceil + 1$. We also developed a recursive fixed-point algorithm (PFix) for computing fixed points of n-dimensional nonexpanding functions with respect to the infinity norm (see Shellman and Sikorski

[38, 39]). We note that the complexity of finding ε-residual solutions for globally expanding functions with $\rho > 1$ is exponential $O((\rho/\varepsilon)^{(n-1)})$, as $\varepsilon \to 0$, [14, 21].

2. Problem formulation

Given the domain $B^n = \{\mathbf{x} \in \Re^n | \ ||\mathbf{x}|| \leq 1\}$, the n-dimensional *Euclidean* unit ball, we consider the class of Lipschitz continuous functions

$$\mathcal{B}^n_{\rho\leq 1} \equiv \{f : B^n \to B^n : ||f(\mathbf{x}) - f(\mathbf{y})|| \leq \rho||\mathbf{x} - \mathbf{y}||, \forall \mathbf{x}, \mathbf{y} \in B^n\}, \tag{2.1}$$

where $n \geq 2, ||\cdot|| = ||\cdot||_2$, and $0 < \rho \leq 1$. In the case when $0 < \rho < 1$, the class of functions is denoted by $\mathcal{B}^n_{\rho<1}$. The existence of fixed points of functions in $\mathcal{B}^n_{\rho\leq 1}$ is assured by the Brouwer's theorem.

In this article, we present the circumscribed ellipsoid (CE) algorithm that, for every $f \in \mathcal{B}^n_{\rho<1}$, computes an *absolute ε-approximation* $\mathbf{x}$ to $\mathbf{x}^*$, $||\mathbf{x} - \mathbf{x}^*|| \leq \varepsilon$, and for every $f \in \mathcal{B}^n_{\rho\leq 1}$, computes a *residual ε-approximation* $\mathbf{x}$, $||f(\mathbf{x}) - \mathbf{x}|| \leq \varepsilon$.

We also extend the applicability of the CE algorithm to larger classes of functions considered by Vassin and Eremin [50]. We stress that the complexity bounds for the CE algorithm do not change in this case. Those larger classes were investigated for problems defined by differential and integral equations originating in geophysics, atmospheric research, material science, and image deblurring [1, 31, 48, 49, 51]. These problems were effectively solved by Feyer-type iterative methods and/or some general optimization techniques; however, no formal complexity bounds were derived. These classes are defined by

$$\begin{aligned}\mathcal{B}^\alpha_{\rho\leq 1} \equiv \{f : B^n \to B^n :& \text{ the set of fixed points of } f, S(f) \neq \emptyset,\\ &\text{and } \forall \mathbf{x} \in B^n, \text{ and } \forall \alpha \in S(f), \text{ we have } ||f(\mathbf{x}) - f(\alpha)|| \leq \rho||\mathbf{x} - \alpha||\},\end{aligned} \tag{2.2}$$

where $n \geq 2$, $||\cdot|| = ||\cdot||_2$, and $\rho \leq 1$. We note that the functions in $\mathcal{B}^\alpha_{\rho\leq 1}$ may be expanding globally, and therefore the class $\mathcal{B}^n_{\rho\leq 1}$ is a proper subclass of $\mathcal{B}^\alpha_{\rho\leq 1}$.

3. Preliminary results

The following lemma is the basis of the *circumscribed ellipsoid* algorithm [7, 40, 41, 42, 47]. The proof of this lemma can be found in [41, 42]; see also [30].

Lemma 3.1. *Let $f \in \mathcal{B}^n_{\rho<1}$. Suppose that $A \subseteq B^n$ contains the fixed-point $\mathbf{x}^*$. Then, for every $\mathbf{x} \in A, \mathbf{x}^* \in A \cap B^n(\mathbf{c}, \gamma)$, where $B^n(\mathbf{c}, \gamma)$ is the ball with center*

$$\mathbf{c} = \mathbf{x} + \frac{1}{1-\rho^2}(f(\mathbf{x}) - \mathbf{x})$$

and radius

$$\gamma = \frac{\rho}{1-\rho^2}||f(\mathbf{x}) - \mathbf{x}||.$$

We stress that this lemma holds for the Euclidean norm.

The above lemma yields

Corollary 3.2. *Let $f \in \mathcal{B}^n_{\rho<1}$. If*

$$||f(\mathbf{x}) - \mathbf{x}|| \leq \frac{1-\rho^2}{\rho}\varepsilon, \tag{3.1}$$

then $\mathbf{x} + (f(\mathbf{x}) - \mathbf{x})/(1-\rho^2)$ is an absolute ε-approximation to the fixed-point $\mathbf{x}^*$.

The idea of CE algorithm is to construct a sequence of ellipsoids decreasing in volume, and converging to a fixed point. More precisely, we let $A = E_i$ be the i-th ellipsoid in that sequence. We let $\mathbf{x} = \mathbf{x}_i$ be the center of E_i. Then we define the half-space H by

$$H = \{\mathbf{y} \in \Re^n : \langle f(\mathbf{x}) - \mathbf{b}, \mathbf{y} - \mathbf{b}\rangle \ \geq 0\}\,, \tag{3.2}$$

where $\mathbf{b} = (f(\mathbf{x})+\rho\mathbf{x})/(1+\rho)$. We observe that the boundary of H is the supporting hyperplane for $B^n(\mathbf{c}, \gamma)$ at point $\mathbf{b}$. Then an immediate conclusion from Lemma 3.1 with $A = E_i$ is

Corollary 3.3. *If the assumptions of Lemma* 3.1 *hold, then*

$$\mathbf{x}^* \in H \cap E_i. \tag{3.3}$$

In the next step of the CE algorithm we construct a new ellipsoid E_{i+1} of smallest volume containing the set $H \cap E_i$.

The following lemma exhibits upper bounds on the number of iterations of the CE algorithm. The proof of this lemma can be found in [8, 23].

Lemma 3.4. *For any $\delta \in (0, 1)$, $f \in \mathcal{B}^n_{\rho\leq 1}$ with $\rho \leq 1$, the CE algorithm requires at most $i = \left\lceil 2n(n+1) \cdot \ln\left(\frac{2+\delta}{\delta}\right)\right\rceil$ iterations to compute an iterate $\mathbf{x}_i \in \Re^n$ such that $||f(\mathbf{x}_i) - \mathbf{x}_i|| \leq \delta$, as $\delta \to 0$.*

We remark that the same bounds hold for the larger class of directionally-nonexpanding functions [8]. We also remark that the bound obtained in the above lemma is better by a factor of n than the bound obtained in [47].

As a direct corollary from this lemma we get:

Corollary 3.5. *If $\rho < 1$, the CE algorithm finds an ε-approximation $\mathbf{x}_i$ of the fixed point $\mathbf{x}^*$ in the absolute sense, $||\mathbf{x}_i - \mathbf{x}^*||_2 \leq \varepsilon$, within $i = O(2n^2(\ln(1/\varepsilon) + \ln(1/(1-\rho)))$ iterations, as $\varepsilon \to 0$.*

We finally note that the number of iterations of the simple iteration algorithm to compute an absolute ε-approximation to the fixed point is $\lceil \ln(1/\varepsilon)/\ln(1/\rho)\rceil$ for contractive functions with $\rho < 1$.

