ELSEVIER Journal of Computational and Applied Mathematics 66 (1996) 239–253

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

A dimension-reducing method for unconstrained optimization

T.N. Grapsa*, M.N. Vrahatis

Department of Mathematics, University of Patras, GR-261.10 Patras, Greece

Received 29 July 1994; revised 8 June 1995

Abstract

A new method for unconstrained optimization in $\mathbb{R}^n$ is presented. This method reduces the dimension of the problem in such a way that it can lead to an iterative approximate formula for the computation of $(n-1)$ components of the optimum while its remaining component is computed separately using the final approximations of the other components. It converges quadratically to a local optimum and it requires storage of order $(n-1)\times(n-1)$. Besides, it does not require a good initial guess for one component of the optimum and it does not directly perform gradient evaluations; thus it can be applied to problems with imprecise gradient values.

Moreover, a procedure for transforming the matrix formed by our method into a symmetric as well as into a diagonal one is presented. Furthermore, the proposed dimension-reducing scheme using finite difference gradient and Hessian is presented.

The methods have been implemented and tested. Performance information for well-known test functions is reported.

Keywords: Dimension-reducing method; Unconstrained optimization; Reduction to one-dimensional equations; Bisection method; Imprecise gradient values; Quadratic convergence

AMS classification: 65K10; 49M37; 90C30; 65H10

1. Introduction

There are a large variety of methods for unconstrained optimization of functions,

$$f:\mathscr{D}\subset\mathbb{R}^n\to\mathbb{R}, \tag{1.1}$$

which require precise function and gradient values. However, in many optimization problems of practical interest the values of the objective functions and the corresponding gradients are known only imprecisely. For example, when the function and gradient values depend on the results of numerical simulations, then it may be difficult or impossible to obtain very precise values. Or, in other cases, it may be necessary to integrate numerically a system of differential equations in order to obtain a function value, so that the precision of the computed value is limited [11].

* Corresponding author. E-mail: grapsa@math.upatras.gr.

SSDI 0377-0427(95)00174-3

Recently, new methods have been proposed [6–9] for the numerical solution of a system of nonlinear algebraic and/or transcendental equations:

$$F(x) = \Theta^n = (0, 0, \ldots, 0), \tag{1.2}$$

where $F = (f_1, \ldots, f_n): \mathscr{D} \subset \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable mapping in an open neighborhood $\mathscr{D}^* \subset \mathscr{D}$ of a solution $x^* \in \mathscr{D}$ of system (1.2). These methods incorporate the advantages of Newton and nonlinear SOR algorithms [14]. Specifically, although these methods use reduction to simpler one-dimensional nonlinear equations, they converge quadratically.

In this paper, we derive and apply a new iterative procedure for the computation of an unconstrained optimum of functions (1.1). This method is based on the methods studied in [6–9] and it incorporates the advantages of Newton and SOR algorithms. Although this new procedure uses reduction to simpler one-dimensional nonlinear equations, it generates a quadratically converging sequence of points in $\mathbb{R}^{n-1}$ which converges to the $n-1$ components of the optimum while the remaining component of the optimum is evaluated separately using the final approximations of the others. For this component an initial guess is not necessary and it is at the user's disposal to choose which will be the remaining component, according to the problem. Also this method does not directly need any gradient evaluation and it compares favorably with quadratically convergent optimization methods.

In the next section the dimension-reducing optimization method is presented, its convergence is studied and the corresponding algorithm is proposed. In Section 3 we perturb the matrix obtained by the dimension-reducing procedure in order to transform it into a symmetric as well as into a diagonal one and we study the convergence of the produced scheme. In Section 4 we present the dimension-reducing optimization method utilizing finite difference approximations. Finally, we give numerical applications and some concluding remarks.

2. The dimension-reducing optimization method and its convergence

Notation. Throughout this paper $\mathbb{R}^n$ is the n-dimensional real space of column vectors x with components $x_1, x_2, \ldots, x_n$; $(y; z)$ represents the column vector with components $y_1, y_2, \ldots, y_m, z_1, z_2, \ldots, z_k$; $\partial_i f(x)$ denotes the partial derivative of $f(x)$ with respect to the ith variable x_i; $g(x) = (g_1(x), \ldots, g_n(x))$ defines the gradient $\nabla f(x)$ of the objective function f at x while $H = [H_{ij}]$ defines the Hessian $\nabla^2 f(x)$ of f at x; $\bar{\mathscr{A}}$ denotes the closure of the set $\mathscr{A}$ and $f(x_1, \ldots, x_{i-1}, \cdot, x_{i+1}, \ldots, x_n)$ defines the mapping obtained by holding $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ fixed.

