A MODIFIED ALGORITHM OF STEEPEST DESCENT METHOD FOR SOLVING UNCONSTRAINED NONLINEAR OPTIMIZATION PROBLEMS

Chein-Shan Liu
Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Jiang-Ren Chang
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

Yung-Wei Chen
Department of Marine Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

Follow this and additional works at: https://jmstt.ntou.edu.tw/journal

Part of the Engineering Commons

A MODIFIED ALGORITHM OF STEEPEST DESCENT METHOD FOR SOLVING UNCONSTRAINED NONLINEAR OPTIMIZATION PROBLEMS

Acknowledgements

A MODIFIED ALGORITHM OF STEEPEST DESCENT METHOD FOR SOLVING UNCONSTRAINED NONLINEAR OPTIMIZATION PROBLEMS

Chein-Shan Liu[1], Jiang-Ren Chang[2], and Yung-Wei Chen[3]

ABSTRACT

The steepest descent method (SDM), which can be traced back to Cauchy (1847), is the simplest gradient method for unconstrained optimization problem. The SDM is effective for well-posed and low-dimensional nonlinear optimization problems without constraints; however, for a large-dimensional system, it converges very slowly. Therefore, a modified steepest decent method (MSDM) is developed to deal with these problems. Under the MSDM framework, the original global minimization problem is transformed into a quadratic-form minimization based on the SDM and the current iterative point. Our starting point is a manifold defined in terms of the quadratic function and a fictitious time variable. Thereafter, we can derive an iterative algorithm by including a parameter in the final stage. Through a Hopf bifurcation, this parameter indeed plays a major role to switch the situation of slow convergence to a new situation that the new algorithm converges faster. Several numerical examples are examined and compared with exact solutions. It is found that the new algorithm of the MSDM has better computational efficiency and accuracy, even for a large-dimensional non-convex minimization problem of the generalized Rosenbrock function.

I. INTRODUCTION

In this paper, we consider the following nonlinear optimi-

Paper submitted 08/15/13; revised 01/21/14; accepted 02/21/14. Author for correspondence: Yung-Wei Chen (e-mail: cyw0710@mail.ntou.edu.tw).

[1] *Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.*

[2] *Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.*

[3] *Department of Marine Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.*

zation problem without constraints:

$$\min f(\mathbf{x}) = 0, \tag{1}$$

where $f: \mathbb{R}^n \mapsto \mathbb{R}$ is a $\mathbb{C}^2$ differentiable function.

For solving (1), there are many approaches of iterative types. If $\mathbf{x}_k$ is the current iterative point, then we denote $f(\mathbf{x}_k)$ by f_k, $\nabla f(\mathbf{x}_k)$ by $\mathbf{g}_k$ and $\nabla^2 f(\mathbf{x}_k)$ by $\mathbf{A}_k$. The second order Taylor expansion of function $f(x)$ at the point $\mathbf{x}_k$ is

$$f(\mathbf{x}) = f_k + \mathbf{g}_k^T \Delta\mathbf{x} + \frac{1}{2}(\Delta\mathbf{x})^T \mathbf{A}_k \Delta\mathbf{x}, \tag{2}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_k$. The superscript T signifies the transpose.

Letting $\mathbf{x} = \mathbf{x}_k - \alpha\mathbf{g}_k$ and inserting it into (2), we can obtain

$$f(\mathbf{x}_k - \alpha\mathbf{g}_k) = f_k - \alpha\mathbf{g}_k^T \mathbf{g}_k + \frac{\alpha^2}{2}\mathbf{g}_k^T \mathbf{A}_k \mathbf{g}_k. \tag{3}$$

Taking a minimization with respect to α, we can solve

$$\alpha = \frac{\|\mathbf{g}_k\|^2}{\mathbf{g}_k^T \mathbf{A}_k \mathbf{g}_k}. \tag{4}$$

Then the well-known steepest descent method (SDM) for solving (1) is obtained:

(i) Give an initial $\mathbf{x}_0$, and then $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$.

(ii) For $k = 0, 1, 2, \dots$ we repeat the following calculations.

If $\|\mathbf{g}_k\| < \varepsilon$, then stop; otherwise, let $k = k + 1$ and find the next $\mathbf{x}_{k+1}$ by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\|\mathbf{g}_k\|^2}{\mathbf{g}_k^T \mathbf{A}_k \mathbf{g}_k}\mathbf{g}_k. \tag{5}$$

Go to step (ii).

Several modifications to the SDM have been addressed.

These modifications have led to a new interest in the SDM that the gradient vector itself is not a bad choice but rather that the original step-length leads to the slow convergence behavior. Barzilai and Browein (1988) were the first to present a new choice of step-length through two-point step-size. Although their method did not guarantee descent of the minimum functional values, it could produce a substantial improvement of the convergence speed for a two dimensional quadratic function. According to their method, many researchers have proposed some choices of the step-length for the gradient method, for example, Raydan (1993; 1997); Friedlander et al. (1999); Dai and Liao (2002); Dai et al. (2002); Raydan and Svaiter (2002); Dai and Yuan (2003); Fletcher (2005), and Yuan (2006). In this research, we will approach this problem from a quite different view of invariant manifold and bifurcation, and propose a new strategy to modify the step-length. Besides the SDM related methods, there were many modifications of the conjugate gradient method for the unconstrained optimization problems, such as Birgin and Martinez (2001); Andrei (2007; 2008; 2010); Shi and Guo (2009); Zhang (2009); Babaie-Kafaki et al. (2010), and references therein.

II. THE BASIC FORMULATION

From the derivation of SDM for solving (1), it is easy to see that the global minimization problem is transformed into a local minimization problem of

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x} + c_0 , \tag{6}$$

where $c_0 = f_k + \mathbf{g}_k^{\mathrm{T}}\mathbf{x}_k + \mathbf{x}_k^{\mathrm{T}}\mathbf{A}_k\mathbf{x}_k/2$ and $\mathbf{b} = \mathbf{A}_k\mathbf{x}_k - \mathbf{g}_k$. Note that the former is a constant scalar and the latter is a constant vector if the coefficient α in $\mathbf{x} = \mathbf{x}_k - \alpha\mathbf{g}_k$ is determined. Here for the general purpose, we omit the subscript k, and then modify the SDM from this quadratic function.

According to the modified SDM proposed by Liu (2012), we consider an evolutional behavior of $\mathbf{x}$ from the ODEs defined on a manifold formed from $\phi(\mathbf{x})$

$$h(\mathbf{x}, t) := Q(t)\phi(\mathbf{x}) = C . \tag{7}$$

Here, we let $\mathbf{x}$ be a function of a fictitious time variable t. We do not need to specify the function $Q(t)$ as a priori, of which $C/Q(t)$ is a measure of the decreasing of ϕ in time. Hence, we expect that in our algorithm $Q(t) > 0$ is an increasing function of t. We let $Q(0) = 1$ and C is determined by the initial condition $\mathbf{x}(0) = \mathbf{x}_0$ with

$$C = \phi(\mathbf{x}(0)) > 0 . \tag{8}$$

We can suitably choose the constant c_0 in (6) such that $\phi(\mathbf{x}) \geq 0$.

