Journal of Computational Mathematics, Vol.23, No.2, 2005, 153–164.

SINGLY DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS COMBINING LINE SEARCH TECHNIQUES FOR UNCONSTRAINED OPTIMIZATION *1)

Xin-long Luo
(*School of Information Engineering, Beijing University of Posts and Telecommunications, P.O. Box 101,Beijing 100876, China*)

Abstract

There exists a strong connection between numerical methods for the integration of ordinary differential equations and optimization problems. In this paper, we try to discover further their links. And we transform unconstrained problems to the equivalent ordinary differential equations and construct the LRKOPT method to solve them by combining the second order singly diagonally implicit Runge-Kutta formulas and line search techniques. Moreover we analyze the global convergence and the local convergence of the LRKOPT method. Promising numerical results are also reported.

Mathematics subject classification: 65K05, 90C30, 65L06.
Key words: Global convergence, Superlinear convergence, Runge-Kutta method, Unconstrained optimization.

1. Introduction

In this paper, we mainly consider numerical methods for the following unconstrained optimization problem

$$\min_{x \in R^n} f(x), \tag{1.1}$$

where f is a continuously differentiable function. The main idea of solving the unconstrained optimization problem (1.1) is that we search for the next iteration point

$$x_{k+1} = x_k + \alpha_k d_k$$

via choosing the descent direction d_k and the step length α_k based on the current iteration point x_k such that $f(x_{k+1})$ satisfies some descent criteria, such as the Armijo line search criterion [13, 27, 35].

It has been extensively studied for choosing the descent direction d_k based on the Newton direction (see [1, 2, 4, 6, 9]), the conjugate gradient direction (see [10, 14, 15]) and the negative gradient direction (see [3, 5, 6, 16, 18, 30]) last decades, where $\nabla f(x_k)$ and $\nabla^2 f(x_k)$ are the gradient and the Hessian matrix of the function f at the current point x_k, respectively. But there are few researches for other descent directions. In the next section, we will consider search directions other than the negative gradient direction or the Newton direction. And we construct the LRKOPT method that has the superlinear convergence and global convergence by discretizing the following initial value problem of ordinary differential equations

$$\frac{dx}{dt} = -\nabla f(x), \tag{1.2}$$

$$x(0) = x_0, \tag{1.3}$$

* Received June 10, 2003, final revised June 23, 2004.
1) This work is partially supported by Chinese National Science Foundation (No.19731010).

where x_0 is an any given initial value. It is well-known that the solution $x(t)$ of differential equations (1.2)-(1.3) converges to the stable point $x^\star$ of the function f as t tends to infinity, namely $\lim_{t\to\infty} \|\nabla f(x(t))\| = 0$ (see [18, 22]).

There are some discussions on numerical methods for solving (1.2)-(1.3) in [30], which point out the importance of studying this class of numerical methods. Schropp [32] applied linear multistep methods to the gradient system (1.2)-(1.3) and studied the qualitative properties of discrete solutions of (1.2)-(1.3). In [23], we give a reasonable explanation that Backward Difference Formulas (BDFs) which are popular methods for solving stiff ordinary differential equations are low efficient for the gradient system (1.2)-(1.3) on the view of unconstrained optimization and we will restate the explanation in Section 2. Thus we mainly consider Runge-Kutta methods for solving the gradient system (1.2)-(1.3).

This paper is organized as follows. In the next section we consider the second order Singly Diagonally Implicit Runge-Kutta methods (SDIRK) for solving the gradient system (1.2)-(1.3) and construct the LRKOPT method with the superlinear convergence for the unconstrained optimization problem (1.1). In Section 3 we analyze the global convergence and the local convergence of the LRKOPT method. Finally, we report some numerical results of the LRKOPT method and the IMPBOT method which is given by Brown and Bartholomew-Biggs (see [6]) in Section 4. Throughout the paper $\|\cdot\|$ denotes the Euclidean vector norm or its induced norm.

2. The LRKOPT Method

We know that the class of methods for solving the gradient system (1.2)-(1.3) need satisfy the L stability via studying the linear test ordinary differential equation if those methods have the good local behavior (see [23]). Because linear multistep methods except for the backward Euler method do not satisfy the L stability, we focus on Runge-Kutta methods for solving the gradient system (1.2)-(1.3).

Runge-Kutta methods for solving the gradient system (1.2)-(1.3) have the following general form

$$K_i = h \cdot g(x_k + \sum_{j=1}^{s} a_{ij} K_j),\ i = 1, 2, \cdots, s, \tag{2.1}$$

$$x_{k+1} = x_k + \sum_{i=1}^{s} b_i K_i, \tag{2.2}$$

where $g(x) = -\nabla f(x)$, $h > 0$ is the time step, a_{ij} and b_i are constants. It is favorable for stiff ordinary differential equations if the numerical method has the A stability. Because the highly nonlinear problem (1.1) can introduce stiff ordinary differential equations (1.2)-(1.3). Thus we consider implicit Runge-Kutta methods for solving the gradient system (1.2)-(1.3).