To obtain a sequence $\{x^p\}$, $p = 0, 1, \ldots,$ of points in $\mathbb{R}^n$ which converges to a local optimum (critical) point $x^* = (x_1^*, \ldots, x_n^*) \in \mathscr{D}$ of the function (1.1), we consider the sets $\mathscr{B}_i$, $i = 1, \ldots, n$, to be those connected components of $g_i^{-1}(0)$ containing x^* on which $\partial_n g_i \neq 0$, for $i = 1, \ldots, n$ respectively. Next, applying the Implicit Function Theorem [14] for each one of the components g_i, $i = 1, \ldots, n$, we can find open neighborhoods $\mathscr{A}_1^* \subset \mathbb{R}^{n-1}$ and $\mathscr{A}_{2,i}^* \subset \mathbb{R}$, $i = 1, \ldots, n$, of the points $y^* = (x_1^*, \ldots, x_{n-1}^*)$ and x_n^* respectively, such that for any $y = (x_1, \ldots, x_{n-1}) \in \bar{\mathscr{A}}_1^*$ there exist unique mappings φ_i defined and continuous in $\mathscr{A}_1^*$ such that

$$x_n = \varphi_i(y) \in \bar{\mathscr{A}}^*_{2,i}, \quad i = 1, \dots, n, \tag{2.1}$$

and

$$g_i(y; \varphi_i(y)) = 0, \quad i = 1, \dots, n. \tag{2.2}$$

Moreover, the partial derivatives $\partial_j \varphi_i$, $j = 1, \dots, n-1$, exist in $\mathscr{A}_1^*$ for each φ_i, $i = 1, \dots, n$, they are continuous in $\bar{\mathscr{A}}_1^*$ and they are given by

$$\partial_j \varphi_i(y) = -\frac{\partial_j g_i(y; \varphi_i(y))}{\partial_n g_i(y; \varphi_i(y))}, \quad i = 1, \dots, n, \; j = 1, \dots, n-1. \tag{2.3}$$

Next, working exactly as in [7], we utilize Taylor's formula to expand the $\varphi_i(y)$, $i = 1, \dots, n$, about y^p. By straightforward calculations, we can obtain the following iterative scheme for the computation of the $n-1$ components of x^*:

$$y^{p+1} = y^p + A_p^{-1} V_p, \quad p = 0, 1, \dots, \tag{2.4}$$

where

$$y^p = [x_i^p], \quad i = 1, \dots, n-1, \tag{2.5}$$

$$A_p = [a_{ij}] = \left[\frac{\partial_j g_i(y^p; x_n^{p,i})}{\partial_n g_i(y^p; x_n^{p,i})} - \frac{\partial_j g_n(y^p; x_n^{p,n})}{\partial_n g_n(y^p; x_n^{p,n})} \right], \quad i, j = 1, \dots, n-1, \tag{2.6}$$

$$V_p = [v_i] = [x_n^{p,i} - x_n^{p,n}], \quad i = 1, \dots, n-1, \tag{2.7}$$

with $x_n^{p,i} = \varphi_i(y^p)$. After a desired number of iterations of (2.4), say $p = m$, the nth component of x^* can be approximated by means of the following relation:

$$x_n^{m+1} = x_n^{m,n} - \sum_{j=1}^{n-1} \left\{ (x_j^{m+1} - x_j^m) \frac{\partial_j g_n(y^m; x_n^{m,n})}{\partial_n g_n(y^m; x_n^{m,n})} \right\}. \tag{2.8}$$

Next, we give a proof of convergence of the dimension-reducing optimization method (2.4).

Theorem 2.1. *Suppose that the objective function $f: \mathscr{D} \subset \mathbb{R}^n \to \mathbb{R}$ is twice-continuously differentiable in an open neighborhood $\mathscr{D}^* \subset \mathscr{D}$ of a point $x^* = (x_1^*, \dots, x_n^*) \in \mathscr{D}$ for which $g(x^*) = \Theta^n$. Let $\mathscr{B}_i$, $i = 1, \dots, n$, be those connected components of $g_i^{-1}(0)$, containing x^* on which $\partial_n g_i \neq 0$ for $i = 1, \dots, n$ respectively. Then the iterations of* (2.4) *and the relationship* (2.8) *will converge to x^* provided the Hessian of f at x^* is nonsingular and also provided the initial guess $y^0 = (x_1^0, \dots, x_{n-1}^0)$ is sufficiently close to $y^* = (x_1^*, \dots, x_{n-1}^*)$. Moreover the iterations y^p, $p = 0, 1, \dots,$ of* (2.4) *have order of convergence two.*

Proof. The determinant of the matrix A_*, obtained from the matrix A_p of (2.6) at x^*, can be written as follows:

$$\det A_* = \frac{\det H(x^*)}{\prod_{i=1}^n \partial_n g_i(x^*)}. \tag{2.9}$$

Now, since the Hessian of f at x^* is nonsingular then obviously A_p is also nonsingular at x^*.

The iterations (2.4) can be written as follows:

$$y^{p+1} = y^p - A_p^{-1} M_p, \quad p = 0, 1, \ldots, \tag{2.10}$$

where

$$M_p = [m_i] = [-x_n^{p,i} + x_n^{p,n}], \quad i = 1, \ldots, n-1. \tag{2.11}$$

Consider now the mapping

$$\begin{aligned} &\Lambda = (\lambda_1, \ldots, \lambda_{n-1}) \colon \mathscr{A}_1^* \subset \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}, \quad \text{with} \\ &\lambda_i(y) = -\varphi_i(y) + \varphi_n(y), \quad i = 1, \ldots, n-1, \end{aligned} \tag{2.12}$$

and denote the corresponding Jacobian by Λ'; then the iterative scheme (2.4) is equivalent to the following one:

$$y^{p+1} = y^p - \Lambda'(y^p)^{-1} \Lambda(y^p), \quad p = 0, 1, \ldots . \tag{2.13}$$

Consequently, by the well-known Newton's convergence theorem [13] for an initial guess y^0 sufficiently close to y^*, the iterations y^p, $p = 0, 1, \ldots,$ of (2.4) converge to y^* and the order of convergence is two.

