

RESEARCH

Open Access



A q -Polak–Ribière–Polyak conjugate gradient algorithm for unconstrained optimization problems

Shashi Kant Mishra¹, Suvra Kanti Chakraborty², Mohammad Esmael Samei^{3*}  and Bhagwat Ram⁴

*Correspondence:
mesamei@gmail.com;
mesamei@basu.ac.ir

³Department of Mathematics, Bu-Ali Sina University, Hamedan, Iran
Full list of author information is available at the end of the article

Abstract

A Polak–Ribière–Polyak (PRP) algorithm is one of the oldest and popular conjugate gradient algorithms for solving nonlinear unconstrained optimization problems. In this paper, we present a q -variant of the PRP (q -PRP) method for which both the sufficient and conjugacy conditions are satisfied at every iteration. The proposed method is convergent globally with standard Wolfe conditions and strong Wolfe conditions. The numerical results show that the proposed method is promising for a set of given test problems with different starting points. Moreover, the method reduces to the classical PRP method as the parameter q approaches 1.

MSC: 34A08; 34B16; 90C26; 39A13

Keywords: Unconstrained optimization; Conjugate gradient method; Global convergence; q -calculus

1 Introduction

The conjugate gradient (CG) methods have played an important role in solving nonlinear optimization problems due to their simplicity of iteration and very low memory requirements [1, 2]. Of course, the CG methods are not among the fastest or most robust optimization algorithms for solving nonlinear problems today, but they are very popular among engineers and mathematicians to solve nonlinear optimization problems [3–5]. The origin of the methods dates back to 1952 when Hestenes and Stiefel introduced a CG method [6] for solving a symmetric positive definite linear system of equations. Further, Fletcher and Reeves modified the same method called FR [7] in the 1960s and developed a conjugate gradient method to solve unconstrained nonlinear optimization problems.

The conjugate gradient methods deflect the steepest descent method [8] by adding to it a positive multiple of the direction used in the previous step. They only require the first-order derivative and overcome the shortcomings of the slow convergence rate of the steepest descent method. By means of conjugacy, the conjugate gradient methods make the steepest descent direction to account for conjugacy and thus enhance the efficiency and reliability of the algorithm. Different conjugate gradient algorithms correspond to different choices of the scalar parameter β_k [6, 7, 9]. The parameter β_k is selected to mini-

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

mize a convex quadratic function in a subspace spanned by a set of mutually conjugate descent directions, but the effectiveness of the algorithm depends on the accuracy of the line searches.

Quantum calculus, known as q -calculus, is the study of calculus without limits, where classical mathematical formulas are obtained as q approaches 1. In q -calculus, the classical derivative is replaced by the q -difference operator. Jackson [10, 11] was the first to have some applications of the q -calculus and introduced the q -analogue of the classical derivative and integral operators. Applications of q -calculus play an important role in various fields of mathematics and physics [12–20].

In 1969, Polak and Ribière [21] and Polyak [22] proposed a conjugate gradient method independently, later it was called Polak, Ribière, and Polyak (PRP) method. In view of the practical computation, the PRP method performed much better than the FR method for many unconstrained optimization problems because it automatically recovered once a small step length was generated, although the global convergence of the PRP method was proved only for the strictly convex functions [23]. For general nonlinear functions, Powell showed that the PRP method could cycle infinitely without approaching a solution even if the step-length was chosen to be the least positive minimizer of the line search function [24]. To change this unbalanced state, Gilbert and Nocedal [25] considered Powell's suggestions [26] to modify the PRP method and showed that this modification of the PRP method is globally convergent for exact and inexact line searches.

In 2019, Yuan et al. proposed a new modified three-term conjugate gradient algorithm based on the modified Armijo line search technique [27]. After that in 2020, they designed a modified conjugate gradient method with a sufficient descent property and a trust region property [28]. The authors in [29] proposed the modified Hestenes–Stiefel (HS) conjugate gradient algorithm in order to solve large-scale complex smooth and nonsmooth optimization problems.

In 2020, Yuan et al. further proposed the PRP method and established the global convergence proof with the modified weak Wolfe–Powell line search technique for nonconvex functions. The numerical results demonstrated the competitiveness of the method compared to the existing methods. The engineering Muskingum model and image restoration problems were used to determine the interesting aspects of the given algorithm [30]. The generalized conjugate gradient algorithms were studied for solving large-scale unconstrained optimization problems within the real world applications, and two open problems were formulated [31–33].

The preliminary experimental optimization results using q -calculus were first shown in the field of global optimization [34]. The idea of this work is utilized in the stochastic q -neurons which are based on activation functions converted into the corresponding stochastic q -activation functions for improving the effectiveness of the algorithm. The q -gradient concept is further utilized in the least mean square algorithm to inherit the fast convergence property with less dependency on the eigenvalue of the input correlation matrix [35]. A modified least mean algorithm using q -calculus was also proposed which automatically adapted the learning rate with respect to the error and was shown to have fast convergence [36]. In optimization, the q -calculus was employed in Newton, modified Newton, BFGS, and limited memory BFGS methods for solving unconstrained nonlinear optimization problems [19, 37–40] with the least number of iterations. In the field of conjugate gradient methods, the q -analogue of the Fletcher–Reeves method was

developed [41] to optimize unimodal and multimodal functions, and the Gaussian perturbations were used in some iterations to ensure the convergence globally in the probabilistic sense only.

In this paper, we propose a q -variant of PRP method, called q -PRP, with the sufficient descent property independent of the line searches and convexity assumption of the objective function. Under a condition on the q -gradient of the objective function and some other appropriate conditions, the proposed method is globally convergent. The numerical experiments are conducted to show the effectiveness of the q -PRP algorithm. For a set of given test functions with different starting points, it was able to escape from many local minima to reach global minima due to q -gradient.

The remainder of this paper is organized as follows: In the next section, we present the essential preliminaries. The main results are presented in Sect. 3, and their convergence proofs are given in Sect. 4. The numerical examples of the theoretical results are analyzed in Sect. 5. The paper is then ended with a conclusion and directions for future work.

2 Essential preliminaries

In this section, the principal terms of q -calculus are formed by assuming $0 < q < 1$, as follows: The q -integer $[n]_q$ is defined by

$$[n]_q = \begin{cases} \frac{1-q^n}{1-q}, & q \neq 1, \\ n, & q = 1, \end{cases}$$

for all $n \in \mathbb{N}$. The q -analogue of $(1+x)^n$ is the polynomial given by

$$(1+x)_q^n = \begin{cases} 1, & n = 0, \\ \prod_{k=0}^{n-1} (1+q^k x), & n \geq 1. \end{cases}$$

The derivative of x^n with respect to x is given by $[n]_q x^{n-1}$. The q -derivative $D_q f$ of a function f is given by

$$D_q f(x) = \frac{f(qx) - f(x)}{qx - x},$$

if $x \neq 0$, and $D_q f(0) = f'(0)$, provided $f'(0)$ exists. Note that

$$\lim_{q \rightarrow 1} D_q f(x) = \lim_{q \rightarrow 1} \frac{f(qx) - f(x)}{(q-1)x} = \frac{df(x)}{dx},$$

if f is differentiable.

Example 2.1 Let the function $f : \mathbb{R} \rightarrow \mathbb{R}$ be such that $f(x) = \ln x$. Then, we have

$$\left(\frac{d}{dx}\right)_q \ln x = \frac{\ln x - \ln(qx)}{(1-q)x} = \frac{\ln \frac{1}{q}}{(1-q)x}.$$

It is obvious that the q -derivative of a function is a linear operator, that is, for any constant a and b , we have [42]

$$D_q\{af(x) + bg(x)\} = aD_qf(x) + bD_qg(x).$$

Let $f(x)$ be a continuous function on $[a, b]$, where $a, b \in \mathbb{R}$. Then, there exist $\hat{q} \in (0, 1)$ and $x \in (a, b)$ [43] such that

$$f(b) - f(a) = D_qf(x)(b - a),$$

for all $q \in (\hat{q}, 1) \cup (1, \hat{q}^{-1})$. The q -partial derivative of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at $x \in \mathbb{R}^n$ with respect to x_i , where scalar $q \in (0, 1)$, is given as [34]

$$D_{q_i x_i} f(x) = \begin{cases} \frac{1}{(1-q)x_i} [f(x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) - f(x_1, x_2, \dots, x_{i-1}, qx_i, x_{i+1}, \dots, x_n)], & x_i \neq 0, q \neq 1, \\ \frac{\partial}{\partial x_i} f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n), & x_i = 0, \\ \frac{\partial}{\partial x_i} f(x_1, x_2, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n), & q = 1. \end{cases}$$

We now choose the parameter q as a vector, that is,

$$q = (q_1, \dots, q_i, \dots, q_n)^T \in \mathbb{R}^n.$$

Then, the q -gradient vector [34] of f is

$$\nabla_q f(x)^T = [D_{q_1 x_1} f(x) \quad \dots \quad D_{q_i x_i} f(x) \quad \dots \quad D_{q_n x_n} f(x)].$$

Let $\{q_i^k\}$ be a real sequence defined by

$$q_i^{k+1} = 1 - \frac{q_i^k}{(k+1)^2}, \tag{1}$$

for each $i = 1, \dots, n$, where $k = 0, 1, 2, \dots$, and a fixed starting number $0 < q_i^0 < 1$. Of course, the sequence $\{q_i^k\}$ converges to $(1, \dots, 1)$ as $k \rightarrow \infty$ [38]. Thus, the q -gradient reduces to a classical derivative. For the sake of convenience, we represent the q -gradient vector of f at x^k as

$$g_{q^k}(x^k) = \nabla_{q^k} f(x^k).$$

Example 2.2 Consider the function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ defined by

$$f(x) = x_1 x_2^2 + 4x_1^2.$$

Then, the q -gradient is given as

$$\nabla_{q^k} f(x)^T = [4(1 + q_1^k)x_1 + x_2^2 \quad x_1(1 + q_1^k)x_2].$$

In the next section, we present the q -PRP method. To improve the efficiency, we utilize the q -gradient in inexact line search methods to generate the step-length which ensures the reduction of the objective function value.

3 On q -Polak–Ribière–Polyak conjugate gradient algorithm

Consider the following unconstrained nonlinear optimization problem:

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously q -differentiable function. The numerical optimization algorithms of general objective functions differ mainly in generating the search directions. In the conjugate gradient algorithms, a sequence of iterates is generated with a given starting point $x^0 \in \mathbb{R}^n$ by the following schema:

$$x^{k+1} = x^k + p^k, \quad p^k = \alpha_k d_{q^k}^k, \tag{2}$$

for all $k \geq 0$, where x^k is the current iterate, $d_{q^k}^k$ is a descent direction of f at x^k and $\alpha_k > 0$ is the step-length. Note that the descent direction $d_{q^k}^k = -g_{q^k}^k$ leads to the q -steepest descent method [34]. In the case q^k approaches

$$(1, 1, \dots, 1)^T$$

as $k \rightarrow \infty$, the method reduces to the classical steepest descent method [7]. The search direction $d_{q^k}^k$ is guaranteed to have a descent direction due to the following:

$$(g_{q^k}^k)^T d_{q^k}^k < 0. \tag{3}$$

The directions $d_{q^k}^k$ are generated in the light of classical conjugate direction methods [7, 9, 21, 44, 45] as

$$d_{q^k}^k = \begin{cases} -g_{q^k}^k, & k = 0, \\ -g_{q^k}^k + \beta_k^{q\text{-PRP}} d_{q^{k-1}}^{k-1}, & k \geq 1, \end{cases} \tag{4}$$

where $\beta_k^{q\text{-PRP}} \in \mathbb{R}$ is modified from a scalar quantity β_k in the PRP method and presented as follows:

$$\beta_k^{q\text{-PRP}} = \frac{(g_{q^k}^k)^T (g_{q^k}^k - g_{q^{k-1}}^{k-1})}{\|g_{q^{k-1}}^{k-1}\|^2}. \tag{5}$$