Before introducing the particular scheme for solving the gradient system (1.2)-(1.3) we give some short descriptions of A-stable, L-stable and B-stable. A numerical method is called A-stable if, for the linear test equation $dx/dt = \mu x$ with $Re(\mu) \le 0$ and for all time steps $h \ge 0$, the stability function $R(z) = 1 + zb^T(I - zA)^{-1}e$ satisfies $|R(z)| \le 1$, where $z = \mu h$, the elements of the matrix A are $a_{ij}(i, j = 1, 2, \cdots, s)$, the vector b equals to $[b_1, b_2, \cdots, b_s]^T$ and all elements of e are one (see [17, 34]). The step length h does not have the stable restriction if the numerical method is A-stable. Furthermore the numerical method is A-stable and satisfies $\lim_{z\to-\infty} R(z) = 0$ then it is called L-stable (see [17, 34]).

Let two sequences $\{x_k\}$ and $\{z_k\}$ of approximation computed by a Runge-Kutta method for the same following autonomous differential equations

$$\frac{dx}{dt} = g(x), \quad g : R^n \to R^n. \tag{2.3}$$

The method is called B-stable if the contractive condition

$$\langle g(x) - g(z), x - z \rangle \leq 0 \tag{2.4}$$

implies $\|x_k - z_k\| \leq \|x_{k-1} - z_{k-1}\|$ for all $h \geq 0$ (see [7, 17]). Clearly, the B-stability is the expansion of the A-stability. We define the matrix

$$M = BA + A^T B - bb^T, \tag{2.5}$$

where the vector b and the matrix A are defined by (2.1)-(2.2) and the diagonal matrix B equals to $diag(b_1, \cdots, b_s)$. We only state the following sufficient condition for the B stability of Runge-Kutta methods and its proof can be found in [8].

Theorem 2.1. *(Burrage and Butcher (1979) [8]) Runge-Kutta methods are B-stable if (2.1)-(2.2) are such that the matrices B and M are positive semi-definite.*

It is not difficult to verify that explicit Runge-Kutta methods are not A-stable. Furthermore we only need coarse approximation solutions x_k in the middle points as we solve the gradient system (1.2)-(1.3) by employing Runge-Kutta methods. Therefore we consider the low order implicit Runge-Kutta methods for solving the gradient system (1.2)-(1.3). In particular, we consider the following second order singly diagonally implicit Runge-Kutta methods at Table 1.

r	r	0
$1-r$	$1-2r$	r
	$\frac{1}{2}$	$\frac{1}{2}$

Table 1: The second order diagonally implicit Runge-Kutta Methods.

The iterative formulas solving ordinary differential equations (1.2)-(1.3) which use the coefficients of Table 1 are

$$K_1 = h \cdot g(x_k + rK_1), \tag{2.6}$$

$$K_2 = h \cdot g(x_k + (1-2r)K_1 + rK_2), \tag{2.7}$$

$$x_{k+1} = x_k + \frac{1}{2}(K_1 + K_2), \tag{2.8}$$

where $g(x) = -\nabla f(x)$ and $h > 0$ is the time step. We can verify that the formulas (2.6)-(2.8) are the third order if the coefficient $r = \frac{3+\sqrt{3}}{6}$ or $r = \frac{3-\sqrt{3}}{6}$ and other cases are the second order by Taylor series expansion.

We study the B stability of singly diagonally implicit Runge-Kutta methods (2.6)-(2.8) in order to determine the parameter r in (2.6)-(2.8). As $r \geq \frac{1}{4}$, we have

$$\begin{aligned} M &= BA + A^T B - bb^T \\ &= \begin{bmatrix} \frac{1}{2}r & 0 \\ \frac{1}{2} - r & \frac{1}{2}r \end{bmatrix} + \begin{bmatrix} \frac{1}{2}r & \frac{1}{2} - r \\ 0 & \frac{1}{2}r \end{bmatrix} - \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix} \\ &= (r - \frac{1}{4}) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\ &\geq 0, \end{aligned} \tag{2.9}$$

where $B = diag(\frac{1}{2}, \frac{1}{2})$ is a diagonal matrix, which is composed of $b = [\frac{1}{2}, \frac{1}{2}]^T$, and the matrix $A = (a_{ij})$ is composed of the right upper coefficients at Table 1. Namely the matrix M is semi-positive definite. Thus Runge-Kutta formulas (2.6)-(2.8) are B-stable as $r \geq \frac{1}{4}$. Therefore the third order Runge-Kutta formula (2.6)-(2.8) satisfies the B stability as $r = \frac{3+\sqrt{3}}{6}$.

Furthermore we require that Runge-Kutta methods (2.6)-(2.8) have the L stability according to the above introductions. If the coefficients of Runge-Kutta methods (2.6)-(2.8) satisfy $b^T A^{-1} e = 1$ the methods are L-stable (see [17, 34]). From

$$b^T A^{-1} e = \frac{1}{2}(\frac{1}{r} + \frac{2r-1}{r^2} + \frac{1}{r}) = 1$$

we obtain $r = 1 + \sqrt{2}/2$ or $r = 1 - \sqrt{2}/2$. Namely the numerical method (2.6)-(2.8) is B-stable and L-stable as $r = 1 + \sqrt{2}/2$ or $r = 1 - \sqrt{2}/2$.