4. Implementation

Each iteration of the CE algorithm requires constructing a minimum-volume ellipsoid that encloses the set containing a fixed point of function f. This set is an intersection of the previous ellipsoid and a half-space H from equation (3.2). This computation is equivalent to computing an eigensystem that defines the ellipsoid. However, directly constructing the updated ellipsoids as in Figure 1 is numerically unstable [6]. Tsay [47] overcame this problem by using a rank-one modification of the symmetric eigensystem presented in [12]. The method is as follows. Each ellipsoid E_i is represented by a symmetric positive definite matrix A_i and the center $\mathbf{x}_i$. The formula for updating ellipsoids is given by

$$A_{i+1} = \beta^2(A_i - (1 - (\alpha/\beta)^2)\mathbf{z}\mathbf{z}^T), \tag{4.1}$$

where α, β, and $\mathbf{z}$ are defined in Figure 1. Suppose the eigendecomposition $A_i = Q_i D_i Q_i^T$ is known. Let $\mathbf{b} = Q_i^T\mathbf{z}$ and $\tau = (1 - (\alpha/\beta)^2)$. Then, (4.1) can be written as

$$A_{i+1} = \beta^2 Q_i(D_i - \tau\mathbf{b}\mathbf{b}^T)Q_i^T.$$

By finding the eigendecomposition

$$D_i - \tau \mathbf{b}\mathbf{b}^T = \hat{Q}\hat{D}\hat{Q}^T,$$

we obtain $A_{i+1} = Q_{i+1}D_{i+1}Q_{i+1}^T$, where $D_{i+1} = \beta^2\hat{D}$ and $Q_{i+1} = Q_i\hat{Q}$. To apply this method, A_i is represented by Q_i and D_i. Initially, $A_0 = I_{n\times n}$, so we assign $Q_0 = I_{n\times n}$ and $D_0 = I_{n\times n}$. The CE algorithm iteratively updates minimal-volume ellipsoids and eigensystems at the same time. Figure 2 presents the numerically stable implementation of the CE algorithm from Figure 1.

```
input $\{\rho \le 1; \varepsilon > 0; i_{\max}$ (maximum number of iterations); function $f \in B^{\alpha}_{\rho\le 1};\}$
$i := 0;$ $\mathbf{x}_0 := 0;$ $A_0 := I_{n\times n};$
while $i \le i_{\max}$ do begin
    if $\sqrt{d_1(A_i)} \le \varepsilon$ then          (1)   ($d_1$ is the largest eigenvalue of $A_i$)
        return $\mathbf{x}_i$ as an absolute $\varepsilon$-approximation;
    $\mathbf{a} := \mathbf{x}_i - f(\mathbf{x}_i);$
    if $||\mathbf{a}|| \le (1-\rho^2)\varepsilon/\rho$ then          (2)
        return $\mathbf{x}_i - \mathbf{a}/(1-\rho^2)$ as an absolute $\varepsilon$-approximation;
    if $||\mathbf{a}|| \le \varepsilon$ then          (3)
        return $\mathbf{x}_i$ as a residual $\varepsilon$-approximation;
    $\xi := \mathbf{a}^T\mathbf{a}/((1+\rho)\sqrt{\mathbf{a}^T A_i \mathbf{a}});$
    $\alpha := n(1-\xi)/(n+1);$
    $\beta := \sqrt{\frac{n^2}{n^2-1}(1-\xi^2)};$
    $\gamma := (n\xi+1)/(n+1);$
    $\mathbf{z} := A_i\mathbf{a}/\sqrt{\mathbf{a}^T A_i \mathbf{a}};$
    $\mathbf{x}_{i+1} := \mathbf{x}_i - \gamma\mathbf{z};$
    $A_{i+1} := \beta^2(A_i - (1-\alpha^2/\beta^2)\mathbf{z}\mathbf{z}^T);$
    $i := i+1;$
end
if $i = i_{\max}$ then
    return failed to compute $\varepsilon$-approximation in $i_{\max}$ iterations
```

Figure 1. The circumscribed ellipsoid algorithm

Our current implementation can be downloaded from http://www.cs.utah.edu/∼sikorski/cea. The numerical stability of our implementation is the result of numerically stable updating of eigensystems as described below.

In the implementation the diagonal matrix $D = D_i$ is stored as a vector $[d_1, d_2, \ldots, d_n]$ and we assume that $d_1 \ge d_2 \ge \cdots \ge d_n$. The rank1 subroutine is the implementation of the rank-one modification of the symmetric eigensystem. The input parameters for this subroutine include $n, \tau, \mathbf{b}, D$, and $Q = Q_i$ and the outputs are the eigenvalues $\hat{D}$ and eigenvectors $\hat{Q}$ of the matrix $D - \tau\mathbf{b}\mathbf{b}^T$. This rank-one modification problem is well studied in numerical linear algebra [12, 15, 19, 20, 44].

Before the rank-one updating of the eigensystem is accomplished, a deflation process proceeds to avoid unnecessary computation. We implemented the deflation algorithm presented by Dongarra and Sorensen [15]. We utilized the LAPACK routines DLAED4, DLAED5 and DLAED6, that are the latest implementation of the algorithm for computing updated eigenvalues. These routines were first implemented by Rutter [32] and then modified by Tisseur and Dongarra [46]. Then the

```
input $\left\{\rho \leq 1; \varepsilon > 0; i_{\max}\,(\text{maximum number of iterations}); \text{function } f \in B^{\alpha}_{\rho \leq 1}\right\}$;
$i := 0;\ \mathbf{c} := 0;\ Q := I_{n\times n}\ D := I_{n\times n};$
while $i \leq i_{\max}$ do begin
    if $\sqrt{d_1} \leq \varepsilon$ then                                (1)
        return $\mathbf{c}$ as an absolute $\varepsilon$-approximation;
    $\mathbf{a} := \mathbf{c} - f(\mathbf{c});$
    if $||\mathbf{a}|| \leq (1-\rho^2)\varepsilon/\rho$ then                  (2)
        return $\mathbf{c} - \mathbf{a}/(1-\rho^2)$ as an absolute $\varepsilon$-approximation;
    if $||\mathbf{a}|| \leq \varepsilon$ then                           (3)
        return $\mathbf{x}_i$ as a residual $\varepsilon$-approximation;
    $\mathbf{u} := Q^T\mathbf{a};$
    $\omega := \sqrt{\mathbf{u}^T D\mathbf{u}};$
    $\xi := \mathbf{a}^T\mathbf{a}/((1+\rho)\omega);$
    $\alpha := n(1-\xi)/(n+1);$
    $\beta := \sqrt{\frac{n^2}{n^2-1}(1-\xi^2)};$
    $\gamma := (n\xi+1)/(n+1);$
    $\mathbf{z} := QD\mathbf{u}/\omega;$
    $\mathbf{c} := \mathbf{c} - \gamma\mathbf{z};$
    $\tau := 1 - \alpha^2/\beta^2;$
    $\mathbf{b} := Q^T\mathbf{z};$
    $[\hat{D}, \hat{Q}] := \mathrm{rank1}(n, \tau, \mathbf{b}, D, Q);$
    $D := \beta^2\hat{D};$
    $Q := Q\hat{Q};$
    $i := i+1;$
end
if $i = i_{\max}$ then
    return failed to compute $\varepsilon$-approximation in $i_{\max}$ iterations
```

FIGURE 2. Stable implementation of the circumscribed ellipsoid algorithm

updated eigenvalues are used to compute the updated eigenvectors by the numerically stable algorithm presented by Gu and Eisenstat [20]. We shall refer to this algorithm as the GE/LAPACK algorithm. The updated eigenvalues and eigenvectors are used to update D and Q as shown in Figure2.

In our implementation, we use explicit computation of updated eigenvalues and eigenvectors for the two-dimensional cases, and use the GE/LAPACK algorithm for higher dimensions. We note that in the worst case the arithmetic cost of each iteration of this algorithm is $O(n^3)$ that is the result of updating of the eigenvector matrix. This cost could be lower than $O(n^2)$ in the average/best case. This suggests that in the worst case this algorithm is efficient for relatively small values of n.

The user specified termination parameter ε is modified in our code according to the following rules. If the function evaluations are carried out in single precision, then

$$\varepsilon = \max(\varepsilon, Macheps(1)), \tag{4.2}$$

and if in double precision, then

$$\varepsilon = \max(\varepsilon, Macheps(2)), \tag{4.3}$$

where $Macheps(j), j = 1, 2$ are, respectively, machine precision in single and double floating point representations.

If the absolute termination is to be used (case $\rho < 1$) the user may request to modify the ε by

$$\varepsilon = \max(\varepsilon, \frac{Macheps(j)}{1-\rho}). \tag{4.4}$$

This is justified by the absolute sensitivity of the problem that is characterized by the condition number equal to $\frac{1}{1-\rho}$.