Suppose now that for some p, for example $p = m$, we obtain $y^m = y^*$. Then, relation (2.8) yields

$$x_n^{m+1} = \varphi_n(y^*) = x_n^* . \tag{2.14}$$

Thus the theorem is proved. □

Remark 2.2. Relative procedures for obtaining x^* can be constructed by replacing x_n with any one of the components $x_1, \ldots, x_{n-1}$, for example x_{int}, and taking $y = (x_1, \ldots, x_{\text{int}-1}, x_{\text{int}+1}, \ldots, x_n)$.

Remark 2.3. The above described method does not require the expressions φ_i but only the values $x_n^{p,i}$ which are given by the solution of the one-dimensional equations $g_i(x_1^p, \ldots, x_{n-1}^p, \cdot) = 0$. So, by holding $y^p = (x_1^p, \ldots, x_{n-1}^p)$ fixed, we can solve the equations

$$g_i(y^p; r_i^p) = 0, \quad i = 1, \ldots, n, \tag{2.15}$$

for r_i^p in the interval (a, b) with an accuracy δ.

We can use any one of the well-known one-dimensional methods [14] to solve the above equations. Here we employ a modified bisection method described in [17, 18]. According to these, for the computation of a root of the equation $\psi(x) = 0$, where $\psi \colon [a, b] \subset \mathbb{R} \to \mathbb{R}$ is continuous, we can use the bisection method which has been modified to the following version:

$$x^{p+1} = x^p + \operatorname{sgn} \psi(x^p) q / 2^{p+1}, \quad p = 0, 1, \ldots, \tag{2.16}$$

where $x^0 = a$, $q = \operatorname{sgn} \psi(a)(b - a)$ and where sgn defines the well-known sign function. This method computes with certainty a root when $\operatorname{sgn} \psi(x^0) \operatorname{sgn} \psi(x^p) = -1$ (see [19] for extensions).

It is evident from (2.16) that the only computable information required by the bisection method is the algebraic signs of the function ψ. We use this method since it is a global convergence method, it always converges within the given interval and it is optimal [16], in the sense that it possesses asymptotically the best rate of convergence. Besides, the number of the iterations v, which are required to obtain an approximate root r^* such that $|r - r^*| \leqslant \delta$ for some $\delta \in (0, 1)$, is given by $v = \lceil \log_2((b-a)\delta^{-1}) \rceil$, where the notation $\lceil \cdot \rceil$ refers to the smallest integer not less than the real number quoted. Moreover, it is the only method that can be applied to problems with imprecise function values. This procedure has been efficiently implemented for the computation of all the zeros and extrema of a function [10].

The proposed method is illustrated in the following algorithm in pseudo-code where $g = (g_1, g_2, \ldots, g_n)$ indicates the gradient of the objective function, x^0 the starting point, $a = (a_1, a_2, \ldots, a_n)$, $b = (b_1, b_2, \ldots, b_n)$ indicate the endpoints in each coordinate direction which are used for the one-dimensional bisection (2.16), δ the predetermined accuracy for applying the procedure (2.16), MIT the maximum number of iterations required and $\varepsilon_1, \varepsilon_2$ the predetermined desired accuracies.

Algorithm 1. Dimension-Reducing Optimization (DROPT)

```
1. Input $\{x^0; a; b; \delta; \mathrm{MIT}; \varepsilon_1; \varepsilon_2\}$.
2. Set $p = -1$.
3. If $p < \mathrm{MIT}$ replace $p$ by $p + 1$ and go to the next step; otherwise, go to Step 14.
4. If $\|g(x^p)\| \leqslant \varepsilon_1$ go to Step 14.
5. Find a coordinate int such that the following relation holds:
       $\operatorname{sgn} g_i(x_1^p, \ldots, x_{\mathrm{int}-1}^p, a_{\mathrm{int}}, x_{\mathrm{int}+1}^p, \ldots, x_n^p) \cdot \operatorname{sgn} g_i(x_1^p, \ldots, x_{\mathrm{int}-1}^p, b_{\mathrm{int}}, x_{\mathrm{int}+1}^p, \ldots, x_n^p) = -1$
   for all $i = 1, 2, \ldots, n$. If this is impossible, apply Armijo's method (see below) and go to Step 4.
6. Compute the approximate solutions $r_i$ for all $i = 1, 2, \ldots, n$ of the equation
       $g_i(x_1^p, \ldots, x_{\mathrm{int}-1}^p, r_i, x_{\mathrm{int}+1}^p, \ldots, x_n^p) = 0$
   by applying the iterative scheme (2.16) in $(a_{\mathrm{int}}, b_{\mathrm{int}})$ within accuracy $\delta$. Set $x_{\mathrm{int}}^{p,i} = r_i$.
7. Set $y^p = (x_1^p, \ldots, x_{\mathrm{int}-1}^p, x_{\mathrm{int}+1}^p, \ldots, x_n^p)$.
8. Set the elements of the matrix $A_p$ of relation (2.6) using $x_{\mathrm{int}}$ instead of $x_n$.
9. Set the elements of the vector $V_p$ of relation (2.7) using $x_{\mathrm{int}}$ instead of $x_n$.
10. Solve the $(n-1) \times (n-1)$ linear system $A_p s^p = -V_p$ for $s^p$.
11. Set $y^{p+1} = y^p + s^p$.
12. Compute $x_{\mathrm{int}}$ by virtue of relation (2.8) and set $x^p = (y^p; x_{\mathrm{int}})$.
13. If $\|s^p\| \leqslant \varepsilon_2$ go to Step 14; otherwise return to Step 3.
14. Output $\{x^p\}$.
```

The criterion in Step 5 ensures the existence of the solution r_i which will be computed at Step 6. If this criterion is not satisfied we apply Armijo's method [1, 20] for a few steps and then try again with our method. Our experience is that in many examples studied in various dimensions as well as for all the problems studied in this paper (see Section 5), the application of such a subprocedure is not necessary. We have merged it in our algorithm for completeness.

