Combining Trust Region Techniques and Rosenbrock Methods for Gradient Systems *

Xin-long Luo † , C. T. Kelley ‡ , Li-Zhi Liao § , Hon Wah Tam ¶

March 28, 2006

Abstract

Rosenbrock methods are popular for solving stiff initial value problems for ordinary differential equations. One advantage is that there is no need to solve a nonlinear equation at every iteration, as compared with other implicit methods such as backward difference formulas and implicit Runge-Kutta methods. In this paper, we introduce some trust region techniques to control the time step in the second order Rosenbrock methods for gradient systems. These techniques are different from the local error control schemes. Both the global and local convergence of the new class of trust region Rosenbrock methods for solving the equilibrium points of gradient systems are addressed. Finally some promising numerical results are presented.

Keywords: Trust region method, unconstrained optimization, Rosenbrock method, gradient system, ordinary differential equation.

1. Introduction

In solving the following unconstrained optimization problem

$$\min_{x \in R^n} \; f(x),$$

there are many methods (see [1, 2, 3, 4, 6, 22, 23, 41, 43, 44]) that convert the optimization problem into the following initial value ordinary differential equation (ODE)

$$\frac{dx(t)}{dt} = F(x(t)), \; t > 0, \quad x(0) = x_0, \tag{1.1}$$

where the function F has the form $F(x) = -\nabla f(x)$ and is continuously differentiable. Equation (1.1) is also called the gradient system (see [47]). The solution $x(t)$ of (1.1) has no closed orbit and tends to the equilibrium point $x^\star$ (i.e. $\nabla f(x^\star) = 0$) as $t \to \infty$ (see [22, 23, 47]). Under some conditions, the equilibrium point $x^\star$ of the gradient system is equivalent to the local minimizer of f.

*This research was supported in part by grant #DMS-0404537 from the United States National Science Foundation, and grant number #W911NF-05-1-0171 from the United States Army Research Office, and the Research Grant Council of Hong Kong.

†School of Information Engineering, Beijing University of Posts and Telecommunications, P. O. Box 101, 100876, P.R. China, xlluo@sci.hkbu.edu.hk or hopeluoxinlong@sina.com.cn.

‡Center for Research in Scientific Computation and Department of Mathematics, North Carolina State University, Box 8205, Raleigh, N. C. 27695-8205, USA, Tim_Kelley@ncsu.edu.

§Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong, liliao@math.hkbu.edu.hk.

¶Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong, hwtam@comp.hkbu.edu.hk.

There are many efficient numerical methods for solving the ODE (1.1) in the finite interval $[0, T]$ such as linear multistep methods and Runge-Kutta methods (see [8, 20, 21, 45, 46]). Given a classical one-step ODE method, at a time point t_k, $0 \leq t_k \leq T$, let x_k be the computed solution that approximates the exact solution $x(t_k)$. Then the error bound has the form

$$\|x_k - x(t_k)\| \leq c e^{lT} h^r, \tag{1.2}$$

where h is the time step and c, r, l are positive constants (see [20, 48]). Notice that this error bound depends on the interval length T. However, sometimes we are interested in the long term behavior of the dynamical system, as mentioned in the above case of solving the optimization problem. In this case, even if $x(t_k)$ tends to the equilibrium $x^\star$, the approximate solution x_k may not converge to $x^\star$ as $t_k \to \infty$.

Humphries and Stuart [27, 49, 50] considered fixed or variable time step Runge-Kutta methods to solve the gradient system (1.1). They obtained approximate solutions that preserve the gradient structure if the time step is sufficiently small and f is a Lyapunov function. However, the convergence rates of these methods are slow because they require a sufficiently small time step. Furthermore, these methods may be unnecessarily computationally costly if we are mainly concerned with the equilibrium and not with the intermediate course of (1.1). On the other hand, Newton's method or quasi-Newton methods for $\nabla f(x) = 0$ alone will usually not suffice if the initial point is too far away from the root. Standard globalization strategies (see [7, 14, 15, 28, 38, 51]), such as line search or trust region methods often converge to nonphysical solution of the steady state (see [9, 10, 16] and the references therein).

Based on the above discussion, we will use Runge-Kutta methods for solving the gradient system (1.1), but consider a new time step control strategy that is different from the local error principle. We are particularly interested in the Rosenbrock methods (see [21, 42, 45, 46]) because they (a) belong to a special class of Runge-Kutta methods, (b) are very efficient for stiff problems since they do not need to solve a nonlinear equation at every iteration comparing with the backward difference formulas (see [8, 21]), and (c) have A-stability. In this paper, we will combine the second order Rosenbrock methods with some trust region techniques in controlling the time step to solve the gradient system (1.1). This algorithm is related both to the pseudo-transient continuation (Ψtc) approach for nonlinear equations [29, 11] and optimization [30, 25]. While we control the time step with a trust region approach, as does [25], we use a second-order Rosenbrock method, which is different from both the nonlinear equations and previous optimization work. The resulting new algorithm will have the desirable properties of global convergence and fast local convergence.

A trust-region method similar to Ψtc was proposed for for gradient flows in [25], and is related to methods from [15, 32, 34]. This method can be viewed as a linearized implicit Euler method or a kind of the first order Rosenbrock methods for solving the ODE (1.1). Thus one advantage of the second order Rosenbrock methods is that they have a more accurate trajectory in the transient state, compared with the Levenberg-Marquardt method. Furthermore, Rosenbrock methods can also achieve superlinear convergence in the steady state if we adjust the time step by trust region techniques. These properties are very desirable in practice.

The rest of this paper is organized as follows. In the next section we provide our trust region Rosenbrock methods that combine the second order Rosenbrock formulas and trust region techniques to solve the gradient system (1.1). In Section 3 we analyze the global and local convergence of our trust region Rosenbrock methods for the gradient system. In Section 4 we report the numerical results of our trust region Rosenbrock methods along with some other popular methods. Finally, some concluding remarks are drawn in Section 5. Throughout the paper we let $\|\cdot\|$ denote the Euclidean vector norm or its induced matrix norm.

2. Trust region Rosenbrock methods

Rosenbrock methods (see [21, 42]) have been extensively studied for stiff ODEs (see [21, 42, 45, 46, 52]). They need not solve a nonlinear equation at every iteration. They possess A-stability and L-stability, which ensure that the step sizes are not limited by numerical stability. These properties are promising as we are mainly interested in the long term behavior of the gradient system (1.1), and do not care about very highly accurate solutions of its intermediate courses. Thus we consider lower order Rosenbrock methods for (1.1). If we use the first order Rosenbrock method to solve (1.1), we get

$$(I + h_k G_k)s_k = -h_k \nabla f(x_k),$$
$$x_{k+1} = x_k + s_k,$$

where $G_k = \nabla^2 f(x_k)$ and $h_k = t_{k+1} - t_k$ is the time step length (see [4, 25]). If we denote $\lambda_k = 1/h_k$, the above iteration is equivalent to the Levenberg-Marquardt method (2.8). If we apply a special second order Rosenbrock method (see [52], pp. 343) to (1.1), we get

$$\begin{aligned}
&\big(I + h_k(1 - \frac{\sqrt{2}}{2})G_k\big)d_k = -h_k \nabla f(x_k), \\
&\big(I + h_k(1 - \frac{\sqrt{2}}{2})G_k\big)s_k = -h_k \nabla f(x_k + \frac{\sqrt{2}-1}{2}d_k), \\
&x_{k+1} = x_k + s_k.
\end{aligned}$$

If we let $\lambda_k = 1/h_k$ in the above iteration, we obtain the search direction s_k which satisfies (2.3) and (2.4).