When $C > 0$ and $Q > 0$, the manifold defined by (7) is continuous. Thus, the following differential operation carried out on the manifold makes sense. For the requirement $\mathbf{x} = \mathbf{x}(t)$, we have

$$\dot{Q}(t)\phi(\mathbf{x}) + Q(t)(\mathbf{A}\mathbf{x} - \mathbf{b}) \cdot \dot{\mathbf{x}} = 0 . \tag{9}$$

We suppose that $\mathbf{x}$ is governed by a gradient-flow:

$$\dot{\mathbf{x}} = -\lambda\frac{\partial\phi}{\partial\mathbf{x}} = -\lambda(\mathbf{A}\mathbf{x} - \mathbf{b}) , \tag{10}$$

where λ is to be determined. Inserting (10) into (9) we can solve

$$\lambda = \frac{q(t)\phi}{\|\mathbf{g}\|^2} , \tag{11}$$

where

$$\mathbf{g} := \mathbf{A}\mathbf{x} - \mathbf{b} , \tag{12}$$

and

$$q(t) := \frac{\dot{Q}(t)}{Q(t)} . \tag{13}$$

Thus inserting (11) into (10), we can obtain an evolution equation for $\mathbf{x}$ defined by a gradient-flow:

$$\dot{\mathbf{x}} = -q(t)\frac{\phi}{\|\mathbf{g}\|^2}\mathbf{g} . \tag{14}$$

Hence, in our algorithm, if $Q(t)$ can be guaranteed to be an increasing function of t, we may have an absolutely convergent property in solving the minimum of ϕ through the following equation:

$$\phi(t) = \frac{C}{Q(t)} . \tag{15}$$

When t increases, the above equation can enforce the function ϕ tending to its minimum.

III. NUMERICAL METHODS

1. Keeping x on the Manifold

From the Euler method for (14), we can obtain the following algorithm:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \beta\frac{\phi}{\|\mathbf{g}\|^2}\mathbf{g} , \tag{16}$$

where

$$\beta = q(t)\Delta t . \tag{17}$$

In order to keep **x** on the manifold defined by (15), we can insert the above $\mathbf{x}(t + \Delta t)$ into

$$\frac{1}{2}\mathbf{x}^{\mathrm{T}}(t + \Delta t)\mathbf{A}\mathbf{x}(t + \Delta t) - \mathbf{b}^{\mathrm{T}}\mathbf{x}(t + \Delta t) = \frac{C}{Q(t + \Delta t)} - c_0 \tag{18}$$

to obtain

$$\frac{C}{Q(t + \Delta t)} - c_0 = \frac{1}{2}\mathbf{x}^{\mathrm{T}}(t)\mathbf{A}\mathbf{x}(t) - \mathbf{b}^{\mathrm{T}}\mathbf{x}(t)$$

$$+ \beta\phi\frac{[\mathbf{b}(t) - \mathbf{A}\mathbf{x}(t)]^{\mathrm{T}}\mathbf{g}}{\|\mathbf{g}\|^2} + \beta^2\phi^2\frac{\mathbf{g}^{\mathrm{T}}\mathbf{A}\mathbf{g}}{2\|\mathbf{g}\|^4} . \tag{19}$$

Thus by (12), (15) and (6) and through some manipulations, we have the following scalar equation:

$$\frac{1}{2}a_0\beta - \beta + 1 = \frac{Q(t)}{Q(t + \Delta t)} , \tag{20}$$

where

$$a_0 := \frac{\phi\mathbf{g}^{\mathrm{T}}\mathbf{A}\mathbf{g}}{\|\mathbf{g}\|^4} . \tag{21}$$

2. A Trial Dynamic

From the approximation of

$$Q(t + \Delta t) = Q(t) + \dot{Q}(t)\Delta t , \tag{22}$$

and by (13) and (17), we can derive

$$\frac{Q(t)}{Q(t + \Delta t)} = \frac{1}{1 + \beta} . \tag{23}$$

Inserting it into (20), we come to a cubic equation for β:

$$a_0\beta^2(1 + \beta) - 2\beta(1 + \beta) + 2(1 + \beta) = 2 . \tag{24}$$

It allows a closed-form solution of β:

$$\beta = \frac{2}{a_0} - 1 . \tag{25}$$

Inserting the above β into (16), we can obtain

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \left[\frac{2}{a_0} - 1\right]\frac{\phi}{\|\mathbf{g}\|^2}\mathbf{g} . \tag{26}$$

However, when a_0 approaches to 2, this algorithm fails and stagnates at a point which is not necessarily a solution. In the following, we should avoid adopting this algorithm, which is based on (20), and furthermore enforce the orbit of **x** being constrained by that manifold.

The above derivation hints us that we must abandon the concept of keeping the orbit of **x** on the manifold and then solve β; otherwise, we only have an algorithm which cannot work. Let $s = Q(t)/Q(t + \Delta t)$. By (20) we can derive

$$\frac{1}{2}a_0\beta^2 - \beta + 1 - s = 0 . \tag{27}$$

From (27), we can take the solution of β to be

$$\beta = \frac{1 - \sqrt{1 - 2(1 - s)a_0}}{a_0} , \text{ if } 1 - 2(1 - s)a_0 \geq 0 . \tag{28}$$

Let

$$1 - 2(1 - s)a_0 = \gamma^2 \geq 0 , \quad s = 1 - \frac{1 - \gamma^2}{2a_0} . \tag{29}$$

Thus we have

$$\beta = \frac{1 - \gamma}{a_0} . \tag{30}$$

Here $0 \leq \gamma < 1$ is a parameter.

It is know that in the SDM, we search the next solution $\mathbf{x}(t + \Delta t)$ from $\mathbf{x}(t)$ by minimizing the functional ϕ along the direction $-\mathbf{g}(t)$, i.e.,

$$\min_\alpha \phi(\mathbf{x}(t) - \alpha\mathbf{g}(t)) . \tag{31}$$

Through some calculations, we can obtain

$$\alpha = \frac{\|\mathbf{g}(t)\|^2}{\mathbf{g}^{\mathrm{T}}(t)\mathbf{A}\mathbf{g}(t)} . \tag{32}$$

Thus we have the following iteration formula:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \frac{\|\mathbf{g}(t)\|^2}{\mathbf{g}^{\mathrm{T}}(t)\mathbf{A}\mathbf{g}(t)}\mathbf{g}(t) . \tag{33}$$

Similarly, from (27) we can choose β, which minimizes s to

obtain

$$\beta = \frac{1}{a_0}. \tag{34}$$

Inserting it into (16) and using (21), we can derive the SDM algorithm again as in (33). Below we will demonstrate that this minimization is not the best choice.