Based on this, we give the following subprocedure, where MAR is the maximum number of Armijo's iterations required, η is an arbitrary assigned positive number and ε the predetermined desired accuracy.

Algorithm 2. Armijo's Modification of Steepest Descent

```
1. Input $\{x^0;$ MAR; $\eta_0; \varepsilon\}$.
2. Set $k = -1$.
3. If $k <$ MAR, replace $k$ by $k + 1$, set $\eta = \eta_0, m = 1$ and go to the next step; otherwise, go to Step 8.
4. If $f(x^k - \eta g(x^k)) - f(x^k) \leqslant -\frac{1}{2}\eta \|g(x^k)\|^2$, go to Step 6; otherwise, set $m = m + 1$ and go to the next step.
5. Set $\eta = \eta/2^{m-1}$ and return to Step 4.
6. Set $x^{k+1} = x^k - \eta g(x^k)$.
7. If $\|g(x^k)\| \leqslant \varepsilon$, go to Step 8; otherwise go to Step 3.
8. Output $\{x^k: f(x^k); g(x^k)\}$.
```

In the case of the application of our method to imprecise problems, Algorithm 2 is replaced by other suitable algorithms as for example the corresponding one proposed in [20].

3. A perturbed dimension-reducing optimization method

The proposed method computes any critical point x^* of f (minimum, maximum or saddle) and it minimizes it if the matrix A_p of (2.6) is symmetric and positive definite so that the critical point is a minimizer. In general this matrix is not symmetric. A case where it is symmetric is given by the following lemma:

Lemma 3.1. *If the elements H_{in}, $i = 1, \dots, n-1$, of the Hessian are equal to each other, then the matrix A_p defined by relation* (2.6) *is symmetric.*

Proof. From relation (2.6) we observe that the matrix A_p is symmetric when $\partial_n g_i = \partial_n g_j$. □

In the sequel we perturb the matrix A_p using proper perturbation parameters Δ_{ij}, $i, j = 1, \dots, n-1$, in order to transform it to a symmetric one. To this end we consider the mapping

$$\begin{aligned} &W = (w_1, \dots, w_{n-1}): \mathscr{A}_1^* \subset \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}, \text{ with} \\ &w_i(y) = -\varphi_i(y) + \varphi_n(y) + \sum_{j=1}^{n-1} \Delta_{ij} x_j, \quad i = 1, \dots, n-1, \end{aligned} \tag{3.1}$$

where the matrix $\Delta = \Delta_{ij}$ of the perturbation parameters is taken such that the inner products

$$\langle x, \Delta_i \rangle = 0, \quad \forall x \in \mathscr{A}_1^* \text{ and } \forall i = 1, \dots, n-1, \tag{3.2}$$

with $\Delta_i = (\Delta_{i1}, \dots, \Delta_{i,n-1})$.

Now, by choosing $\frac{1}{2}n(n-1)$ arbitrary parameters Δ_{ij}, $i > j$, and taking

$$\Delta_{ij} = a_{ji} + \Delta_{ji} - a_{ij}, \quad i < j, \tag{3.3}$$

while the remaining parameters Δ_{ii} are computed so that $\langle x, \Delta_i \rangle = 0$, the matrix $A_p = [a_{ij}]$ is transformed into a symmetric one. Since Δ_{ij}, $i > j$, are arbitrarily chosen, we are able to transform

A_p to a diagonal one. Thus, by taking

$$\Delta_{ij} = -a_{ij} = \left[-\frac{\partial_j g_i(y^p; x_n^{p,i})}{\partial_n g_i(y^p; x_n^{p,i})} + \frac{\partial_j g_n(y^p; x_n^{p,n})}{\partial_n g_n(y^p; x_n^{p,n})} \right], \quad i = 1, \ldots, n-1, \quad i > j, \tag{3.4}$$

from relations (3.3) and (3.2) our method becomes

$$y^{p+1} = y^p - D_p^{-1} M_p, \quad p = 0, 1, \ldots, \tag{3.5}$$

where D_p is a diagonal matrix with elements d_{ii}, $i = 1, \ldots, n-1$, given by

$$\begin{aligned} d_{ii} = a_{ii} + \Delta_{ii} &= \left[\frac{\partial_i g_i(y^p; x_n^{p,i})}{\partial_n g_i(y^p; x_n^{p,i})} - \frac{\partial_i g_n(y^p; x_n^{p,n})}{\partial_n g_n(y^p; x_n^{p,n})} \right] \\ &+ \frac{1}{x_i} \sum_{\substack{j=1 \\ j \neq i}}^{n-1} \left\{ x_j \left[\frac{\partial_j g_i(y^p; x_n^{p,i})}{\partial_n g_i(y^p; x_n^{p,i})} - \frac{\partial_j g_n(y^p; x_n^{p,n})}{\partial_n g_n(y^p; x_n^{p,n})} \right] \right\}. \end{aligned} \tag{3.6}$$

Finally, after a desired number of iterations of the above scheme, say $p = m$, the nth component of x^* is approximated using relation (2.8).