Because Runge-Kutta methods (2.6)-(2.8) require solving the system of nonlinear equations every iteration. Thus we consider the following linear model

$$\frac{dx}{dt} = -\nabla f(x_k) - \nabla^2 f(x_k)(x - x_k), \tag{2.10}$$

$$x(0) = x_k. \tag{2.11}$$

The solution of differential equations (2.10)-(2.11) is

$$x(t) = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) + exp(-\nabla^2 f(x_k)t) [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \tag{2.12}$$

if $\nabla^2 f(x_k)$ is nonsingular. From (2.12) we get

$$\lim_{t\to\infty} x(t) = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k), \tag{2.13}$$

namely the solution of (2.10)-(2.11) tends to the Newton step as $t \to \infty$, if $\nabla^2 f(x_k)$ is a positive definite matrix.

Applying Runge-Kutta methods (2.6)-(2.8) to the linear model (2.10)-(2.11) we obtain following iteration formulas

$$(\lambda_k I + r\ G_k)K_1 = -\nabla f(x_k), \tag{2.14}$$

$$(\lambda_k I + r\ G_k)K_2 = -\nabla f(x_k) - (1-2r)G_k K_1, \tag{2.15}$$

$$s_k = \frac{1}{2}(K_1 + K_2), \tag{2.16}$$

$$x_{k+1} = x_k + s_k, \tag{2.17}$$

where $\lambda_k = 1/h$ and $G_k = \nabla^2 f(x_k)$. Thus we obtain the curvilinear search direction s_k in (2.14)-(2.17) and it requires to determine the step size parameter λ_k. We choose the parameter λ_k such that s_k satisfies the Armijo line search criterion [13, 27, 35]

$$f(x_k + s_k) \leq f(x_k) + \alpha s_k^T \nabla f(x_k),\ 0 < \alpha < 1. \tag{2.18}$$

According to the above analysis we can give the following algorithm for solving the unconstrained optimization problem (1.1).

Algorithm 2.2. *The LRKOPT Method*

Step 0: Give initial parameters. *Give an initial point x_1. The constants $\lambda_1 = \min\ \{\|\nabla f(x_k)\|,\ 10\}$, $\alpha = 10^{-4}$, $r = 1 + \frac{\sqrt{2}}{2}$ or $r = 1 - \frac{\sqrt{2}}{2}$ and the tolerable error $Tol = 10^{-6}$ are also given.*

Step 1: Test the terminal criterion. *Compute $\nabla f(x_k)$ and $G_k = \nabla^2 f(x_k)$. If $\nabla f(x_k)$ satisfies the terminal criterion*

$$\|\nabla f(x_k)\| \leq Tol, \tag{2.19}$$

then stop. Otherwise go to Step 2.

Step 2: Solve the subproblem. *Solve (2.14)-(2.17). If $\lambda_k I + r\, G_k \succ 0$ and s_k satisfies the Armijo line search criterion (2.18), then set*

$$x_{k+1} = x_k + s_k, \tag{2.20}$$

$$\lambda_{k+1} = \frac{1}{2}\lambda_k. \tag{2.21}$$

Otherwise set

$$x_{k+1} = x_k, \tag{2.22}$$

$$\lambda_{k+1} = 4\lambda_k. \tag{2.23}$$

Step 3: Continue iterations. *Increase the index k by one and go to Step 1.*

Brown and Bartholomew-Biggs (see [6]) gave the following IMPBOT method. It requires to solve linear equations

$$(\lambda_k I + \nabla^2 f(x_k))s_k = -\nabla f(x_k), \tag{2.24}$$

$$x_{k+1} = x_k + s_k, \tag{2.25}$$

such that

$$f(x_{k+1}) < f(x_k) \tag{2.26}$$

holds. The IMBOT method is the essentially same as the Levenberg-Marquardt method (see [13, 21, 24, 19, 27, 35]). And it obtains the search direction s_k by applying the backward Euler method to the linear model (2.10)-(2.11). However the LRKOPT method obtains the search direction s_k by applying the second order singly diagonally implicit Runge-Kutta method to the linear model (2.10)-(2.11). Thus the LRKOPT method approximates the linear model (2.10)-(2.11) better than the IMPBOT method. Namely the search direction of the LRKOPT method tends to the Newton step better than the search direction of the IMPBOT method and this property is desirable for solving the unconstrained optimization. It is worth noting that the sequence $\{x_k\}$ which is generated by the IMPBOT method (2.24)-(2.26) may not converge to the local minimizer of the function f (see [23]).

3. Analysis of the LRKOPT Method

In this section, we will analyze the convergence of the LRKOPT method. First, we discuss some properties of the LRKOPT method. From (2.14)-(2.16) we obtain

$$\begin{aligned} s_k(\lambda_k) &= \frac{1}{2}(K_1 + K_2) \\ &= -(\lambda_k I + rG_k)^{-1}\nabla f(x_k) + \frac{1-2r}{2}(\lambda_k I + rG_k)^{-1}G_k(\lambda_k I + rG_k)^{-1}\nabla f(x_k). \end{aligned} \tag{3.1}$$

Therefore, $s_k(\lambda_k)$ moves towards the negative gradient direction as $\lambda_k \to \infty$. If the Hessian matrix G_k is nonsingular, we let $\lambda_k = 0$ in (3.1) and obtain

$$\begin{aligned} s_k(0) &= -(\frac{1}{r} - \frac{1-2r}{2r^2})G_k^{-1}\nabla f(x_k) \\ &= -G_k^{-1}\nabla f(x_k). \end{aligned} \tag{3.2}$$

Namely the search direction $s_k(\lambda_k)$ moves towards the Newton step as $\lambda_k \to 0$.