Some well-known conjugate gradient methods are available, such as FR (Fletcher–Reeves) [7], PRP (Polak–Ribière–Polyak) [9, 21], and HS (Hestenes–Stiefel) [6] conjugate gradient method, respectively. Among these, the PRP method is considered the best in practical computation. In order to guarantee the global convergence, we choose $d_{q^k}^k$ to satisfy the sufficient descent condition:

$$(g_{q^k}^k)^T d_{q^k}^k \leq -c \|g_{q^k}^k\|^2, \tag{6}$$

where $c > 0$ is a constant. There are several approaches to find the step-length. Among them, the exact line search [46, 47] is time consuming and sometimes difficult to carry out. Therefore, the researchers adopt the approaches of some inexact line search techniques such as Wolfe line search [48], Goldstein line search [49], or Armijo line search with backtracking [50]. The most used line search conditions for determining the step-length are the so-called standard Wolfe line search conditions:

$$f(x^k + \alpha_k d_{q^k}^k) \leq f(x^k) + \delta \alpha_k (g_{q^k}^k)^T d_{q^k}^k \tag{7}$$

and

$$g_{q^k}^k(x^k + \alpha_k d_{q^k}^k)^T d_{q^k}^k \geq \sigma (g_{q^k}^k)^T d_{q^k}^k, \tag{8}$$

where $0 < \delta < \sigma < 1$. The first condition (7) is called the Armijo condition, which ensures a sufficient reduction of the objective function value, while the second condition (8) is called the curvature condition, which ensures nonacceptance of short step-length. To investigate the global convergence property of the PRP method, a modified Armijo line search method was proposed [51]. For given constants $\mu > 0, \delta, \rho \in (0, 1)$, the line search aims to find

$$\alpha_k = \max \left\{ \rho^j \frac{\mu |(g_{q^k}^k)^T d_{q^k}^k|}{\|d_{q^k}^k\|^2} : j = 0, 1, \dots \right\}$$

such that (2) and (4) satisfy

$$f(x^{k+1}) \leq f(x^k) - \delta \alpha_k^2 \|d_{q^k}^k\|^2, \tag{9}$$

and

$$-C_1 \|g_{q^{k+1}}(x^{k+1})\|^2 \leq (g_{q^{k+1}}(x^{k+1}))^T d_{q^{k+1}}^{k+1} \leq -C_2 \|g_{q^{k+1}}(x^{k+1})\|^2,$$

where $0 < C_2 < 1 < C_1$ are constants. Accordingly, since $\{f(x^k)\}_{k \geq 0}$ is a nonincreasing sequence, we have

$$\sum_{k=0}^{\infty} \alpha_k^2 \|d_{q^k}^k\|^2 < \infty.$$

Equivalently,

$$\lim_{k \rightarrow \infty} \alpha_k \|d_{q^k}^k\| = 0. \tag{10}$$

It is worth mentioning that a step-length computed by the standard Wolfe line search conditions (7)–(8) may not be sufficiently close to a minimizer of (P) . Instead, the strong Wolfe line search conditions can be used, which consist of (7) and, instead of (8), the following strengthened version:

$$|g_{q^k}^k(x^k + \alpha_k d_{q^k}^k)^T d_{q^k}^k| \leq -\sigma (g_{q^k}^k)^T d_{q^k}^k \tag{11}$$

is used. From (11), we see that if $\sigma \rightarrow \infty$, then the step-length satisfying (7) and (11) tends to be the optimal step-length [2]. Note that appropriate choices for a starting point have a positive effect on computational cost and convergence speed of the algorithm. The modified PRP conjugate gradient-like method introduced by [52] is presented in the context of q -calculus as:

$$d_{q^k}^k = \begin{cases} -g_{q^k}^k, & k = 0, \\ -g_{q^k}^k + \beta_k^{q\text{-PRP}} d_{q^{k-1}}^{k-1} - \theta^k (g_{q^k}^k - g_{q^{k-1}}^{k-1}), & k > 0. \end{cases} \tag{12}$$

With the q -gradient, we can have a modification of [52] by taking

$$\theta^k = \frac{(g_{q^k}^k)^T d_{q^{k-1}}^{k-1}}{\|g_{q^{k-1}}^{k-1}\|^2}. \tag{13}$$

From (12) and (13) for $k \geq 1$, we obtain

$$d_{q^k}^k = -g_{q^k}^k + \frac{(g_{q^k}^k)^T (g_{q^k}^k - g_{q^{k-1}}^{k-1})}{\|g_{q^{k-1}}^{k-1}\|^2} d_{q^{k-1}}^{k-1} - \frac{(g_{q^k}^k)^T d_{q^{k-1}}^{k-1}}{\|g_{q^{k-1}}^{k-1}\|^2} (g_{q^k}^k - g_{q^{k-1}}^{k-1}),$$

that is,

$$(d_{q^k}^k)^T g_{q^k}^k = -\|g_{q^k}^k\|^2. \tag{14}$$

This implies that $d_{q^k}^k$ provides a q -descent direction of the objective function at x^k . It is worth mentioning that if exact line search [53] is used to compute the step-length α_k , then $\theta^k = 0$, and $q^k \rightarrow (1, 1, \dots, 1)^T$ for $k \rightarrow \infty$. Then, finally, the q -PRP method reduces to the classical PRP method.

The number of steps taken by the algorithm to a large extent determines the number of iterations which always differs from one problem to another. Thus, we present the following Algorithm 1 to solve the problem (P).

4 Global convergence

In this section, we prove the global convergence of Algorithm 1 under the following assumptions.

Assumption 4.1 The level set

$$\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\},$$

is bounded, where x^0 is a starting point.

Assumption 4.2 In some neighborhood N of Ω , f has a continuous q -derivative and there exists a constant $L > 0$ such that

$$\|g_q(x) - g_q(y)\| \leq L\|x - y\|, \tag{15}$$

for $x, y \in N$.

Algorithm 1 q -PRP conjugate gradient algorithm

Input: Given a starting point $x^0 \in \mathbb{R}^n$, $0 < \epsilon < 0$.

- 1: Set $d_{q^0}^0 = -g_{q^0}^0$.
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: **if** $\|g_{q^k}^k\| < \epsilon$ **then**
- 4: Stop.
- 5: **else**
- 6: Compute $\beta_k^{q\text{-PRP}}$ using (5).
- 7: Find $d_{q^k}^k$ using (12)
- 8: **end if**
- 9: Determine α_k either by (7)–(8) or (7) and (11).
- 10: Set $x^{k+1} = x^k + \alpha_k d_{q^k}^k$.
- 11: **for** $i = 1, \dots, n$ **do**
- 12: Set $q_i^{k+1} = 1 - \frac{q_i^k}{(k+1)^2}$.
- 13: **end for**
- 14: **end for**

Output: Optimal point $x^* = x^k$.

Since $\{f(x)\}$ is nonincreasing, it is clear that the sequence $\{x^k\}$ generated by Algorithm 1 is contained in Ω . From Assumptions 4.1 and 4.2, there is a constant $\eta > 0$ such that

$$\|g_{q^k}(x)\| \leq \eta, \tag{16}$$

for each $x \in \Omega$. Based on Assumption 4.1, there exists a positive constant \mathcal{B} such that $\|x\| \leq \mathcal{B}$, for all $x \in \Omega$. Without any specification, let $\{x^k\}$ and $\{d_{q^k}^k\}$ be the iterative sequence and q -descent direction sequence generated by Algorithm 1. To this point, we present the following lemma.

Lemma 4.1 *If there exists a constant $\epsilon > 0$, and $\{q^k\}$ generated by (1) is such that*

$$\|g_{q^k}^k\| \geq \epsilon, \tag{17}$$

for all k , then there exists a constant $\mathcal{M} > 0$ such that the q -descent direction satisfies

$$\|d_{q^k}^k\| \leq \mathcal{M}, \tag{18}$$

for all k .

Proof From (12) and (16) for $k \geq 1$, we obtain

$$d_{q^k}^k = -g_{q^k}^k + \frac{(g_{q^k}^k)^T (g_{q^k}^k - g_{q^{k-1}}^{k-1})}{\|g_{q^{k-1}}^{k-1}\|^2} d_{q^{k-1}}^{k-1} - \frac{(g_{q^k}^k)^T d_{q^{k-1}}^{k-1}}{\|g_{q^{k-1}}^{k-1}\|^2} (g_{q^k}^k - g_{q^{k-1}}^{k-1}).$$

Taking the norm of both sides of the above equation and using (16), we get

$$\|d_{q^k}^k\| \leq \eta + 2\eta \frac{\|g_{q^k}^k - g_{q^{k-1}}^{k-1}\| \|d_{q^{k-1}}^{k-1}\|}{\|g_{q^{k-1}}^{k-1}\|^2}.$$

From Assumption 4.2 and (17), we have

$$\|d_{q^k}^k\| \leq \eta + 2\eta \frac{L\alpha_{k-1}\|d_{q^{k-1}}^{k-1}\|}{\epsilon^2} \|d_{q^{k-1}}^{k-1}\|. \tag{19}$$

From (10), $\alpha_{k-1}d_{q^{k-1}}^{k-1} \rightarrow 0$ and since $\{q^k\}$ approaches $(1, \dots, 1)^T$ as $k \rightarrow \infty$, there exist a constant $s \in (0, 1)$ and an integer k_0 such that the following inequality holds for all $k \geq k_0$:

$$2\eta \frac{L\alpha_{k-1}}{\epsilon^2} \|d_{q^{k-1}}^{k-1}\| \leq s.$$

From (19), we get for any $k > k_0$,

$$\begin{aligned} \|d_{q^k}^k\| &\leq \eta + s\|d_{q^{k-1}}^{k-1}\| \\ &\leq \eta(1 + s) + s^2\|d_{q^{k-2}}^{k-2}\| \\ &\quad \vdots \\ &\leq \eta(1 + s + s^2 + \dots + s^{k-k_0-1}) + s^{k-k_0}\|d_{q^{k_0}}^{k_0}\|. \end{aligned}$$

For k sufficiently large with $s \in (0, 1)$, the second term of the above inequality can satisfy

$$s^{k-k_0}\|d_{q^{k_0}}^{k_0}\| < \|d_{q^{k_0}}^{k_0}\|.$$

Thus, we get

$$\|d_{q^k}^k\| < \frac{\eta}{1-s} + \|d_{q^{k_0}}^{k_0}\|.$$

Choosing

$$\mathcal{M} = \max \left\{ \|d_{q^1}^1\|, \|d_{q^2}^2\|, \dots, \|d_{q^{k_0}}^{k_0}\|, \frac{\eta}{1-s} + \|d_{q^{k_0}}^{k_0}\| \right\},$$

thus we get (18). □

We now present that the modified q -PRP method with modified Armijo-type line search introduced by [51] due to the q -gradient is globally convergent.

Theorem 4.2 *Assume that Assumptions 4.1 and 4.2 hold, then Algorithm 1 generates an infinite sequence $\{x^k\}$ such that*

$$\lim_{k \rightarrow \infty} \|g_{q^k}^k\| = 0. \tag{20}$$

Proof For the sake of obtaining a contradiction, we suppose that the given conclusion is not true. Then, there exists a constant $\epsilon > 0$ such that

$$\|g_{q^k}^k\| \geq \epsilon, \tag{21}$$

for all k . If $\liminf_{k \rightarrow \infty} \alpha_k > 0$, then from (10) and (14), we get

$$\liminf_{k \rightarrow \infty} \|g_{q^k}^k\| = 0.$$

This contradicts the assumption (22). Suppose that $\liminf_{k \rightarrow \infty} \alpha_k = 0$, that is, there is an infinite index set \mathcal{K} such that

$$\lim_{\substack{k \rightarrow \infty, \\ k \in \mathcal{K}}} \alpha_k = 0.$$

Suppose that step-9 of Algorithm 1 utilizes (9) to generate the step-length. When $k \in \mathcal{K}$ is sufficiently large, and $\rho^{-1}\alpha_k$ for $\rho \in (0, 1)$ [52] does not satisfy (9), then we must have

$$f(x^k + \rho^{-1}\alpha_k d_{q^k}^k) - f(x^k) > -\delta \rho^{-2} \alpha_k^2 \|d_{q^k}^k\|^2. \tag{22}$$

From the q -mean value theorem, there is $\gamma_k \in (0, 1)$ such that

$$f(x^k + \rho^{-1}\alpha_k d_{q^k}^k) - f(x^k) = \rho^{-1}\alpha_k g_{q^k}(x^k + \gamma_k \rho^{-1}\alpha_k d_{q^k}^k)^T d_{q^k}^k,$$

that is,

$$\begin{aligned} f(x^k + \rho^{-1}\alpha_k d_{q^k}^k) - f(x^k) &= \rho^{-1}\alpha_k (g_{q^k}^k)^T d_{q^k}^k \\ &\quad + \rho^{-1}\alpha_k (g_{q^k}(x^k + \gamma_k \rho^{-1}\alpha_k d_{q^k}^k) - g_{q^k}(x^k))^T d_{q^k}^k. \end{aligned}$$

From Lemma 4.1 and Assumption 4.2, we have

$$f(x^k + \rho^{-1}\alpha_k d_{q^k}^k) - f(x^k) \leq \rho^{-1}\alpha_k (g_{q^k}^k)^T d_{q^k}^k + L \rho^{-2} \alpha_k^2 \|d_{q^k}^k\|^2, \tag{23}$$

where $L > 0$. From (22) and (23),

$$-\delta \rho^{-2} \alpha_k^2 \|d_{q^k}^k\|^2 \leq \rho^{-1}\alpha_k (g_{q^k}^k)^T d_{q^k}^k + L \rho^{-2} \alpha_k^2 \|d_{q^k}^k\|^2.$$

Using (14), we get

$$\|g_{q^k}^k\|^2 \leq \alpha_k (\delta + L) \rho^{-1} \|d_{q^k}^k\|^2.$$

Since $\{d_{q^k}^k\}$ is bounded and $\lim_{k \in \mathcal{K}, k \rightarrow \infty} \alpha_k = 0$,

$$\lim_{\substack{k \rightarrow \infty, \\ k \in \mathcal{K}}} \|g_{q^k}^k\| = 0.$$

This gives a contradiction. The proof is complete. □

The following important result introduced by Zoutendijk [54] can be expressed in the sense of q -calculus as follows:

Lemma 4.3 *Suppose that Assumptions 4.1 and 4.2 hold. Consider the iteration methods (2) and (4), where $d_{q^k}^k$ satisfies (3) and α_k is obtained by standard Wolfe line search conditions (7)–(8) and strong Wolfe line search conditions (7) and (11). Then,*

$$\sum_{k=0}^{\infty} \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} < +\infty. \tag{24}$$

We now present the convergence analysis of Algorithm 1 with standard Wolfe conditions, which is a modification of [55, 56] in the sense of q -calculus. In this case, the step-lengths are bounded below by a positive constant.