In the termination phase, due to the restriction of stability, the Rosenbrock method will normally consume an unnecessary amount of time if the local error principle is used to control the time step. To avoid this situation, we adopt some trust region techniques to adjust the time step and expect to achieve rapid convergence near the equilibrium point. By combining the above second order Rosenbrock methods with some trust region techniques, we suggest the following methods for (1.1).

Algorithm 2.1 ***Trust Region Rosenbrock Methods (TRRM)***

Step 0: Initialize the parameters. *Choose an initial point x_0, an initial parameter λ_0, constants τ, η_1, η_2, γ_1, and γ_2 satisfying*

$$0 < \tau < \eta_1 \le \frac{1}{2} \le \eta_2 < 1 \text{ and } 0 < \gamma_1 < 1 < \gamma_2, \tag{2.1}$$

where τ is a small positive number (such as $\tau = 10^{-4}$). Compute $f(x_0)$ and set $k = 0$.

Step 1: Define the approximate model. *Compute $g_k = \nabla f(x_k)$ and $G_k = \nabla^2 f(x_k)$, and define the approximate function*

$$q_k(s) = s^T g_k + \frac{1}{2}s^T G_k s. \tag{2.2}$$

Step 2: Obtain the search step. *If $\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k \succ 0$, solve*

$$\big(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k\big)d_k = -g_k, \tag{2.3}$$

$$\big(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k\big)s_k = -\nabla f(x_k + \frac{\sqrt{2}-1}{2}d_k) \tag{2.4}$$

for s_k and go to Step 3, else set $\rho_k = -1$ and go to Step 4.

Step 3: Compute the ratio. *Compute $q_k(s_k)$. If*

$$q_k(0) - q_k(s_k) \geq \tau \|g_k\| \min\{\|s_k\|, \|g_k\|/\|G_k\|\}, \tag{2.5}$$

then compute $f(x_k + s_k)$ and set

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{q_k(0) - q_k(s_k)}. \tag{2.6}$$

Otherwise set $\rho_k = -1$ and go to Step 4.

Step 4: Accept the trial point. *If $\rho_k \leq 0$, then set $x_{k+1} = x_k$. Otherwise set $x_{k+1} = x_k + s_k$.*

Step 5: Adjust the parameter λ_{k+1}. *Set*

$$\lambda_{k+1} = \begin{cases} 10\lambda_k, & \text{if } \rho_k < 0, \\ \gamma_2 \lambda_k, & \text{if } 0 \leq \rho_k < \eta_1, \\ \lambda_k, & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ \gamma_1 \lambda_k, & \text{if } \rho_k \geq \eta_2. \end{cases} \tag{2.7}$$

Increase k by one and go to Step 1.

The stopping criterion for the above methods is that $\|g_k\|$ is sufficiently small. We replace the time step h_k with the parameter $\lambda_k = 1/h_k$ in the classical Rosenbrock methods for convenience. We can choose the parameters $\gamma_1 = \frac{1}{2}$, $\gamma_2 = 2$, $\tau = 10^{-4}$, $\eta_1 = 0.25$ and $\eta_2 = 0.75$ in the numerical computation. Our numerical results indicate that the choice of these parameters has little impact on the numerical performance.

It should be mentioned that Step 3 in Algorithm 2.1 requires the search step s_k to satisfy the inequality (2.5). This requirement is not needed in the Levenberg-Marquardt method or the trust region method (see [14, 7, 32, 34, 38, 51]), whose search step s_k is obtained from

$$(\lambda_k I + G_k) s_k = -g_k, \tag{2.8}$$

$$(\lambda_k I + G_k) \succ 0, \tag{2.9}$$

where G_k is a symmetric matrix, $g_k = \nabla f(x_k)$ is the gradient of the function f, and λ_k is a positive parameter. Note that the solution s_k of (2.8)-(2.9) is also the solution of the subproblem (see [17, 18])

$$\min_{s \in R^n} g_k^T s + \frac{1}{2} s^T G_k s, \tag{2.10}$$

$$\text{subject to } \|s\| \leq \|s_k\|. \tag{2.11}$$

Thus, from Theorem 4 in [40] we have

$$q_k(0) - q_k(s_k) \geq \frac{1}{2} \|g_k\| \min\{\|s_k\|, \|g_k\|/\|G_k\|\}. \tag{2.12}$$

Therefore the solution s_k of (2.8)-(2.9) also satisfies (2.5).

It is instructive to consider the case $f(x) = x^4 - x^2$, $x_k = \sqrt{6}/6$, and $\lambda_k = \frac{\sqrt{2}-1}{6}$. From (2.3)-(2.4), we get $g_k = -\frac{2\sqrt{6}}{9}$, $G_k = 0$, and $s_k = -\frac{220(\sqrt{12}+\sqrt{6})}{3}$. Thus $s_k^T g_k > 0$, so that s_k is not a descent direction and therefore does not satisfy (2.5). However, as guaranteed by the following lemma, there always exists a sufficiently large λ_k such that s_k satisfies (2.5):

Lemma 2.2 *Assume that s_k is the solution of (2.3)-(2.4) and $g(x) = \nabla f(x)$ is uniformly continuous. If $\|g_k\| > 0$, we have*

$$\lim_{\lambda_k \to \infty} \frac{-s_k^T g_k}{\|s_k\| \cdot \|g_k\|} = 1. \tag{2.13}$$

Proof. From (2.3) we have

$$\begin{aligned} \|d_k\| &= \|(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1} g_k\| \\ &\leq \frac{\|g_k\|}{\lambda_k - (1 - \sqrt{2}/2)\|G_k\|} \to 0, \text{ as } \lambda_k \to \infty. \end{aligned} \tag{2.14}$$

Thus we get

$$\lim_{\lambda_k \to \infty} \|g(x_k + \frac{\sqrt{2}-1}{2} d_k) - g(x_k)\| = 0 \tag{2.15}$$

because $g(x)$ is uniformly continuous. From (2.3)-(2.4) and (2.14)-(2.15), we have

$$\begin{aligned} \|s_k\| &= \| - (\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1} g(x_k + \frac{\sqrt{2}-1}{2} d_k)\| \\ &\leq \frac{\|g(x_k + (\sqrt{2}/2 - 1/2)d_k) - g(x_k)\| + \|g_k\|}{\lambda_k - (1 - \sqrt{2}/2)\|G_k\|}, \end{aligned} \tag{2.16}$$

and

$$\begin{aligned} -s_k^T g_k &= g_k^T (\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1} g(x_k + \frac{\sqrt{2}-1}{2} d_k) \\ &= g_k^T (\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1} g_k + g_k^T (\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1} [g(x_k + \frac{\sqrt{2}-1}{2} d_k) - g_k] \\ &\geq \frac{\|g_k\|^2}{\lambda_k + (1 + \sqrt{2}/2)\|G_k\|} - \frac{\|g_k\| \cdot \|g(x_k + (\sqrt{2}/2 - 1/2)d_k) - g_k\|}{\lambda_k - (1 - \sqrt{2}/2)\|G_k\|} \end{aligned} \tag{2.17}$$

as $\lambda_k > (1 - \sqrt{2}/2)\|G_k\|$. Therefore, from (2.16) and (2.17) we obtain

$$\begin{aligned} \frac{-s_k^T g_k}{\|s_k\| \cdot \|g_k\|} &\geq \frac{\frac{\|g_k\|}{\lambda_k + (1+\sqrt{2}/2)\|G_k\|} - \frac{\|g(x_k + (\sqrt{2}/2 - 1/2)d_k) - g_k\|}{\lambda_k - (1-\sqrt{2}/2)\|G_k\|}}{\frac{\|g(x_k + (\sqrt{2}/2-1/2)d_k) - g(x_k)\| + \|g_k\|}{\lambda_k - (1-\sqrt{2}/2)\|G_k\|}} \\ &= \frac{\|g_k\| \frac{\lambda_k - (1-\sqrt{2}/2)\|G_k\|}{\lambda_k + (1-\sqrt{2}/2)\|G_k\|} - \|g(x_k + (\sqrt{2}/2 - 1/2)d_k) - g(x_k)\|}{\|g(x_k + (\sqrt{2}/2 - 1/2)d_k) - g(x_k)\| + \|g_k\|}, \end{aligned} \tag{2.18}$$

which gives

$$\lim_{\lambda_k \to \infty} \frac{-s_k^T g_k}{\|s_k\| \cdot \|g_k\|} \geq 1. \tag{2.19}$$