Lemma 3.1. *The function $d(\lambda) = \|s_k(\lambda)\|$ decreases monotonically and it has the maximizer $\|G_k^{-1}\nabla f(x_k)\|$ as $r = 1 + \frac{\sqrt{2}}{2}$, $\lambda \geq 0$ and G_k is a symmetric positive definite matrix, where $s_k(\lambda)$ is defined by (3.1). And $d(\lambda)$ decreases monotonically as $r = 1 - \frac{\sqrt{2}}{2}$ and $\lambda \geq (1-3r)\mu_{max}$, where G_k is a symmetric positive definite matrix and μ_{max} is the maximum eigenvalue of G_k.*

Proof. Because the matrix G_k is symmetric it exists an orthogonal matrix Q_k such that $Q_k^T G_k Q_k = diag(\mu_1, \mu_2, \cdots, \mu_n)$, where μ_i are the eigenvalues of G_k. We define

$$z = Q_k^T \nabla f(x_k). \tag{3.3}$$

From (3.1) and (3.3) we have

$$\begin{aligned} d(\lambda) &= \|s_k(\lambda)\| \\ &= \|Q_k^T s_k(\lambda)\| \\ &= \sqrt{\sum_{i=1}^{n} [\frac{1}{\lambda + r\mu_i} - \frac{1-2r}{2}\frac{\mu_i}{(\lambda + r\mu_i)^2}]^2 z_i^2}. \end{aligned} \tag{3.4}$$

We define

$$\begin{aligned} \varphi(\lambda, \mu) &= \frac{1}{\lambda + r\mu} - \frac{1-2r}{2}\frac{\mu}{(\lambda + r\mu)^2} \\ &= \frac{2\lambda + (4r-1)\mu}{2(\lambda + r\mu)^2}. \end{aligned} \tag{3.5}$$

And we have $4r - 1 > 0$ as $r = 1 + \sqrt{2}/2$ or $r = 1 - \sqrt{2}/2$. Therefore $\varphi(\lambda, \mu) > 0$ as $\lambda \geq 0$ and $\mu > 0$. From (3.5) we have

$$\varphi_\lambda(\lambda, \mu) = -\frac{\lambda + (3r-1)\mu}{(\lambda + r\mu)^3} \leq 0, \tag{3.6}$$

as $\lambda \geq (1 - 3r)\mu$. Thus $\varphi(\lambda, \mu)$ decreases monotonically on the variable λ as $r = 1 + \sqrt{2}/2$, $\mu > 0$ and $\lambda \geq 0$, namely $d(\lambda)$ decreases monotonically and it has the maximizer $\|G_k^{-1}\nabla f(x_k)\|$ as $\lambda = 0$. And $\varphi(\lambda, \mu)$ also decreases monotonically on λ as $r = 1 - \sqrt{2}/2$, $\lambda \geq (1 - 3r)\mu$ and $\mu > 0$. Therefore we also get the second part of the lemma.

Lemma 3.2. *The function $\phi(\lambda) = -\nabla^T f(x_k)s_k(\lambda)$ is greater than zero and decreases monotonically as G_k is a symmetric positive definite matrix and $r = 1 + \frac{\sqrt{2}}{2}$, $\lambda \geq 0$ or $r = 1 - \frac{\sqrt{2}}{2}$, $\lambda \geq (1 - 3r)\mu_{max}$, where μ_{max} is the maximum eigenvalue of G_k and $s_k(\lambda)$ is defined by (3.1). Furthermore $\lim_{\lambda\to\infty} -\frac{\nabla^T f(x_k)s_k(\lambda)}{\|s_k(\lambda)\|\cdot\|\nabla f(x_k)\|} = 1$, namely the search direction $s_k(\lambda)$ moves towards the negative gradient direction as $\lambda \to \infty$.*

Proof. From (3.1) we obtain

$$\begin{aligned} \phi(\lambda) &= -s_k^T(\lambda)\nabla f(x_k) \\ &= \sum_{i=1}^{n} [\frac{1}{\lambda + r\mu_i} - \frac{(1-2r)\mu_i}{2(\lambda + r\mu_i)^2}] z_i^2 \\ &= \sum_{i=1}^{n} \frac{2\lambda + (4r-1)\mu_i}{2(\lambda + r\mu_i)^2} z_i^2 \\ &> 0, \end{aligned} \tag{3.7}$$

where the vector z is defined by (3.3) and μ_i are the eigenvalues of the positive matrix G_k. Therefore, from (3.5)-(3.7) we know that $\phi(\lambda)$ decreases monotonically and $\phi(\lambda) > 0$ as $r = 1 + \frac{\sqrt{2}}{2}$, $\lambda \geq 0$ or $r = 1 - \frac{\sqrt{2}}{2}$, $\lambda \geq (1 - 3r)\mu_{max}$, where μ_{max} is the maximum eigenvalue of G_k.