5. Numerical results

In this section we compare the numerical results for several test functions using the CE, SI, and Newton-Raphson (NR) methods. The NR method is used to solve corresponding root-finding problems of the form $f(\mathbf{x}) - \mathbf{x} = \mathbf{0}$. We stress that the CE algorithm is designed to converge globally for noncontinuous, nondifferentiable functions, whenever Newton's method requires C^1 smoothness and starting points that are sufficiently close to the solution. We include the tests of Newton's method for illustrative comparison. We utilize the ACM TOMS implementation of Newton's method [10]. In each table for each algorithm (CE, SI, NR) we exhibit the total CPU time (in seconds), number of iterations and the number of the stopping criterion that resulted in termination. For the stopping criterion, the numbers 1 and 2 indicate absolute termination and number 3, residual termination (see Figure 2). In addition for the CE and SI, we exhibit in parentheses the upper bounds on the number of iterations (see Lemma 3.4, where $\delta = \varepsilon(1-\rho)$ for the absolute termination case). The speedup factors that represent the ratio in CPU time when using the CE algorithm instead of the SI or NR algorithm are also included. All tests are carried out on a Redhat Linux operating system on an AMD Athlon 64 Processor 3400+ machine.

In the tests of functions 2 and 3, some initial balls are not the unit balls. In these cases, the problems are defined on a general ball $B^n(\mathbf{c}, \gamma)$. They can be transformed to the unit ball $B^n(\mathbf{0}, 1)$ as follows.

Let $f : B^n(\mathbf{c}, \gamma) \to B^n(\mathbf{c}, \gamma)$ denote the original function defined on a ball $B^n(\mathbf{c}, \gamma)$. The modified function $\overline{f} : B^n(\mathbf{0}, 1) \to B^n(\mathbf{0}, 1)$ defined on a unit ball $B^n(\mathbf{0}, 1)$ is

$$\overline{f}(\mathbf{x}) = \frac{1}{\gamma}\left(f(\mathbf{x} \cdot \gamma + \mathbf{c}) - \mathbf{c}\right). \tag{5.1}$$

It turns out that if $f \in \mathcal{B}^n_{\rho \le 1}$, then $\overline{f} \in \mathcal{B}^n_{\rho \le 1}$. Indeed, if $\mathbf{x}_1, \mathbf{x}_2 \in B^n(\mathbf{0}, 1)$, then

$$\begin{aligned} ||\overline{f}(\mathbf{x}_1) - \overline{f}(\mathbf{x}_2)|| &= \frac{1}{\gamma}||f(\mathbf{x}_1 \cdot \gamma + \mathbf{c}) - f(\mathbf{x}_2 \cdot \gamma + \mathbf{c})|| \\ &\le \frac{1}{\gamma} \cdot \rho||\gamma \cdot (\mathbf{x}_1 - \mathbf{x}_2)|| \\ &= \rho||\mathbf{x}_1 - \mathbf{x}_2||, \end{aligned}$$

i.e., they satisfy the Lipschitz condition with the same ρ and

$$||\overline{f}(\mathbf{x})|| = \frac{1}{\gamma}||f(\mathbf{x}\cdot\gamma+\mathbf{c})-\mathbf{c}|| \le \frac{1}{\gamma}\cdot\gamma = 1,$$

i.e., $\overline{f} : B^n(\mathbf{0},1) \to B^n(\mathbf{0},1)$.

In all tables we exhibit the user specified ε. In all cases these ε's were then modified according to equations (4.2)-(4.4); however, in only a few cases their values were changed.

Test 1. This test function is a simple affine mapping $f : \Re^n \to \Re^n$ given by

$$T_1 : \qquad f(\mathbf{x}) = \rho\mathbf{x} + (1-\rho)\mathbf{s}, \tag{5.2}$$

where the constant vector $\mathbf{s}$ is randomly chosen from $B^n(\mathbf{0},1)$. Obviously, $\mathbf{s}$ is the unique fixed point of f for $\rho < 1$. We select this affine function since it is an almost worst-case test for the SI algorithm, as well as it shows how much faster the CE method can be than the Newton-Raphson algorithm. For example, Table 2 shows that for $n = 5$, 17–19 iterations of the CE method are 3.7–4.6 times faster than the corresponding 2 iterations of the NR method. Table 1 shows the results when $n = 5$ and the Lipschitz constant is varied from 0.9–0.999999 using the SI and NR methods, while Table 2 shows the results of the CE method.

TABLE 1. T_1: $n = 5, \varepsilon = 10^{-6}$ and $\mathbf{x}^* = [0.1, 0.3, 0.4, 0.1, 0.2]^T$

ρ	SI	NR
$1-10^{-1}$	7.332×10^{-6}, 119 (132), 2	2.620×10^{-5}, 2, 2
$1-10^{-2}$	7.299×10^{-5}, 1247 (1375), 2	3.094×10^{-5}, 3, 2
$1-10^{-3}$	7.199×10^{-4}, 12531 (13809), 2	2.663×10^{-5}, 2, 2
$1-10^{-4}$	7.299×10^{-3}, 125361 (138149), 2	1.694×10^{-5}, 2, 2
$1-10^{-5}$	7.149×10^{-2}, 1253671 (1381545), 2	1.647×10^{-5}, 2, 2
$1-10^{-6}$	7.139×10^{-1}, 12536780 (13815504), 2	1.660×10^{-5}, 2, 2

TABLE 2. T_1: $n = 5, \varepsilon = 10^{-6}$ and $\mathbf{x}^* = [0.1, 0.3, 0.4, 0.1, 0.2]^T$

ρ	CE	Speedup wrt SI	Speedup wrt NR
$1-10^{-1}$	6.479×10^{-6}, 17 (1009), 2	1.1	4.0
$1-10^{-2}$	6.799×10^{-6}, 18 (1147), 2	10.7	4.6
$1-10^{-3}$	7.159×10^{-6}, 19 (1285), 2	100.6	3.7
$1-10^{-4}$	2.803×10^{-5}, 30 (1424), 2	260.4	0.6
$1-10^{-5}$	4.291×10^{-4}, 123 (1562), 2	166.6	0.04
$1-10^{-6}$	7.570×10^{-5}, 41 (1700), 2	943.1	0.2

Test 2. This test function is a complex function from [22], given by

$$T_2 : \qquad f(\mathbf{z}) = g(g(z)), \tag{5.3}$$

where

$$g(z) = \frac{z^2 + c\cos^2 z}{z + \sin z \cos z}, \tag{5.4}$$

z is a complex variable and c is a complex constant. We include this test function since it comes from a practical electrical engineering application and exhibits slow convergence of the SI and NR algorithms. This complex test function is considered as a two-dimensional real function $f : \Re^2 \to \Re^2$, i.e., $n = 2$, but we evaluate the function as a one-dimensional complex function. The problem is tested with two values of the constant $c : c = 1.025$ and $c = \pi/4 + 1.2 + i(\pi - 1.17)$. The fixed points of this problem are $[0, 0.69032769]^T$ and $[2.14062, -2.50683]^T$, respectively. The results obtained by using the SI and NR method are exhibited in Table 3, while Table 4 shows the results from the CE method. In the case of ball B_1 and $\varepsilon = 10^{-6}$, the NR method exhibits oscillatory behavior with very slow convergence to the fixed point. This might be due to numerical instability of this specific implementation of the NR method for this test case.