Theorem 4.4 *Assume that the line search fulfills the standard Wolfe conditions (7)–(8). If there exists a positive constant $\alpha_0 \in (0, 1]$ such that $\alpha_k \geq \alpha_0$ for all $k \geq 0$, then*

$$\lim_{k \rightarrow \infty} \|g_{q^k}^k\| = 0. \tag{25}$$

Proof From (3) and the first Wolfe condition (7), we have

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \delta \alpha_k (g_{q^k}^k)^T d_{q^k}^k \\ &\leq f(x^k) \leq f(x^{k-1}) \leq \dots \leq f(x^0). \end{aligned}$$

This means that the sequence $\{f(x^k)\}_{k \geq 0}$ is bounded. From the second standard Wolfe condition (8) and Assumption 4.2, we get

$$\begin{aligned} -(1 - \sigma)(g_{q^k}^k)^T d_{q^k}^k &\leq (g_{q^{k+1}}^{k+1} - g_{q^k}^k)^T d_{q^k}^k \\ &\leq \|g_{q^{k+1}}^{k+1} - g_{q^k}^k\| \|d_{q^k}^k\| \leq \alpha_k L \|d_{q^k}^k\|^2, \end{aligned}$$

that is,

$$-\frac{(1 - \sigma)(g_{q^k}^k)^T d_{q^k}^k}{L \|d_{q^k}^k\|^2} \leq \alpha_k.$$

Post-multiplying both sides by $\delta (g_{q^k}^k)^T d_{q^k}^k$, we get

$$-\frac{(1 - \sigma)\delta((g_{q^k}^k)^T d_{q^k}^k)^2}{L \|d_{q^k}^k\|^2} \leq \alpha_k \delta (g_{q^k}^k)^T d_{q^k}^k.$$

From the first standard Wolfe condition (7), we have

$$\frac{\delta(1 - \sigma)}{L} \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} \leq f(x^k) - f(x^{k+1}).$$

Since $\{f(x^k)\}_{k \geq 0}$ is bounded,

$$\begin{aligned} \frac{\delta(1-\sigma)}{L} \sum_{k=0}^{\infty} \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} &\leq f(x^0) - f(x^1) + (f(x^1) - f(x^2)) + \dots \\ &= f(x^0) - \lim_{k \rightarrow \infty} f(x^k) < +\infty. \end{aligned}$$

Thus, Zoutendijk condition (24) holds, that is,

$$\sum_{k=0}^{\infty} \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} < +\infty. \tag{26}$$

From Assumption 4.1, there exists a constant \mathcal{B} such that

$$\|p^k\| = \|\alpha_k d_{q^k}^k\| \leq \mathcal{B}.$$

Since $\alpha_k \geq \alpha_0$, we get

$$\|d_{q^k}^k\| \leq \frac{\mathcal{B}}{\alpha_0}.$$

This, together with (6) and (26), leads to (25). □

We present the following theorem which is a modification of that in [57] using the q -gradient for q -PRP method with strong Wolfe conditions.

Theorem 4.5 *Suppose that x^0 is a starting point and Assumptions 4.1 and 4.2 hold. Let $\{x^k\}$ be the sequence generated by Algorithm 1. If $\beta_k^{q\text{-PRP}}$ is such that the step-length α_k satisfies the strong Wolfe conditions (7) and (11), then either*

$$\lim_{k \rightarrow \infty} \|g_{q^k}^k\| = 0 \quad \text{or} \quad \sum_{k=1}^{\infty} \frac{\|g_{q^k}^k\|^4}{\|d_{q^k}^k\|^2} < \infty. \tag{27}$$

Proof From (4), for all $k \geq 1$, we have

$$d_{q^k}^k + g_{q^k}^k = \beta_k^{q\text{-PRP}} d_{q^{k-1}}^{k-1}.$$

Squaring both sides of the above equation, we get

$$\|d_{q^k}^k\|^2 + \|g_{q^k}^k\|^2 + 2(g_{q^k}^k)^T d_{q^k}^k = (\beta_k^{q\text{-PRP}})^2 \|d_{q^{k-1}}^{k-1}\|^2.$$

Since $d_{q^k}^k$ satisfies the descent condition $(g_{q^k}^k)^T d_{q^k}^k < 0$,

$$\|d_{q^k}^k\|^2 \geq -\|g_{q^k}^k\|^2 + (\beta_k^{q\text{-PRP}})^2 \|d_{q^{k-1}}^{k-1}\|^2. \tag{28}$$

Pre-multiplying (4) for $k \geq 1$ by $g_{q^k}^k$, we get

$$(g_{q^k}^k)^T d_{q^k}^k = -\|g_{q^k}^k\|^2 + \beta_k^{q\text{-PRP}} (g_{q^k}^k)^T d_{q^k}^k. \tag{29}$$

From (29) and the second strong Wolfe condition (11), one obtains

$$|g_{q^k}^k d_{q^k}^k| + \sigma |\beta_k^{q-PRP}| |(g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1}| \geq \|g_{q^k}^k\|^2. \tag{30}$$

From the inequality

$$(a + \sigma b)^2 \leq (1 + \sigma^2)(a^2 + b^2),$$

for all $a, b, \sigma \geq 0$, with

$$a = |(g_{q^k}^k)^T d_{q^k}^k|$$

and

$$b = |\beta_k^{q-PRP}| |(g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1}|,$$

we can express (30) as

$$((g_{q^k}^k)^T d_{q^k}^k)^2 + (\beta_k^{q-PRP})^2 ((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2 \geq c_1 \|g_{q^k}^k\|^4, \tag{31}$$

where $c_1 = \frac{1}{(1+\sigma^2)}$. Note that

$$\begin{aligned} & \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} + \frac{((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2}{\|d_{q^{k-1}}^{k-1}\|^2} \\ &= \frac{1}{\|d_{q^k}^k\|^2} \left[((g_{q^k}^k)^T d_{q^k}^k)^2 + \frac{\|d_{q^k}^k\|^2}{\|d_{q^{k-1}}^{k-1}\|^2} ((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2 \right]. \end{aligned}$$

From (28) one gets

$$\begin{aligned} & \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} + \frac{((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2}{\|d_{q^{k-1}}^{k-1}\|^2} \\ & \geq \frac{1}{\|d_{q^k}^k\|^2} \left[((g_{q^k}^k)^T d_{q^k}^k)^2 + (\beta_k^{q-PRP})^2 ((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2 \right. \\ & \quad \left. - \frac{(g_{q^{k-1}}^{k-1} d_{q^{k-1}}^{k-1})^2}{\|d_{q^{k-1}}^{k-1}\|^2} \|g_{q^k}^k\|^2 \right]. \end{aligned}$$

Using (31), we obtain

$$\begin{aligned} & \frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} + \frac{((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2}{\|d_{q^{k-1}}^{k-1}\|^2} \\ & \geq \frac{1}{\|d_{q^k}^k\|^2} \left[c_1 \|g_{q^k}^k\|^4 - \frac{(g_{q^k}^{k-1} d_{q^{k-1}}^{k-1})^2}{\|d_{q^{k-1}}^{k-1}\|^2} \|g_{q^k}^k\|^2 \right]. \end{aligned} \tag{32}$$

If (27) is not true, then from the Zoutendijk condition (24) with (32) we obtain the following inequality:

$$\frac{((g_{q^k}^k)^T d_{q^k}^k)^2}{\|d_{q^k}^k\|^2} + \frac{((g_{q^{k-1}}^{k-1})^T d_{q^{k-1}}^{k-1})^2}{\|d_{q^{k-1}}^{k-1}\|^2} \geq \frac{c}{2} \frac{\|g_{q^k}^k\|^4}{\|d_{q^k}^k\|^2} \tag{33}$$

which holds for k sufficiently large with q^k approaching $(1, \dots, 1)^T$. From (32) and (33), one immediately recovers (30). □

The following lemma immediately follows from the above convergence theorem.

Lemma 4.6 *Suppose that Assumptions 4.1 and 4.2 hold and, from Algorithm 1, the step-length is determined using strong Wolfe conditions. If*

$$\sum_{k=1}^{\infty} \frac{\|g_{q^k}^k\|^r}{\|d_{q^k}^k\|^2} = +\infty, \tag{34}$$

for any $r \in [0, 4]$, then the method converges in the sense that (27) is true.

Proof If (27) is not true, then from Theorem 4.5, it follows that

$$\sum_{k=1}^{\infty} \frac{\|g_{q^k}^k\|^4}{\|d_{q^k}^k\|^2} < +\infty. \tag{35}$$

Because $\|g_{q^k}^k\|$ is bounded away from zero and $r \in [0, 4]$, it is easy to see that (35) contradicts (34). Therefore, the lemma is true. □

The above inequality shows that if a conjugate gradient method fails to converge, then the length of the search direction will diverge to infinity. Observe that in the above developments, the sufficient descent condition is assumed. This lemma is very useful for proving the global convergence of some conjugate gradient methods without assuming the sufficient descent condition.

5 Numerical illustration

In this section, we investigate the computational efficiency of Algorithm 1 using standard Wolfe conditions (7) and (8), and strong Wolfe conditions (7) and (11), respectively, in contrast to the classical PRP method under the same two conditions.

All codes of Algorithm 1 and classical PRP method are written in R version 3.6.1 installed on a laptop having Intel(R) Core(TM) i3-4005U, 1.70 GHz CPU processor and 4 GB RAM. The iteration was set to terminate if it exceeded 1000 or the gradient of a function was within 10^{-6} .

Example 5.1 Consider a function (Mishra 6) [58] $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ given by

$$f(x) = -\log(\sin^2((\cos x_1 + \cos x_2)^2)) - \cos^2((\sin x_1 + \sin x_2)^2) + 0.1[(x_1 - 1)^2 + (x_2 - 1)^2].$$

We find the q -gradient of the above function at the point

$$x = (2.88, 1.82)^T,$$

with the starting parameter value

$$q^1 = (0.32, 0.32)^T.$$

We run the q -gradient algorithm [39] for $k = 1, \dots, 50$ iterations so that q^{50} approaches

$$(0.999607921, 0.999607921)^T,$$

and in the 50th iteration we get the q -gradient

$$g_{q^{50}}^{50} = (-0.41348771, -0.63704079)^T.$$

The complete computational details are given in Table 1 which is depicted graphically through Fig. 1. Note that Fig. 2 provides the three-dimensional view of Mishra 6 test function.

Example 5.2 Consider a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ given by

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2.$$

The Rosenbrock function, also called Rosenbrock's valley or banana function, is a non-convex, unimodal, and nonseparable function. Finding its global minimum numerically is difficult. It has only one global minimizer located at the point

$$x^* = (1, 1)^T,$$

with the search range $[-100, 100]$ for x_1 and x_2 . For performing the experiment, we first generated 37 different starting points from the interval $[-5, 5]$ for the above Rosenbrock function. The numerical results are shown in Table 2 for Algorithm 1 and Table 3 for the classical PRP Algorithm. From these tables, we observe that the number of iterations (NI) is smaller in the case of Algorithm 1 in comparison to the classical PRP method. The meanings of columns of both tables are well-defined. Figure 3 shows the results of comparisons in the number of iterations.

Example 5.3 Consider the following Rastrigin function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, that is,

$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1) + \cos 2\pi x_2.$$

The Rastrigin test function is a nonconvex, multimodal, and separable function, which has several local minimizers arranged in a regular lattice, but it has only one global minimizer located at the point

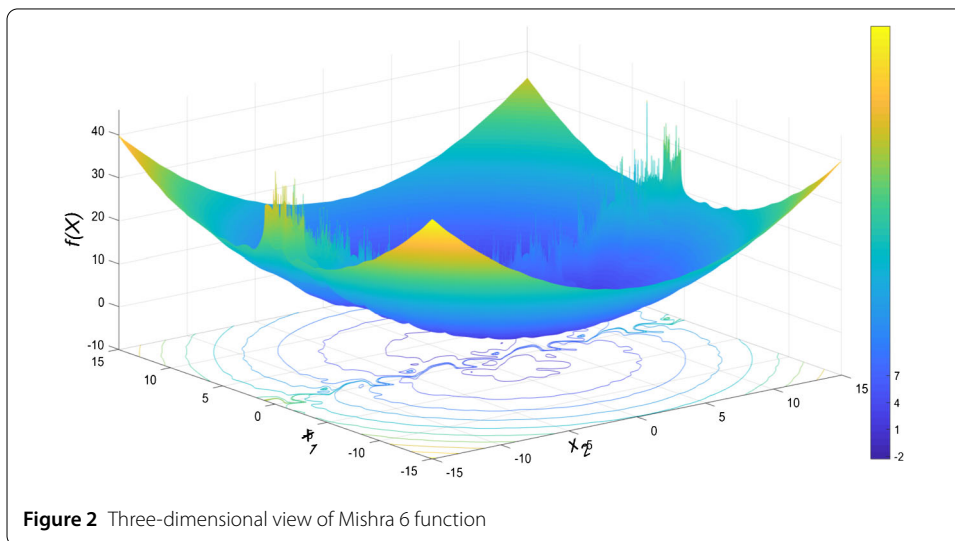
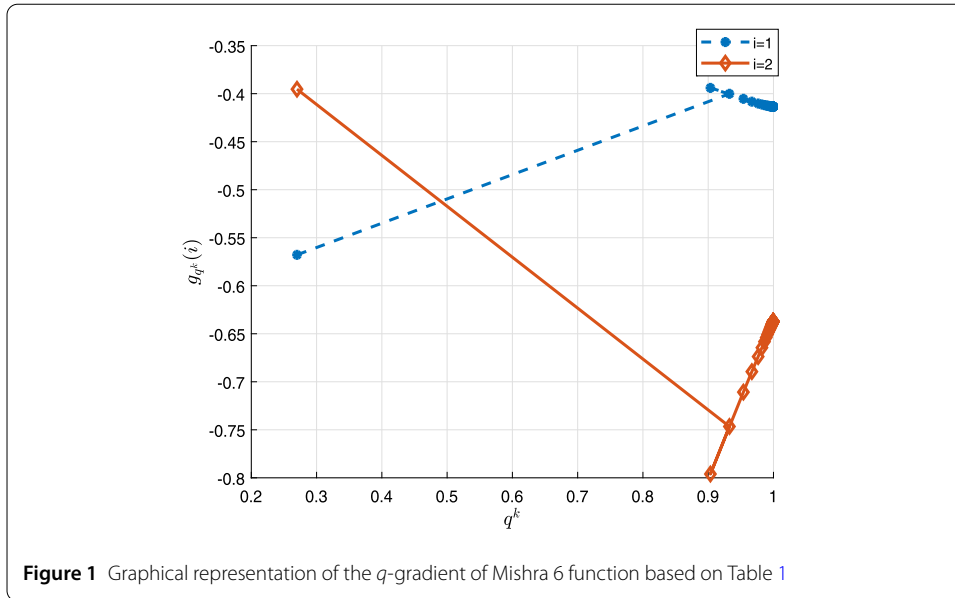
$$x^* = (0, 0)^T.$$