On the other hand, we have

$$\frac{-s_k^T g_k}{\|s_k\| \cdot \|g_k\|} \leq 1 \tag{2.20}$$

by the Cauchy-Schwartz inequality. Therefore inequalities (2.19) and (2.20) yield (2.13). □

From Lemma 2.2 and Algorithm 2.1, we can find a sufficiently large parameter λ_k such that s_k satisfies (2.5), if the iterative point x_k is far away from the equilibrium point $x^\star$ of the gradient system (1.1). On the other hand, we would adopt a small parameter λ_k if possible so that Algorithm 2.1 can achieve rapid convergence as x_k falls into a neighborhood of $x^\star$.

3. The convergence analysis

In this section, we will analyze the global and local convergence of Algorithm 2.1. The following theorem shows that Algorithm 2.1 is globally convergent.

Theorem 3.1 *Suppose that the sequence $\{x_k\}$ generated by Algorithm 2.1 satisfies $x_k \in S$ for all k, where S is a closed and bounded convex set in R^n, and the potential function f is twice differentiable in S. Then $\{x_k\}$ is not bounded away from stationary point of f, that is*

$$\lim_{k\to\infty} \inf \|\nabla f(x_k)\| = 0, \tag{3.1}$$

or

$$\lim_{k\to\infty} \inf \|\nabla f(x_k + \frac{\sqrt{2}-1}{2} d_k)\| = 0, \tag{3.2}$$

where d_k is defined in (2.3).

Proof. We will prove this theorem by contradiction. If both (3.1) and (3.2) were not true there would exist a positive constant δ such that

$$\|g_k\| \geq \delta \tag{3.3}$$

and

$$\|\nabla f(x_k + \frac{\sqrt{2}-1}{2} d_k)\| \geq \delta \tag{3.4}$$

hold for all k, where $g_k = \nabla f(x_k)$. It is convenient to distinguish between the following two cases:

$$\text{(i)}\ \sup \lambda_k = \infty, \qquad \text{(ii)}\ \sup \lambda_k \leq W \ \text{ for some constant } W. \tag{3.5}$$

Case (i). From (2.7) there exists an infinite subsequence whose indices form a set S_1 such that $\rho_k < \eta_1$ and $\lim_{k\to\infty} \lambda_k = \infty$ for $k \in S_1$. Because g_k and G_k are bounded, from (2.3) we have

$$\lim_{k\to\infty} \|d_k\| = 0, \quad k \in S_1. \tag{3.6}$$

Thus, combining (2.4) we get

$$\begin{aligned} \|s_k\| &= \|(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1} \nabla f(x_k + \frac{\sqrt{2}-1}{2} d_k)\| \\ &\leq \|(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)^{-1}\| \cdot \|g_k + \frac{\sqrt{2}-1}{2} G_k d_k + o(\|d_k\|)\| \\ &\leq \frac{\|g_k\| + \frac{\sqrt{2}-1}{2}\|G_k\| \cdot \|d_k\| + o(\|d_k\|)}{\lambda_k - (1 - \frac{\sqrt{2}}{2})\|G_k\|} \to 0, \quad \text{as} \quad k \to \infty, \quad k \in S_1. \end{aligned} \tag{3.7}$$

From Lemma 2.2 and (2.5)-(2.6) we have $\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k \succ 0$ and

$$\begin{aligned}|\rho_k - 1| &= |\frac{f(x_k) - f(x_k+s_k)}{q_k(0) - q_k(s_k)} - 1| \\ &= |\frac{o(\|s_k\|^2)}{s_k^T g_k + \frac{1}{2}s_k^T G_k s_k}| \\ &\le \frac{o(\|s_k\|^2)}{\tau\|s_k\|\cdot\|g_k\|} \to 0, \quad \text{as} \quad k\to\infty, \quad k\in S_1,\end{aligned} \tag{3.8}$$

so that $\rho_k > \eta_1$ for sufficiently large k and $k \in S_1$, which contradicts the definition of S_1.

Case (ii). From (2.7) there exists an infinite subsequence whose indices form a set S_2 such that $\rho_k \ge \eta_1$ for $k \in S_2$. Thus, using the monotonically decreasing property of f_k and from (2.5)-(2.7), we have

$$\begin{aligned}\sum_{k\in S_2} \tau\eta_1\|g_k\|\min\{\|s_k\|, \|g_k\|/\|G_k\|\} &\le \sum_{k\in S_2}\eta_1(-s_k^T g_k - \frac{1}{2}s_k^T G_k s_k) \\ &\le \sum_{k\in S_2}(f_k - f_{k+1}) \le \sum_{k=1}^{\infty}(f_k - f_{k+1}) < \infty,\end{aligned} \tag{3.9}$$

which gives

$$\|s_k\| \to 0, \quad \text{as} \quad k\to\infty, \quad k\in S_2 \tag{3.10}$$

because of the assumption (3.3). Combining with (2.4) we get

$$\begin{aligned}\|\nabla f(x_k + \frac{\sqrt{2}-1}{2}d_k)\| &\le \|\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k\|\cdot\|s_k\| \\ &\le (W + (1-\frac{\sqrt{2}}{2})\|G_k\|)\|s_k\| \to 0, \quad \text{as} \quad k\to\infty, \quad k\in S_2,\end{aligned} \tag{3.11}$$

which contradicts the assumption (3.4). $\square$

From Theorem 3.1 we know that Algorithm 2.1 converges to an equilibrium point of the gradient system (1.1) for any initial point x_0. The time step $h_k = 1/\lambda_k$ is small and is governed by stability and accuracy properties of the integration scheme if the iteration point x_k is far away from the equilibrium point of the gradient system (1.1). Conversely, $h_k = 1/\lambda_k$ could be large when the iteration point x_k is near an equilibrium point. In the following we will give some interpretations on the local behavior of Algorithm 2.1. Firstly, we explore the properties of some special functions before we analyze the superlinear convergence of Algorithm 2.1.

Lemma 3.2 *Define*

$$\psi(\lambda,\mu) = \frac{\lambda + \frac{3-2\sqrt{2}}{2}\mu}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^2}, \quad \lambda\ge 0, \quad \mu > 0, \tag{3.12}$$

$$\phi(\lambda,\mu) = \frac{\lambda(\lambda + \frac{3-2\sqrt{2}}{2}\mu)}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^2}, \quad \lambda\ge 0, \quad \mu > 0, \tag{3.13}$$

and

$$\begin{aligned}\varphi(\lambda,\mu) &= (\lambda + \frac{\frac{3-2\sqrt{2}}{4}\mu^2}{\lambda + \frac{3-2\sqrt{2}}{2}\mu})\frac{(\lambda + \frac{3-2\sqrt{2}}{2}\mu)^2}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^4} \\ &= \frac{\lambda^3 + (3-2\sqrt{2})\mu\lambda^2 + \frac{10-7\sqrt{2}}{2}\mu^2\lambda + \frac{17-12\sqrt{2}}{8}\mu^3}{(\lambda + \frac{2-\sqrt{2}}{2}\mu)^4}, \quad \lambda\ge 0, \quad \mu > 0.\end{aligned} \tag{3.14}$$