TABLE 3. $T_2 : B_1 = B^2([0.0, 0.1]^T, 1)$, $B_2 = B^2([2.2, -2.2]^T, 1)$, $c_1 = 1.025$, $\rho_1 = 0.9989885$, $c_2 = \pi/4 + 1.0 + i(\pi - 1.17)$, and $\rho_2 = 0.9984$

ε	ρ	c	Ball	SI	NR
10^{-2}	ρ_1	c_1	B_1	2.143×10^{-4}, 583 (4551), 2	7.120×10^{-5}, 8, 2
10^{-3}	ρ_1	c_1	B_1	8.513×10^{-4}, 2313 (6826), 2	7.762×10^{-5}, 9, 2
10^{-4}	ρ_1	c_1	B_1	1.679×10^{-3}, 4572 (9102), 2	7.739×10^{-5}, 9, 2
10^{-5}	ρ_1	c_1	B_1	2.510×10^{-3}, 6846 (11377), 2	8.356×10^{-5}, 10, 2
10^{-6}	ρ_1	c_1	B_1	3.345×10^{-3}, 9122 (13652), 2	3.551×10^{-1}, 41392, 2
10^{-2}	ρ_2	c_2	B_2	1.379×10^{-3}, 2877 (2876), 1	3.487×10^{-5}, 4, 2
10^{-3}	ρ_2	c_2	B_2	2.065×10^{-3}, 4315 (4314), 1	3.490×10^{-5}, 4, 2
10^{-4}	ρ_2	c_2	B_2	2.889×10^{-3}, 5753 (5752), 1	4.124×10^{-5}, 5, 2
10^{-5}	ρ_2	c_2	B_2	3.601×10^{-3}, 7191 (7190), 1	4.134×10^{-5}, 5, 2
10^{-6}	ρ_2	c_2	B_2	4.353×10^{-3}, 8629 (8628), 1	3.505×10^{-5}, 4, 2

Test 3. We define T_3 as a function transforming the whole $\Re^2$ into $[1 - \rho/2, 1]^2$ as a periodic, parabolic function from [42], given by

$$T_3 : \qquad f(x_1, x_2) = [f_1(x_1, x_2), f_2(x_1, x_2)]^T, \tag{5.5}$$

where

$$f_i(x_1, x_2) = \frac{\rho}{2}(x_i - 2m)^2 + 1 - \frac{\rho}{2} \tag{5.6}$$

for $i = 1, 2$ whenever x_i satisfies $2m - 1 < x_i \leq 2m + 1$ for some integer m. As such, T_3 is contractive with the factor ρ $(0 < \rho < 1)$ on the whole $\Re^2$ and has a unique fixed point at $[1, 1]^T$, at which each component of T_3 is not continuously differentiable (since the right and left partial derivatives are $-\rho$ and $+\rho$). The results from the SI and NR methods are exhibited in Table 5. The results from the CE method are exhibited in Table 6. It is worth noting that the CE algorithm can terminate with absolute error criterion even when $\rho = 1$; i.e., when the function becomes nonexpanding.

TABLE 4. $T_2 : B_1 = B^2([0.0, 0.1]^T, 1)$, $B_2 = B^2([2.2, -2.2]^T, 1)$, $c_1 = 1.025$, $\rho_1 = 0.9989885$, $c_2 = \pi/4 + 1.0 + i(\pi - 1.17)$, and $\rho_2 = 0.9984$

ε	ρ	c	Ball	CE	Speedup wrt SI	Speedup wrt NR
10^{-2}	ρ_1	c_1	B_1	2.670×10^{-6}, 3 (147), 2	80.3	26.7
10^{-3}	ρ_1	c_1	B_1	5.440×10^{-6}, 7 (174), 2	156.5	14.3
10^{-4}	ρ_1	c_1	B_1	1.034×10^{-5}, 14 (202), 2	162.4	7.5
10^{-5}	ρ_1	c_1	B_1	1.455×10^{-5}, 20 (230), 2	172.5	5.7
10^{-6}	ρ_1	c_1	B_1	1.869×10^{-5}, 26 (257), 2	179.0	1.9×10^4
10^{-2}	ρ_2	c_2	B_2	1.517×10^{-5}, 14 (141), 1	90.9	2.3
10^{-3}	ρ_2	c_2	B_2	2.158×10^{-5}, 20 (169), 1	95.7	1.6
10^{-4}	ρ_2	c_2	B_2	2.699×10^{-5}, 25 (197), 1	107.0	1.5
10^{-5}	ρ_2	c_2	B_2	3.350×10^{-5}, 31 (224), 1	107.5	1.2
10^{-6}	ρ_2	c_2	B_2	3.981×10^{-5}, 37 (252), 1	109.3	0.9

TABLE 5. $T_3 : B_1 = B^2([0, 0]^T, 2)$, $B_2 = B^2([0.1, 0.2]^T, 2)$, $\rho_1 = 1 - 10^{-3}$, $\rho_2 = 1 - 10^{-5}$, and $\rho_3 = 1 - 10^{-15}$

ε	ρ	Ball	SI	NR
10^{-3}	ρ_1	B_1	6.060×10^{-5}, 1120 (7598), 2	2.361×10^{-5}, 3, 2
10^{-3}	ρ_2	B_1	6.389×10^{-4}, 11874 (760087), 2	6.337×10^{-5}, 7, 2
10^{-4}	ρ_1	B_2	1.500×10^{-4}, 2768 (9899), 2	4.017×10^{-5}, 5, 2
10^{-4}	ρ_2	B_2	2.020×10^{-3}, 37376 (990344), 2	7.131×10^{-5}, 7, 2
10^{-6}	ρ_2	B_2	1.500×10^{-2}, 277744 (1450859), 2	7.131×10^{-5}, 7, 2
$10^{-6}*$	ρ_3	B_2	5.893×10^{0}, 1.1×10^{8} (1.2×10^{16}), 2	2.954×10^{-5}, 3, 2
$10^{-6}*$	1	B_2	9.274×10^{-5}, 1673, 3	2.977×10^{-5}, 3, 2

TABLE 6. $T_3 : B_1 = B^2([0,0]^T, 2)$, $B_2 = B^2([0.1, 0.2]^T, 2)$, $\rho_1 = 1 - 10^{-3}$, $\rho_2 = 1 - 10^{-5}$, and $\rho_3 = 1 - 10^{-15}$

ε	ρ	Ball	CE	Speedup wrt SI	Speedup wrt NR
10^{-3}	ρ_1	B_1	2.240×10^{-5}, 34 (183), 2	2.7	1.1
10^{-3}	ρ_2	B_1	3.030×10^{-5}, 45 (238), 1	21.1	2.9
10^{-4}	ρ_1	B_2	3.160×10^{-5}, 47 (211), 2	4.7	1.3
10^{-4}	ρ_2	B_2	3.590×10^{-5}, 54 (266), 2	56.3	2.0
10^{-6}	ρ_2	B_2	5.260×10^{-5}, 79 (321), 2	285.2	1.4
$10^{-6}*$	ρ_3	B_2	5.780×10^{-5}, 87 (597), 1	1.02×10^5	0.5
$10^{-6}*$	1	B_2	5.830×10^{-5}, 87 (∞), 1	1.6 **	0.5

$*$ In these tests we removed the modification of ε according to equation (4.4), to test the limits of numerical precision of our algorithm. We note that the algorithm computed correct results even in the case $\rho = 1$.

$**$ Residual termination for SI and absolute termination for CE.

Test 4. This fixed-point test function is a saw-like, periodical function from [42], given by

$$T_4 : \qquad f(x_1, x_2) = \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} \tag{5.7}$$

where

$$f_i(x_1, x_2) = \min_{j=1,99} (\rho |x_i - m - 10^{-2} j| + i/3) \tag{5.8}$$

for $i = 1, 2$ and $m \leq x_i \leq m + 1$ and m is an arbitrary integer. This function has a unique fixed point at $[-0.01946, 0.75933]^T$. We select this function since it is nondifferentiable at many points. The results obtained by using the SI and NR methods are exhibited in Table 7. It is worth noting that the speedup factor is increased by 4 orders of magnitude when ρ is closer to 1. Table 8 shows the results from the CE method. It is also worth noting that for this test function the CE method is in all cases superior to the Newton-Raphson method.