Next, we give a proof of convergence of the perturbed dimension-reducing optimization method (3.5).

Theorem 3.2. *Suppose that the objective function $f: \mathscr{D} \subset \mathbb{R}^n \to \mathbb{R}$ is twice-continuously differentiable in an open neighborhood $\mathscr{D}^* \subset \mathscr{D}$ of a point $x^* = (x_1^*, \ldots, x_n^*) \in \mathscr{D}$ for which $g(x^*) = \Theta^n$. Let $\mathscr{B}_i$, $i = 1, \ldots, n$, be those connected components of $w_i^{-1}(0)$, containing x^* on which $\partial_n w_i \neq 0$ for $i = 1, \ldots, n$ respectively where the functions w_i are defined by* (3.1). *Suppose further that*

$$\|A_*^{-1}\| \, \|\Delta\| < 1, \tag{3.7}$$

where the matrix A_* is obtained from the matrix A_p of (2.6) *at x^* and Δ is the perturbation matrix. Then the iterations* (3.5) *and relation* (2.8) *will converge to x^* provided the Hessian of f at x^* is nonsingular and the initial guess $y^0 = (x_1^0, \ldots, x_{n-1}^0)$ is sufficiently close to $y^* = (x_1^*, \ldots, x_{n-1}^*)$. Moreover the iterations y^p, $p = 0, 1, \ldots,$ of* (3.5) *have order of convergence two.*

Proof. Following the proof of Theorem 2.1, since the Hessian of f at x^* is nonsingular then A_p is also nonsingular at x^*. Moreover, since relation (3.7) is fulfilled, then by the Banach Permutation Lemma [14] the matrix $(A_* + \Delta)$ is also nonsingular.

Using the mapping (3.1) the scheme (3.5) can be written in the following form:

$$y^{p+1} = y^p - W'(y^p)^{-1} W(y^p), \quad p = 0, 1, \ldots, \tag{3.8}$$

where W' denotes the corresponding Jacobian. Clearly, we have $W' = (A_* + \Delta)$ which is nonsingular at x^* and, consequently, by the well-known Newton's convergence theorem [13], for an initial guess y^0 sufficiently close to y^*, the iterations y^p, $p = 0, 1, \ldots,$ of (3.5) converge to y^* and the order of convergence is two.

Suppose now that for some p, for example $p = m$, we obtain $y^m = y^*$. Then, relation (2.8) yields

$$x_n^{m+1} = \varphi_n(y^*) = x_n^*. \tag{3.9}$$

Thus the theorem is proved. □

4. A finite-difference derivative dimension-reducing optimization method

In this section we consider the dimension-reducing method for unconstrained optimization using finite difference gradients and Hessian. So, suppose that the function (1.1) is continuously differentiable in the open convex domain $\mathscr{D} \subset \mathbb{R}^n$, $x \in \mathscr{D}$, and let $g(x)$ be Lipschitz continuous at x in the neighborhood $\mathscr{D}$. Assume $x + he_i \in \mathscr{D}$, $i = 1, \ldots, n$, for a small quantity h, where e_i denotes the ith unit vector; then using forward finite differences we obtain

$$g_i(x) = \partial_i f \simeq \frac{f(x + he_i) - f(x)}{h}. \tag{4.1}$$

Furthermore, the Hessian $H(x)$ can be approximated using only values of $f(x)$. To this end, suppose that the previous conditions regarding f are satisfied and assume x, $x + he_i$, $x + he_j$, $x + he_i + he_j \in \mathscr{D}$, $1 \leqslant i, j \leqslant n$, then

$$\partial_j g_i(x) = \partial_{ij}^2 f \simeq \frac{f(x + he_i + he_j) - f(x + he_i) - f(x + he_j) + f(x)}{h^2}. \tag{4.2}$$

For error estimates for these approximations see [4].

Using the above approximations to (2.4) and (2.8) we propose the following iterative scheme for the computation of the $n - 1$ components of an optimum x^*:

$$y^{p+1} = y^p + U_p^{-1} V_p, \quad p = 0, 1, \ldots, \tag{4.3}$$

where, applying (4.1) for g_i instead of f, the elements of the matrix $U_p = [u_{ij}]$ become

$$u_{ij} = \left[\frac{g_i(y^p + he_j; x_n^{p,i}) - g_i(y^p; x_n^{p,i})}{g_i(y^p; x_n^{p,i} + he_n) - g_i(y^p; x_n^{p,i})} - \frac{g_n(y^p + he_j; x_n^{p,n}) - g_n(y^p; x_n^{p,n})}{g_n(y^p; x_n^{p,n} + he_n) - g_n(y^p; x_n^{p,n})} \right], \tag{4.4}$$

while using (4.2) we have

$$\begin{aligned} u_{ij} = \Bigg[& \frac{f(y^p + he_i + he_j; x_n^{p,i}) - f(y^p + he_i; x_n^{p,i}) - f(y^p + he_j; x_n^{p,i}) + f(y^p; x_n^{p,i})}{f(y^p + he_i; x_n^{p,i} + he_n) - f(y^p + he_i; x_n^{p,i}) - f(y^p; x_n^{p,i} + he_n) + f(y^p; x_n^{p,i})} \\ & - \frac{f(y^p + he_j; x_n^{p,n} + he_n) - f(y^p + he_j; x_n^{p,n}) - f(y^p; x_n^{p,n} + he_n) + f(y^p; x_n^{p,n})}{f(y^p; x_n^{p,n} + 2he_n) - 2f(y^p; x_n^{p,n} + he_n) + f(y^p; x_n^{p,n})} \Bigg]. \end{aligned} \tag{4.5}$$