Table 1 q -Gradient of Example 5.1

k	q^k	$x = (2.88, 1.82)^T$			
		$f(x)$	$f(q^k x)$	$g_{q^k}(1)$	$g_{q^k}(2)$
1	$(0.32, 0.32)^T$	0.430480518	1.076549247	-0.74986965	-0.52203356
2	$(0.92, 0.92)^T$	0.430480518	0.542260247	-0.39724386	-0.76771792
3	$(0.905955556, 0.905955556)^T$	0.430480518	0.565980670	-0.39444591	-0.79165371
4	$(0.953527001, 0.953527001)^T$	0.430480518	0.490652335	-0.40526797	-0.71141177
5	$(0.966933215, 0.966933215)^T$	0.430480518	0.471966971	-0.40833281	-0.68935502
6	$(0.976520984, 0.976520984)^T$	0.430480518	0.459272296	-0.41026389	-0.67377859
7	$(0.982303255, 0.982303255)^T$	0.430480518	0.451881863	-0.41128298	-0.66447140
8	$(0.986210799, 0.986210799)^T$	0.430480518	0.446999451	-0.41189872	-0.65822064
9	$(0.988952271, 0.988952271)^T$	0.430480518	0.443627482	-0.41229233	-0.65385448
10	$(0.990951471, 0.990951471)^T$	0.430480518	0.441196138	-0.41255827	-0.65068070
11	$(0.990951471, 0.992453623)^T$	0.430480518	0.439384557	-0.41274593	-0.64830175
12	$(0.993610749, 0.993610749)^T$	0.430480518	0.437997983	-0.41288317	-0.64647262
13	$(0.994520907, 0.994520907)^T$	0.430480518	0.436912784	-0.41298657	-0.64503597
14	$(0.995249658, 0.995249658)^T$	0.430480518	0.436047321	-0.41306642	-0.64388700
15	$(0.995842177, 0.995842177)^T$	0.430480518	0.435345901	-0.41312939	-0.64295371
16	$(0.996330408, 0.996330408)^T$	0.430480518	0.434769453	-0.41317996	-0.64218527
17	$(0.996737456, 0.996737456)^T$	0.430480518	0.434289902	-0.41322119	-0.64154502
18	$(0.997080367, 0.997080367)^T$	0.430480518	0.433886651	-0.41325527	-0.64100595
19	$(0.997371937, 0.997371937)^T$	0.430480518	0.433544306	-0.41328378	-0.64054780
20	$(0.997621924, 0.997621924)^T$	0.430480518	0.433251172	-0.41330787	-0.64015514
21	$(0.997837873, 0.997837873)^T$	0.430480518	0.432998239	-0.41332843	-0.63981606
22	$(0.998025695, 0.998025695)^T$	0.430480518	0.432778468	-0.41334611	-0.63952123
23	$(0.998190068, 0.998190068)^T$	0.430480518	0.432586299	-0.41336143	-0.63926328
24	$(0.998334740, 0.998334740)^T$	0.430480518	0.432417292	-0.41337480	-0.63903629
25	$(0.998462736, 0.998462736)^T$	0.430480518	0.432267864	-0.41338653	-0.63883551
26	$(0.998576525, 0.998576525)^T$	0.430480518	0.432135102	-0.41339690	-0.63865705
27	$(0.998678133, 0.998678133)^T$	0.430480518	0.432016614	-0.41340609	-0.63849771
28	$(0.998769240, 0.998769240)^T$	0.430480518	0.431910422	-0.41341429	-0.63835486
29	$(0.998851242, 0.998851242)^T$	0.430480518	0.431814882	-0.41342163	-0.63822631
30	$(0.998925315, 0.998925315)^T$	0.430480518	0.431728614	-0.41342823	-0.63811020
31	$(0.998992449, 0.998992449)^T$	0.430480518	0.431650455	-0.41343419	-0.63800497
32	$(0.999053483, 0.999053483)^T$	0.430480518	0.431579419	-0.41343959	-0.63790932
33	$(0.999109135, 0.999109135)^T$	0.430480518	0.431514666	-0.41344449	-0.63782211
34	$(0.999160020, 0.999160020)^T$	0.430480518	0.431455475	-0.41344896	-0.63774237
35	$(0.999206667, 0.999206667)^T$	0.430480518	0.431401227	-0.41345305	-0.63766929
36	$(0.999249534, 0.999249534)^T$	0.430480518	0.431351386	-0.41345679	-0.63760212
37	$(0.999289018, 0.999289018)^T$	0.430480518	0.431305487	-0.41346023	-0.63754027
38	$(0.999325466, 0.999325466)^T$	0.430480518	0.431263125	-0.41346340	-0.63748317
39	$(0.999359181, 0.999359181)^T$	0.430480518	0.431223946	-0.41346633	-0.63743035
40	$(0.999390431, 0.999390431)^T$	0.430480518	0.431187638	-0.41346903	-0.63738140
41	$(0.999419450, 0.999419450)^T$	0.430480518	0.431153928	-0.41347154	-0.63733595
42	$(0.999446445, 0.999446445)^T$	0.430480518	0.431122572	-0.41347387	-0.63729367
43	$(0.999471599, 0.999471599)^T$	0.430480518	0.431093358	-0.41347603	-0.63725427
44	$(0.999495078, 0.999495078)^T$	0.430480518	0.431066094	-0.41347805	-0.63721750
45	$(0.999517026, 0.999517026)^T$	0.430480518	0.431040610	-0.41347994	-0.63718313
46	$(0.999537573, 0.999537573)^T$	0.430480518	0.431016755	-0.41348170	-0.63715095
47	$(0.999556836, 0.999556836)^T$	0.430480518	0.430994392	-0.41348335	-0.63712078
48	$(0.999574921, 0.999574921)^T$	0.430480518	0.430973400	-0.41348490	-0.63709246
49	$(0.999591921, 0.999591921)^T$	0.430480518	0.430953669	-0.41348635	-0.63706584
50	$(0.999607921, 0.999607921)^T$	0.430480518	0.430935100	-0.41348771	-0.63704079

The search range for the Rastrigin function is $[-5.12, 5.12]$ in both x_1 and x_2 . This function poses a fairly difficult problem due to its large search space and its large number of local minima. With a chosen starting point $(0.2, 0.2)^T$, we minimize this function through Algorithm 1 using strong Wolfe conditions. Note that q -PRP terminates in 5 iterations as

$$g_q^5 = (0.0001900418, 0.0001900418)^T,$$



with step-length $\alpha_5 = 0.252244535$. Thus, we get the global minimizer

$$x^* = x^5 = (-2.05643E - 08, -2.05643E - 08)^T,$$

with minimum function value

$$f(x^*) = 1.669775E - 13,$$

while running the classical PRP method using strong Wolfe conditions from the same chosen starting point, it terminates in 5 iterations with

$$g_{q^5}^5 = (1.776357E - 10, 2.66453E - 10)^T,$$

Table 2 Numerical results of Example 5.2 using Algorithm 1

Sl. no.	Starting point		q-PRP algorithm		x*	f(x*)	NI
	Step-length	q-gradient	Step-length	q-gradient			
1	(-4.1448E + 00, -2.9324E + 00) ^T	(1.4775E + 00, -7.0264E - 01) ^T	2.3367E - 02	(1.4775E + 00, -7.0264E - 01) ^T	(1.0415E + 00, 1.0814E + 00) ^T	2.9218E - 03	26
2	(-4.2752E + 00, -2.8376E + 00) ^T	(-4.9287E - 02, -1.8236E - 03) ^T	4.3899E - 05	(-4.9287E - 02, -1.8236E - 03) ^T	(1.0046E + 00, 1.0093E + 00) ^T	2.1377E - 05	23
3	(3.0101E + 00, -5.8300E - 02) ^T	(4.7749E - 02, -6.2513E - 02) ^T	7.9493E - 03	(4.7749E - 02, -6.2513E - 02) ^T	(9.9178E - 01, 9.8337E - 01) ^T	7.4429E - 05	19
4	(2.4348E + 00, -2.8090E + 00) ^T	(8.0440E - 01, -4.0639E - 01) ^T	2.6283E - 01	(8.0440E - 01, -4.0639E - 01) ^T	(1.0202E + 00, 1.0388E + 00) ^T	7.9812E - 04	25
5	(2.4944E + 00, 8.7790E - 01) ^T	(-4.9341E - 03, -4.9406E - 03) ^T	1.8736E - 42	(-4.9341E - 03, -4.9406E - 03) ^T	(1.0255E + 00, 1.0517E + 00) ^T	6.5168E - 04	18
6	(2.3033E + 00, 4.0372E + 00) ^T	(3.1725E - 03, -1.0430E - 03) ^T	1.5548E - 02	(3.1725E - 03, -1.0430E - 03) ^T	(1.0344E + 00, 1.0701E + 00) ^T	1.1861E - 03	20
7	(-1.1390E - 01, 3.9092E + 00) ^T	(-4.4790E - 02, -9.2646E - 04) ^T	3.9223E - 37	(-4.4790E - 02, -9.2646E - 04) ^T	(1.0081E + 00, 1.0163E + 00) ^T	6.5406E - 05	24
8	(-2.7857E + 00, 1.0782E + 00) ^T	(-7.2107E - 03, -1.6834E - 02) ^T	4.2172E - 38	(-7.2107E - 03, -1.6834E - 02) ^T	(1.0111E + 00, 1.0222E + 00) ^T	1.2234E - 04	18
9	(-2.6272E + 00, 1.9875E + 00) ^T	(9.6266E - 01, -4.4836E - 01) ^T	8.1062E - 01	(9.6266E - 01, -4.4836E - 01) ^T	(1.0471E + 00, 1.0941E + 00) ^T	2.6921E - 03	16
10	(-4.1150E - 01, -3.0219E + 00) ^T	(-2.7351E - 02, -9.0127E - 03) ^T	8.0059E - 39	(-2.7351E - 02, -9.0127E - 03) ^T	(1.0087E + 00, 1.0175E + 00) ^T	7.5592E - 05	16
11	(4.6309E + 00, -4.6946E + 00) ^T	(-2.4447E - 02, -3.1051E - 02) ^T	7.1986E - 04	(-2.4447E - 02, -3.1051E - 02) ^T	(9.8658E - 01, 9.7324E - 01) ^T	1.8118E - 04	29
12	(4.6810E - 01, 2.4407E + 00) ^T	(-1.8246E + 00, 8.8302E - 01) ^T	4.3738E - 02	(-1.8246E + 00, 8.8302E - 01) ^T	(1.0756E + 00, 1.1614E + 00) ^T	7.7209E - 03	12
13	(2.1140E - 01, 2.0000E - 04) ^T	(1.6658E - 01, -7.9139E - 02) ^T	1.0000E + 00	(1.6658E - 01, -7.9139E - 02) ^T	(1.0354E + 00, 1.0717E + 00) ^T	1.2627E - 03	19
14	(-2.6841E + 00, -2.0080E - 01) ^T	(-3.8200E - 02, -8.7571E - 03) ^T	4.3478E - 37	(-3.8200E - 02, -8.7571E - 03) ^T	(1.0031E + 00, 1.0061E + 00) ^T	9.3228E - 06	12
15	(-1.1100E - 01, 4.0472E + 00) ^T	(-1.3487E - 02, -2.8364E - 02) ^T	2.3756E - 03	(-1.3487E - 02, -2.8364E - 02) ^T	(9.9525E - 01, 9.9043E - 01) ^T	2.3413E - 05	19
16	(1.2406E + 00, 1.0987E + 00) ^T	(-3.8603E - 01, 1.9025E - 01) ^T	4.4170E - 03	(-3.8603E - 01, 1.9025E - 01) ^T	(1.0391E + 00, 1.0806E + 00) ^T	1.6267E - 03	6
17	(1.7914E + 00, 1.1767E + 00) ^T	(-1.2518E - 02, 4.0719E - 03) ^T	7.7842E - 42	(-1.2518E - 02, 4.0719E - 03) ^T	(1.0316E + 00, 1.0643E + 00) ^T	9.9856E - 04	13
18	(-1.0448E + 00, 3.5944E + 00) ^T	(-5.0676E - 01, 2.1352E - 01) ^T	1.4973E - 03	(-5.0676E - 01, 2.1352E - 01) ^T	(9.8677E - 01, 9.7483E - 01) ^T	2.9988E - 04	32
19	(-1.3256E + 00, 3.0549E + 00) ^T	(-3.8494E - 02, -1.5711E - 02) ^T	4.9450E - 04	(-3.8494E - 02, -1.5711E - 02) ^T	(9.9534E - 01, 9.9068E - 01) ^T	2.1754E - 05	26
20	(4.8798E + 00, 7.6720E - 01) ^T	(1.7832E + 00, -8.3048E - 01) ^T	2.7269E - 02	(1.7832E + 00, -8.3048E - 01) ^T	(1.0529E + 00, 1.1046E + 00) ^T	4.4787E - 03	25