TABLE 7. $T_4 : \varepsilon = 10^{-6}$, $\mathbf{x}^* = [-0.04, 0.74]^T$, $B_1 = B^2([0,0]^T, 1)$, $B_2 = B^2([0,0]^T, 2)$, and $B_3 = B^2([0.1, 0.2]^T, 2)$

ρ	Ball	SI	NR
$1 - 10^{-2}$	B_1	1.124×10^{-3}, 1109 (1375), 2	4.484×10^{-4}, 20, 2
$1 - 10^{-2}$	B_2	1.126×10^{-3}, 1109 (1375), 2	4.486×10^{-4}, 20, 2
$1 - 10^{-2}$	B_3	1.152×10^{-3}, 1141 (1375), 2	2.587×10^{-4}, 14, 2
$1 - 10^{-6}$	B_1	1.395×10^{1}, 13815504 (13815504), 1	2.121×10^{-4}, 13, 3
$1 - 10^{-6}$	B_2	1.394×10^{1}, 13815504 (13815504), 1	2.127×10^{-4}, 13, 3
$1 - 10^{-6}$	B_3	1.396×10^{1}, 13815504 (13815504), 1	2.847×10^{-4}, 16, 3

TABLE 8. $T_4 : \varepsilon = 10^{-6}$, $\mathbf{x}^* = [-0.04, 0.74]^T$, $B_1 = B^2([0,0]^T, 1)$, $B_2 = B^2([0,0]^T, 2)$, and $B_3 = B^2([0.1, 0.2]^T, 2)$

ρ	Ball	CE	Speedup wrt SI	Speedup wrt NR
$1 - 10^{-2}$	B_1	6.159×10^{-5}, 36 (230), 1	18.3	7.3
$1 - 10^{-2}$	B_2	6.949×10^{-5}, 40 (238), 1	16.2	6.5
$1 - 10^{-2}$	B_3	7.039×10^{-5}, 41 (238), 1	16.4	3.7
$1 - 10^{-6}$	B_1	6.069×10^{-5}, 36 (340), 1	2.299×10^5	3.5
$1 - 10^{-6}$	B_2	7.049×10^{-5}, 41 (349), 1	1.978×10^5	3.0
$1 - 10^{-6}$	B_3	7.069×10^{-5}, 41 (349), 1	1.975×10^5	4.0

Test 5. This fixed-point test function is a nine-dimensional function from [53], given by

$$T_5 : \qquad f(\mathbf{x}) = x - tF(\mathbf{x}) \tag{5.9}$$

where

$$F(\mathbf{x}) = A\mathbf{x} + G(\mathbf{x}) - \underline{,} \tag{5.10}$$

$$A = \begin{bmatrix} B & -I & 0 \\ -I & B & -I \\ 0 & -I & B \end{bmatrix}, \tag{5.11}$$

$$B = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \tag{5.12}$$

$$G(x) = \frac{1}{8}\left[g\left(\frac{1}{4}, \frac{1}{4}, x_1\right), g\left(\frac{2}{4}, \frac{2}{4}, x_2\right), \ldots, g\left(\frac{3}{4}, \frac{3}{4}, x_9\right)\right]^T, \tag{5.13}$$

and

$$\begin{aligned} \mathbf{b} = \quad & \left[\phi\left(0, \tfrac{1}{4}\right) + \phi\left(\tfrac{1}{4}, 0\right), \phi\left(\tfrac{2}{4}, 0\right), \phi\left(\tfrac{3}{4}, 0\right) + \phi\left(1, \tfrac{1}{4}\right), \phi\left(0, \tfrac{2}{4}\right), 0, \phi\left(1, \tfrac{2}{4}\right), \right. \\ & \left. \phi\left(0, \tfrac{3}{4}\right) + \phi\left(\tfrac{1}{4}, 1\right), \phi\left(\tfrac{2}{4}, 1\right), \phi\left(\tfrac{3}{4}, 1\right) + \phi\left(1, \tfrac{3}{4}\right)\right]^T, \end{aligned}$$

with

$$g(s, t, u) = \frac{t}{s} + |u - 1|^3, \qquad \phi(s, t) = \sin(0.5\pi st).$$

We choose the value of $t = \sqrt{L^2(1-\rho^2)}/L^2$ with $L = 0.8234$ and select the Lipschitz constant ρ close to 1. This function has a unique fixed point at $[8.89 \times 10^{-4}, 6.61 \times 10^{-2}, 0.184, 6.61 \times 10^{-2}, 0.209, 0.415, 0.184, 0.415, 0.690]^T$. This choice implies that f is contractive [47, 53]. We include this function since it has a higher dimension than the other test functions and also comes from an application of numerical solutions of partial differential equations. The numerical results are exhibited in Table 9 and Table 10. It turns out that the SI algorithm is faster than the CE algorithm when ρ is not too close to 1. This is because the cost of each iteration of the CE algorithm increases with the dimension as $O(n^3)$. However, when ρ is close to 1, the CE algorithm is much faster than the SI algorithm although it is much slower than the NR algorithm.

TABLE 9. T_5: $\varepsilon = 10^{-6}$ and ρ is varied from $1 - 10^{-2}$ to $1 - 10^{-15}$

ρ	SI	NR
$1 - 10^{-2}$	2.476×10^{-5}, 70 (1375), 2	1.333×10^{-4}, 5, 2
$1 - 10^{-3}$	8.549×10^{-5}, 256 (13809), 2	1.134×10^{-4}, 4, 3
$1 - 10^{-4}$	2.780×10^{-4}, 883 (138149), 2	1.613×10^{-4}, 6, 2
$1 - 10^{-5}$	1.046×10^{-3}, 2990 (1381545), 2	1.585×10^{-4}, 6, 2
$1 - 10^{-6}$	3.290×10^{-3}, 10041 (13815504), 2	1.566×10^{-4}, 6, 2
$1 - 10^{-7}$	1.140×10^{-2}, 33571 (138155099), 2	1.298×10^{-4}, 6, 2
$1 - 10^{-8}$	3.852×10^{-2}, 111858 (1.4×10^{9}), 2	1.460×10^{-4}, 6, 2
$1 - 10^{-9}$	1.156×10^{-1}, 371639 (1.4×10^{10}), 2	1.242×10^{-4}, 5, 2
$1 - 10^{-10}$	3.806×10^{-1}, 1229543 (1.4×10^{11}), 2	1.205×10^{-4}, 5, 2
$1 - 10^{-11}$	1.321×10^{0}, 4024132 (1.4×10^{12}), 2	1.192×10^{-4}, 5, 2
$1 - 10^{-12}$	3.815×10^{0}, 12391327 (1.4×10^{13}), 2	1.397×10^{-4}, 6, 2
$1 - 10^{-13}$	1.193×10^{1}, 38756581 (1.4×10^{14}), 2	1.446×10^{-4}, 6, 2
$1 - 10^{-14}$	3.654×10^{1}, 118682979 (1.4×10^{15}), 2	1.795×10^{-4}, 7, 2
$1 - 10^{-15}$	1.101×10^{2}, 357331990 (1.4×10^{16}), 2	1.954×10^{-4}, 8, 2

$*$ In these tests we removed the modification of ε according to equation (4.4), to test the limits of numerical precision of our algorithm.

TABLE 10. T_5: $\varepsilon = 10^{-6}$ and ρ is varied from $1 - 10^{-2}$ to $1 - 10^{-15}$

ρ	CE	Speedup wrt SI	Speedup wrt NR
$1 - 10^{-2}$	2.151×10^{-2}, 1534 (3441), 2	1.151×10^{-3}	6.197×10^{-3}
$1 - 10^{-3}$	2.922×10^{-2}, 2102 (3855), 2	2.926×10^{-3}	3.881×10^{-3}
$1 - 10^{-4}$	3.115×10^{-2}, 2258 (4270), 1	8.925×10^{-3}	5.178×10^{-3}
$1 - 10^{-5}$	3.224×10^{-2}, 2311 (4684), 1	0.03	4.916×10^{-3}
$1 - 10^{-6}$	3.234×10^{-2}, 2325 (5099), 1	0.1	4.842×10^{-3}
$1 - 10^{-7}$	3.245×10^{-2}, 2330 (5513), 1	0.4	4.000×10^{-3}
$1 - 10^{-8}$	3.245×10^{-2}, 2327 (5928), 1	1.2	4.499×10^{-3}
$1 - 10^{-9}$	3.226×10^{-2}, 2326 (6342), 1	3.6	3.850×10^{-3}
$1 - 10^{-10}$	3.247×10^{-2}, 2342 (6757), 1	11.7	3.711×10^{-3}
$1 - 10^{-11}$	3.167×10^{-2}, 2337 (7171), 1	41.7	3.764×10^{-3}
$1 - 10^{-12}$	3.229×10^{-2}, 2320 (7586), 1	118.1	4.326×10^{-3}
$1 - 10^{-13}$	3.137×10^{-2}, 2338 (8000), 1	380.3	4.609×10^{-3}
$1 - 10^{-14}$	3.024×10^{-2}, 2331 (8415), 1	1208.0	5.936×10^{-3}
$1 - 10^{-15}$	3.089×10^{-2}, 2337 (8829), 1	3564.0	6.326×10^{-3}

$*$ In these tests we removed the modification of ε according to equation (4.4), to test the limits of numerical precision of our algorithm.