From (3.5) we have

$$\begin{aligned} \varphi_\mu(\lambda, \mu) &= -\frac{\lambda + r(4r-1)\mu}{2(\lambda + r\mu)^3} \\ &\leq 0, \end{aligned} \tag{3.8}$$

as $\mu > 0$, $\lambda \geq 0$ and $r = 1 + \sqrt{2}/2$ or $r = 1 - \sqrt{2}/2$. Thus $\varphi(\lambda, \mu)$ decreases monotonically on μ. Combining (3.4) and (3.7)-(3.8) we obtain

$$\begin{aligned} -\frac{s_k^T(\lambda)\nabla f(x_k)}{\|\nabla f(x_k)\| \cdot \|s_k(\lambda)\|} &= \sum_{i=1}^{n} \frac{2\lambda + (4r-1)\mu_i}{2(\lambda + r\mu_i)^2} z_i^2 / [\|z\| \sqrt{\sum_{i=1}^{n} (\frac{2\lambda + (4r-1)\mu_i}{2(\lambda + r\mu_i)^2})^2 z_i^2}\ ] \\ &\geq [\frac{2\lambda + (4r-1)\mu_{max}}{2(\lambda + r\mu_{max})^2} \|z\|^2] / [\frac{2\lambda + (4r-1)\mu_{min}}{2(\lambda + r\mu_{min})^2} \|z\|^2] \\ &= [\frac{2\lambda + (4r-1)\mu_{max}}{2(\lambda + r\mu_{max})^2}] / [\frac{2\lambda + (4r-1)\mu_{min}}{2(\lambda + r\mu_{min})^2}] \end{aligned} \quad (3.9)$$

by using the monotonically decreasing property of $\varphi(\lambda, \mu)$ on μ, where μ_{max} and μ_{min} are the maximum and minimum eigenvalues of G_k, respectively. Thus, from (3.9) we have

$$\lim_{\lambda \to \infty} -\frac{s_k^T(\lambda)\nabla f(x_k)}{\|\nabla f(x_k)\| \cdot \|s_k(\lambda)\|} \geq 1. \quad (3.10)$$

Furthermore we get

$$-\frac{s_k^T(\lambda)\nabla f(x_k)}{\|\nabla f(x_k)\| \cdot \|s_k(\lambda)\|} \leq 1 \quad (3.11)$$

by the Cauchy-Schwartz inequality. (3.10) and (3.11) yield $\lim_{\lambda \to \infty} -\frac{s_k^T(\lambda)\nabla f(x_k)}{\|\nabla f(x_k)\| \cdot \|s_k(\lambda)\|} = 1$.

Using the above lemmas, we give the following global convergence analysis of the LRKOPT method.

Theorem 3.3. *Assume that f is twice continuously differentiable and has the lower bound. And suppose that the sequence x_k is generated by Algorithm 2.2 such that $x_k \in B$ and $\nabla f(x_k) \neq 0$ for all k, where B is a closed convex set in R^n. If G_k are symmetric positive definite matrices. Then $\{x_k\}$ converges to the local minimizer or the stable point of $f(x)$, namely* $\inf_{k \to \infty} \|\nabla f(x_k)\| = 0$.

Proof. It is convenient to distinguish between two cases:

$$\text{(i)}\ \sup \lambda_k = \infty, \qquad \text{(ii)}\ \sup \lambda_k \leq W \ \text{ for some constant W.} \quad (3.12)$$

Case (i). From (2.21) and (2.23), there must be an infinite subsequence whose indices form a set S_1 such that (2.23) is satisfied, namely s_k does not satisfy the Armijo line search criterion (2.18) or $\lambda_k I + G_k \preceq 0$ for $k \in S_1$. Also, using the bounds of G_k and $\nabla f(x_k)$, from (3.1) we have

$$\lim_{k \to \infty} \|s_k\| = 0, \quad k \in S_1, \quad (3.13)$$

because $\lim_{k \to \infty} \lambda_k = \infty$ for $k \in S_1$. In the following proof we will proceed it by contradiction. If the conclusion were not true there would exist a positive constant δ such that

$$\|\nabla f(x_k)\| \geq \delta > 0, \quad (3.14)$$

holds for all k. Then we get

$$\lim_{k \to \infty} \frac{f(x_k + s_k) - f(x_k)}{s_k^T \nabla f(x_k)} = 1, \quad k \in S_1, \quad (3.15)$$

which gives

$$\frac{f(x_k + s_k) - f(x_k)}{s_k^T \nabla f(x_k)} \geq \alpha \quad (3.16)$$

for the sufficiently large $k \in S_1$. From (3.7) we have $s_k^T \nabla f(x_k) < 0$. Combining (3.16) we obtain

$$f(x_k + s_k) \leq f(x_k) + \alpha s_k^T \nabla f(x_k), \quad (3.17)$$

and $\lambda_k I + G_k \succ 0$ for sufficiently large $k \in S_1$, which contradict the definition of S_1.

Case (ii). From (2.21) there must be an infinite subsequence whose indices form a set S_2 such that s_k satisfies the Armijo line search criterion (2.18) for $k \in S_2$. Thus we have

$$\sum_{k\in S_2}[-\alpha s_k^T \nabla f(x_k)] \le \sum_{k\in S_2}[f(x_k) - f(x_{k+1})] \le \sum_{k=1}^{\infty}[f(x_k) - f(x_{k+1})]. \tag{3.18}$$