Then the function ψ decreases monotonically in μ and function ϕ increases monotonically in λ if $\lambda \geq 0$ and $\mu > 0$. Furthermore the function φ decreases monotonically in λ if $0 \leq \lambda \leq \frac{3\sqrt{2}-4}{2}\mu$ or $\lambda \geq (\sqrt{2}-1)\mu > 0$ and increases monotonically in λ if $0 < \frac{3\sqrt{2}-4}{2}\mu \leq \lambda \leq (\sqrt{2}-1)\mu$. Moreover,

$$\varphi(\lambda, \mu) \geq \varphi(\frac{\sqrt{2}}{2}\mu, \mu) = \varphi(\frac{3\sqrt{2}-4}{2}\mu, \mu) = \frac{2\sqrt{2}+1}{8\mu}, \quad \textit{if} \quad 0 \leq \lambda \leq \frac{\sqrt{2}}{2}\mu. \tag{3.15}$$

Proof. From (3.12) we have

$$\begin{aligned}\psi_\mu(\lambda, \mu) &= \frac{\frac{3-2\sqrt{2}}{2}(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^2 - \frac{4-2\sqrt{2}}{2}(\lambda + \frac{3-2\sqrt{2}}{2}\mu)(\lambda + (1-\frac{\sqrt{2}}{2})\mu)}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^4} \\ &= \frac{-\frac{1}{2}(\lambda + \frac{10-7\sqrt{2}}{2}\mu)}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^3} \leq 0, \end{aligned} \tag{3.16}$$

if $\lambda \geq 0$ and $\mu > 0$. Thus the function ψ decreases monotonically in μ if $\lambda \geq 0$ and $\mu > 0$.

From (3.13) we get

$$\begin{aligned}\phi_\lambda(\lambda, \mu) &= \frac{(2\lambda + \frac{3-2\sqrt{2}}{2}\mu)(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^2 - 2\lambda(\lambda + \frac{3-2\sqrt{2}}{2}\mu)(\lambda + (1-\frac{\sqrt{2}}{2})\mu)}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^4} \\ &= \frac{\frac{1}{2}\mu\lambda + \frac{10-7\sqrt{2}}{4}\mu^2}{(\lambda + (1-\frac{\sqrt{2}}{2})\mu)^3} \geq 0, \quad \text{as} \quad \lambda > 0, \quad \mu > 0. \end{aligned} \tag{3.17}$$

Therefore the function ϕ increases monotonically in λ if $\lambda \geq 0$ and $\mu > 0$.

From (3.14) we have

$$\begin{aligned}\varphi_\lambda(\lambda, \mu) &= \frac{-\lambda(\lambda^2 + \frac{6-5\sqrt{2}}{2}\mu\lambda + \frac{10-7\sqrt{2}}{2}\mu^2)}{(\lambda + \frac{2-\sqrt{2}}{2}\mu)^5} \\ &= \frac{-\lambda(\lambda - \frac{3\sqrt{2}-4}{2}\mu)(\lambda - (\sqrt{2}-1)\mu)}{(\lambda + \frac{2-\sqrt{2}}{2}\mu)^5}, \end{aligned} \tag{3.18}$$

which gives

$$\varphi_\lambda(\lambda, \mu) \leq 0, \quad \text{if} \quad 0 \leq \lambda \leq \frac{3\sqrt{2}-4}{2}\mu \quad \text{or} \quad \lambda \geq (\sqrt{2}-1)\mu, \tag{3.19}$$

and

$$\varphi_\lambda(\lambda, \mu) \geq 0, \quad \text{if} \quad \frac{3\sqrt{2}-4}{2}\mu \leq \lambda \leq (\sqrt{2}-1)\mu. \tag{3.20}$$

Therefore the function φ decreases monotonically in λ if $0 \leq \lambda \leq \frac{3\sqrt{2}-4}{2}\mu$ or $\lambda \geq (\sqrt{2}-1)\mu$ and increases monotonically in λ if $\frac{3\sqrt{2}-4}{2}\mu \leq \lambda \leq (\sqrt{2}-1)\mu$. From (3.18) and using the monotonicity property of φ, we know that function φ has a local minimizer $\frac{2\sqrt{2}+1}{8\mu}$ at $\lambda = \frac{3\sqrt{2}-4}{2}\mu$ in λ. In addition, it is not difficult to verify that $\varphi(\frac{\sqrt{2}}{2}\mu, \mu) = \varphi(\frac{3\sqrt{2}-4}{2}\mu, \mu)$ and $\frac{\sqrt{2}}{2}\mu > (\sqrt{2}-1)\mu$. Thus we get (3.15). □

The following theorem shows that Algorithm 2.1 has good local behavior.

Theorem 3.3 *Assume that $\{x_k\}$, which is generated by Algorithm 2.1, converges to a local minimizer $x^\star$ of the function f and $\nabla^2 f(x^\star)$ is positive definite. Then there exist positive constants ε, m and M such that*

$$m\|z\|^2 \le z^T\nabla^2 f(x_k)z \le M\|z\|^2, \quad \forall z \in R^n, \quad x_k \in B(\varepsilon) = \{x \mid \|x - x^\star\| \le \varepsilon\}, \tag{3.21}$$

where the update matrix $G_k = \nabla^2 f(x_k)$ in Algorithm 2.1. Furthermore the parameter λ_k of Algorithm 2.1 satisfies

$$\lim_{k\to\infty} \lambda_k = 0, \tag{3.22}$$

and x_k converges superlinearly to $x^\star$.

Proof. Because $x_k \to x^\star$ and $\nabla f(x^\star) = 0$, we have $\lim_{k\to\infty}\|g_k\| = 0$, where $g_k = \nabla f(x_k)$. From (2.3) and (3.21) we get $\lim_{k\to\infty}\|d_k\| = 0$. Combining (2.3)-(2.4) we obtain

$$\begin{aligned}(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)s_k &= -\nabla f(x_k + \frac{\sqrt{2}-1}{2}d_k)\\ &= -g_k - \frac{\sqrt{2}-1}{2}G_k d_k + o(\|d_k\|)\\ &= -[I - \frac{\sqrt{2}-1}{2}G_k(\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k)^{-1}]g_k + o(\|d_k\|)\\ &= -(\lambda_k I + \frac{3-2\sqrt{2}}{2}G_k)(\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k)^{-1}g_k + o(\|d_k\|)\\ &= -(\lambda_k I + \frac{3-2\sqrt{2}}{2}G_k)(\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k)^{-1}g_k + o(\|s_k\|).\end{aligned} \tag{3.23}$$

In the above arguments, we have used the relationship $o(\|d_k\|) = o(\|s_k\|)$, which can be derived as follows. From (2.3)-(2.4) we have

$$\begin{aligned}(\lambda_k I + (1 - \frac{\sqrt{2}}{2})G_k)(d_k - s_k) &= \nabla f(x_k + \frac{\sqrt{2}-1}{2}d_k) - \nabla f(x_k)\\ &= \frac{\sqrt{2}-1}{2}G_k d_k + o(\|d_k\|),\end{aligned} \tag{3.24}$$

which gives

$$\begin{aligned}\|d_k - s_k\| &= \|(\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k)^{-1}(\frac{\sqrt{2}-1}{2}G_k d_k + o(\|d_k\|))\|\\ &\le \|\frac{\sqrt{2}-1}{2}(\lambda_k I + (1-\frac{\sqrt{2}}{2})G_k)^{-1}G_k\| \cdot \|d_k\| + o(\|d_k\|)\\ &\le \frac{\sqrt{2}}{2}\|d_k\| + o(\|d_k\|),\end{aligned}$$

so that

$$(1 - \frac{\sqrt{2}}{2})\|d_k\| + o(\|d_k\|) \le \|s_k\| \le (1 + \frac{\sqrt{2}}{2})\|d_k\| + o(\|d_k\|). \tag{3.25}$$

Therefore, we have $o(\|d_k\|) = o(\|s_k\|)$.