3. A Modified Steepest Descent Method (MSDM)

Let $\mathbf{x}_k$ denote the numerical value of $\mathbf{x}$ at the k-th step, and return $\mathbf{g}$ to $\mathbf{g}_k$ and $\mathbf{A}$ to $\mathbf{A}_k$. Thus by inserting (30) for β into (16) and using (21), we can derive an iterative algorithm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \frac{\|\mathbf{g}_k\|^2}{\mathbf{g}_k^{\mathrm{T}} \mathbf{A}_k \mathbf{g}_k} \mathbf{g}_k, \tag{35}$$

where

$$\eta = 1 - \gamma. \tag{36}$$

Therefore, we have the following algorithm:
(i) Give an initial $\mathbf{x}_0$, and then $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$.
(ii) For $k = 0, 1, 2 \dots$ we repeat the following calculations.

If $\|\mathbf{g}_k\| < \varepsilon$ then stop; otherwise, let $k = k + 1$ and find the next $\mathbf{x}_{k+1}$ by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (1 - \gamma) \frac{\|\mathbf{g}(t)\|^2}{\mathbf{g}_k^{\mathrm{T}} \mathbf{A}_k \mathbf{g}_k} \mathbf{g}_k. \tag{37}$$

Go to step (ii). $0 \le \gamma < 1$ is a parameter determined by the user. If $\gamma = 0$, the above algorithm is reduced to the steepest descent method (SDM).

IV. NUMERICAL EXAMPLES

In order to assess the performance of the newly developed method, let us investigate the following examples. Some results are compared with those obtained from the steepest descent method (SDM). In order to emphasize the difference of our new algorithm from the SDM, we might call the present modification as a modified steepest descent method (MSDM).

Example 1

We will first consider a simple case:

$$\min f = \frac{1}{3} x_1^3 + x_1^2 + \frac{1}{2} x_2^2 + 3x_2. \tag{38}$$

The minimum of f is -4.5 occurring at $(x_1, x_2) = (0, 3)$. We apply the MSDM to this problem starting at $(x_1, x_2) = (10, 10)$ under a convergence criterion $\varepsilon = 10^{-15}$. When $\gamma = 0$, the
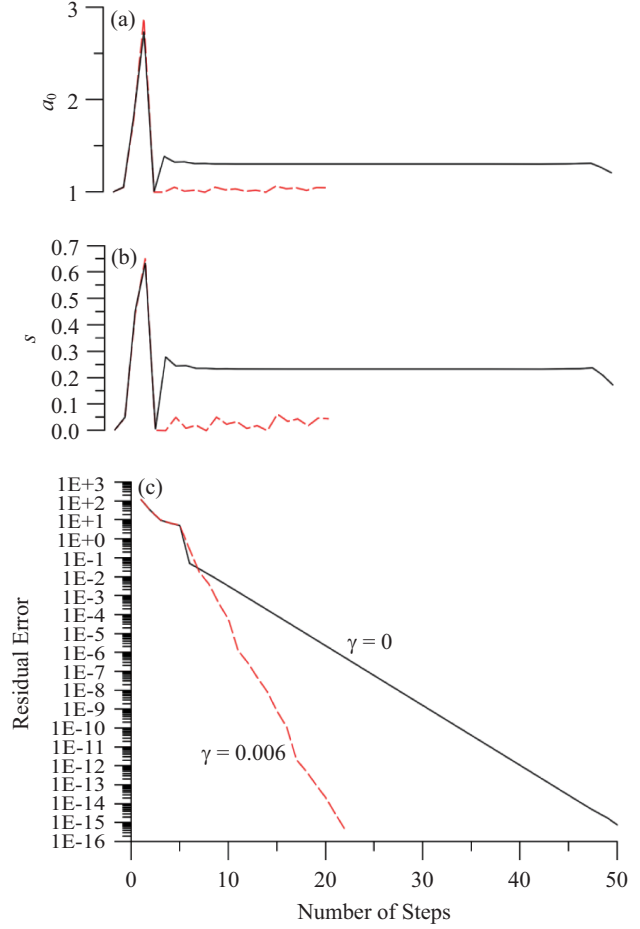

Fig. 1. For a simple case both the SDM and MSDM can converge very fast.

MSDM is reduced to the SDM. Under the above stopping criterion, the SDM is convergent with 50 steps as shown in Fig. 1 by solid lines for showing a_0, s and residual error. The SDM can reach a very accurate value of $(x_1, x_2) = (3.25 \times 10^{-16}, 3)$. At the same time, the MSDM with $\gamma = 0.006$ converges very fast with only 22 steps, with a_0, s and residual error shown in Fig. 1 by dashed lines. The MSDM is 2 times faster than the SDM, and furthermore we can get $(x_1, x_2) = (-1.65 \times 10^{-17}, 3)$. From Figs. 1(a) and (b), we can understand that the converging speed of the MSDM is faster than that of the SDM, because a_0 and s of the MSDM are much smaller than those of the SDM.

Example 2

As a comparison with SDM, we use the following function given by Rosenbrock (1960):

$$\min f = 100(x_2 + x_1^2)^2 + (1 - x_1)^2. \tag{39}$$

In mathematical optimization, the Rosenbrock function is a non-convex function used as a performance test case for optimization algorithms. It is also known as Rosenbrock's valley
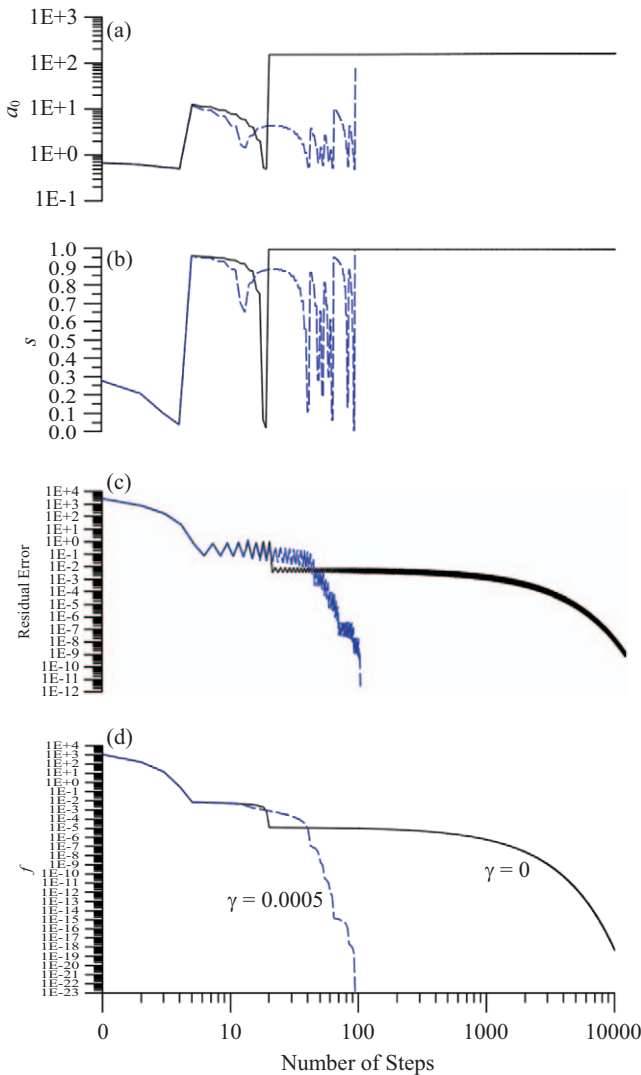
Fig. 2. For a Rosenbrock optimization problem the new algorithm is one hundred times faster than the classical steepest descent method.