In the following proof we will proceed it by contradiction. If the conclusion were not true there would exist a positive constant δ such that (3.14) holds for all k. From (3.7) we have $s_k^T \nabla f(x_k) < 0$. Thus $\{f(x_k)\}$ is a decreasing sequence. Moreover the sequence $\{f(x_k)\}$ has the lower bound we get

$$\sum_{k\in S_2}[-s_k^T \nabla f(x_k)] < \infty, \tag{3.19}$$

which gives

$$\lim_{k\to\infty} -s_k^T \nabla f(x_k) = 0, \quad k \in S_2. \tag{3.20}$$

From Lemma 3.1 and Lemma 3.2 we know that $\varphi(\lambda, \mu)$ is a monotonically decreasing function on variables λ and μ respectively, where $\varphi(\lambda, \mu)$ is defined by (3.5). Therefore we have

$$\begin{aligned} -s_k^T \nabla f(x_k) &= \sum_{i=1}^{n} \frac{2\lambda_k + (4r-1)\mu_i}{2(\lambda_k + r\mu_i)^2} z_i^2 \\ &\ge \sum_{i=1}^{n} \frac{2W + (4r-1)M}{2(W + rM)^2} z_i^2 \\ &= \frac{2W + (4r-1)M}{2(W + rM)^2} \|Q_k^T \nabla f(x_k)\| \\ &= \frac{2W + (4r-1)M}{2(W + rM)^2} \|\nabla f(x_k)\|, \end{aligned} \tag{3.21}$$

where the vector z is defined by (3.3), μ_i are the eigenvalues of G_k and M is a constant such that $\|G_k\| \le M$. From (3.20) and (3.21) we also get $\inf_{k\to\infty} \|\nabla f(x_k)\| = 0$ which contradicts the assumption (3.14).

It is a natural conjecture that the LRKOPT method has the local superlinear convergence from (3.2). First, we state the characters of the superlinear convergence before we give the analysis of local convergence for the LRKOPT method.

Theorem 3.4. *(Moré and Dennis (1974) [11]) Let $f : R^n \to R$ be twice continuously differentiable in an open convex set D, and assume that $\nabla^2 f(x)$ is Lipschitz continuous in D. Consider a sequence $\{x_k\}$ generated by $x_{k+1} = x_k + \lambda_k p_k$, where $\nabla f(x_k)^T p_k < 0$ for all k and λ_k is chosen to satisfy $f(x_{k+1}) \le f(x_k) + \alpha \nabla f(x_k)^T (x_{k+1} - x_k)$ and $\nabla f(x_{k+1})^T (x_{k+1} - x_k) \ge \beta \nabla f(x_k)^T (x_{k+1} - x_k)$, where $0 < \alpha < \frac{1}{2} < \beta < 1$. If $\{x_k\}$ converges to a point $x^\star \in D$ at which $\nabla^2 f(x^\star)$ is positive definite, and if*

$$\lim_{k\to\infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k) p_k\|}{\|p_k\|} = 0, \tag{3.22}$$

then there is an index k_0 such that for all $k \ge k_0$, $\lambda_k = 1$ is admissible. Furthermore, $\nabla f(x^\star) = 0$, and if $\lambda_k = 1$ for all $k \ge k_0$, then x_k converges q-superlinearly to $x^\star$.

Using Theorem 3.4 we obtain the following the local convergence of the LRKOPT method.

Theorem 3.5. *Let $f : R^n \to R$ be twice continuously differentiable in an open convex set D and assume that $G(x) = \nabla^2 f(x)$ is Lipschitz continuous in D. If the sequence $\{x_k\}$ is generated by Algorithm 2.2 and converges to a point $x^\star \in D$ at which $\nabla^2 f(x^\star)$ is positive definite, then x_k converges q-superlinearly to $x^\star$.*

Proof. Because $\{x_k\}$ converges to $x^\star$ we have $\lim_{k\to\infty}\|s_k\| = 0$. By $\nabla^2 f(x^\star) \succ 0$, we get

$$m\|x\|^2 \le x^T\nabla^2 f(x_k)x \le M\|x\|^2, \quad \forall x \in R^n, \tag{3.23}$$

for sufficiently large k.

Because the matrix G_k is symmetric positive definite for sufficiently large k it exists an orthogonal matrix Q_k such that $Q_k^T G_k Q_k = diag(\mu_1, \cdots, \mu_n)$, where u_i are the eigenvalues of G_k. We denote

$$\bar{s} = Q_k^T s_k, \quad z = Q_k^T \nabla f(x_k). \tag{3.24}$$

Thus, from (3.1) and (3.24) we have

$$z_i = -\frac{(\lambda_k + r\mu_i)^2}{\lambda_k + (2r - \frac{1}{2})\mu_i}\bar{s}_i. \tag{3.25}$$

Therefore (3.24) and (3.25) yield

$$\begin{aligned}
-s_k^T\nabla f(x_k) - \frac{1}{2}s_k^T G_k s_k &= -\frac{2-\sqrt{2}}{4}s_k^T\nabla f(x_k) - \frac{2+\sqrt{2}}{4}s_k^T\nabla f(x_k) - \frac{1}{2}s_k^T G_k s_k \\
&= -\frac{2-\sqrt{2}}{4}s_k^T\nabla f(x_k) + \sum_{i=1}^{n}\frac{\frac{2+\sqrt{2}}{4}\lambda_k^2 + (\frac{2+\sqrt{2}}{2}r - \frac{1}{2})\lambda_k\mu_i + (\frac{2+\sqrt{2}}{4}r^2 - r + \frac{1}{4})\mu_i^2}{\lambda_k + (2r-\frac{1}{2})\mu_i}\bar{s}_i^2 \\
&\ge -\frac{2-\sqrt{2}}{4}s_k^T\nabla f(x_k) \\
&= \frac{2-\sqrt{2}}{4}\sum_{i=1}^{n}\frac{(\lambda_k + r\mu_i)^2}{\lambda_k + (2r-\frac{1}{2})\mu_i}\bar{s}_i^2 \ge 0,
\end{aligned} \tag{3.26}$$