Table 2 (Continued)

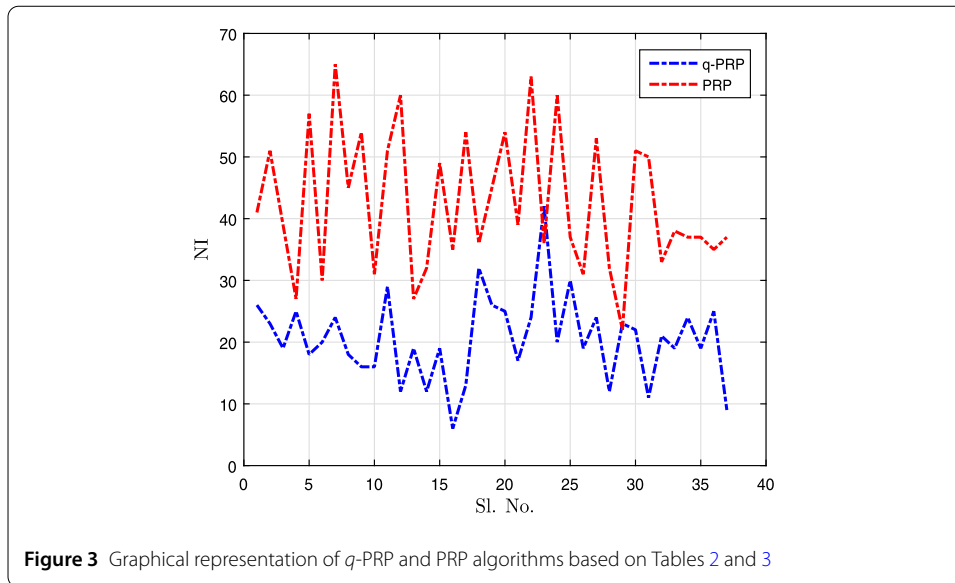
Sl. no.	Starting point	q-PPR algorithm	q-gradient	x^*	$f(x^*)$	NI
		Step-length				
21	$(-4.6226E+00, -3.1708E+00)^T$	1.0522E-43	$(-6.8375E-03, -2.4455E-03)^T$	$(1.0273E+00, 1.0554E+00)^T$	$7.4551E-04$	17
22	$(3.8517E+00, -2.6007E+00)^T$	5.8588E-04	$(-2.8237E-02, -2.4777E-02)^T$	$(9.9117E-01, 9.8235E-01)^T$	$7.8430E-05$	24
23	$(4.1329E+00, 3.8651E+00)^T$	7.9595E-04	$(-3.1541E-02, -2.1954E-02)^T$	$(9.9240E-01, 9.8481E-01)^T$	$5.8049E-05$	42
24	$(-6.4660E-01, -9.5770E-01)^T$	1.4184E-37	$(-2.6830E-02, -2.2030E-02)^T$	$(9.9485E-01, 9.8967E-01)^T$	$2.6847E-05$	20
25	$(-4.0129E+00, -1.0100E-01)^T$	2.2635E-43	$(-4.4286E-03, -4.6080E-04)^T$	$(1.0309E+00, 1.0628E+00)^T$	$9.5458E-04$	30
26	$(-2.3813E+00, -3.3207E+00)^T$	5.2263E-40	$(-1.9932E-02, -1.0626E-02)^T$	$(1.0110E+00, 1.0220E+00)^T$	$1.2022E-04$	19
27	$(-1.6464E+00, 4.7868E+00)^T$	1.4188E-03	$(-3.0516E-02, -1.4897E-02)^T$	$(1.0005E+00, 1.0011E+00)^T$	$3.4798E-07$	24
28	$(1.7973E+00, 2.1269E+00)^T$	5.9605E-03	$(1.1950E+00, -5.6006E-01)^T$	$(1.0465E+00, 1.0925E+00)^T$	$2.9181E-03$	12
29	$(-3.6345E+00, 4.7000E-03)^T$	2.1807E-03	$(-4.5867E-01, 2.2021E-01)^T$	$(1.0314E+00, 1.0650E+00)^T$	$1.1214E-03$	23
30	$(2.2123E+00, -2.8910E-01)^T$	8.0544E-41	$(-2.0036E-02, -4.1951E-03)^T$	$(1.0181E+00, 1.0365E+00)^T$	$3.2608E-04$	22
31	$(-8.9670E-01, 2.5902E+00)^T$	6.0000E-04	$(2.4092E-01, 2.4092E-01)^T$	$(1.0351E+00, 1.0727E+00)^T$	$1.3913E-03$	11
32	$(4.7941E+00, -1.7500E+00)^T$	2.3004E-39	$(-2.2169E-02, -1.1757E-02)^T$	$(1.0085E+00, 1.0171E+00)^T$	$7.2274E-05$	21
33	$(1.3976E+00, 4.5380E+00)^T$	3.1194E-01	$(1.1220E+00, -5.4273E-01)^T$	$(1.0338E+00, 1.0661E+00)^T$	$1.8503E-03$	19
34	$(3.6020E+00, 2.7511E+00)^T$	8.8939E-03	$(1.1476E-01, -1.0254E-01)^T$	$(9.8567E-01, 9.7108E-01)^T$	$2.2681E-04$	24
35	$(-1.0068E+00, 3.3042E+00)^T$	2.7821E-03	$(6.8507E-01, -3.1611E-01)^T$	$(1.0468E+00, 1.0943E+00)^T$	$2.4221E-03$	19
36	$(4.4101E+00, 7.4560E-01)^T$	6.3994E-04	$(-4.2013E-02, -7.5740E-03)^T$	$(1.0023E+00, 1.0046E+00)^T$	$5.1517E-06$	25
37	$(4.7164E+00, -8.0170E-01)^T$	1.4569E-36	$(-3.8348E-02, -9.5965E-03)^T$	$(1.0021E+00, 1.0041E+00)^T$	$4.2179E-06$	9

Table 3 Numerical results of Example 5.2 using classical PRP Algorithm

Sl. no.	PRP algorithm		Gradient	x*	f(x*)	NI
	Starting point	Step-length				
1	(-4.1448E+00, -2.9324E+00) ^T	1.0472E-03	(1.0287E-07, -8.3547E-09) ^T	(1.0000E+00, 1.0000E+00) ^T	5.3298E-16	41
2	(-4.2752E+00, -2.8376E+00) ^T	1.9542E-03	(3.1505E-07, -9.3225E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	6.3375E-13	51
3	(3.0101E+00, -5.8300E-02) ^T	1.0721E-03	(-2.7082E-07, -1.2131E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	7.6609E-14	39
4	(2.4348E+00, -2.8090E+00) ^T	1.1164E-03	(9.4822E-08, 2.0076E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	2.2560E-15	27
5	(2.4944E+00, 8.7790E-01) ^T	1.4495E-03	(-3.8184E-07, -8.9269E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	9.0135E-14	57
6	(2.3033E+00, 4.0372E+00) ^T	1.2433E-03	(-1.5568E-09, -5.8448E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	3.6719E-13	30
7	(-1.1390E-01, 3.9092E+00) ^T	1.5078E-03	(-2.0814E-07, 7.1389E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	3.4916E-13	65
8	(-2.7857E+00, 1.0782E+00) ^T	1.2141E-03	(-4.8623E-07, 3.7559E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	1.3003E-14	45
9	(-2.6272E+00, 1.9875E+00) ^T	1.2900E-03	(-8.2621E-07, 5.7665E-09) ^T	(1.0000E+00, 1.0000E+00) ^T	1.8262E-13	54
10	(-4.1150E-01, -3.0219E+00) ^T	1.0185E-03	(1.7739E-09, -2.8126E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	2.2335E-15	31
11	(4.6309E+00, -4.6946E+00) ^T	1.6958E-03	(1.0000E+00, 1.0000E+00) ^T	(7.0029E-07, 1.8475E-07) ^T	2.6520E-13	51
12	(4.6810E-01, 2.4407E+00) ^T	1.5815E-03	(-5.8657E-07, 5.9774E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	8.1809E-14	60
13	(2.1140E-01, 2.0000E-04) ^T	1.1507E-03	(-8.3793E-07, 7.7501E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	1.3067E-13	27
14	(-2.6841E+00, -2.0080E-01) ^T	1.1155E-03	(3.5193E-07, 1.1527E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	7.3605E-14	32
15	(-1.1100E-01, 4.0472E+00) ^T	1.2035E-03	(5.3046E-09, -3.8049E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	1.5864E-13	49
16	(1.2406E+00, 1.0987E+00) ^T	1.0736E-03	(-1.5669E-07, 3.9305E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	3.4898E-15	35
17	(1.7914E+00, 1.167E+00) ^T	1.1084E-03	(-4.2829E-07, -3.1031E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	7.0320E-14	54
18	(-1.0448E+00, 3.5944E+00) ^T	1.0011E-03	(3.3377E-08, 4.1644E-10) ^T	(1.0000E+00, 1.0000E+00) ^T	8.3811E-18	36
19	(-1.3256E+00, 3.0549E+00) ^T	1.3002E-03	(-3.4159E-07, 1.5132E-08) ^T	(1.0000E+00, 1.0000E+00) ^T	3.0857E-14	45
20	(4.8798E+00, 7.6720E-01) ^T	1.0964E-03	(1.0605E-07, 6.6228E-07) ^T	(1.0000E+00, 1.0000E+00) ^T	4.8455E-13	54

Table 3 (Continued)

Sl. no.	Starting point		PRP algorithm	Step-length	Gradient	x^*	$f(x^*)$	NI
21	$(-4.6226E + 00, -3.1708E + 00)^T$	$(-9.7930E - 07, -1.3147E - 07)^T$	1.1278E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	4.1108E - 13	39	
22	$(3.8517E + 00, -2.6007E + 00)^T$	$(2.6266E - 07, -3.7627E - 07)^T$	1.2356E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	7.0545E - 14	63	
23	$(4.1329E + 00, 3.8651E + 00)^T$	$(-1.1176E - 07, -2.4588E - 07)^T$	9.9963E - 04		$(1.0000E + 00, 1.0000E + 00)^T$	1.0368E - 13	36	
24	$(-6.4660E - 01, -9.5770E - 01)^T$	$(-2.7319E - 07, -1.3338E - 07)^T$	1.2494E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	8.4128E - 14	60	
25	$(-4.0129E + 00, -1.0100E - 01)^T$	$(5.1172E - 10, 8.1593E - 07)^T$	1.0451E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	6.3558E - 13	37	
26	$(-2.3813E + 00, -3.3207E + 00)^T$	$(1.7723E - 07, -3.3755E - 09)^T$	1.0864E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	4.2562E - 15	31	
27	$(-1.6464E + 00, 4.7868E + 00)^T$	$(2.7136E - 07, -7.2242E - 07)^T$	1.4132E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	3.6945E - 13	53	
28	$(1.7973E + 00, 2.1269E + 00)^T$	$(7.7808E - 09, -7.7013E - 08)^T$	1.0040E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	8.6866E - 15	32	
29	$(-3.6345E + 00, 4.7000E - 03)^T$	$(8.2682E - 07, 2.2167E - 07)^T$	1.0004E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	3.7845E - 13	22	
30	$(2.2123E + 00, -2.8910E - 01)^T$	$(3.6809E - 08, -7.3394E - 08)^T$	1.0979E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	5.6368E - 15	51	
31	$(-8.9670E - 01, 2.5902E + 00)^T$	$(-7.9028E - 07, 6.0042E - 07)^T$	1.3091E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	3.5229E - 14	50	
32	$(4.7941E + 00, -1.7500E + 00)^T$	$(-4.7745E - 08, 2.1303E - 07)^T$	1.0964E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	2.8729E - 14	33	
33	$(1.3976E + 00, 4.5380E + 00)^T$	$(2.2973E - 07, 1.9974E - 08)^T$	1.1476E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	1.3189E - 14	38	
34	$(3.6020E + 00, 2.7511E + 00)^T$	$(4.0635E - 07, -7.6268E - 08)^T$	1.0413E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	1.1443E - 14	37	
35	$(-1.0068E + 00, 3.3042E + 00)^T$	$(5.3450E - 07, 1.5490E - 07)^T$	1.2183E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	1.6179E - 13	37	
36	$(4.4101E + 00, 7.4560E - 01)^T$	$(-4.8905E - 08, 2.0217E - 07)^T$	1.0686E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	2.4978E - 14	35	
37	$(4.7164E + 00, -8.0170E - 01)^T$	$(-9.3419E - 07, -5.7970E - 08)^T$	1.2984E - 03		$(1.0000E + 00, 1.0000E + 00)^T$	2.9711E - 13	37	



$\alpha_5 = 0.002547382$, but fails to achieve the global minimizer as

$$x^* = x^5 = (-1.990911, -1.990911)^T,$$

and

$$f(x^*) = 7.967698,$$

which are not true. This is one of the advantages of using the q -gradient in our proposed method over the classical method.