Fig. 3. For a Rosenbrock optimization problem comparing the iterative paths of the SDM and the MSDM.

or Rosenbrock's banana function. The minimum is zero occurring at $(x_1, x_2) = (1, 1)$. This function is difficult to minimize because it has a steep sided valley following the parabolic curve $x_1^2 = x_2$. Kuo et al. (2006) have used the particle swarm method to solve this problem; however, the numerical procedures are rather complex. Liu and Atluri (2008) have applied a fictitious time integration method to solve the above problem under the constraints of $x_1 \geq 0$ and $x_2 \geq 0$, whose accuracy can reach to the fifth order.

We apply the MSDM to this problem starting at (2, 0.5) under a convergence criterion $\varepsilon = 10^{-10}$ or a maximum number 10000 of iterations. The SDM is run over 10000 steps without convergence as shown in Fig. 2 by solid lines for showing a_0 and s, residual error and f. The SDM can reach a very accurate value of f with 4.95×10^{-19}. The MSDM with $\gamma = 0.0005$ converges very fast only with 94 steps, with a_0, s, residual error and f shown in Fig. 2 by dashed lines. The
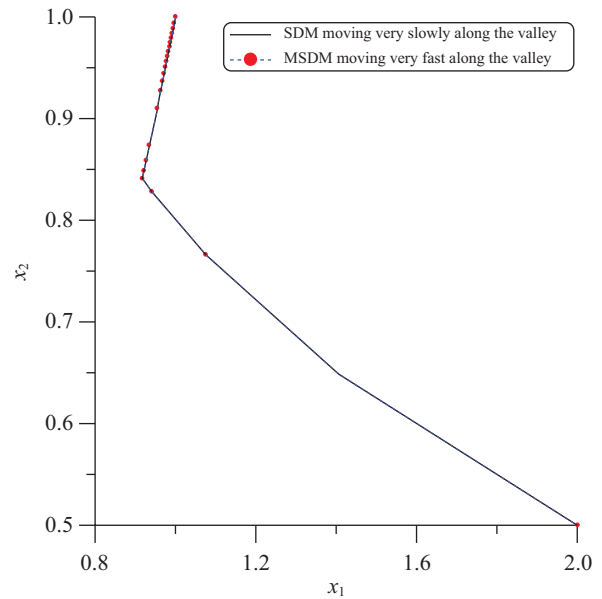
MSDM is 100 times faster than the SDM, and furthermore f can be reduced to 1.12×10^{-23}.

In Fig. 3, we compare the iterative paths generated by the SDM and the MSDM. It is found that both algorithms are fast approaching to the valley. In addition, when the SDM is moving very slowly along the valley, the MSDM is moving very fast to the solution.

Now, we can explain the parameter appeared in (37). In Fig. 2(a), we compare a_0 for $\gamma = 0$ and $\gamma = 0.0005$. It can be seen that for the case with $\gamma = 0$, the values of a_0 tend to a constant and keep unchanged. By (21) it means that there exists an attracting set for the iterative orbit of $\mathbf{x}$ described by the following manifold:

$$a_0 = \frac{f\mathbf{g}^\mathbf{T}\mathbf{A}\mathbf{g}}{\|\mathbf{g}\|^4} = \text{Constant} . \tag{40}$$

When the iterative orbit approaches to this manifold, the residual error is reduced slowly as shown in Fig. 2(c) by solid line, whereas the ratio of s is also keeping near to 1 as shown in Fig. 2(b) by the solid line. Conversely, for the case $\gamma = 0.0005$, a_0 is no more tending to a constant as shown in Fig. 2(a) by the dashed line. Because the iterative orbit is not attracted by a constant manifold, the values of f as shown in Fig. 2(d) by the dashed line can be reduced step by step, whereas the ratio of s is often leaving the value near to 1 as shown in Fig. 2(b) by the dashed line. Thus, we can observe that when $\mathbf{s}$ varies from zero to a positive value, the iterative dynamics as given by (37) undergoes a Hopf bifurcation, like as the ODEs behavior observed by Liu (2000; 2007). The original stable manifold existent for $\gamma = 0$ now becomes a ghost manifold for $\gamma = 0.0005$,

Table 1. Comparison of x_i by different methods.

x_{EX}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
x_{GA}	0.9942	0.9875	0.977	0.9604	0.9274	0.9056	0.8474	0.729	0.5308	0.2649	0.0756
x_{LCGA}	0.9937	1.0	1.0	1.0	0.9984	0.9969	0.9937	0.9906	0.9813	0.962	0.9254
x_{MSDM}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
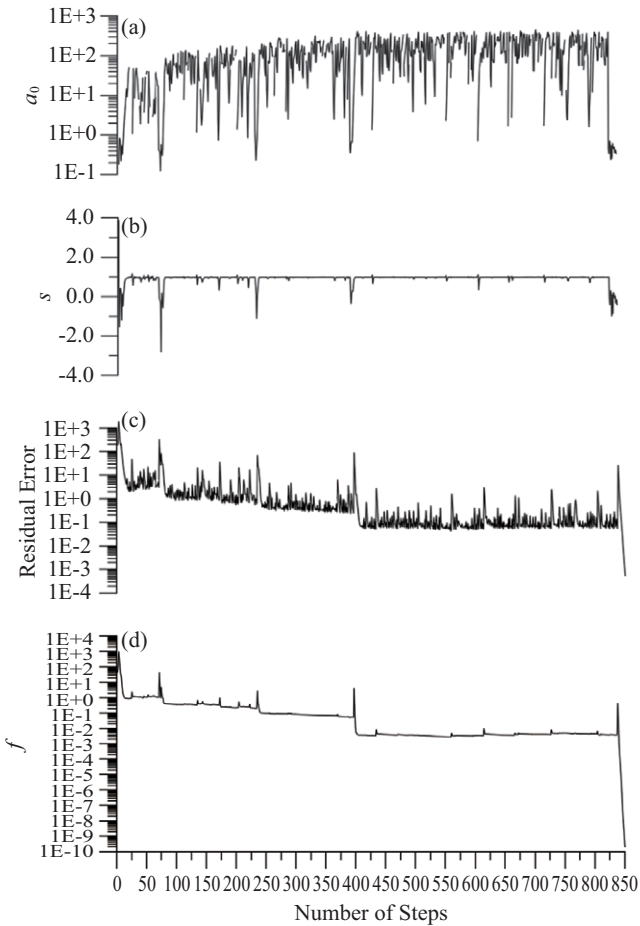
Fig. 4. For a generalized Rosenbrock optimization problem with $n = 11$ showing (a) a_0, (b) s, (c) residual error and (d) f calculated by using the MSDM.

and thus the iterative orbit generated from the algorithm of the MSDM with $\gamma = 0.0005$ is not attracted by that manifold again. Instead of the intermittency occupy, an irregularly jumping behavior of a_0 and the residual error are shown respectively in Figs. 2(a) and 2(c) by dashed lines.