After a desired number of iterations of (4.3), say $p = m$, the nth component of x^* is approximated by means of one of the following relations:

$$x_n^{m+1} = x_n^{m,n} - \sum_{j=1}^{n-1} \left\{ (x_j^{m+1} - x_j^m) \frac{g_n(y^m + he_j; x_n^{m,n}) - g_n(y^m; x_n^{m,n})}{g_n(y^m; x_n^{m,n} + he_n) - g_n(y^m; x_n^{m,n})} \right\}, \tag{4.6}$$

or

$$\begin{aligned} x_n^{m+1} = x_n^{m,n} - \sum_{j=1}^{n-1} \Bigg\{ & (x_j^{m+1} - x_j^m) \\ & \times \frac{f(y^m + he_j; x_n^{m,n} + he_n) - f(y^m + he_j; x_n^{m,n}) - f(y^m; x_n^{m,n} + he_n) + f(y^m; x_n^{m,n})}{f(y^m; x_n^{m,n} + 2he_n) - 2f(y^m; x_n^{m,n} + he_n) + f(y^m; x_n^{m,n})} \Bigg\}. \end{aligned} \tag{4.7}$$

The signs of the gradient which are required by our method are obtained by virtue of (4.1).

5. Numerical applications

The procedures described in this paper have been implemented using a new FORTRAN program named DROPT (Dimension-Reducing OPTimization). DROPT was tested on the University of Patras HP-715 system as well as on a PC IBM compatible with random problems of various dimensions. Our experience is that the algorithm behaved predictably and reliably and the results were quite satisfactory. Some typical computational results are given below. For the following problems, the reported parameters indicate: n dimension, $x^0 = (x_1, x_2, \dots, x_n)$ starting point, $h = (h_1, h_2, \dots, h_n)$ starting stepsizes in each coordinate direction, $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ approximate local optimum computed within an accuracy of $\varepsilon = 10^{-8}$, IT the total number of iterations required to obtain x^*, FE the total number of function evaluations (including derivatives), ASG the total number of algebraic signs of the components of the gradient that are required for applying the iterative schemes (2.16).

In Tables 1–3 we compare the numerical results obtained, for various starting points, by applying Armijo's steepest descent method [1] as well as conjugate gradient methods and variable metric methods, with the corresponding numerical results of the method presented in this paper obtained on the University of Patras HP-715 system. The index α indicates the classical starting point. Furthermore, D indicates divergence or nonconvergence while FR, PR and BFGS indicate the corresponding results obtained by Fletcher–Reeves [15], Polak–Ribiere [15] and Broyden–Fletcher–Goldfarb–Shanno (BFGS) [4] algorithms, respectively. Also we compare our method with well-known and efficient root-finding methods such as Brown's method [3], a Brent–Gay modification of Brown's method, specifically the modification of Brown's method suggested by Brent [2] and followed by Gay (BBG) [5], as well as with Brent and Choleski–Newton methods (CN) [15]. These root-finding methods as well as our method, for the examples examined below, are applied to the following system of equations:

$$g_i(x) = \sum_{j=1}^{n} \left(\frac{\partial f_j(x)}{\partial x_i} \right) f_j(x) = 0, \quad 1 \leqslant j \leqslant n, \tag{5.1}$$

where f is the objective function.

Example 5.1 (*Rosenbrock function* [12]). This example gives the optimum for the objective function f given by

$$f(x) = \sum_{i=1}^{2} f_i^2(x), \tag{5.2}$$

where

$$f_1(x) = 10\,(x_2 - x_1^2), \qquad f_2(x) = 1 - x_1, \tag{5.3}$$

with $f(x^*) = 0$ at $x^* = (1, 1)$. As our starting values we utilized $x^0 = (-1.2, 1)$ and $h = (2, 2)$. We obtained the $x^* = (1, 1)$ after IT = 1, FE = 4 and ASG = 20.

Table 1
Rosenbrock function

x^0	Armijo		FR		PR		BFGS		DROPT		
	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	ASG
$(-1.2, 1)^\alpha$	1881	21 396	142	2545	19	364	22	343	1	4	20
$(-3, 6)$	5960	74 560	194	4462	23	455	28	436	4	16	80
$(-2, -2)$	1828	20 852	29	480	15	290	20	305	4	16	80
$(3, 3)$	5993	74 364	130	2939	26	509	25	384	4	16	80
$(1, 20)$	D	D	259	5732	32	689	32	689	1	4	20
$(10, 10)$	18 416	251 611	310	7469	26	526	32	505	4	16	80
$(100, 100)$	D	D	D	D	33	746	54	822	2	8	40
$(-2000, -2000)$	2542	35 743	D	D	93	2466	173	2667	2	8	40

x^0	Brown		BBG		Brent		CN		DROPT		
	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	ASG
$(-1.2, 1)^\alpha$	22	110	16	80	17	85	8	48	1	4	20
$(-7, 1)$	D	D	31	155	29	145	6	36	1	4	20
$(-3, 45)$	D	D	D	D	D	D	7	42	4	16	80
$(10, 10)$	70	350	29	145	27	135	6	36	4	16	80
$(-100, 1)$	D	D	D	D	46	230	6	36	1	4	20
$(100, -100)$	D	D	50	250	43	215	6	36	2	8	40
$(100, 100)$	D	D	49	245	41	205	6	36	2	8	40
$(-2000, -2000)$	D	D	D	D	D	D	6	36	2	8	40