We now execute Algorithm 1 on a set of test functions taken from the CUTer library [59] with 51 different starting points under standard and strong Wolfe conditions, respectively. Note that direction d_q^k generated by the proposed method is the q -descent direction due to the involvement of the q -gradient. Tables 4 and 5 list the numerical results of Algorithm 1 for 51 different starting points on a set of test problems, and Figs. 4 and 5 show this comparison graphically for both q -PRP and classical PRP methods under standard and strong Wolfe conditions, respectively. We conclude that our method is better than the classical method with the smaller number of iterations for the selected set of test problems.

6 Conclusion and future work

This paper proposed the q -PRP conjugate gradient method, which is an improvement of classical PRP conjugate gradient methods. The global convergence of the proposed method is established under the standard and strong Wolfe line searches. The effectiveness of the proposed method has been shown by some numerical examples. We find that the proposed method due to the q -gradient converges fast for a set of test problems with different starting points. The inclusion of q -calculus in other conjugate gradient methods deserves further investigation.

Table 4 Numerical results using Algorithm 1

Sl. no.	Test problem	Starting point		q-PRP		Using standard Wolfe		Using strong Wolfe	
		Starting point	NI	$f(x^*)$	NI	$f(x^*)$	x^*	NI	$f(x^*)$
1	Ackley	(2.000E-01, 1.000E-01) ^T	20	4.3168E-06	20	(-5.553E-08, -1.525E-06) ^T	20	(-5.782E-08, 8.525E-08) ^T	2.9134E-07
2	Beale	(-1.370E+00, -1.433E+00) ^T	16	8.6289E-02	16	(3.173E+00, 5.909E-01) ^T	16	(3.000E+00, 5.000E-01) ^T	1.4030E-09
3	Beale	(1.882E+00, -3.130E-01) ^T	20	1.0665E-06	20	(3.003E+00, 5.007E-01) ^T	20	(3.002E+00, 5.004E-01) ^T	3.7846E-07
4	Beale	(1.476E+00, 5.286E-01) ^T	16	1.0818E-06	16	(3.003E+00, 5.007E-01) ^T	16	(3.000E+00, 5.000E-01) ^T	6.5751E-09
5	Beale	(6.077E-01, -1.118E+00) ^T	37	1.8662E-08	37	(3.000E+00, 5.001E-01) ^T	37	(3.000E+00, 5.000E-01) ^T	2.8397E-09
6	Booth	(1.000E+00, 4.000E+00) ^T	16	5.4094E-06	16	(9.987E-01, 3.000E+00) ^T	16	(1.000E+00, 3.000E+00) ^T	1.1595E-09
7	Camel: Three Hump	(1.472E-01, -4.025E+00) ^T	28	1.5777E-13	28	(-1.113E-07, -3.133E-07) ^T	28	(-5.381E-08, 9.050E-08) ^T	9.1115E-15
8	Camel: Three Hump	(4.058E+00, -2.215E+00) ^T	26	8.7677E-13	26	(7.076E-07, -3.795E-07) ^T	26	(1.160E-07, -6.956E-07) ^T	4.3008E-13
9	Camel: Three Hump	(-3.730E+00, 4.688E-01) ^T	33	3.5296E-12	33	(9.617E-07, -1.863E-06) ^T	33	(-1.593E-07, -1.680E-08) ^T	5.3740E-14
10	Camel: Three Hump	(4.134E+00, 4.575E+00) ^T	36	3.3240E-11	36	(8.942E-07, 5.196E-06) ^T	36	(-5.209E-07, -4.531E-07) ^T	9.8400E-13
11	Camel: Three Hump	(1.324E+00, 4.649E+00) ^T	41	2.7932E-14	41	(7.574E-08, -1.716E-07) ^T	41	(-1.350E-07, -8.584E-09) ^T	3.7701E-14
12	Camel: Six Hump	(0.000E+00, 9.000E-01) ^T	14	-1.0316E+00	14	(-8.985E-02, 7.127E-01) ^T	14	(-8.984E-02, 7.127E-01) ^T	-1.0316E+00
13	Camel: Six Hump	(1.000E+00, -1.000E+00) ^T	22	-1.0316E+00	22	(8.970E-02, -7.125E-01) ^T	22	(8.987E-02, -7.127E-01) ^T	-1.0316E+00
14	Camel: Six Hump	(-4.864E-01, -8.967E-01) ^T	16	-1.0316E+00	16	(-8.987E-02, 7.127E-01) ^T	16	(0.000E+00, 0.000E+00) ^T	0.0000E+00
15	Camel: Six Hump	(4.561E-01, 1.377E+00) ^T	15	-1.0316E+00	15	(8.988E-02, -7.127E-01) ^T	15	(-8.985E-02, 7.127E-01) ^T	-1.0316E+00
16	Camel: Six Hump	(1.036E+00, -7.086E-01) ^T	21	-1.0316E+00	21	(8.985E-02, -7.127E-01) ^T	21	(-8.983E-02, 7.127E-01) ^T	-1.0316E+00
17	Chung Reynolds	(-1.225E+00, -2.047E-01) ^T	12	1.1819E-09	12	(-5.060E-03, -4.790E-03) ^T	12	(4.930E-03, -5.851E-03) ^T	1.7627E-09
18	Cube	(-1.000E+00, 2.000E+00) ^T	9	5.5657E-03	9	(1.074E+00, 1.240E+00) ^T	9	(1.067E+00, 1.216E+00) ^T	4.5261E-03
19	Cube	(-7.000E-01, 1.700E+00) ^T	19	4.8337E-03	19	(1.069E+00, 1.223E+00) ^T	19	(9.790E-01, 9.381E-01) ^T	4.4353E-04
20	Dixon-Price	(1.892E+00, 4.133E+00) ^T	25	2.1449E-07	25	(1.000E+00, 7.072E-01) ^T	25	(1.000E+00, -7.072E-01) ^T	2.0993E-08
21	Dixon-Price	(2.482E+00, -1.476E+00) ^T	19	1.9147E-06	19	(1.001E+00, 7.070E-01) ^T	19	(1.000E+00, 7.072E-01) ^T	1.2528E-08
22	Egg Crate	(1.000E-01, 5.000E-01) ^T	18	1.5788E-11	18	(-2.165E-07, -7.486E-07) ^T	18	(-5.116E-07, -4.764E-07) ^T	1.2706E-11
23	Himmelblau	(1.236E+00, 1.941E+00) ^T	13	6.8617E-07	13	(3.000E+00, 2.000E+00) ^T	13	(3.000E+00, 2.000E+00) ^T	2.7077E-07
24	Himmelblau	(2.809E+00, 1.169E+00) ^T	12	1.2192E-06	12	(3.000E+00, 2.000E+00) ^T	12	(3.000E+00, 2.000E+00) ^T	8.0157E-07
25	Himmelblau	(1.339E+00, 3.097E+00) ^T	8	1.0315E-06	8	(3.000E+00, 2.000E+00) ^T	8	(3.000E+00, 2.000E+00) ^T	5.6783E-07

Table 4 (Continued)

Sl. no.	Test problem	Starting point	q-PRP		Using standard Wolfe		Using strong Wolfe	
			x^*	NI	$f(x^*)$	NI	x^*	$f(x^*)$
26	Himmelblau	(2.723E + 00, 2.701E + 00) ^T	(3.000E + 00, 2.000E + 00) ^T	1.8308E - 06	14	(3.000E + 00, 2.000E + 00) ^T	6.8798E - 07	8
27	Leon	(5.007E - 01, 1.187E + 00) ^T	(9.967E - 01, 9.902E - 01) ^T	1.0976E - 05	17	(1.027E + 00, 1.083E + 00) ^T	7.3372E - 04	9
28	Leon	(1.083E + 00, 1.080E + 00) ^T	(1.030E + 00, 1.094E + 00) ^T	9.1674E - 04	7	(1.031E + 00, 1.095E + 00) ^T	9.4325E - 04	2
29	Leon	(9.337E - 01, 4.431E - 01) ^T	(1.007E + 00, 1.021E + 00) ^T	4.9947E - 05	20	(1.007E + 00, 1.021E + 00) ^T	4.8354E - 05	17
30	Rotated Ellipse 2	(2.242E + 00, -2.944E + 00) ^T	(5.030E - 07, -1.723E - 06) ^T	4.0889E - 12	20	(-5.710E - 08, -7.529E - 08) ^T	4.6296E - 15	5
31	Rotated Ellipse 2	(-8.821E - 01, 3.536E + 00) ^T	(-1.186E - 06, -9.251E - 07) ^T	1.1653E - 12	18	(-1.117E - 06, -8.961E - 07) ^T	1.0495E - 12	9
32	Rotated Ellipse 2	(-2.067E + 00, 3.649E + 00) ^T	(-4.149E - 07, -5.158E - 07) ^T	2.2421E - 13	23	(-6.791E - 07, -7.858E - 07) ^T	5.4504E - 13	8
33	Rotated Ellipse 2	(-7.687E - 01, 6.017E - 01) ^T	(-8.170E - 08, -7.721E - 08) ^T	6.3286E - 15	18	(4.064E - 08, -2.384E - 08) ^T	3.1887E - 15	5
34	Hosaki	(3.235E + 00, 1.287E + 00) ^T	(4.000E + 00, 2.000E + 00) ^T	-2.3458E + 00	6	(4.000E + 00, 2.000E + 00) ^T	-2.3458E + 00	11
35	Leon	(1.238E + 00, 7.653E - 01) ^T	(1.003E + 00, 1.006E + 00) ^T	9.3647E - 06	19	(1.002E + 00, 1.003E + 00) ^T	2.6746E - 06	16
36	Leon	(7.768E - 01, -5.870E - 01) ^T	(9.932E - 01, 9.881E - 01) ^T	3.6659E - 04	32	(1.002E + 00, 1.004E + 00) ^T	3.3665E - 06	25
37	Full Hessian 2	(1.000E + 00, 1.000E + 00) ^T	(5.000E + 00, -4.000E + 00) ^T	3.8588E - 08	16	(5.000E + 00, -4.000E + 00) ^T	4.5348E - 09	4
38	Full Hessian 2	(2.231E + 00, 1.675E + 00) ^T	(5.000E + 00, -4.001E + 00) ^T	2.3319E - 07	13	(5.000E + 00, -4.000E + 00) ^T	3.7499E - 08	3
39	Full Hessian 1	(-2.244E + 00, -1.776E + 00) ^T	(3.000E + 00, -2.982E + 00) ^T	2.5418E - 07	33	(3.001E + 00, -2.977E + 00) ^T	6.1723E - 07	20
40	Full Hessian 1	(2.864E + 00, 2.857E + 00) ^T	(3.000E + 00, -2.979E + 00) ^T	4.0601E - 07	21	(3.000E + 00, -2.980E + 00) ^T	3.4617E - 07	27
41	Full Hessian 1	(1.683E + 00, -1.268E + 00) ^T	(3.000E + 00, -2.992E + 00) ^T	1.2249E - 08	17	(3.000E + 00, -2.982E + 00) ^T	2.4862E - 07	17
42	Extended Maratos	(3.683E + 00, 4.268E + 00) ^T	(4.999E - 03, 1.113E - 06) ^T	-1.0000E + 02	13	(-5.000E - 03, -1.175E - 08) ^T	-1.0000E + 02	3
43	Generalized White & Holst	(1.010E + 00, 2.000E - 02) ^T	(1.016E + 00, 1.050E + 00) ^T	2.7167E - 04	20	(1.063E + 00, 1.202E + 00) ^T	4.0101E - 03	13
44	Fletcher	(3.150E + 00, 4.693E + 00) ^T	(-2.995E + 00, 4.979E + 00) ^T	6.9824E - 04	6	(-2.611E + 00, 3.205E + 00) ^T	6.0116E - 09	4
45	Zakharov	(1.786E + 00, 8.985E - 02) ^T	(-9.500E - 04, 1.178E - 03) ^T	2.7825E - 06	6	(-4.934E - 03, -1.568E - 02) ^T	5.9939E - 04	6
46	Zakharov	(-1.812E + 00, 7.856E - 01) ^T	(5.014E - 06, -1.573E - 06) ^T	2.8491E - 11	12	(-2.703E - 03, -1.435E - 02) ^T	4.6007E - 04	3
47	HIMMELH	(1.200E + 00, 1.000E - 01) ^T	(1.000E + 00, 9.999E - 01) ^T	-1.0000E + 00	20	(1.000E + 00, 1.000E + 00) ^T	-1.0000E + 00	9
48	HIMMELH	(1.200E + 00, -2.100E + 00) ^T	(1.000E + 00, 1.000E + 00) ^T	-1.0000E + 00	11	(1.000E + 00, 1.000E + 00) ^T	-1.0000E + 00	7
49	DIXON3DQ	(2.245E + 00, 2.306E + 00) ^T	(1.000E + 00, 1.000E + 00) ^T	1.8385E - 08	13	(1.000E + 00, 1.000E + 00) ^T	1.3340E - 10	3
50	DIXON3DQ	(7.012E - 01, 1.999E + 00) ^T	(1.000E + 00, 1.000E + 00) ^T	1.3656E - 08	27	(1.000E + 00, 1.000E + 00) ^T	1.1506E - 10	4
51	DIXON3DQ	(1.236E + 00, 5.929E - 02) ^T	(1.000E + 00, 1.000E + 00) ^T	2.4082E - 09	16	(1.000E + 00, 1.000E + 00) ^T	2.4809E - 10	4