Example 3

Next we consider a generalization of the Rosenbrock function (Bouvry et al., 2000; Kok and Sandrock, 2009):

$$\min f = \sum_{k=1}^{n-1} \left[100(x_{k+1} + x_k^2)^2 + (1 - x_k)^2 \right]. \tag{41}$$

This variant has been shown to have exactly one minimum

for $n = 3$ at $(x_1, x_2, x_3) = (1, 1, 1)$ and exactly two minima for $4 \le n \le 7$. The global minimum of all ones and a local minimum are near $(x_1, x_2, \ldots, x_n) = (-1, 1, \ldots, 1)$. This result is obtained by setting the gradient of the function equal to zero. Note that the resulting equation is a rational function of x_i.

For small n, the polynomials can be determined exactly and Sturm's theorem can be used to determine the number of real roots, while the roots can be bounded in the region of $|x_i| < 2.4$ (Kok and Sandrock, 2009). For larger n, this method breaks down due to the size of the coefficients involved.

In order to compare the results obtained by the MSDM with those calculated by the genetic algorithm (GA) and the loosely coupled genetic algorithm (LCGA) as reported by Bouvry et al. (2000), we take $n = 11$ and start from $x_i = 0.7$ as those used in the above paper. In Fig. 4, we show a_0, s, residual error and f obtained by the MSDM with $\gamma = 0.2$. We found that the MSDM is convergent with 850 steps under a convergence criterion $\varepsilon = 10^{-3}$, of which the final f is very small with the value of 2.02×10^{-10} as shown in Fig. 4(d). We compare the values of x_i with those calculated by the GA and the LCGA in Table 1, where the GA and the LCGA are run to 300 generations with $f = 1.2019$ for the GA and $f = 0.0188$ for the LCGA. Obviously, the MSDM is much more accurate than the GA and the LCGA.

Then we apply the MSDM to this problem with $n = 20$ starting at $x_i = 0$ under a convergence criterion $\varepsilon = 10^{-8}$ or a maximum number 20000 of iterations. Under the above stopping criterion, the SDM is run over 20000 steps without convergence as shown in Fig. 5 by solid lines for showing a_0, s, residual error and f. The SDM can reach a very accurate value of $f = 4.97 \times 10^{-15}$. The MSDM with $\gamma = 0.0008$ converges with 8663 steps, with a_0, s, residual error and f shown in Fig. 5 by dashed lines. The MSDM is 3 times faster than the SDM, and similarly f can be reduced to 1.3×10^{-16}.

Example 4

We consider a case due to Powell (1962):

$$\min f = (x_1 + 10x_2)^2 + 5(x_3 + 10x_4)^2 + (x_2 - 2x_3)^4$$
$$+ 10(x_1 - 2x_4)^4. \tag{42}$$

The minimum of f is zero occurring at $(x_1, x_2, x_3, x_4) = (0, 0, 0, 0)$. We apply the MSDM to this problem starting at $(x_1, x_2, x_3, x_4) = (3, -1, 0, 1)$ under a convergence criterion of $\varepsilon = 10^{-8}$. The SDM is convergent very slowly with 49846 steps as shown in Fig. 6 by solid lines for showing a_0, s, and f. The
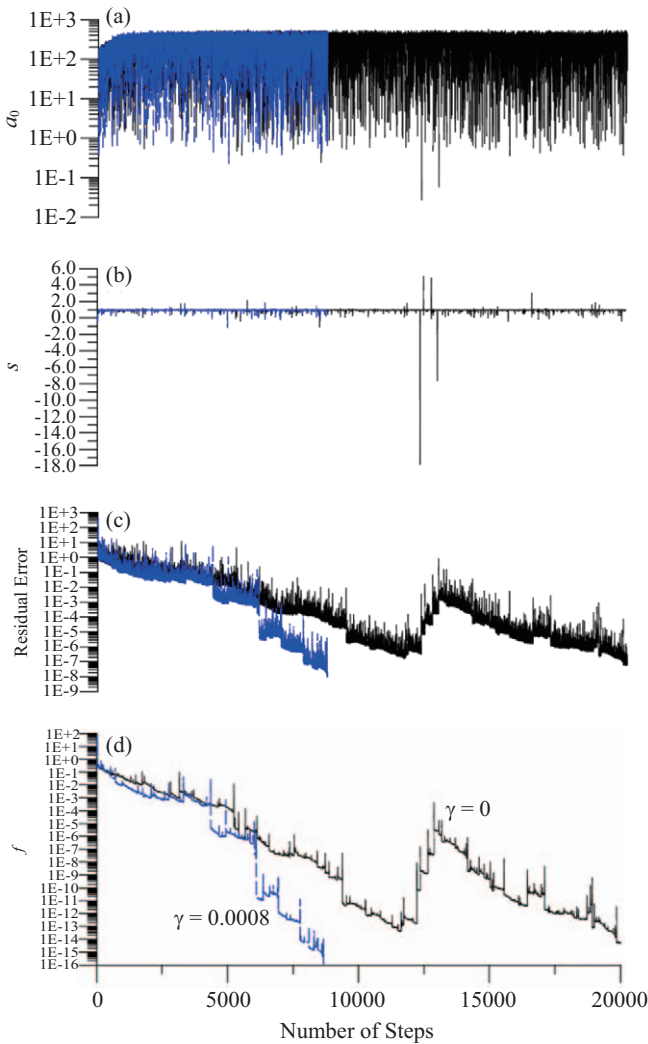
Fig. 5. For a generalized Rosenbrock optimization problem with $n = 20$ the new algorithm is three times faster than the classical steepest descent method.

Fig. 6. For the Powell case comparing (a) a_0, (b) s and (c) f obtained by the SDM and the MSDM.

SDM can reach a very accurate value of $f = 10^{-7}$. At the same time, the MSDM with $\gamma = 0.15$ converges with 1301 steps, with a_0, s, and f shown in Fig. 6 by dashed lines. The MSDM is 38 times faster than the SDM, and furthermore we can get a more accurate $f = 9.96 \times 10^{-9}$.