TABLE 11. $T_6 : \mathbf{x}^* = [0.5, 0.5]^T, B_1 = B^2([0, 0.1]^T, 1)$

ε	SI	NR
10^{-2}	2×10^{-6}, 8, 3	3.7×10^{-5}, 2, 3
10^{-3}	3×10^{-6}, 33, 3	3.6×10^{-5}, 2, 3
10^{-4}	6×10^{-6}, 113, 3	5.1×10^{-5}, 4, 3
10^{-5}	1.7×10^{-5}, 369, 3	5.0×10^{-5}, 4, 3
10^{-6}	4.9×10^{-5}, 1181, 3	5.1×10^{-5}, 4, 3
10^{-7}	0.000166, 3751, 3	5.1×10^{-5}, 4, 3
10^{-8}	0.000494, 11881, 3	6.8×10^{-5}, 6, 3
10^{-9}	0.001750, 37594, 3	9.1×10^{-5}, 8, 3
10^{-10}	0.004969, 118908, 3	Failed
10^{-11}	0.016410, 376045, 3	Failed
10^{-12}	0.050150, 1189162, 3	Failed
10^{-13}	0.162500, 3761413, 3	Failed
10^{-14}	0.490700, 11898141, 3	Failed
10^{-15}	1.625000, 37578817, 3	Failed

Test 6. This fixed-point test function is given by

$$T_6 : \qquad f_i(\mathbf{x}) = g_i(\mathbf{x})^2 + \frac{1}{4} \tag{5.14}$$

where

$$g_i(\mathbf{x}) = \frac{1}{4} + \frac{x_i - \frac{1}{4}}{4||\mathbf{x} - [\frac{1}{4}, \frac{1}{4}]||_\infty}. \tag{5.15}$$

for $i = 1, 2$. This function has a unique fixed point at $[0.5, 0.5]^T$ and it is nondifferentiable at the fixed point. The results obtained by using the SI and NR methods are exhibited in Table 11. This test function has the Lipschitz constant $\rho = 1$. Therefore, we set up parameters so that both methods terminate with the residual error criterion. Table 12 shows the results from the CE method. It is worth noting that in this test function the CE method is up to 7 times faster than the NR method. In addition, the NR method fails when ε is equal to and less than 10^{-10}. We also observe that e.g. when $\varepsilon = 10^{-7}$ each iteration of the NR method is about 12 times more expensive than each CE iteration.

We select "rotational-type" functions 7 and 8 from the larger class of directionally nonexpanding functions. They have discontinuities on the boundary and/or inside the domain.

Test 7. This test function is in the larger class $\mathcal{B}^n_{\rho\leq 1}$, so it is nonexpanding ($\rho = 1$) in the direction of the unique fixed point $[0.5, 0.5]^T$. This is why for this test we utilize the residual stopping criterion. We observe that this function is discontinuous, and it is a rotation by angle $\theta = 10$ degrees around $[0.5, 0.5]^T$ of all points in the disk $D = \{(x_1, x_2) : r \leq R\}$, is a projection of any boundary point $\mathbf{x}, \mathbf{x} \neq [1/\sqrt{2}, 1/\sqrt{2}]^T$ of the unit circle C onto the point $[1/\sqrt{2}, 1/\sqrt{2}]^T$, and is a clockwise rotation around $[0.5, 0.5]^T$ of any interior point of $C - D$ onto the

TABLE 12. $T_6 : \mathbf{x}^* = [0.5, 0.5]^T, B_1 = B^2([0, 0.1]^T, 1)$

ε	CE	Speedup wrt SI	Speedup wrt NR
10^{-2}	5.0×10^{-6}, 5 (38), 3	0.667	7.4
10^{-3}	1.4×10^{-5}, 13 (57), 3	0.750	2.6
10^{-4}	2.4×10^{-5}, 23 (75), 3	2.000	2.1
10^{-5}	3.2×10^{-5}, 32 (94), 3	4.250	1.7
10^{-6}	4.0×10^{-5}, 40 (112), 3	12.25	1.3
10^{-7}	4.8×10^{-5}, 48 (130), 3	33.20	1.1
10^{-8}	5.9×10^{-5}, 57 (149), 3	123.5	1.2
10^{-9}	6.5×10^{-5}, 65 (167), 3	583.3	1.4
10^{-10}	7.5×10^{-5}, 75 (186), 3	1242	N/A
10^{-11}	8.2×10^{-5}, 83 (204), 3	5470	N/A
10^{-12}	9.1×10^{-5}, 92 (223), 3	12538	N/A
10^{-13}	9.9×10^{-5}, 100 (241), 3	40625	N/A
10^{-14}	1.07×10^{-4}, 108 (259), 3	163570	N/A
10^{-15}	1.14×10^{-4}, 117 (278), 3	325000	N/A

boundary of C. It is given by

$$T_7: \qquad f(x_1,x_2) = \begin{cases} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \ r \le R, \\ \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}; \ \|\mathbf{x}\|_2 = 1, \mathbf{x} \ne [1/\sqrt{2}, 1/\sqrt{2}]^T, \\ \begin{bmatrix} y \\ (1-y^2)^{1/2} \end{bmatrix}; \ R < r \le 0.5\sqrt{10}, \\ \begin{bmatrix} y \\ -(1-y^2)^{1/2} \end{bmatrix}; \ 0.5\sqrt{10} < r < \sqrt{1.5+\sqrt{2}}, \end{cases}$$

where

$$r = \|\mathbf{x} - [0.5, 0.5]^T\|_2, R = \sqrt{1.5 - \sqrt{2}}, C = 1.5 - r^2, y = \frac{C - \sqrt{2 - C^2}}{2}.$$

Table 13 shows the results from using the SI and NR algorithms, and Table 14 show the results from the CE algorithm. The solutions computed by CE are within absolute distance $= 10^{-2}, ..., 10^{-15}$, to the fixed point $[0.5, 0.5]^T$ whenever $\varepsilon = 10^{-2}, ..., 10^{-15}$. For this test function the SI algorithm fails to obtain the fixed point. Although the NR method is mostly faster than the CE algorithm, the computed approximation is far away from the fixed-point (absolute distance from the fixed point is approximately equal to 0.707). The NR method converges to the boundary of the unit circle, in whose neighborhood the function has arbitrarily small residual value. In addition, the NR method fails to converge when $\varepsilon \le 10^{-9}$.

TABLE 13. $T_7 : \mathbf{x}^* = [0.5, 0.5]^T, B_1 = B^2([0,0]^T, 1.5)$

ε	SI	NR
10^{-2}	Failed	2.6×10^{-5}, 2, 3
10^{-3}	Failed	2.6×10^{-5}, 2, 3
10^{-4}	Failed	3.2×10^{-5}, 3, 3
10^{-5}	Failed	3.3×10^{-5}, 3, 3
10^{-6}	Failed	3.3×10^{-5}, 3, 3
10^{-7}	Failed	4.1×10^{-5}, 4, 3
10^{-8}	Failed	4.1×10^{-5}, 4, 3
10^{-9}	Failed	Failed
10^{-10}	Failed	Failed
10^{-11}	Failed	Failed
10^{-12}	Failed	Failed
10^{-13}	Failed	Failed
10^{-14}	Failed	Failed
10^{-15}	Failed	Failed