Because G_k is a symmetric matrix, there exists an orthogonal matrix Q_k such that

$$Q_k^T G_k Q_k = diag(\mu_1, \cdots, \mu_n), \tag{3.26}$$

where μ_i are the eigenvalues of G_k. We denote

$$\bar{s} = Q_k^T s_k, \quad \bar{g} = Q_k^T g_k. \tag{3.27}$$

Thus, from (3.23) and (3.25)-(3.27) we get

$$\bar{s}_i = \frac{\lambda_k + \frac{3-2\sqrt{2}}{2}\mu_i}{(\lambda_k + \frac{2-\sqrt{2}}{2}\mu_i)^2}\bar{g}_i + o(\|s_k\|), \quad i = 1, 2, \cdots, n. \tag{3.28}$$

From (2.2) and (3.26)-(3.28) we have

$$\begin{aligned} q_k(0) - q_k(s_k) &= -s_k^T g_k - \frac{1}{2} s_k^T G_k s_k \\ &= -(Q_k^T s_k)^T (Q_k^T g_k) - \frac{1}{2}(Q_k^T s_k)^T (Q_k^T G_k Q_k)(Q_k^T s_k) \\ &= \sum_{i=1}^{n} \frac{\lambda_k^2 + \frac{3-2\sqrt{2}}{2}\mu_i\lambda_k + \frac{3-2\sqrt{2}}{4}\mu_i^2}{\lambda_k + \frac{3-2\sqrt{2}}{2}\mu_i}\bar{s}_i^2 + o(\|s_k\|^2) \\ &= \sum_{i=1}^{n}(\lambda_k + \frac{\frac{3-2\sqrt{2}}{4}\mu_i^2}{\lambda_k + \frac{3-2\sqrt{2}}{2}\mu_i})\bar{s}_i^2 + o(\|s_k\|^2) \end{aligned} \tag{3.29}$$

by using orthogonal similarity transformation.

We will consider two distinct cases as we show that s_k satisfies (2.5) for a sufficiently large k, namely, (i) $0 \le \lambda_k < \frac{\sqrt{2}}{2}\mu_{max}$ and (ii) $\lambda_k \ge \frac{\sqrt{2}}{2}\mu_{max}$, where μ_{max} is the maximum eigenvalue of G_k. We assume that k is sufficiently large such that $x_k \in B(\varepsilon)$ holds in the following proof.

Case (i). From Lemma 3.2 we know that the function φ decreases monotonically in λ if $\lambda \ge \frac{\sqrt{2}}{2}\mu$, and $\varphi(\lambda, \mu) \ge \varphi(\frac{\sqrt{2}}{2}\mu, \mu) = \frac{2\sqrt{2}+1}{8\mu}$ if $0 \le \lambda \le \frac{\sqrt{2}}{2}\mu$. Because $0 \le \mu_i \le \mu_{max}$, we have

$$\varphi(\lambda_k, \mu_i) \ge \varphi(\frac{\sqrt{2}}{2}\mu_i, \mu_i) \ge \varphi(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i), \quad \text{if} \quad 0 \le \lambda_k \le \frac{\sqrt{2}}{2}\mu_i, \tag{3.30}$$

and

$$\varphi(\lambda_k, \mu_i) \ge \varphi(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i), \quad \text{if} \quad \frac{\sqrt{2}}{2}\mu_i \le \lambda_k \le \frac{\sqrt{2}}{2}\mu_{max}, \quad i = 1, 2, \cdots, n, \tag{3.31}$$

so that

$$\varphi(\lambda_k, \mu_i) \ge \varphi(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i), \quad \text{if} \quad 0 \le \lambda_k \le \frac{\sqrt{2}}{2}\mu_{max}, \quad i = 1, 2, \cdots, n. \tag{3.32}$$

Thus, from (3.28)-(3.29), (3.14), (3.32), and (3.12) we obtain

$$\begin{aligned} q_k(0) - q_k(s_k) &= \sum_{i=1}^{n} \varphi(\lambda_k, \mu_i)\bar{g}_i^2 + o(\|s_k\|^2) \\ &\ge \sum_{i=1}^{n} \varphi(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i)\bar{g}_i^2 + o(\|s_k\|^2) \\ &= \sum_{i=1}^{n}(\frac{\sqrt{2}}{2}\mu_{max} + \frac{\frac{3-2\sqrt{2}}{4}\mu_i^2}{\frac{\sqrt{2}}{2}\mu_{max} + \frac{3-\sqrt{2}}{2}\mu_i})\psi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i)\bar{g}_i^2 + o(\|s_k\|^2) \\ &\ge \frac{\sqrt{2}}{2}\mu_{max} \sum_{i=1}^{n} \psi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i)\bar{g}_i^2 + o(\|s_k\|^2), \quad \text{if} \quad 0 \le \lambda_k \le \frac{\sqrt{2}}{2}\mu_{max}. \end{aligned} \tag{3.33}$$

From Lemma 3.2 we know that function ψ decreases monotonically in μ if $\mu > 0$, $\lambda \geq 0$. Combining this result with (3.33), we have

$$\begin{aligned} q_k(0) - q_k(s_k) &\geq \frac{\sqrt{2}}{2}\mu_{max} \sum_{i=1}^{n} \psi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_{max})\bar{g}_i^2 + o(\|s_k\|^2) \\ &= \frac{\sqrt{2}}{2}\mu_{max}\psi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_{max})\|g_k\|^2 + o(\|s_k\|^2) \\ &= \frac{11\sqrt{2}-12}{8} \frac{\|g_k\|^2}{\|G_k\|} + o(\|g_k\|^2), \quad \text{if} \quad 0 \leq \lambda_k \leq \frac{\sqrt{2}}{2}\mu_{max}. \end{aligned} \tag{3.34}$$

In the above arguments, we have used the fact that $o(\|g_k\|) = o(\|s_k\|)$, which is derived from (3.23) and $0 \leq \lambda_k \leq \frac{\sqrt{2}}{2}\mu_{max}$. Therefore s_k satisfies (2.5) for a sufficiently large k if $0 \leq \lambda_k \leq \frac{\sqrt{2}}{2}\mu_{max}$.

Case (ii). From (3.28)-(3.29) and (3.13), we have

$$\begin{aligned} q_k(0) - q_k(s_k) &= \sum_{i=1}^{n}(\lambda_k + \frac{\frac{3-2\sqrt{2}}{4}\mu_i^2}{\lambda_k + \frac{3-2\sqrt{2}}{2}\mu_i})\bar{s}_i^2 + o(\|s_k\|^2) \\ &\geq \lambda_k\|\bar{s}\|^2 + o(\|s_k\|^2) \\ &= (\sum_{i=1}^{n} \lambda_k^2\bar{s}_i^2)^{\frac{1}{2}} \ \|s_k\| + o(\|s_k\|^2) \\ &= (\sum_{i=1}^{n} \lambda_k^2(\frac{\lambda_k + \frac{3-2\sqrt{2}}{2}\mu_i}{(\lambda_k + \frac{2-\sqrt{2}}{2}\mu_i)^2}\bar{g}_i + o(\|s_k\|))^2)^{\frac{1}{2}} \ \|s_k\| + o(\|s_k\|^2) \\ &= (\sum_{i=1}^{n} \phi^2(\lambda_k, \mu_i)\bar{g}_i^2)^{\frac{1}{2}} \ \|s_k\| + o(\|s_k\|^2). \end{aligned} \tag{3.35}$$