Table 2
Freudenstein and Roth function

x^0	Armijo		FR		PR		BFGS		DROPT		
	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	ASG
$(0.5, -2)$[a]	1827	24 155	18	356	8	187	7	138	3	12	60
$(0.5, 1000)$	1380	18 770	D	D	D	D	D	D	3	12	60
$(-2, -2)$	1119	14 625	19	336	8	180	7	121	4	16	80
$(-20, 20)$	1851	24 986	24	451	10	211	9	149	5	20	100
$(4.5, 4.5)$	1239	16 289	18	342	9	196	8	129	3	12	60
$(10, 100)$	1845	24 664	10	200	9	194	9	170	3	12	60
$(12, 2)$	2027	26 886	70	1145	8	130	7	103	3	12	60
$(4, -1000)$	1886	25 597	D	D	D	D	D	D	3	12	60

x^0	Brown		BBG		Brent		CN		DROPT		
	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	ASG
$(0.5, -2)$[a]	5	25	6	30	6	30	8	48	3	12	60
$(0.5, 1000)$	D	D	D	D	D	D	32	192	3	12	60
$(-20, -200)$	D	D	D	D	D	D	25	150	5	20	100
$(4.5, -8)$	D	D	D	D	15	75	14	84	3	12	60
$(10, -20)$	D	D	21	105	D	D	18	108	3	12	60
$(12, -24)$	38	190	D	D	D	D	18	108	3	12	60
$(4, -1000)$	D	D	D	D	D	D	35	210	3	12	60
$(10, 100)$	D	D	D	D	D	D	22	132	3	12	60

Example 5.2 (*Freudenstein and Roth function* [12]). In this example the objective function f is given by

$$f(x) = \sum_{i=1}^{2} f_i^2(x), \tag{5.4}$$

where

$$f_1(x) = -13 + x_1 + ((5 - x_2)x_2 - 2)x_2, \qquad f_2(x) = -29 + x_1 + ((x_2 + 1)x_2 - 14)x_2, \tag{5.5}$$

with $f(x^*) = 0$ at $x^* = (5, 4)$ and $f(x^*) = 48.9842\ldots$ at $x^* = (11.41\ldots, -0.8968\ldots)$. As our starting values we utilized $x^0 = (0.5, -2)$ and $h = (2, 2)$. We obtained the $x^* = (5, 4)$ after IT = 3, FE = 12 and ASG = 60.

Example 5.3 (*Brown almost-linear function* [12]). In this case the objective function f is given by

$$f(x) = \sum_{i=1}^{n} f_i^2(x), \tag{5.6}$$

Table 3
Brown almost-linear function

x^0	Armijo		FR		PR		BFGS		DROPT		
	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	ASG
$(0.5, 0.5, 0.5)^\top$	177	1330	12	177	6	93	6	91	6	54	180
(0, 0, 3)	221	1612	27	389	9	143	9	131	1	9	30
(−1, 0, 3)	233	1691	34	491	42	612	12	180	1	9	30
(0.1, 0.1, −2)	181	1349	52	742	9	149	9	145	4	36	120
(1.2, 1.2, 0)	183	1368	23	334	9	138	8	121	3	27	90
(0.8, 0.7, −2)	193	1446	70	996	23	336	14	216	4	36	120
(−0.1, −0.1, −0.1)	173	1295	20	290	7	109	6	92	3	27	90
(1.2, 1.2, 10)	64	563	47	711	10	165	9	139	3	27	90

x_0	Brown		BBG		Brent		CN		DROPT		
	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	ASG
$(0.5, 0.5, 0.5)^\top$	7	63	8	72	8	72	9	108	6	54	180
(0, 0, 17)	13	117	22	198	34	306	D	D	1	9	30
(−1, 0, −1)	10	90	17	153	D	D	9	108	1	9	30
(−1, 0, 23)	11	99	33	297	22	198	D	D	1	9	30
(0.8, 0.7, −0.7)	30	270	D	D	D	D	D	D	4	36	120
(0.8, 0.7, −1.7)	7	63	33	297	13	117	15	180	4	36	120
(1.2, 1.2, −10)	50	450	19	171	D	D	D	D	3	27	90
(1.2, 1.2, −23)	14	126	30	270	23	207	D	D	3	27	90

Table 4
Rosenbrock function

x^0	Brown		BFGS		DROPT			FDDROPT		
	IT	FE	IT	FE	IT	FE	ASG	IT	FE	ASG
$(-1.2, 1)^T$	22	210	22	343	1	4	20	1	6	20
$(-3, 6)$	D	D	28	436	4	16	80	21	126	420
$(-3, 45)$	D	D	51	771	4	16	80	23	138	460
$(100, 1)$	D	D	26	410	1	4	20	1	6	20
$(0.7, -4)$	15	75	45	682	4	16	80	9	54	180
$(0.5, -5)$	16	80	51	783	6	24	120	6	36	120
$(0.8, 3)$	D	D	16	235	4	16	80	9	54	180
$(1, 2)$	17	85	13	193	1	4	20	1	6	20

where

$$f_i(x) = x_i + \sum_{j=1}^{n} x_j - (n+1), \quad 1 \leqslant i < n, \quad f_n(x) = \left(\prod_{j=1}^{n} x_j \right) - 1, \tag{5.7}$$

with $f(x^*) = 0$ at $x^* = (a, \ldots, a, a^{1-n})$ where a satisfies the equation $na^n - (n+1)a^{n-1} + 1 = 0$ and $f(x^*) = 1$ at $x^* = (0, \ldots, 0, n+1)$. As our starting values we utilized $n = 3$, $x^0 = (0.5, 0.5, 0.5)$ and $h = (2, 2, 2)$. We obtained the $x^* = (0.76759187, 0.76759187, 1.69722437)$ after IT = 6, FE = 54 and ASG = 180.