where $r = 1 + \frac{\sqrt{2}}{2}$ or $r = 1 - \frac{\sqrt{2}}{2}$. Thus we obtain

$$\begin{aligned}
\frac{f(x_k) - f(x_k + s_k)}{-s_k^T\nabla f(x_k)} &= \frac{-s_k^T\nabla f(x_k) - \frac{1}{2}s_k^T G_k s_k + o(\|s_k\|^2)}{-s_k^T\nabla f(x_k)} \\
&> \frac{2-\sqrt{2}}{4} + \frac{o(\|s_k\|^2)}{-s_k^T\nabla f(x_k)} \\
&= \frac{2-\sqrt{2}}{8} + \frac{2-\sqrt{2}}{8} + \frac{o(\|s_k\|^2)}{-s_k^T\nabla f(x_k)} \\
&> \frac{2-\sqrt{2}}{8}
\end{aligned} \tag{3.27}$$

for sufficiently large k, because

$$\frac{o(\|s_k\|^2)}{-s_k^T\nabla f(x_k)} \le \frac{o(\|s_k\|^2)}{\frac{1}{2}s_k^T G_k s_k} \le \frac{1}{2}\frac{o(\|s_k\|^2)}{m\|s_k\|^2} \to 0, \quad \text{as} \quad k \to \infty. \tag{3.28}$$

In the above the inequality $-s_k^T\nabla f(x_k) \ge \frac{1}{2}s_k^T G_k s_k$ is derived from (3.26). Therefore we have

$$\frac{f(x_k) - f(x_k + s_k)}{-s_k^T\nabla f(x_k)} \ge \alpha \tag{3.29}$$

for sufficiently large k, because $0 < \alpha = 10^{-4} < \frac{2-\sqrt{2}}{8}$ from Step 0 of Algorithm 2.2. From (3.26) and (3.29) we get

$$f(x_k + s_k) \le f(x_k) + \alpha s_k^T\nabla f(x_k), \tag{3.30}$$

namely s_k satisfies the Armijo line search criterion for sufficiently large k. Therefore, from (2.21) we have $\lim_{k\to\infty}\lambda_k = 0$.

From Theorem 3.4, we need only verify $\|s_k - s_k^N\|/\|s_k\| \to 0$ as $k \to \infty$ when we analyze the local convergence of the LRKOPT method, where $s_k^N = -G_k^{-1}\nabla f(x_k)$ is the Newton step. From (3.1) we have

$$\begin{aligned} s_k - s_k^N &= (\lambda_k I + G_k)^{-1}[(r-1)I + \lambda_k G_k^{-1} + \frac{1-2r}{2}G_k(\lambda_k I + rG_k)^{-1}]\nabla f(x_k) \\ &= (\lambda_k I + G_k)^{-1}[(2r-1)\lambda_k I + \lambda_k^2 G_k^{-1}](\lambda_k I + G_k)^{-1}\nabla f(x_k), \end{aligned} \tag{3.31}$$

where the constant $r = 1 + \sqrt{2}/2$ or $r = 1 - \sqrt{2}/2$. Therefore, from (3.31) and $\lim_{k\to\infty}\lambda_k = 0$ we have $\lim_{k\to\infty}\|s_k - s_k^N\|/\|s_k\| = 0$, namely the LRKOPT method has the superlinear convergence.

It requires to solve the systems of linear equations for K_1 and K_2 in (2.14) and (2.15). The computational complexity of K_1 is $O(n^3)$, but the computational complexity of K_2 is $O(n^2)$ because (2.14) and (2.15) have the same coefficient matrix, we need only factor the matrix once and perform two back-substitutions, one for each right-hand side. It is worth noting that we can update the matrices G_k of Algorithm 2.2 by quasi-Newton formulas, particularly the BFGS update formula

$$G_{k+1} = G_k - \frac{G_k s_k s_k^T G_k}{s_k^T G_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{3.32}$$

if the curvature condition $s_k^T y_k > 0$ holds, where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

4. Numerical Tests

We have implemented Algorithm 2.2 (i.e. the LRKOPT method) and compared it with the IMPBOT method (2.24)-(2.26) given in [6]. They require to give an initial parameter λ_1 for the LRKOPT method and the IMPBOT method. As the initial parameter λ_1 is very small, these methods behave like the Newton method and may require more iteration steps such that the search direction s_1 satisfies the Armijo line search criterion. On the other hand, these methods behave like the negative gradient method as the initial parameter λ_1 is very large. Thus we choose an initial parameter $\lambda_1 = 0.1, 1, 10, 100$ for small problems.

Our test problems are from [25] and they are least square problems $f(x) = \sum_{i=1}^m f_i^2(x)$. And we give the following test functions in reason for convenience.