TABLE 14. $T_7 : \mathbf{x}^* = [0.5, 0.5]^T, B_1 = B^2([0,0]^T, 1.5)$

ε	CE	Speedup wrt SI	Speedup wrt NR
10^{-2}	2.5×10^{-5}, 9 (38), 3	N/A	1.04
10^{-3}	4.1×10^{-5}, 27 (57), 3	N/A	0.63
10^{-4}	5.5×10^{-5}, 40 (75), 3	N/A	0.58
10^{-5}	6.7×10^{-5}, 53 (94), 3	N/A	0.49
10^{-6}	7.8×10^{-5}, 66 (112), 3	N/A	0.42
10^{-7}	9.4×10^{-5}, 81 (130), 3	N/A	0.44
10^{-8}	1.2×10^{-4}, 94 (149), 3	N/A	0.34
10^{-9}	1.1×10^{-4}, 107 (167), 3	N/A	N/A
10^{-10}	1.3×10^{-4}, 120 (186), 3	N/A	N/A
10^{-11}	1.5×10^{-4}, 134 (204), 3	N/A	N/A
10^{-12}	1.7×10^{-4}, 147 (223), 3	N/A	N/A
10^{-13}	1.8×10^{-4}, 161 (241), 3	N/A	N/A
10^{-14}	1.9×10^{-4}, 174 (259), 3	N/A	N/A
10^{-15}	2.1×10^{-4}, 187 (278), 3	N/A	N/A

Test 8. This test function is similar to T_7, directionally nonexpanding function, and is given by

$$T_8 : \qquad f(\mathbf{x}) = \begin{cases} \mathbf{y_1};\ \mathbf{r \leq R}, \\ \\ \mathbf{y_2};\ \mathbf{r > R}\ \textbf{and}\ ||\mathbf{y_2}||_{\mathbf{2}} \leq \mathbf{1}, \\ \\ \mathbf{y}_2/||\mathbf{y}_2||_2;\ ||\mathbf{y}_2||_2 > 1, \end{cases}$$

where

$$\mathbf{y}_i = R_i\mathbf{x}, R_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}, i = 1, 2; r = ||\mathbf{x} - [0.5, 0.5]^T||_2, R = \sqrt{1.5 - \sqrt{2}}.$$

The fixed point of this problem is $[0.5, 0.5]^T$. In this test we use $\theta_1 = 0.1$ degree and $\theta_2 = 1$ degree. Table 15 shows the results from using the SI and NR algorithms, and Table 16 shows the results from the CE method. In this case, the NR method converged to the correct fixed point.

TABLE 15. $T_8 : \mathbf{x}^* = [0.5, 0.5]^T, B_1 = B^2([0, 0]^T, 1.5)$

ε	SI	NR
10^{-2}	Failed	2.3×10^{-5}, 1, 3
10^{-3}	Failed	2.3×10^{-5}, 1, 3
10^{-4}	Failed	3.2×10^{-5}, 2, 3
10^{-5}	Failed	3.1×10^{-5}, 2, 3
10^{-6}	Failed	3.8×10^{-5}, 3, 3
10^{-7}	Failed	3.9×10^{-5}, 3, 3
10^{-8}	Failed	4.7×10^{-5}, 4, 3
10^{-9}	Failed	4.6×10^{-5}, 4, 3
10^{-10}	Failed	4.8×10^{-5}, 4, 3
10^{-11}	Failed	4.6×10^{-5}, 4, 3
10^{-12}	Failed	4.7×10^{-5}, 4, 3
10^{-13}	Failed	4.7×10^{-5}, 4, 3
10^{-14}	Failed	4.8×10^{-5}, 4, 3
10^{-15}	Failed	4.8×10^{-5}, 4, 3

We note that in all our tests the CE algorithm terminated with the absolute error criterion (1) or (2) much faster than indicated by the upper bounds. We also note that each iteration of the Newton-Raphson method is about 8–12 times more expensive than the iteration of the CE algorithm for two-dimensional functions.

TABLE 16. $T_8 : \mathbf{x}^* = [0.5, 0.5]^T, B_1 = B^2([0,0]^T, 1.5)$

ε	CE	Speedup wrt SI	Speedup wrt NR
10^{-2}	2.5×10^{-5}, 4 (38), 3	N/A	0.92
10^{-3}	4.1×10^{-5}, 6 (57), 3	N/A	0.56
10^{-4}	5.5×10^{-5}, 25 (75), 3	N/A	0.58
10^{-5}	6.7×10^{-5}, 40 (94), 3	N/A	0.46
10^{-6}	7.8×10^{-5}, 60 (112), 3	N/A	0.49
10^{-7}	9.4×10^{-5}, 74 (130), 3	N/A	0.41
10^{-8}	1.2×10^{-4}, 94 (149), 3	N/A	0.39
10^{-9}	1.1×10^{-4}, 109 (167), 3	N/A	0.42
10^{-10}	1.3×10^{-4}, 129 (186), 3	N/A	0.37
10^{-11}	1.5×10^{-4}, 144 (204), 3	N/A	0.31
10^{-12}	1.7×10^{-4}, 164 (223), 3	N/A	0.28
10^{-13}	1.8×10^{-4}, 184 (241), 3	N/A	0.26
10^{-14}	1.9×10^{-4}, 198 (259), 3	N/A	0.25
10^{-15}	2.1×10^{-4}, 218 (278), 3	N/A	0.23

6. CONCLUSION

We developed a numerically stable FORTRAN 77 implementation of the circumscribed ellipsoid (CE) algorithm for approximating fixed points of directionally nonexpanding functions. Our implementation of the CE algorithm terminates with the absolute error criterion (1) or (2) much faster than indicated by the upper bounds, for all of the tested functions. The implementation of the CE method is much more efficient than the simple iteration algorithm when n is small and ρ is close or equal to 1. Our experiments indicate that a hybrid method based on circumscribed ellipsoid and Newton-Raphson algorithms could be very efficient for solving our problems and should be developed in future research.

ACKNOWLEDGMENTS

We would like to thank the reviewers for detailed comments that significantly improved our paper.

REFERENCES

1. A. Ageev, V. Vassin, E.N. Bessonova, and V.M. Markusevich, *Radiosounding ionosphere on two frequencies. Algorithmic analysis of fredholm-stiltjes integral equation, in: Theoretical problems in geophysics (in Russian)*, (1997), 100–118.
2. E. L. Allgower, *Application of a fixed point search algorithm to nonlinear boundary value problems having several solutions*, Fixed Points Algorithms and Applications (New York) (S. Karamardian, ed.), Academic Press, 1977, pp. 87–111. MR0448858 (56:7163)
3. E. L. Allgower and K. Georg, *Numerical continuation methods*, Springer-Verlag, New York, 1990. MR1059455 (92a:65165)
4. J. Andreas, *Topological fixed point principles for boundary value problems*, Kluwer Academic Publishers, 2003. MR1998968 (2005a:47102)