Next, we apply our method using finite differences (FDDROPT) and we compare these results with the corresponding ones obtained by Brown, DROPT and BFGS methods without finite differences for various starting guesses. These results are shown in Table 4.

6. Concluding remarks

This paper describes a new efficient numerical method for computing an unconstrained local optimum. This method rapidly minimizes general functions and it appears to be superior to other well-known optimization and root-finding methods on a variety of classical test functions. We have tested our method for higher dimensions and we have observed relative results. In the case of large sparse problems, if, for some reason, the component x_{int} which corresponds to the dimension-reducing component is missing for some component g_i, we replace it by the function $g_i + g_k$ where g_k includes x_{int}.

Although the method of this paper uses reduction to simpler one-dimensional equations, it converges quadratically to $n - 1$ components of the optimum, while the remaining component of the optimum is evaluated separately using the final approximations of the other components. Thus, it does not require a good initial estimate for one component of the optimum. Besides, this method does not directly perform gradient evaluations, since it uses the modified one-dimensional bisection method. It requires only that the algebraic signs of the function and gradient values be correct, so that it can be applied to problems with imprecise function values.

Since in general the matrix of our reduced system is not symmetric, we have transformed it to a symmetric one by using proper perturbations. Also, applying this transformation we have been able to obtain analytical forms of the equivalent diagonal matrix.

Furthermore, we have substituted finite difference approximations for the elements of the matrix A_p and we give the corresponding dimension-reducing scheme. By means of this, we have been able to compute optimum points utilizing only values of the objective function.

Acknowledgements

We wish to acknowledge the referee for useful suggestions.

References

[1] L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific. J. Math.* **16** (1966) 1–3.

[2] R.P. Brent, Some efficient algorithms for solving systems of nonlinear equations, *SIAM J. Numer. Anal.* **10** (1973) 327–366.

[3] K.M. Brown, Computer oriented algorithms for solving systems of simultaneous nonlinear algebraic equations, in: G.D. Byrne and C.A. Hall, Eds., *Numerical Solution of Systems of Nonlinear Algebraic Equations* (Academic Press, New York, 1973).

[4] J.E. Dennis Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[5] D.M. Gay, Brown's method and some generalizations with applications to minimization problems, Ph.D. Thesis, Cornell Univ., 1975.

[6] T.N. Grapsa and M.N. Vrahatis, The implicit function theorem for solving systems of nonlinear equations in $\mathbb{R}^2$, *Internat. J. Comput. Math.* **28** (1989) 171–181.

[7] T.N. Grapsa and M.N. Vrahatis, A dimension-reducing method for solving systems of nonlinear equations in $\mathbb{R}^n$, *Internat. J. Comput. Math.* **32** (1990) 205–216.

[8] T.N. Grapsa and M.N. Vrahatis, A new dimension-reducing method for solving systems of nonlinear equations, *Internat. J. Comput. Math.* **55** (1995) 235–244.

[9] T.N. Grapsa, M.N. Vrahatis and T.C. Bountis, Solving systems of nonlinear equations in $\mathbb{R}^n$ using a rotating hyperplane in $\mathbb{R}^{n+1}$, *Internat. J. Comput. Math.* **35** (1990) 133–151.

[10] D. Kavvadias and M.N. Vrahatis, Locating and computing all the simple roots and extrema of a function, *SIAM J. Sci. Comput.*, accepted.

[11] M. Kupferschmid and J.G. Ecker, A note on solution of nonlinear programming problems with imprecise function and gradient values, *Math. Programming Stud.* **31** (1987) 129–138.

[12] B.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization, *ACM Trans. Math. Software* **7** (1981) 17–41.

[13] J.M. Ortega, *Numerical Analysis* (Academic Press, New York, 1972).

[14] J.M. Ortega and W.C. Rheinbolt, *Iteration Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).

[15] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes, The Art of Scientific Computing* (Cambridge Univ. Press, New York, 2nd ed., 1992).

[16] K. Sikorski, Bisection is optimal, *Numer. Math.* **40** (1982) 111–117.

[17] M.N. Vrahatis, Solving systems of nonlinear equations using the nonzero value of the topological degree, *ACM Trans. Math. Software* **14** (1988) 312–329.

[18] M.N. Vrahatis, CHABIS: a mathematical software package for locating and evaluating roots of systems of nonlinear equations, *ACM Trans. Math. Software* **14** (1988) 330–336.

[19] M.N. Vrahatis, A short proof and a generalization of Miranda's existence theorem, *Proc. Amer. Math. Soc.* **107** (1989) 701–703.

[20] M.N. Vrahatis, G.S. Androulakis and G.E. Manoussakis, A new unconstrained optimization method for imprecise function and gradient values, *J. Math. Anal. Appl.*, accepted.