From Lemma 3.2 we know that the function ϕ increases monotonically in λ and the function ψ decreases monotonically in μ if $\lambda \geq 0$, $\mu > 0$. Combining these monotonicity properties of ϕ and ψ, from (3.35) and (3.12)-(3.13), we obtain

$$\begin{aligned} q_k(0) - q_k(s_k) &\geq (\sum_{i=1}^{n} \phi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i)\bar{g}_i^2)^{\frac{1}{2}} \ \|s_k\| + o(\|s_k\|^2) \\ &= (\sum_{i=1}^{n} \frac{1}{2}\mu_{max}^2\psi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_i)\bar{g}_i^2)^{\frac{1}{2}} \ \|s_k\| + o(\|s_k\|^2) \\ &\geq (\sum_{i=1}^{n} \frac{1}{2}\mu_{max}^2\psi^2(\frac{\sqrt{2}}{2}\mu_{max}, \mu_{max})\bar{g}_i^2)^{\frac{1}{2}} \ \|s_k\| + o(\|s_k\|^2) \\ &= \frac{\sqrt{2}}{2}\mu_{max}\psi(\frac{\sqrt{2}}{2}\mu_{max}, \mu_{max})\|g_k\| \ \|s_k\| + o(\|s_k\|^2) \\ &= \frac{3\sqrt{2}-2}{4}\|g_k\| \ \|s_k\| + o(\|s_k\|^2), \quad \text{if} \quad \lambda_k \geq \frac{\sqrt{2}}{2}\mu_{max}. \end{aligned} \tag{3.36}$$

We have $o(\|s_k\|^2) \leq o(\|g_k\| \ \|s_k\|)$ from (3.23). Applying this result to (3.36), we obtain that s_k also satisfies (2.5) for a sufficiently large k as $\lambda_k \geq \frac{\sqrt{2}}{2}\mu_{max}$.

From (3.29) and using the inequality $a^2+b^2 \geq 2ab$, we get

$$
\begin{aligned}
q_k(0)-q_k(s_k) &= \sum_{i=1}^{n}(\lambda_k + \frac{3-2\sqrt{2}}{2}\mu_i + \frac{\frac{3-2\sqrt{2}}{4}\mu_i^2}{\lambda_k+\frac{3-2\sqrt{2}}{2}\mu_i} - \frac{3-2\sqrt{2}}{2}\mu_i)\bar{s}_i^2 + o(\|s_k\|^2) \\
&\geq \sum_{i=1}^{n}\frac{4\sqrt{2}-5}{2}\mu_i\bar{s}_i^2 + o(\|s_k\|^2) \\
&= \frac{4\sqrt{2}-5}{2}s_k^TG_ks_k + o(\|s_k\|^2) \\
&\geq \frac{4\sqrt{2}-5}{2}m\|s_k\|^2 + o(\|s_k\|^2), \qquad (3.37)
\end{aligned}
$$

for a sufficiently large k. Using (3.34) and (3.36), we also know that s_k satisfies (2.5) for a sufficiently large k. Thus, from (3.37) we obtain

$$
\begin{aligned}
|\rho_k - 1| &= |\frac{f(x_k)-f(x_k+s_k)}{q_k(0)-q_k(s_k)} - 1| \\
&= |\frac{-s_k^Tg_k - \frac{1}{2}s_k^TG_ks_k + o(\|s_k\|^2)}{-s_k^Tg_k - \frac{1}{2}s_k^TG_ks_k} - 1| \\
&\leq \frac{o(\|s_k\|^2)}{\frac{4\sqrt{2}-5}{2}m\|s_k\|^2 + o(\|s_k\|^2)} \to 0, \quad \text{as} \quad k \to \infty, \qquad (3.38)
\end{aligned}
$$

which gives

$$\rho_k \geq \eta_2, \qquad (3.39)$$

for a sufficiently large k. From (2.7) and (3.39), we get (3.22).

Using (3.23) and (3.22), we have

$$
\begin{aligned}
\frac{\|s_k - s_k^N\|}{\|s_k\|} &= \frac{\|\lambda_k(\lambda_kI + \frac{2-\sqrt{2}}{2}G_k)^{-1}(\lambda_kG_k^{-1} - (\sqrt{2}-1)I)(\lambda_kI + \frac{2-\sqrt{2}}{2}G_k)^{-1}g_k + o(\|s_k\|)\|}{\|(\lambda_kI + \frac{2-\sqrt{2}}{2}G_k)^{-1}(\lambda_kI + \frac{3-2\sqrt{2}}{2}G_k)(\lambda_kI + \frac{2-\sqrt{2}}{2}G_k)^{-1}g_k + o(\|s_k\|)\|} \\
&\leq \frac{\lambda_k(\lambda_k + \frac{2-\sqrt{2}}{2}M)^2(\frac{\lambda_k}{m} + (\sqrt{2}-1)) + o(\|s_k\|)}{(\lambda_k + \frac{2-\sqrt{2}}{2}m)^2(\lambda_k + \frac{3-2\sqrt{2}}{2}m) + o(\|s_k\|)} \\
&\to 0, \quad \text{as} \quad k \to \infty, \qquad (3.40)
\end{aligned}
$$

where $s_k^N = -G_k^{-1}g_k$ is the Newton step. Thus the sequence $\{x_k\}$ satisfies the well-known characterization result of Dennis and Moré ([12, 13]). Hence $\{x_k\}$ converges superlinearly to $x^\star$. □

4. Numerical experiments

In this section, we will conduct some numerical experiments for Algorithm 2.1. The test problems are the standard 18 unconstrained optimization problems in Moré, Garbow and Hillstrom [36].

Because we are mainly concerned with the equilibrium point of the gradient system (1.1), we compare Algorithm 2.1 with other numerical methods for unconstrained optimization problems. In order to show the advantages of the time step control of Algorithm 2.1, we compare Algorithm 2.1 with the second order Rosenbrock methods using the conventional local error bound for time step control (**ode232s** in Matlab). We also compare Algorithm 2.1 to two variants of pseudo-transient continuation (Ψtc). In the first variant Ψtc controls the time step via

$$\lambda_k = \lambda_{k-1}\|g_k\|/\|g_{k-1}\|. \qquad (4.1)$$

This approach is used to compute steady-state solutions of time-dependent partial differential equations (see [9, 11, 31, 29, 16]) and the solutions of nonlinear equations (see [10]). Since Ψtc generates the pseudo-trajectory for the gradient system and has good theoretical properties (see [29]), we report the numerical results of Ψtc for the above problems as well. The second variant of Ψtc is the trust-region approach from [25]. In this approach, the time step is controlled with a trust-region paradigm, but not the same one as Algorithm 2.1. One can think of this second variant as a Levenberg-Marquardt method as well.

There are four methods in our comparison:

a) Conventional Rosenbrock Method: This is just **ode23s** in Matlab. For detailed description of the method, see [46, 35].

b) Trust Region Rosenbrock Method: This is our Algorithm 2.1.

c) Ψtc: This is the method developed in [29] and applied to optimization problems in [30]. In this approach the time step is controlled by (4.1). We considered other methods for control of the time step [29], but these methods did not perform as well as the one we used, and are not supported by any convergence theory.

d) Ψtc-TR: This is a conventional trust region approach which uses ideas from pseudo-transient continuation. The global and local convergence are addressed in [25]. This approach uses (2.8) instead of (2.3) and (2.4) in Step 2 of Algorithm 2.1.

Our tests were conducted in MATLAB 6.5 with machine precision 10^{-16}. Algorithm 2.1 was implemented with parameters $\eta_1 = 0.25$, $\eta_2 = 0.75$, $\gamma_1 = 0.5$, and $\gamma_2 = 2$. We simply adopt initial time step $h = 1/\lambda_0$, where

$$\lambda_0 = \min\{\|\nabla f(x_0)\|, 10\} \tag{4.2}$$

as the default choice for Algorithm 2.1 and the Ψtc method. The approximate Hessian matrices $\nabla^2 f(x_k)$ are obtained by the numerical difference method.