Example 5

In this example, we design an office block inside a structure with a curved roof given by $x = 100 - y^2$. Suppose that the number of total cuboids is n and each cuboid can have different size. We attempt to find the dimensions of all cuboids with maximum volume which would fit inside the given roof structure, that is,

$$\max f = y_1 \left[100 - y_1^2 \right] + y_2 \left[100 - (y_1 + y_2)^2 \right] + \ldots$$

$$+ y_n \left[100 - (y_1 + \ldots + y_n)^2 \right], \tag{43}$$

where $y_1 > 0$ is the height of the i-th cuboid.

The maximum of f is tending to 2000/3 when n is increasing. When $n = 95$, we apply the MSDM to this problem starting at $y_i = 0.05$ under a convergence criterion of $\varepsilon = 10^{-5}$. The SDM is convergent with 6868 steps as shown in Fig. 7 by solid lines for showing a_0, s, residual error and f. At the same time, the MSDM with $\gamma = 0.35$ converges with 502 steps, with a_0, s, residual error and f shown in Fig. 7 by dashed lines. Both f of the SDM and the MSDM are tending to 661.9945. The MSDM is 13 times faster than the SDM. The heights and widths of the cuboids with respect to the number of floors are plotted in Fig. 8.

Example 6

In this example, we design an office block inside a structure with a circular roof given by $x = \sqrt{1296 - y^2}$. Here we fix $n = 95$, and consider

$$\max f = y_1 \sqrt{1296 - y^2} + y_2 \sqrt{1296 - (y_1 + y_2)^2} + \ldots$$

$$+ y_n \sqrt{1296 - (y_1 + \ldots + y_n)^2}. \tag{44}$$

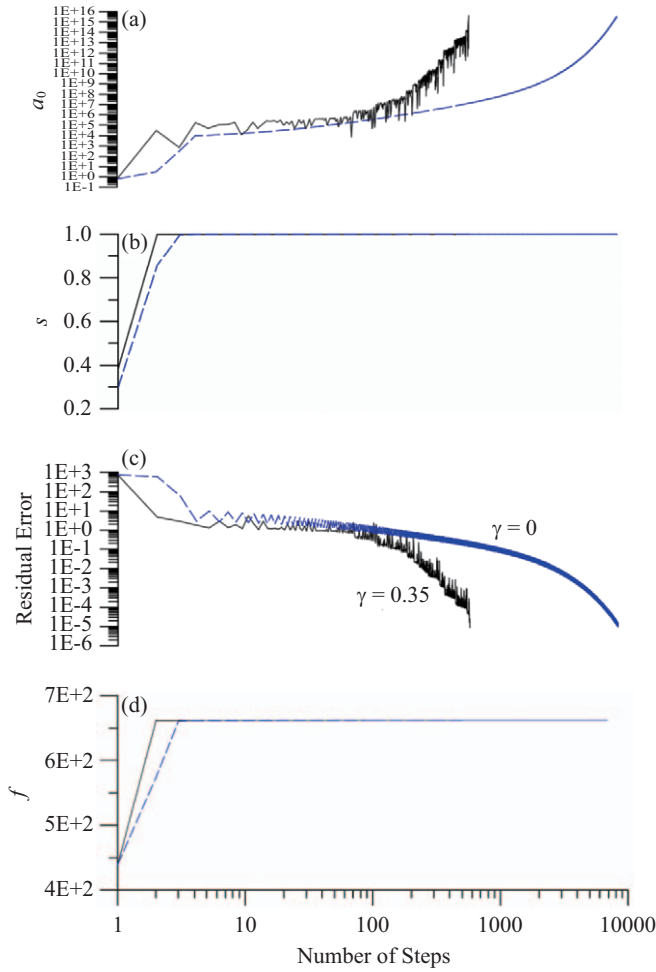This problem is more difficult than that in Example 5.

Fig. 7. For the maximum area under a given curve comparing (a) a_0, (b) s, (c) residual error and (d) f obtained by the SDM and the MSDM.
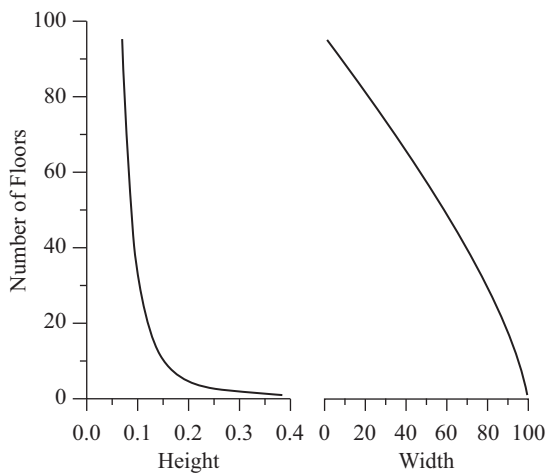

Fig. 8. Showing the heights and widths of the floors with respect to the number of floors

The maximum of f is tending to $324\pi = 1017.88$ when n is increasing. We apply the MSDM to this problem starting at $y_i = 1$ under a convergence criterion of $\varepsilon = 10^{-3}$. The SDM is
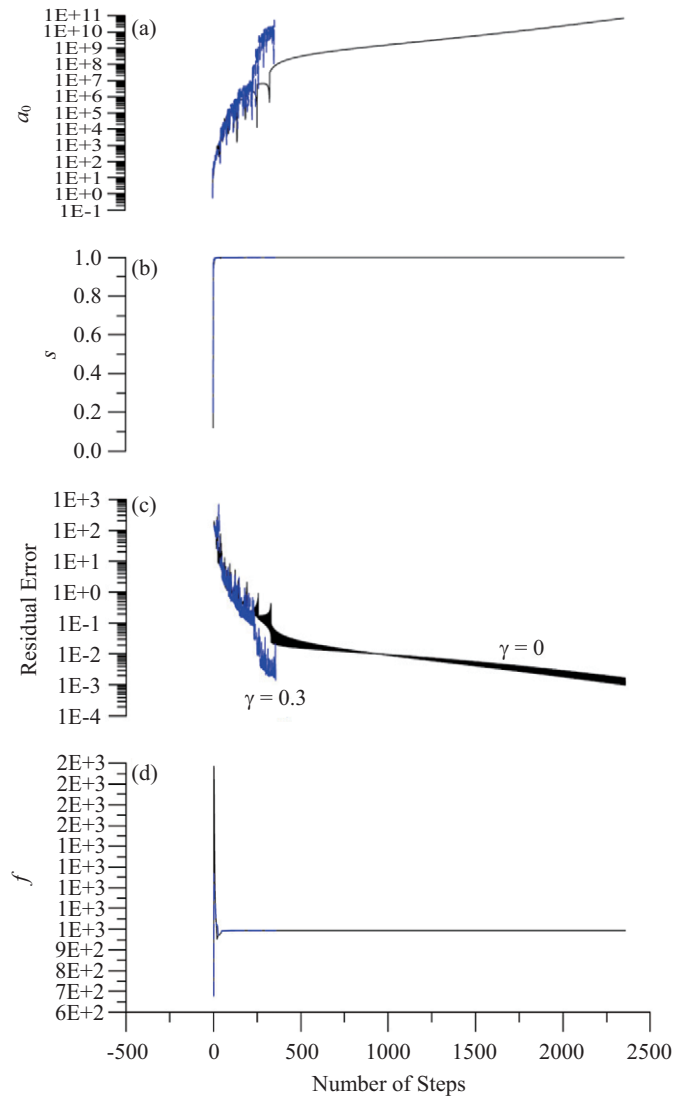