1. Rosenbrock function: $f(x) = (10(x_2 - x_1^2))^2 + (1 - x_1)^2$, $x_0 = (-1.2, 1)$.

2. Powell badly scaled function: $f(x) = (10^4 x_1 x_2 - 1)^2 + (e^{-x_1} + e^{-x_2} - 1.0001)^2$, $x_0 = (0, 1)$.

3. Brown badly scaled function: $f(x) = (x_1 - 10^6)^2 + (x_2 - 2\cdot 10^{-6})^2 + (x_1 x_2 - 2)^2$, $x_0 = (1, 1)$.

4. Wood function: $f(x) = (10(x_2 - x_1^2))^2 + (1 - x_1)^2 + (\sqrt{90}(x_4 - x_3^2))^2 + (1 - x_3)^2 + (\sqrt{10}(x_2 + x_4 - 2))^2 + (\frac{1}{\sqrt{10}}(x_2 - x_4))^2$, $x_0 = (-3, -1, -3, -1)$.

5. Helical valley function: $f(x) = (10(x_3 - 10\theta(x_1, x_2)))^2 + (10(\sqrt{x_1^2 + x_2^2} - 1))^2 + x_3^2$, where

$$\theta(x_1, x_2) = \begin{cases} \frac{1}{2\pi} arctan(\frac{x_2}{x_1}), & \text{if } x_1 > 0, \\ \frac{1}{2\pi} arctan(\frac{x_2}{x_1}) + 0.5, & \text{if } x_1 < 0, \end{cases}$$

   and $x_0 = (-1, 0, 0)$.

We use analytical expressions of test functions and their gradient functions as we implement the LRKOPT method or the IMPBOT method. But we obtain the second order derivatives of test functions by using the difference method. We choose the parameters $r = 1 - \sqrt{2}/2$ for the

LRKOPT method and $\alpha = 1.0 \times 10^{-4}$ for the Armijo line search criterion (2.18). The iterative termination criterion is

$$\|\nabla f(x)\|_2 \leq 1.0 \times 10^{-6}.$$

Numerical results are given at Table 2. ANITR denotes average iterations for different initial parameters λ_1 and AEFE denotes the average number of the equivalent evaluation number of $f(x)$ for different initial parameters λ_1. We observe that the LRKOPT method performed better than the IMPBOT method from Table 2, in terms of function evaluations. Therefore methods based on ordinary differential equations are worth further exploration.

Table 2: Numerical results of the IMPBOT method and the LRKOPT method

PRB		ANITR	AEFE
1	LRKOPT	21.25	201.75
	IMPBOT	21.75	206.75
2	LRKOPT	91.5	881
	IMPBOT	97.75	940.75
3	LRKOPT	17.25	157.75
	IMPBOT	16	146.25
4	LRKOPT	38.75	917.5
	IMPBOT	41	968.25
5	LRKOPT	17	255
	IMPBOT	20	300

Acknowledgments. Authors are grateful to anonymous referees for their helpful suggestions.

References

[1] E. Allgower and K. Georg, Simplicial and continuous methods for approximating fixed points and solutions to systems of equations, *SIAM Review*, **22** (1980), 28-85.

[2] C.A. Botsaris and D.H. Jacobson, A Newton-type curvilinear search method for optimization, *Journal of Mathematical Analysis and Applications*, **54** (1976), 217-229.

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

[4] C.A. Botsaris, A curvilinear optimization method based on iterative estimation of the eigensystem of the Hessian matrix, *Journal of Mathematical Analysis and Applications*, **63** (1978b), 396-411.

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

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

[7] J.C. Butcher, A stability property of implicit Runge-Kutta methods, *BIT*, **15** (1975), 358-361.

[8] K. Burrage and J.C. Butcher, Stability criteria for implicit Runge-Kutta methods, *SIAM J. Numer. Anal.*, **16** (1979), 46-57.

[9] A. Cichocki, R. Unbehauen, Neural Networks for Optimization and Signal Processing, Chichester, Wiley, 1993.

[10] I.D. Coope and C.J. Price, A Direct Search Frame-based Conjugate Gradients Method, *J. Comput. Math.,* **22** (2004), 489-500.

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

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

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

[14] Y.H. Dai and Q. Ni, Testing different conjugate gradient methods for large-scale unconstrained optimization, *J. Comput. Math.,* **21** (2003), 331-320.

[15] Y.H. Dai and Y. Yuan, A note on the nonlinear conjugate gradient method, *J. Comput. Math.,* **20** (2002), 575-582.

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

[17] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, Springer-Verlag, (1996).

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

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

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

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

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

[23] X.L. Luo, Convergence analysis of methods based on BDF for unconstrained optimization , *Mathematica Numerica Sinica*, **25** (2003), 177-184.

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

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

[26] J.J. Moré, Recent developments in algorithms and software for trust region methods, in: A. Bachem, M. Grotschel and B. Korte, eds., Mathematical Programming: The State of the Art, Springer-Verlag, Berlin, (1983), 258-287.

[27] J. Nocedal and S.J. Wright, Numerical Optimization, Springer-Verlag, (1999).

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

[29] M.J.D. Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line search, in: R.W. Cottle and C.E.Lemke eds., Nonlinear Programming, SIAM-AMS Proceedings IX, SIAM publications, (1976), 53-72.

[30] M. J. D. Powell ed., Discussions on Unconstrained Optimization, Nonlinear Optimization 1981 , Academic Press, 59-64.

[31] S. Schaffler and H. Warsitz, A trajectory-following method for unconstrained optimization, *Journal of Optimization Theory and Applications*, **67** (1990), 133-140.

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

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

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

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