All test examples start at the standard initial points given in [36]. All methods terminate when the iteration point x_k satisfies

$$\|g_k\| \leq 10^{-7}. \tag{4.3}$$

The numerical results are reported in Table 1. Iter (f-g-G) stands for the number of iterations, the number of function evaluations, gradient evaluations, and Hessian evaluations, respectively. These numbers include the equivalent function and gradient evaluations when the Hessian matrices were computed by the numerical difference method.

From Table 1, we can see that the conventional Rosenbrock method (namely ode23s.m) requires more number of iterations and gradient evaluations than the other methods. Algorithm 2.1 performs better than the Levenberg-Marquardt method and the conventional Rosenbrock method for most test problems, and is as good as the pseudo-transient continuation method. However, Ψtc may fail to converge to the equilibrium point of the gradient system if we do not choose a proper initial time step. Moreover, Algorithm 2.1 is second order while Ψtc is only first order. Thus Algorithm 2.1 has higher accuracy than Ψtc in the transient state for any gradient system.

In order to compare the robustness of ode23s, Algorithm 2.1, Ψtc method and Levenberg-Marquardt method we analyze the solutions of some test problems further.

Problem 4 is the Powell badly scaled function with a global minimum $f = 0$ at the point $[1.09815933 \times 10^{-5}; 9.106146738]$. The conventional Rosenbrock method (ode23s.m), our trust region Rosenbrock method and the Levenberg-Marquardt method all failed. However, the Ψtc managed to stop at $[-0.0001000300; -0.0001000291]$ which is still far away from the optimal solution.

Problem 12 is the Gulf Research and Development function with a global minimum $f = 0$ at the point $[50, 25, 1.5]$ and the local minimum $f = 0.038$ at the point $x = [99.89537834; 60.61453903; 9.16124389]$ or $x = [201.66258949; 60.616331505; 10.22489116]$. The numerical solutions of ode23s, Ψtc, and the Levenberg-Marquardt method are $[21.10943580; 28.70651498; 1.29687116]$, $[20.35757478; 28.86321556; 1.28813195]$, and $[5.267893580; 2.53685592; -5.06048565]$, respectively, which are far away from the global minimum or local minimum. But the numerical solution of Algorithm 2.1 is $[49.94376437; 25.00476559; 1.49973934]$ which is close to the global optimal solution.

Problem 13 is the Trigonometric function which is known to have a global minimum $f = 0$ at the point $[0.042965; 0.043976; 0.045093; 0.046339; 0.047744; 0.049355; 0.051237; 0.195209; 0.164978; 0.060149]$ and a local minimum $f = 2.79506e{-5}$ at the point $[0.055151; 0.056841; 0.058764; 0.060991; 0.063626; 0.066843; 0.208162; 0.164363; 0.085007; 0.091431]$. The numerical solutions of ode23s, Algorithm 2.1, and Levenberg-Marquardt method are $[0.055151; 0.056841; 0.058764; 0.060991; 0.063626; 0.066843; 0.208162; 0.164363; 0.085007; 0.091431]$, $[0.055151; 0.056841; 0.058764; 0.060991; 0.063626; 0.066843; 0.208161; 0.164363; 0.085008; 0.091432]$, and $[0.055151; 0.056841; 0.058764; 0.060991; 0.063626; 0.066843; 0.208162; 0.164363; 0.085007; 0.091432]$, respectively, which are close to the local solution. The numerical solution of Ψtc is $[0.062492; 0.064473; 0.066706; 0.069245; 0.275398; 0.075446; 0.079030; 0.082273; 0.083639; 0.082089]$ which is far away from the global solution or local solution.

It is worth noting that we only factor the matrix once and perform two back-substitutions in Algorithm 2.1 as equations (2.3) and (2.4) have the same coefficient matrix. Because Algorithm 2.1 does not require more iterations and Hessian matrix evaluations than Ψtc or the Levenberg-Marquardt method for most test problems, Algorithm 2.1 can be used as a workhorse to solve the gradient system if we are mainly concerned with the equilibrium state.

Table 1 – Numerical results of the 18 problems in [36]

		Conventional Rosenbrock Method	Trust Region Rosenbrock Method	Ψtc	Ψtc-TR
	n	Iter (f-g-G)	Iter (f-g-G)	Iter (f-g-G)	Iter (f-g-G)
1	3	90 (0-543-90)	16 (17-78-15)	15 (0-61-15)	18 (19-70-17)
2	6	110 (0-971-107)	19 (20-153-19)	28 (0-197-28)	25 (26-170-24)
3	3	6 (0-39-6)	3 (3-15-3)	3 (0-13-3)	2 (3-9-2)
4	2	(Failed)	> 700 (Failed)	34 (0-103-34) (False sol.)	> 700 (Failed)
5	3	80 (0-483-80)	23 (24-116-23)	40 (0-161-40)	29 (30-114-28)
6	10	37 (0-484-37)	10 (10-120-10)	13 (0-144-13)	14 (15-155-14)
7	12	> 700	25 (26-351-25)	12 (0-157-12)	25(26-326-25)
8	10	80 (0-987-75)	28 (28-336-28)	21 (0-232-21)	42 (43-423-38)
9	4	147 (0-1011-147)	90 (91-481-75)	18 (0-91-18)	140 (141-609-117)
10	2	195 (0-971-195)	55 (55-198-44)	> 700	347 (348-1038-345)
11	4	58 (0-408-58)	7 (8-43-7)	26 (0-131-26)	9 (10-46-9)
12	3	56 (0-338-56) (Far away sol.)	121 (122-546-101) (Close to global sol.)	40 (0-161-40) (Far away sol.)	1 (2-5-1) (Far way sol.)
13	10	32 (0-418-32) (Close to local sol.)	13 (13-146-12) (Close to local sol.)	10 (0-111-10) (Far away sol.)	12 (13-123-11) (Close to local sol.)
14	50	153 (0-767-153)	16 (17-833-16)	26 (0-1327-26)	27 (28-1228-24)
15	64	98 (0-6568-98)	19 (20-1255-19)	27 (0-1756-27)	22 (23-1431-22)
16	2	48 (0-239-47)	13 (14-53-13)	11 (0-34-11)	17 (18-50-16)
17	4	289 (0-1946-273)	51 (52-275-43)	18 (0-91-18)	56 (57-245-47)
18	8	33 (0-356-32)	16 (17-145-14)	11 (0-100-11)	16 (17-129-14)

5. Final remarks

We have demonstrated that combining certain optimization techniques (e.g. line search methods or trust region methods) with traditional numerical ODE methods have advantages in calculating the equilibrium points of gradient systems or unconstrained optimization problems. One advantage is that these methods have good global and local convergence characteristics in comparison to the traditional error control. Secondly, these methods have explicit physical solutions and are more robust, compared with conventional unconstrained optimization methods (e.g. line search methods or trust region methods). Thus such a combination is a promising area for future research. Furthermore we believe that this class of techniques of controlling the time step can be applied to the field of integro-differential equations if we are mainly interested in the equilibrium point. For recent results and numerical methods of integro-differential equations, one can refer to the book of Brunner (2005) [5].

References

[1] W.-J. Beyn and J. Schropp (2000), *Runge-Kutta discretizations of singularly perturbed gradient equations*, BIT, Vol. 40, pp. 415-433.

[2] C. A. Botsaris (1978), *Differential gradient methods*, Journal of Mathematical Analysis and Applications, Vol. 63, pp. 177-198.

[3] C. A. Botsaris (1978), *A class of methods for unconstrained minimization based on stable numerical integration techniques*, Journal of Mathematical Analysis and Applications, Vol. 63, pp. 729-749.

[4] A. A. Brown and M. C. Bartholomew-Biggs (1989), *Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations*, Journal of Optimization and Theory Applications, Vol. 62, pp. 211-224.