Table 5 Numerical results using classical PRP

Sl. no.	Test problem	Starting point	PRP			Using standard Wolfe			Using strong Wolfe		
			x^*	$f(x^*)$	NI	x^*	$f(x^*)$	NI	x^*	$f(x^*)$	NI
1	Ackley	(2.000E - 01, 1.000E - 01) ^T	(8.0700E - 15, -1.127E - 12) ^T	3.1872E - 12	35	(-1.691E - 02, -3.888E - 02) ^T	4.2395E - 07	14			
2	Beale	(-1.370E + 00, -1.433E + 00) ^T	(3.0000E + 00, 5.000E - 01) ^T	9.0393E - 14	59	(3.000E + 00, 5.000E - 01) ^T	1.3600E - 13	34			
3	Beale	(1.882E + 00, -3.130E - 01) ^T	(3.0000E + 00, 5.000E - 01) ^T	2.2124E - 13	43	(3.000E + 00, 5.000E - 01) ^T	2.4689E - 13	47			
4	Beale	(1.476E + 00, 5.286E - 01) ^T	(3.0000E + 00, 5.000E - 01) ^T	2.1515E - 14	42	(3.000E + 00, 5.000E - 01) ^T	9.4967E - 14	54			
5	Beale	(6.077E - 01, -1.118E + 00) ^T	(3.0000E + 00, 5.000E - 01) ^T	1.8879E - 10	43	(3.000E + 00, 5.000E - 01) ^T	1.5070E - 16	21			
6	Booth	(1.000E + 00, 4.000E + 00) ^T	(1.000E + 00, 3.000E + 00) ^T	3.8496E - 15	25	(1.000E + 00, 3.000E + 00) ^T	6.0707E - 21	2			
7	Camel: Three Hump	(1.472E - 01, -4.025E + 00) ^T	(5.7530E - 09, -2.537E - 07) ^T	6.2992E - 14	27	(2.026E - 08, -5.473E - 08) ^T	2.7071E - 15	7			
8	Camel: Three Hump	(4.058E + 00, -2.215E + 00) ^T	(-3.0604E - 07, 3.778E - 07) ^T	2.1442E - 13	38	(2.385E - 08, -1.432E - 07) ^T	1.8236E - 14	13			
9	Camel: Three Hump	(-3.730E + 00, 4.688E - 01) ^T	(1.2745E - 07, -8.439E - 08) ^T	2.8855E - 14	33	(-1.373E - 07, 2.043E - 07) ^T	5.1397E - 14	9			
10	Camel: Three Hump	(4.134E + 00, 4.575E + 00) ^T	(-1.6209E - 07, 3.520E - 07) ^T	1.1940E - 13	33	(4.917E - 09, -4.958E - 08) ^T	2.2623E - 15	10			
11	Camel: Three Hump	(1.324E + 00, 4.649E + 00) ^T	(-7.3313E - 08, 1.078E - 07) ^T	1.4461E - 14	26	(2.647E - 09, -6.108E - 09) ^T	3.5154E - 17	7			
12	Camel: Six Hump	(0.000E + 00, 9.000E - 01) ^T	(-8.9842E - 02, 7.127E - 01) ^T	-1.0316E + 00	19	(-8.984E - 02, 7.127E - 01) ^T	-1.0316E + 00	8			
13	Camel: Six Hump	(1.000E + 00, -1.000E + 00) ^T	(8.9842E - 02, -7.127E - 01) ^T	-1.0316E + 00	29	(8.984E - 02, -7.127E - 01) ^T	-1.0316E + 00	11			
14	Camel: Six Hump	(-4.864E - 01, -8.967E - 01) ^T	(-8.9842E - 02, 7.127E - 01) ^T	-1.0316E + 00	26	(-8.984E - 02, 7.127E - 01) ^T	-1.0316E + 00	9			
15	Camel: Six Hump	(4.561E - 01, 1.377E + 00) ^T	(8.9842E - 02, -7.127E - 01) ^T	-1.0316E + 00	25	(-8.984E - 02, 7.127E - 01) ^T	-1.0316E + 00	11			
16	Camel: Six Hump	(1.036E + 00, -7.086E - 01) ^T	(8.9842E - 02, -7.127E - 01) ^T	-1.0316E + 00	31	(-8.984E - 02, 7.127E - 01) ^T	-1.0316E + 00	13			
17	Chung Reynolds	(-1.225E + 00, -2.047E - 01) ^T	(-5.0624E - 03, -4.786E - 03) ^T	1.1816E - 09	12	(4.931E - 03, -5.851E - 03) ^T	1.7631E - 09	12			
18	Cube	(-1.000E + 00, 2.000E + 00) ^T	(1.0180E + 00, 1.054E + 00) ^T	3.6294E - 04	14	(1.000E + 00, 1.000E + 00) ^T	1.9171E - 13	38			
19	Cube	(-7.000E - 01, 1.700E + 00) ^T	(1.000E + 00, 1.000E + 00) ^T	1.2254E - 13	76	(-2.669E - 07, -1.256E - 07) ^T	1.6958E - 13	40			
20	Dixon-Price	(1.892E + 00, 4.133E + 00) ^T	(9.9172E - 01, 7.070E - 01) ^T	2.0024E - 04	17	(1.000E + 00, -7.071E - 01) ^T	7.3023E - 15	12			
21	Dixon-Price	(2.482E + 00, -1.476E + 00) ^T	(1.000E + 00, 7.071E - 01) ^T	2.8514E - 15	24	(1.000E + 00, 7.071E - 01) ^T	4.3728E - 16	12			
22	Egg Crate	(1.000E - 01, 5.000E - 01) ^T	(-6.4083E - 10, 1.830E - 08) ^T	8.7177E - 15	18	(1.759E - 11, 2.400E - 12) ^T	8.1916E - 21	5			
23	Himmelblau	(1.236E + 00, 1.941E + 00) ^T	(3.000E + 00, 2.000E + 00) ^T	1.8810E - 16	21	(3.000E + 00, 2.000E + 00) ^T	7.6171E - 15	12			
24	Himmelblau	(2.809E + 00, 1.169E + 00) ^T	(3.000E + 00, 2.000E + 00) ^T	5.6991E - 15	23	(3.000E + 00, 2.000E + 00) ^T	2.9048E - 17	12			
25	Himmelblau	(1.339E + 00, 3.097E + 00) ^T	(3.000E + 00, 2.000E + 00) ^T	4.5473E - 15	20	(3.000E + 00, 2.000E + 00) ^T	1.4686E - 16	16			

Table 5 (Continued)

Sl. no.	Test problem	Starting point	PRP		Using standard Wolfe		Using strong Wolfe	
			x^*	NI	$f(x^*)$	NI	x^*	$f(x^*)$
26	Himmelblau	(2.723E + 00, 2.701E + 00) ^T	(3.0000E + 00, 2.000E + 00) ^T	32	(3.000E + 00, 2.000E + 00) ^T	2.6335E - 18	11	
27	Leon	(5.007E - 01, 1.187E + 00) ^T	(1.0000E + 00, 1.000E + 00) ^T	43	(1.000E + 00, 1.000E + 00) ^T	2.4088E - 14	20	
28	Leon	(1.083E + 00, 1.080E + 00) ^T	(1.0000E + 00, 1.000E + 00) ^T	51	(1.000E + 00, 1.000E + 00) ^T	6.3897E - 13	30	
29	Leon	(9.337E - 01, 4.431E - 01) ^T	(1.0000E + 00, 1.000E + 00) ^T	53	(1.000E + 00, 1.000E + 00) ^T	9.0974E - 13	41	
30	Rotated Ellipse 2	(2.242E + 00, -2.944E + 00) ^T	(-6.8708E - 07, -6.668E - 07) ^T	20	(-5.928E - 07, -5.525E - 07) ^T	3.2912E - 13	4	
31	Rotated Ellipse 2	(-8.821E - 01, 3.536E + 00) ^T	(-2.1865E - 07, 5.103E - 08) ^T	20	(-3.800E - 08, -3.864E - 08) ^T	1.4688E - 15	10	
32	Rotated Ellipse 2	(-2.067E + 00, 3.649E + 00) ^T	(1.9903E - 08, -3.484E - 07) ^T	25	(8.176E - 08, -2.497E - 08) ^T	9.3490E - 15	7	
33	Rotated Ellipse 2	(-7.687E - 01, 6.017E - 01) ^T	(1.6098E - 07, -2.524E - 07) ^T	17	(-8.662E - 08, -9.228E - 08) ^T	8.0248E - 15	4	
34	Hosaki	(3.235E + 00, 1.287E + 00) ^T	(4.0000E + 00, 2.000E + 00) ^T	24	(4.000E + 00, 2.000E + 00) ^T	-2.3458E + 00	9	
35	Leon	(1.238E + 00, 7.653E - 01) ^T	(9.7925E - 01, 9.571E - 01) ^T	19	(1.000E + 00, 1.000E + 00) ^T	2.2956E - 13	23	
36	Leon	(7.768E - 01, -5.870E - 01) ^T	(9.0943E - 01, 7.994E - 01) ^T	33	(1.000E + 00, 1.000E + 00) ^T	1.9307E - 15	33	
37	Full Hessian 2	(1.000E + 00, 1.000E + 00) ^T	(5.0000E + 00, -4.000E + 00) ^T	18	(5.000E + 00, -4.000E + 00) ^T	2.5310E - 20	2	
38	Full Hessian 2	(2.231E + 00, 1.675E + 00) ^T	(5.0000E + 00, -4.000E + 00) ^T	19	(5.000E + 00, -4.000E + 00) ^T	2.8648E - 20	2	
39	Full Hessian 1	(-2.244E + 00, -1.776E + 00) ^T	(3.0000E + 00, -2.997E + 00) ^T	61	(3.000E + 00, -2.996E + 00) ^T	4.3440E - 10	52	
40	Full Hessian 1	(2.864E + 00, 2.857E + 00) ^T	(2.9994E + 00, -3.022E + 00) ^T	27	(3.000E + 00, -3.004E + 00) ^T	3.2194E - 10	40	
41	Full Hessian 1	(1.683E + 00, -1.268E + 00) ^T	(3.0000E + 00, -2.996E + 00) ^T	49	(3.000E + 00, -2.991E + 00) ^T	1.3520E - 08	41	
42	Extended Maratos	(3.683E + 00, 4.268E + 00) ^T	(-5.0000E - 03, -2.646E - 09) ^T	17	(-5.000E - 03, 6.277E - 09) ^T	-1.0000E + 02	2	
43	Generalized White & Holst	(1.010E + 00, 2.000E - 02) ^T	(1.0000E + 00, 1.000E + 00) ^T	100	(1.000E + 00, 1.000E + 00) ^T	9.0708E - 13	37	
44	Fletcher	(3.150E + 00, 4.693E + 00) ^T	(-2.9958E + 00, 4.979E + 00) ^T	14	(-2.615E + 00, 3.223E + 00) ^T	8.9434E - 19	4	
45	Zakharov	(1.786E + 00, 8.985E - 02) ^T	(-9.5498E - 04, 1.184E - 03) ^T	7	(-5.000E - 03, -1.585E - 02) ^T	6.1296E - 04	6	
46	Zakharov	(-1.812E + 00, 7.856E - 01) ^T	(-8.0501E - 08, 7.469E - 08) ^T	13	(-2.701E - 03, -1.436E - 02) ^T	4.6014E - 04	3	
47	HIMMELH	(1.200E + 00, 1.000E - 01) ^T	(1.0000E + 00, 1.000E + 00) ^T	17	(1.000E + 00, 1.000E + 00) ^T	-1.0000E + 00	9	
48	HIMMELH	(1.200E + 00, -2.100E + 00) ^T	(1.0000E + 00, 1.000E + 00) ^T	21	(1.000E + 00, 1.000E + 00) ^T	-1.0000E + 00	8	
49	DIXON3DQ	(2.245E + 00, 2.306E + 00) ^T	(1.0000E + 00, 1.000E + 00) ^T	17	(1.000E + 00, 1.000E + 00) ^T	5.6225E - 19	2	
50	DIXON3DQ	(7.012E - 01, 1.999E + 00) ^T	(1.0000E + 00, 1.000E + 00) ^T	47	(1.000E + 00, 1.000E + 00) ^T	1.1851E - 20	2	
51	DIXON3DQ	(1.236E + 00, 5.929E - 02) ^T	(1.0000E + 00, 1.000E + 00) ^T	42	(1.000E + 00, 1.000E + 00) ^T	2.4317E - 21	2	

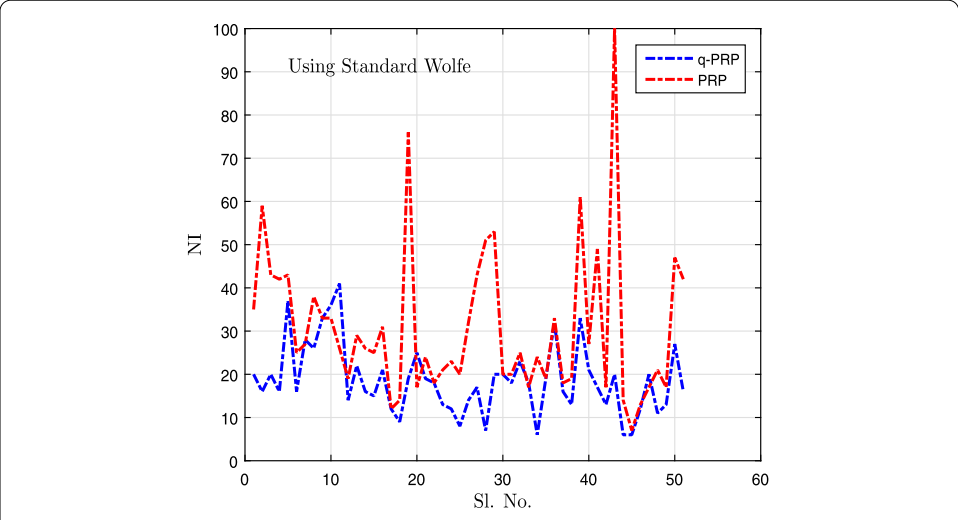


Figure 4 Graphical representation of q -PRP and PRP algorithms under standard Wolfe conditions based on Tables 4 and 5