5. S. Banach, *Sur les opérations dans les ensembles abstraits et leur application aux équations integrales*, Fund. Math. **3** (1922), 133–181.
6. R. G. Bland, D. Goldfarb, and M. Todd, *The ellipsoid method: A survey*, Operations Research **6** (1981), 1039–1090. MR641676 (83e:90081)
7. C. Boonyasiriwat, *Circumscribed ellipsoid algorithm for fixed point problems*, Master's thesis, University of Utah, Salt Lake City, UT, 2004.
8. C. Boonyasiriwat, K. Sikorski, and C. Xiong, *A note on two fixed point problems*, J. Complexity **23** (2007), 952–961. MR2372002 (2008k:65107)
9. K. C. Border, *Fixed point theorems with applications to economics and game theory*, Cambridge University Press, New York, 1985. MR790845 (86j:90002)
10. A. Bouaricha and R. B. Schnabel, *Algorithm 768: Tensolve: a software package for solving systems of nonlinear equations and nonlinear least-squares problems using tensor methods*, ACM Trans. Math. Soft. **23** (1997), no. 2, 174–195. MR1671795
11. L. E. Brouwer, *Über abbildungen von mannigfaltigkeiten*, Mathematische Annalen (1912), no. 71, 97–115.
12. J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math. **31** (1978), 31–48. MR508586 (80g:65038)
13. R. M. Corless, G. W. Frank, and J. G. Monroe, *Chaos and continued fractions*, Physica D **46** (1990), 241–253. MR1083721 (92c:58074)
14. X. Deng and Xi. Chen, *On algorithms for discrete and approximate Brouwer fixed points*, Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005 (Harold N. Gabov and Ronald Fagin, eds.), ACM, 2005. MR2181632
15. J. J. Dongarra and D. C. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput. **8** (1987), no. 2, 139–154. MR879400 (88f:65054)
16. B. C. Eaves, *Homotopies for computation of fixed points*, Math. Programming **3** (1972), no. 1, 1–12. MR0303953 (46:3089)
17. B. C. Eaves and R. Saigal, *Homotopies for computation of fixed points on unbounded regions*, Math. Programming **3** (1972), no. 2, 225–237. MR0314028 (47:2580)
18. C. B. Garcia, C. E. Lemke, and H. Luethi, *Simplicial approximation of an equilibrium point for non-cooperative n-person games*, Mathematical Programming (New York) (T. C. Hu and S. M. Robinson, eds.), Academic Press, 1973, pp. 227–260. MR0439238 (55:12135)
19. G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev. **15** (1973), 318–334. MR0329227 (48:7569)
20. M. Gu and S. C. Eisenstat, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl. **15** (1994), no. 4, 1266–1276. MR1293916 (96c:65057)
21. M. D. Hirsch, C. Papadimitriou, and S. Vavasis, *Exponential lower bounds for finding Brouwer fixed points*, Journal of Complexity **5** (1989), 379–416. MR1028904 (91c:90021)
22. J. L. Howland and R. Vaillancourt, *Attractive cycle in the iteration of meromorphic functions*, Numer. Math. **46** (1985), 323–337. MR791694 (86g:30034)
23. Z. Huang, L. Khachiyan, and K. Sikorski, *Approximating fixed points of weakly contracting mappings*, J. Complexity **15** (1999), 200–213. MR1693888 (2000e:65055)
24. M. M. Jeppson, *A search for the fixed points of a continuous mapping*, Mathematical Topics in Economic Theory and Computation (Philadelphia) (R. H. Day and S. M. Robinsons, eds.), SIAM, 1972, pp. 122–129. MR0408236 (53:12001)
25. L. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl. **20** (1979), no. 1, 191–194. MR639296 (83h:90058)
26. ———, *Polynomial algorithms in linear programming*, USSR Computational Mathematics and Mathematical Physics (1980), no. 20, 53–72. MR564776 (81j:90079)
27. O. H. Merril, *Application and extensions of an algorithm that computes fixed points of a certain upper semi-continuous point of set mapping*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1972.
28. ———, *A summary of techniques for computing fixed points of continuous mapping*, Mathematical Topics in Economic Theory and Computation (Philadelphia, PA) (R. H. Day and S. M. Robinson, eds.), SIAM, 1972, pp. 130–149. MR0408235 (53:12000)
29. J. J. More, B. S. Garbow, and K. E. Hillstrom, *User guide for minpack-1*, Argonne National Laboratory Report ANL-80-74, Argonne, 1980.

30. A. Neumaier, *A better estimate of fixed points of contractions*, Z. Angew. Math. Mech. **62** (1982), 627. MR694062 (84c:65076)
31. G. Perestonina, I.L. Prutkin, L.Ya. Timerkhanova, and V. Vassin, *Solving three-dimensional inverse problems of gravimetry and magnetometry for three layer medium (in Russian)*, Math. Modeling **15** (2003), no. 2, 69–76.
32. J. Rutter, *A serial implementation of cuppen's divide and conquer algorithm for the symmetric eigenvalue problem*, Technical Report CS-94-225, Department of Computer Science, University of Tennessee, Knoxville, TN, USA, March 1994. LAPACK Working Note 69, 1994.
33. H. Scarf, *The approximation of fixed point of a continuous mapping*, SIAM J. Appl. Math. **35** (1967), 1328–1343. MR0242483 (39:3814)
34. ———, *The core of an n person game*, Econometrica **35** (1967), 50–69. MR0234735 (38:3051)
35. H. E. Scarf and T. Hansen, *Computation of economic equilibria*, Yale University Press, 1973. MR0391909 (52:12727)
36. S. Shellman and K. Sikorski, *A two-dimensional bisection envelope algorithm for fixed points*, J. Complexity **18** (2002), no. 2, 641–659. MR1919453 (2003g:65070)
37. ———, *Algorithm 825: A deep-cut bisection envelope algorithm for fixed points*, ACM Trans. Math. Soft. **29** (2003), no. 3, 309–325. MR2002734 (2004h:65051)
38. ———, *A recursive algorithm for the infinity-norm fixed point problem*, J. Complexity **19** (2003), no. 6, 799–834. MR2040430 (2005a:65053)
39. ———, *Algorithm 848: A recursive fixed point algorithm for the infinity-norm case*, ACM Trans. Math. Soft. **31** (2005), no. 4, 580–587. MR2272347 (2007j:65063)
40. K. Sikorski, *Fast algorithms for the computation of fixed points*, Robustness in Identification and Control (New York) (R. Tempo M. Milanese and A. Vicino, eds.), Plenum Press, 1989, pp. 49–59. MR1041089 (91d:65067)
41. ———, *Optimal solution of nonlinear equations*, Oxford Press, New York, 2001. MR1827804 (2002h:65002)
42. K. Sikorski, C. W. Tsay, and H. Woźniakowski, *An ellipsoid algorithm for the computation of fixed points*, J. Complexity **9** (1993), 181–200. MR1213495 (94b:65182)
43. K. Sikorski and H. Woźniakowski, *Complexity of fixed points i.*, J. Complexity (1987), 388–405. MR919096 (89a:65100)
44. D. C. Sorensen and P. T. P. Tang, *On the orthogonality of eigenvectors computed by divide-and-conquer techniques*, SIAM J. Numer. Anal. **28** (1991), 1752–1775. MR1135764 (92h:65065)
45. R. Szrednicki, *A generalization of the lefschetz fixed point theorem and detection of chaos*, Proc. Amer. Math. Soc. **128** (2000), 1231–1239. MR1691005 (2000i:55015)
46. F. Tisseur and J. Dongarra, *Parallelizing the divide and conquer algorithm for the symmetric tridiagonal eigenvalue problem on distributed memory architectures*, SIAM J. Sci. Comput. **20** (1999), no. 6, 2223–2236. MR1703274 (2000e:65045)
47. C. W. Tsay, *Fixed point computation and parallel algorithms for solving wave equations*, Ph.D. thesis, University of Utah, Salt Lake City, UT, 1994.
48. V. Vassin, *Ill-posed problems with a priori information: Methods and applications*, Institute of Mathematics and Mechanics, Russian Academy of Sciences, Ural Subdivision, 2005.
49. V. Vassin and A.L. Ageev, *Ill-posed problems with a priori information*, VSP, Utrecht, The Netherlands, 1995. MR1374573 (97j:65100)
50. V. Vassin and E. Eremin, *Feyer type operators and iterative processes (in Russian)*, Russian Academy of Sciences, Ural Subdivision, Ekaterinburg, 2005.
51. V. Vassin and T.I. Sereznikova, *Two stage method for approximation of nonsmooth solutions and reconstruction of noisy images (in Russian)*, Automatica and Telemechanica (2004), no. 2.
52. L. T. Watson, *A globally convergent algorithm for computing fixed points of c_2 maps*, Appl. Math. Comput. **5** (1979), 297–311. MR544868 (81b:55006)
53. Z. B. Xu and X. Z. Shi, *A comparision of point and ball iterations in the contractive mapping case*, Computing **49** (1992), 75–85. MR1182443 (93h:65085)
54. N. Yamamoto, *A numerical verification method for solutions of boundary value problems with local uniqueness by banach's fixed-point theorem*, SIAM J. Num. Anal. **35** (1998), 2004–2013. MR1639986 (99f:65180)
55. Z. Yang, *Computing equilibria and fixed points: The solution of nonlinear inequalities*, Kluwer Academic Publishers, Dordrecht, 1999. MR1788059 (2001h:90003)

School of Computing, University of Utah, 50 S. Central Campus Dr., Salt Lake City, Utah 84112

E-mail address: chaiwoot@yahoo.com

School of Computing, University of Utah, 50 S. Central Campus Dr., Salt Lake City, Utah 84112

E-mail address: sikorski@cs.utah.edu

Computer Science and Information Management, Providence University 200 Chung-chi Rd., Shalu Taichung 43301, Taiwan

E-mail address: cwtsay@pu.edu.tw