Fig. 9. For the maximum area under a circular roof comparing (a) a_0, (b) s, (c) residual error and (d) f obtained by the SDM and the MSDM.

convergent with 2358 steps as shown in Fig. 9 by solid lines for showing a_0, s, residual error and f. At the same time, the MSDM with $\gamma = 0.3$ converges with 356 steps, with a_0, s, residual error and f shown in Fig. 9 by dashed lines. Each f of the two methods is tending to 994.2315. The MSDM is 7 times faster than the SDM. The heights and widths of the cuboids with respect to the number of floors are plotted in Fig. 10.

Example 7

In this example, we test the minimization of the Schwefel function with $n = 100$:

$$\min f = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2. \tag{45}$$
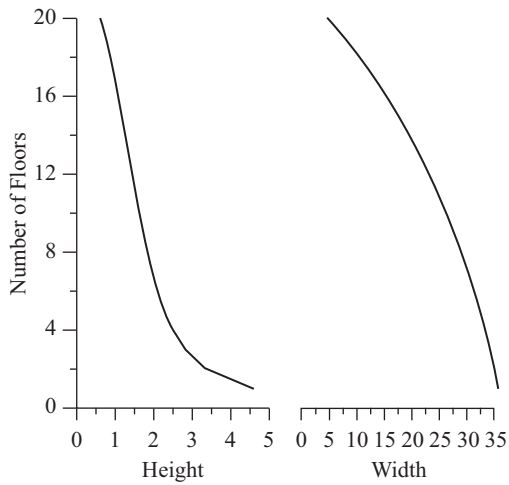
Fig. 10. Showing the heights and widths of the floors with respect to the number of floors for a circular roof.
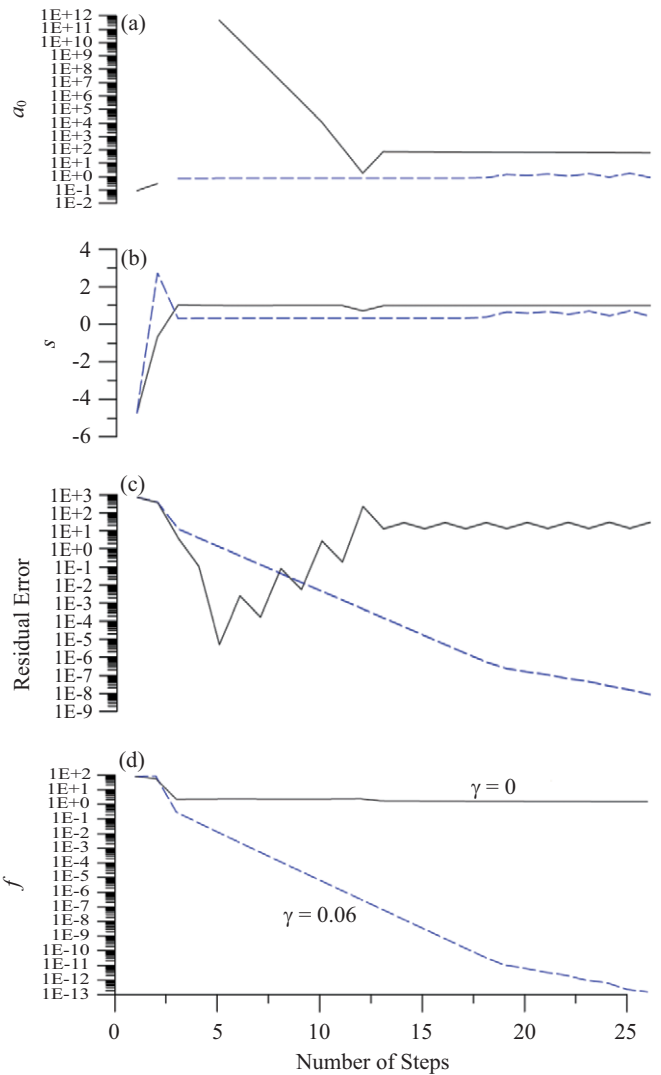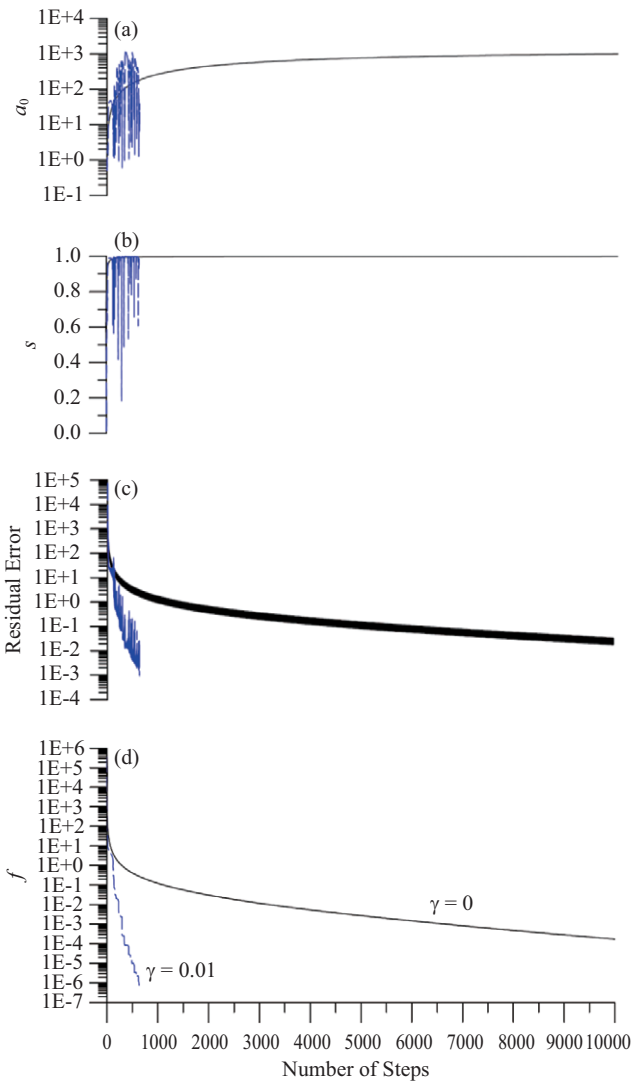

Fig. 11. For the minimization of Schwefel function comparing (a) a_0, (b) s, (c) residual error and (d) f obtained by the SDM and the MSDM.