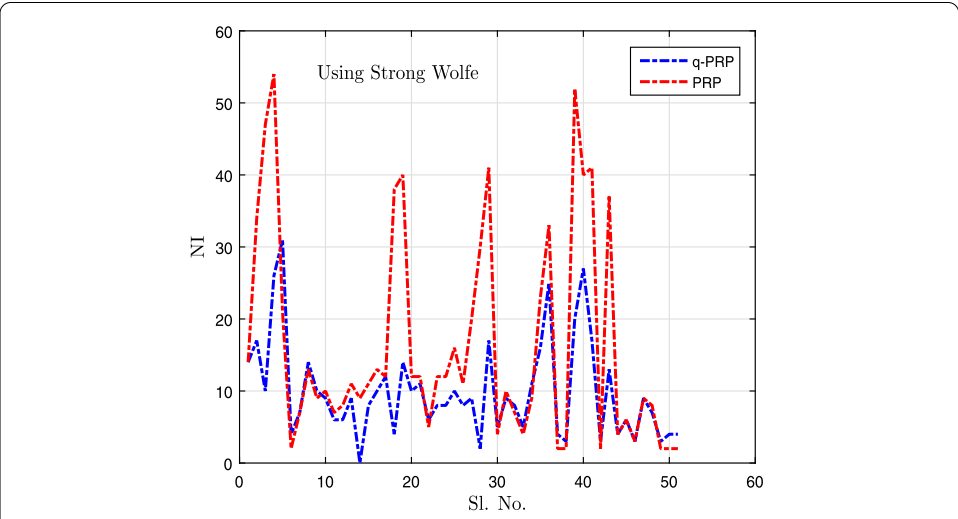


Figure 5 Graphical representation of q -PRP and PRP algorithms under strong Wolfe conditions based on Tables 4 and 5

Acknowledgements

The first author was supported by the Science and Engineering Research Board (Grant No. DST-SERB-MTR-2018/000121). The third author was supported by Bu-Ali Sina University. The fourth author was supported by the University Grants Commission (IN) (Grant No. UGC-2015-UTT-59235). Constructive comments by the referees are gratefully acknowledged.

Funding

Not applicable.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Authors' contributions

The authors declare that the study was realized in collaboration with equal responsibility. All authors read and approved the final manuscript.

Author details

¹Department of Mathematics, Institute of Science, Banaras Hindu University, Varanasi 221005, India. ²Department of Mathematics, Sir Gurudas Mahavidyalaya, Kolkata 700067, India. ³Department of Mathematics, Bu-Ali Sina University, Hamedan, Iran. ⁴DST-Centre for Interdisciplinary Mathematical Sciences, Institute of Science, Banaras Hindu University, Varanasi 221005, India.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 21 September 2020 Accepted: 14 January 2021 Published online: 28 January 2021

References

1. Mishra, S.K., Ram, B.: Conjugate gradient methods. In: Introduction to Unconstrained Optimization with R, pp. 211–244. Springer, Singapore (2019)
2. Andrei, N.: Nonlinear Conjugate Gradient Methods for Unconstrained Optimization. Springer, Berlin (2020)
3. Li, Y., Chen, W., Zhou, H., Yang, L.: Conjugate gradient method with pseudospectral collocation scheme for optimal rocket landing guidance. *Aerosp. Sci. Technol.* **104**, 105999 (2020)
4. Liu, J., Du, S., Chen, Y.: A sufficient descent nonlinear conjugate gradient method for solving m -tensor equations. *J. Comput. Appl. Math.* **371**, 112709 (2020)
5. Yuan, G., Li, T., Hu, W.: A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems. *Appl. Numer. Math.* **147**, 129–141 (2020)
6. Hestenes, M.R., Stiefel, E., et al.: Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **49**(6), 409–436 (1952)
7. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. *Comput. J.* **7**(2), 149–154 (1964)
8. Mishra, S.K., Ram, B.: Steepest descent method. In: Introduction to Unconstrained Optimization with R, pp. 131–173. Springer, Singapore (2019)
9. Polak, E., Ribiere, G.: Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Math. Model. Numer. Anal.* **3**(R1), 35–43 (1969)
10. Jackson, F.H.: On q -functions and a certain difference operator. *Earth Environ. Sci. Trans. R. Soc. Edinb.* **46**(2), 253–281 (1909)
11. Jackson, D.O., Fukuda, T., Dunn, O., Majors, E.: On q -definite integrals. *Q. J. Pure Appl. Math.* **41**, 193–203 (1910)
12. Ernst, T.: A method for q -calculus. *J. Nonlinear Math. Phys.* **10**(4), 487–525 (2003)
13. Awan, M.U., Talib, S., Kashuri, A., Noor, M.A., Chu, Y.-M.: Estimates of quantum bounds pertaining to new q -integral identity with applications. *Adv. Differ. Equ.* **2020**, 424 (2020)
14. Samei, M.E.: Existence of solutions for a system of singular sum fractional q -differential equations via quantum calculus. *Adv. Differ. Equ.* **2020**, 23 (2020). <https://doi.org/10.1186/s13662-019-2480-y>
15. Liang, S., Samei, M.E.: New approach to solutions of a class of singular fractional q -differential problem via quantum calculus. *Adv. Differ. Equ.* **2020**, 14 (2020). <https://doi.org/10.1186/s13662-019-2489-2>
16. Ahmadian, A., Rezapour, S., Salahshour, S., Samei, M.E.: Solutions of sum-type singular fractional q -integro-differential equation with m -point boundary value problem using quantum calculus. *Math. Methods Appl. Sci.* **43**(15), 8980–9004 (2020). <https://doi.org/10.1002/mma.6591>
17. Samei, M.E., Hedayati, H., Rezapour, S.: Existence results for a fraction hybrid differential inclusion with Caputo–Hadamard type fractional derivative. *Adv. Differ. Equ.* **2019**, 163 (2019). <https://doi.org/10.1186/s13662-019-2090-8>
18. Samei, M.E., Ranjbar, G.K., Hedayati, V.: Existence of solutions for equations and inclusions of multiterm fractional q -integro-differential with nonseparated and initial boundary conditions. *J. Inequal. Appl.* **2019**, 273 (2019). <https://doi.org/10.1186/s13660-019-2224-2>
19. Mishra, S.K., Panda, G., Ansary, M.A.T., Ram, B.: On q -Newton's method for unconstrained multiobjective optimization problems. *J. Appl. Math. Comput.* **63**, 391–410 (2020)
20. Lai, K.K., Mishra, S.K., Ram, B.: On q -quasi-Newton's method for unconstrained multiobjective optimization problems. *Mathematics* **8**(4), 616 (2020)
21. Polyak, B.T.: The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.* **9**(4), 94–112 (1969)
22. Polyak, B.T.: The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.* **9**(4), 94–112 (1969)
23. Yuan, Y.: Numerical Methods for Nonlinear Programming. Shanghai Sci. Technol., Shanghai (1993)
24. Powell, M.J.: Nonconvex minimization calculations and the conjugate gradient method. In: Numerical Analysis, pp. 122–141. Springer, Berlin (1984)
25. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.* **2**(1), 21–42 (1992)
26. Powell, M.J.: Convergence properties of algorithms for nonlinear optimization. *SIAM Rev.* **28**(4), 487–500 (1986)
27. Yuan, G., Li, T., Hu, W.: A conjugate gradient algorithm and its application in large-scale optimization problems and image restoration. *J. Inequal. Appl.* **2019**, 247 (2019). <https://doi.org/10.1186/s13660-019-2192-6>

28. Yuan, G., Li, T., Hu, W.: A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems. *Appl. Numer. Math.* **147**, 129–141 (2020). <https://doi.org/10.1016/j.apnum.2019.08.022>
29. Hu, W., Wu, J., Yuan, G.: Some modified Hestenes–Stiefel conjugate gradient algorithms with application in image restoration. *Appl. Numer. Math.* **158**, 360–376 (2020). <https://doi.org/10.1016/j.apnum.2020.08.009>
30. Yuan, G., Lu, J., Wang, Z.: The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems. *Appl. Numer. Math.* **152**, 1–11 (2020). <https://doi.org/10.1016/j.apnum.2020.01.019>
31. Yuan, G., Wei, Z., Yang, Y.: The global convergence of the Polak–Ribière–Polyak conjugate gradient algorithm under inexact line search for nonconvex functions. *J. Comput. Appl. Math.* **362**, 262–275 (2019)
32. Yuan, G., Wang, X., Zhou, S.: The projection technique for two open problems of unconstrained optimization problems. *J. Optim. Theory Appl.* **186**, 590–619 (2020)
33. Zhang, M., Zhou, Y., Wang, S.: A modified nonlinear conjugate gradient method with the Armijo line search and its application. *Math. Probl. Eng.* **2020**, 6210965 (2020)
34. Soterroni, A.C., Galski, R.L., Ramos, F.M.: The q -gradient vector for unconstrained continuous optimization problems. In: *Operations Research Proceedings 2010*, pp. 365–370. Springer, Berlin (2011)
35. Sadiq, A., Usman, M., Khan, S., Naseem, I., Moinuddin, M., Al-Saggaf, U.M.: q -LMF: quantum calculus-based least mean fourth algorithm. In: *Fourth International Congress on Information and Communication Technology*, pp. 303–311. Springer, Berlin (2020)
36. Sadiq, A., Khan, S., Naseem, I., Togneri, R., Bennamoun, M.: Enhanced q -least mean square. *Circuits Syst. Signal Process.* **38**(10), 4817–4839 (2019)
37. Chakraborty, S.K., Panda, G.: q -line search scheme for optimization problem. arXiv preprint, [arXiv:1702.01518](https://arxiv.org/abs/1702.01518) (2017)
38. Chakraborty, S.K., Panda, G.: Newton like line search method using q -calculus. In: *International Conference on Mathematics and Computing*, pp. 196–208. Springer, Berlin (2017)
39. Lai, K.K., Mishra, S.K., Panda, G., Chakraborty, S.K., Samei, M.E., Ram, B.: A limited memory q -BFGS algorithm for unconstrained optimization problems. *J. Appl. Math. Comput.* (2020). <https://doi.org/10.1007/s12190-020-01432-6>
40. Lai, K.K., Mishra, S.K., Panda, G., Ansary, M.A.T., Ram, B.: On q -steepest descent method for unconstrained multiobjective optimization problems. *AIMS Math.* **5**(6), 5521–5540 (2020)
41. Gouvêa, É.J., Regis, R.G., Soterroni, A.C., Scarabello, M.C., Ramos, F.M.: Global optimization using q -gradients. *Eur. J. Oper. Res.* **251**(3), 727–738 (2016)
42. Aral, A., Gupta, V., Agarwal, R.P., et al.: *Applications of q -Calculus in Operator Theory*. Springer, New York (2013)
43. Rajković, P., Stanković, M., Marinković, S.D.: Mean value theorems in q -calculus. *Mat. Vesn.* **54**(3–4), 171–178 (2002)
44. Dai, Y.-H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **10**(1), 177–182 (1999)
45. Fletcher, R.: *Practical Methods of Optimization*, vol. 80, 4. Wiley, New York (1987)
46. Nocedal, J., Wright, S.: *Numerical Optimization*. Springer, New York (2006)
47. Mishra, S.K., Ram, B.: *Introduction to Unconstrained Optimization with R*. Springer, Singapore (2019)
48. Wolfe, P.: Convergence conditions for ascent methods, II: some corrections. *SIAM Rev.* **13**(2), 185–188 (1971)
49. Goldstein, A.A.: On steepest descent. *J. Soc. Ind. Appl. Math., A, on Control* **3**(1), 147–151 (1965)
50. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. *Pac. J. Math.* **16**(1), 1–3 (1966)
51. Grippo, L., Lucidi, S.: A globally convergent version of the Polak–Ribière conjugate gradient method. *Math. Program.* **78**(3), 375–391 (1997)
52. Zhang, L., Zhou, W., Li, D.-H.: A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **26**(4), 629–640 (2006)
53. Mishra, S.K., Ram, B.: One-dimensional optimization methods. In: *Introduction to Unconstrained Optimization with R*, pp. 85–130. Springer, Boston (2019)
54. Zoutendijk, G.: Nonlinear programming, computational methods. In: *Integer and Nonlinear Programming*, pp. 37–86 (1970)
55. Yuan, G.: Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. *Optim. Lett.* **3**(1), 11–21 (2009)
56. Aminifard, Z., Babaie-Kafaki, S.: A modified descent Polak–Ribière–Polyak conjugate gradient method with global convergence property for nonconvex functions. *Calcolo* **56**(2), 16 (2019)
57. Dai, Y., Han, J., Liu, G., Sun, D., Yin, H., Yuan, Y.-X.: Convergence properties of nonlinear conjugate gradient methods. *SIAM J. Optim.* **10**(2), 345–358 (2000)
58. Mishra, S.K.: Global optimization by differential evolution and particle swarm methods: evaluation on some benchmark functions. Available at SSRN 933827 (2006)
59. Gould, N.I.M., Orban, D., Toint, P.L.: CUTER (and SifDec), a constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Softw.* **29**, 373–394 (2003)