[5] H. Brunner (2005), *Collocation Methods for Volterra Integral and Related Functional Differential Equations*, Cambridge University Press, Cambridge.

[6] Yu. G. Evtushenko (1985), *Numerical Optimization Techniques*, Optimization Software Inc., Publications division, New York.

[7] A. R. Conn, N. Gould and Ph. L. Toint (2000), *Trust-Region Methods*, SIAM, Philadelphia, USA.

[8] S. D. Cohen and A. C. Hindmarsh (1994), *CVODE User Guide*, Lawrence Livermore National Laboratory technical report UCRL-MA-118618.

[9] T. S. Coffey, R. J. McMullan, C. T. Kelley, and D. S. McRae (2003), *Globally convergent algorithms for nonsmooth nonlinear equations in computational fluid dynamics* J. Comp. Appl. Math., Vol.152, pp. 69-81.

[10] T. S. Coffey, C. T. Kelley and D. E. Keyes (2003), *Pseudotransient continuation and differential-algebraic equations*, SIAM Journal on Scientific Computing, Vol. 25, pp. 553-569.

[11] P. Deuflhard (2002), *Adaptive Pseudo-transient Continuation for Nonlinear Steady State Problems*, Konrad-Zuse-Zentrum Berlin. ZIB-Report 02-14.

[12] J. E. Dennis and J. J. Moré (1974), *A characterization of superlinear convergence and its application to quasi-Newton methods*, Mathematics of Computation, Vol. 28, pp. 549-560.

[13] J. E. Dennis and J.J. Moré (1977), *Quasi-Newton methods, motivation and theory*, SIAM Review, Vol. 19, pp. 46-89.

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

[15] R. Fletcher (1987), *Practical Methods of Optimization*, John Wiley and Sons, New York.

[16] K. R. Fowler and C. T. Kelley, *Pseudo-transient continuation for nonsmooth nonlinear equations*, SIAM J. Numer. Anal., 43 (2005), pp. 1385–1406.

[17] D. M. Gay (1981), *Computing optimal locally constrained steps*, SIAM Journal on Scientific and Statistical Computing, Vol. 2, pp. 186-197.

[18] S. M. Goldfeld, R. E. Quandt, and H. F. Trotter (1966), *Maximisation by quadratic hill-climbing*, Econometrica, Vol. 34, pp. 541-551.

[19] O. Gonzalez, D. J. Higham and A. M. Stuart (1999), *Qualitative properties of modified equations*, IMA Journal of Numerical Analysis, Vol. 19, pp. 169-190.

[20] E. Hairer, S. P. Nørsett and G. Wanner (1993), *Solving Ordinary Differential Equations I, Nonstiff Problems*, 2nd ed., Springer-Verlag.

[21] E. Hairer and G. Wanner (1996), *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin.

[22] Q. M. Han, L. -Z. Liao, H. D. Qi, and L. Q. Qi (2001), *Stability analysis of gradient-based neural networks for optimization problems*, Journal of Global Optimization, Vol. 19, pp. 363-381.

[23] U. Helmke and J. B. Moore (1994), *Optimization and Dynamical Systems*, Springer-Verlag, Berlin.

[24] B. S. He (2000), *Solving trust region problem in large scale optimization*, Journal of Computational Mathematics, Vol. 18, pp. 1-12.

[25] D. J. Higham (1999), *Trust region algorithms and timestep selection*, SIAM J. Numer. Anal., Vol. 37, pp. 194-210.

[26] D. J. Higham and A. M. Stuart (1998), *Analysis of the dynamics of local error control via a piecewise continuous residual*, BIT, Vol. 38, pp. 44-57.

[27] A. R. Humphries and A. M. Stuart (1994), *Runge-Kutta methods for dissipative and gradient dynamical systems*, SIAM J. Numer. Anal., Vol. 31, pp. 1452-1485.

[28] C. T. Kelley (1999), *Iterative Methods for Optimization*, SIAM, Philadelphia, USA.

[29] C. T. Kelley and D. E. Keyes (1998), *Convergence analysis of pseudo-transient continuation*, SIAM J. Numer. Anal., Vol. 35, pp. 508-523.

[30] C. T. Kelley, L. Qi, L.-Z. Liao, J. P. Reese, and C. Winton, *Pseudo-transient continuation and optimization*, o Submitted for publication, 2006.

[31] D. A. Knoll and P. R. Mchugh (1998), *Enhanced nonlinear iterative techniques applied to a nonequillibrium plasma flow*, SIAM J. Sci. Comput., Vol. 19, pp. 291-301.

[32] K. Levenberg (1944), *A method for the solution of certain problems in least squares*, Quart. Appl. Math., Vol. 2, pp.164-168.

[33] X. L. Luo (2001), *Unconstrained Optimization Methods Based on Dynamic Systems* (in Chinese), Ph. D. Thesis, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences.

[34] D. Marquardt (1963), *An algorithm for least-squares estimation of nonlinear parameters*, SIAM J. Appl. Math., Vol. 11, pp.431-441.

[35] MATLAB 6.5 (2003), The MathWorks Inc., http://www. mathworks. com.

[36] J. J. Moré, B. S. Garbow and K. E. Hillstrom (1981), *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, Vol. 7, pp. 17-41.

[37] J. J. Moré and D. C. Sorensen (1983), *Computing a trust region step*, SIAM Journal on Scientific and Statistical Computing, Vol. 4, pp. 553-572.

[38] J. Nocedal and S. J. Wright (1999), *Numerical Optimization*, Springer.

[39] J. Nocedal and Y. Yuan (1998), *Combining trust region and line search techniques*, in: Y. Yuan, ed. Advances in Nonlinear Programming, Kluwer, pp. 153–175.

[40] M. J. D. Powell (1975), *Convergence properties of a class of minimization algorithms*, in: O.L. Mangasarian, R. R. Meyer and S. M. Robinson eds., *Nonlinear Programming 2*, Academic Press, New York, pp. 1-27.

[41] M. J. D. Powell (1982), ED., *Nonlinear Optimization 1981: Proceedings of the NATO Advanced Research Institute*, Academic Press, New York.

[42] H. H. Rosenbrock (1963), *Some general implicit processes for the numerical solution of differential equations*, The Computer Journal, Vol. 5, pp. 329-331.

[43] J. Schropp (1995), *Using dynamical systems methods to solving minimization problems*, Appl. Numer. Math., Vol. 18, pp. 321-335.

[44] J. Schropp (1997). *A note on minimization problems and multistep methods*, Numer. Math., Vol. 78, pp. 87-101.

[45] L. F. Shampine (1982), *Implementation of Rosenbrock methods*, ACM Transaction on Mathematical Software, Vol. 8, pp. 93-113.

[46] L. F. Shampine and M. W. Reichelt (1997), *The matlab ODE suite*, SIAM Journal on Scientific Computing, Vol. 18, pp. 1-22.

[47] S. H. Strogatz (1994), *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*, Addison-Wesley Publishing Company.

[48] A. M. Stuart (1994), *Numerical analysis of dynamical systems*, Acta Numerica, Vol. 3, pp. 467-572.

[49] A. M. Stuart and A. R. Humphries (1994), *Model problems in numerical stability theory for initial value problems*, SIAM Review, Vol. 36, pp. 226-257.

[50] A. M. Stuart and A. R. Humphries (1995), *The essential stability of local error control for dynamical systems*, SIAM J. Numer. Anal., Vol. 32, pp. 1940-1971.

[51] Y. Yuan and W. Sun (1997), *Optimization Theories and Methods* (in Chinese), Science Press, Beijing.

[52] Z. D. Yuan, J. G. Fei and D. G. Liu (1987), *The Numerical Solutions for Initial Value Problems of Stiff Ordinary Differential Equations* (in Chinese), Science Press, Beijing.