Fig. 12. For the minimization of Whitley function comparing (a) a_0, (b) s, (c) residual error and (d) f obtained by the SDM and the MSDM.

The minimum is zero at $x_j = 0, j = 1, \ldots, n$.

We apply the MSDM to this problem starting at $x_i = 1$ under a convergence criterion of $\varepsilon = 10^{-3}$. The SDM does not converge with 10000 steps as shown in Fig. 11 by solid lines for showing a_0, s, residual error and f. At the same time, the MSDM with $\gamma = 0.01$ converges with 634 steps, with a_0, s, residual error and f shown in Fig. 11 by dashed lines. The MSDM is over 13 times faster than the SDM, and f is tending to 7.65×10^{-7}, which is more accurate than 1.733×10^{-4} obtained by the SDM.

Example 8

In this example, we test the minimization of the Whitley function with $n = 8$:

$$\min f = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\frac{y_{ji}^2}{4000} - \cos y_{ji} + 1 \right], \qquad (46)$$

where $y_{ji} = 100(x_i - x_j^2)^2 + (x_j - 1)^2$. The minimum is zero at $x_j = 1, j = 1, \ldots, n$.

It is very difficult to optimize. We apply the MSDM to this problem starting at $x_i = 1.12$ under a convergence criterion of $\varepsilon = 10^{-8}$. The SDM diverges as shown in Fig. 12 by solid lines for showing a_0, s, residual error and f. At the same time, the MSDM with $\gamma = 0.06$ converges with 26 steps, with a_0, s, residual error and f shown in Fig. 12 by dashed lines, and f is tending to 1.54×10^{-13}. It can be found that the SDM fails.

V. CONCLUSIONS

By embedding the minimization problem into a continuous manifold with a fictitious time, we can derive a governing ODE for the unknown vector. Then by employing the Euler scheme, we have derived an iterative algorithm, which is naturally rendered to a modification of the classical steepest descent method (SDM) with a critical parameter $0 \le \gamma < 1$. This novel algorithm might be named a modified steepest descent method (MSDM). We have proved that the minimizations in the SDM and in our formulation lead to the same algorithm, and are not the best ones, which usually result in a quite slow convergence of finding solution. The parameter γ is a bifurcation parameter, which played the role to change the situation from a slow convergence with $\gamma = 0$ to a quick convergence with $\gamma > 0$. This bifurcation is indeed an intermittent chaos which destabilizes the original invariant manifold existent for $\gamma = 0$ in the SDM algorithm and is also the main reason to cause a slow convergence of the SDM for solving optimization problems. Through several tests, we have found that the MSDM outperforms very well as compared with the SDM.

ACKNOWLEDGMENTS

REFERENCES

Andrei, N. (2007). Scaled conjugate gradient algorithms for unconstrained optimization. Computational Optimization and Applications 38, 401-416.

Andrei, N. (2008). A Dai-Yuan conjugate gradient algorithm with sufficient descent and conjugacy conditions for unconstrained optimization. Applied Mathematics Letters 21, 165-171.

Andrei, N. (2010). New accelerated conjugate gradient algorithms as a modification of Dai-Yuan's computational scheme for unconstrained optimization. Journal of Computational and Applied Mathematics 234, 3397-3410.

Babaie-Kaffaki, S., R. Ghanbari and N. Mahdavi-Amiri (2010). Two new conjugate gradient methods based on modified secant equations. Journal of Computational and Applied Mathematics 234, 1374-1386.

Barzilai, J. and J. M. Borwein (1988). Two point step size gradient methods. IMA Journal of Numerical Analysis 8, 141-148.

Birgin, E. G. and J. M. Martinez (2001). A spectral conjugate gradient method for unconstrained optimization. Applied Mathematics & Optimization 43, 117-128.

Bouvry, P., F. Arbab and F. Seredynski (2000). Distributed evolutionary optimization, in manifold: Rosenbrock's function case study. Information Computer Science 122, 141-159.

Dai, Y. H. and L. H. Liao (2002). R-linear convergence of the Barzilai and Borwein gradient method. IMA Journal of Numerical Analysis 22, 1-10.

Dai, Y. H., J. Y. Yuan and Y. Yuan (2002). Modified two-point stepsize gradient methods for unconstrained optimization. Computational Optimization and Applications 22, 103-109.

Dai, Y. H. and Y. Yuan (2003). Alternate minimization gradient method," IMA Journal of Numerical Analysis 23, 377-393.

Fletcher, R (2005). On the Barzilai-Borwein gradient method. In: Optimization and Control with Applications, edited by L. Qi, K. Teo and X. Yang, Springer, New York, 235-256.

Friedlander, A., J. M. Martinez, B. Molina and M. Raydan (1999). Gradient method with retards and generalizations. SIAM Journal on Numerical Analysis 36, 275-289.

Kok, S. and C. Sandrock (2009). Locating and characterizing the stationary points of the extended Rosenbrock function. Evolutionary computation 17, 437-453.

Kuo, H. C., J. R. Chang and C. H. Liu (2006). Particle swarm optimization for global optimization problems. Journal of Marine Science and Technology 14, 170-181.

Liu, C. S. (2000). Intermittent transition to quasi periodicity demonstrated via a circular differential equation. International Journal of Non-Linear Mechanics 35, 931-946.

Liu, C. S. (2007). A study of type I intermittency of a circular differential equation under a discontinuous right-hand side. Journal of Mathematical Analysis and Applications, 331, 547-566.

Liu, C. S. (2012). Modifications of steepest descent method and conjugate gradient method against noise for ill-posed linear system. Communications in Numerical Analysis 1, Doi: 10.5899/2012/can-00115.

Liu, C. S. and S. N. Atluri (2008). A fictitious time integration method (FTIM) for solving mixed complementarity problems with applications to non-linear optimization. CMES: Computer Modeling in Engineering and Sciences 34, 155-178.

Powell, M. J. D. (1962). An iterative method for finding stationary values of a function of several variables. The Computer Journal 5, 147-151.

Raydan, M. (1993). On the Barzilai and Borwein choice of step length for the gradient method. IMA Journal of Numerical Analysis 13, 321-326.

Raydan, M. (1997). The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM Journal on Optimization 7, 26-33.

Raydan, M. and B. F. Svaiter (2002). Relaxed steepest descent and Cauchy-Barzilai-Borwein method. Computational Optimization and Applications 21, 155-167.

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. The Computer Journal 3, 175-184.

Shi, Z. J. and J. Guo (2009). A new family of conjugate gradient methods. Journal of Computational and Applied Mathematics 224, 444-457.

Yuan, Y. (2006). A new step size for the steepest descent method. Journal of Computational Mathematics 24, 149-156.

Zhang, L. (2009). A new Liu-Storey type nonlinear conjugate gradient method for unconstrained optimization problems. Journal of Computational and Applied Mathematics 225, 146-157.