Kybernetika

Youcef Elhamam Hemici; Samia Khelladi; Djamel Benterki

New hybrid conjugate gradient method for nonlinear optimization with application to image restoration problems

Kybernetika, Vol. 60 (2024), No. 4, 535–552

Persistent URL: http://dml.cz/dmlcz/152618

KYBERNETIKA — VOLUME *60* (2024), NUMBER 4, PAGES 535–552

NEW HYBRID CONJUGATE GRADIENT METHOD FOR NONLINEAR OPTIMIZATION WITH APPLICATION TO IMAGE RESTORATION PROBLEMS

Youcef Elhamam Hemici, Samia Khelladi and Djamel Benterki

The conjugate gradient method is one of the most effective algorithm for unconstrained nonlinear optimization problems. This is due to the fact that it does not need a lot of storage memory and its simple structure properties, which motivate us to propose a new hybrid conjugate gradient method through a convex combination of β_k^{RMIL} and β_k^{HS}. We compute the convex parameter θ_k using the Newton direction. Global convergence is established through the strong Wolfe conditions. Numerical experiments show the superior efficiency of our algorithm to solve unconstrained optimization problem compared to other considered methods. Applied to image restoration problem, our algorithm is competitive with existing algorithms and performs even better when the level of noise in the image is significant.

Keywords: unconstrained optimization, conjugate gradient method, descent direction, line search, image restoration

Classification: 65K05, 90C26, 90C30

1. INTRODUCTION

The conjugate gradient method remains one of the most widely used iterative techniques for solving linear systems and nonlinear optimization problems due to its efficiency, numerical stability and versatility across different domains of science and engineering.

In this paper, we are interested in solving unconstrained optimization problems in the form

$$\begin{cases} \min \ f(x) \\ x \in \mathbb{R}^n, \end{cases} \tag{1}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function.

The nonlinear conjugate gradient methods are efficient to solve problem (1), especially for large scale problems. So, the classical conjugate gradient method is given by:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

DOI: 10.14736/kyb-2024-4-0535

where x_k is the current iterate point, $\alpha_k > 0$ is the stepsize which can be found by one of the line search methods and d_k is the search direction defined by:

$$d_0 = -g_0, \ d_{k+1} = -g_{k+1} + \beta_k d_k, \ \text{for } k \geq 0, \tag{3}$$

where $g_k = \nabla f(x_k)$ is the gradient of f at x_k and β_k is a scalar conjugacy coefficient. The choice of the parameter β_k in the conjugate gradient method determines the search direction d_k which leads to different variants of the conjugate gradient method.

In recent years, researchers have continued to explore and refine the conjugate gradient method, proposing new β_k to address specific problem characteristics and computational challenges.

Some of the well known parameters β_k are those of Hesteness–Steifel (HS) [15], Fletcher and Reeves (FR) [13], Polak–Ribiere–Polyak (PRP) [21, 22], Conjugate Descent (CD) [12], Liu–Storey (LS) [17], Dai–Yuan (DY) [6, 7] and Rivaie–Mustafa–Ismail–Leong (RMIL) [24]. Hager and Zhang gave a good survey of nonlinear conjugate gradient methods in [14].

The formulas of the β_k mentioned above are:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \ \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \ \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \ \beta_k^{CD} = -\frac{\|g_{k+1}\|^2}{g_k^T d_k},$$

$$\beta_k^{LS} = -\frac{g_{k+1}^T y_k}{g_k^T d_k}, \ \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \ \beta_k^{RMIL} = \frac{g_{k+1}^T y_k}{\|d_k\|^2}.$$

Where $y_k = g_{k+1} - g_k$ and $\|\cdot\|$ denotes the Euclidean norm.

Many researchers proposed new families and combinations of the conjugate gradient methods, specifically the hybrid methods. The idea of the hybrid methods is to combine the standard conjugate gradient methods and exploit the attractive features of each of one and to avoid the jamming phenomenon. This combination can be convex or nonconvex. Among these hybrid methods, we can cite those proposed by Dalladji et al. [8] and Mtagulwa and Kaelo [20], Djordjevic [9, 10], Ben Hanachi et al. [3], Yang et al. [28], Rivaie et al. [25] and the families of conjugate methods proposed by Sellami and Chaib [26, 27].

Recently, a great contribution in the area of conjugate gradient methods and its application has been done by Andrei in [2].

The aim of this paper is to propose an efficient conjugate gradient method for nonlinear optimization using a new parameter β_k which leads to a new descent direction.

The contents of the remaining part of this paper are organized as follows. In section 2, we give the new formula of β_k and describe the corresponding algorithm. In addition, we present a complete analysis of the descent condition of the obtained direction then, we show the global convergence of the new algorithm using the strong Wolfe line search. Section 3 includes numerical experiments on the obtained algorithm based on our new parameter β_k on several unconstrained optimization problems compared to other considered methods, using the well known test functions in the literature and an image processing problems called image restoration problems. Finally, we end with a conclusion in section 4.

2. NEW HYBRID CONJUGATE GRADIENT METHOD AND DESCRIPTION OF THE CORRESPONDING ALGORITHM

Based on the previously mentioned hybridization ideas and considering the good numerical performance of the RMIL and HS algorithms, we present in this section a new formula of β_k which is a convex combination of β_k^{RMIL} and β_k^{HS}. Our new parameter β_k, denoted by β_k^{RMILHS} is given as follows:

$$\beta_k^{RMILHS} = (1-\theta_k)\beta_k^{RMIL} + \theta_k \beta_k^{HS}. \tag{4}$$

So, we can write

$$d_{k+1} = -g_{k+1} + \beta_k^{RMILHS} d_k, \tag{5}$$

where θ_k is a scalar parameter which satisfies $0 \leq \theta_k \leq 1$.

To compute the stepsize α_k, of (2), we use the strong Wolfe conditions, given by:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k \tag{6}$$

$$\left| g_{k+1}^T d_k \right| \leq -\sigma g_k^T d_k, \tag{7}$$

where $0 < \rho < \sigma < \frac{1}{2}$.

It is obvious that, if $\theta_k = 0$, then $\beta_k^{RMILHS} = \beta_k^{RMIL}$ and if $\theta_k = 1$, then $\beta_k^{RMILHS} = \beta_k^{HS}$.

On the other hand, if $0 < \theta_k < 1$, then β_k^{RMILHS} is proper convex combination of β_k^{RMIL} and β_k^{HS}.

Having in view the formulas of $\beta_k^{RMIL}$and β_k^{HS}, the relation (4) becomes

$$\beta_k^{RMILHS} = (1-\theta_k)\frac{g_{k+1}^T y_k}{\|d_k\|^2} + \theta_k \frac{g_{k+1}^T y_k}{d_k^T y_k}. \tag{8}$$

Hence the relation (5) becomes

$$d_{k+1} = -g_{k+1} + (1-\theta_k)\frac{g_{k+1}^T y_k}{\|d_k\|^2} d_k + \theta_k \frac{g_{k+1}^T y_k}{d_k^T y_k} d_k. \tag{9}$$

As known, if the point x_{k+1} is close enough to a local minimizer x^*, then the Newton direction is the best direction to follow.

The parameter θ_k is chosen such that the search direction d_{k+1} aligns with the Newton direction. So, assuming that the inverse of the hessian $\nabla^2 f(x_{k+1})$ exists at each iterative point x_{k+1} for the objective function f. We are going to choose the parameter θ_k such that the search direction d_k defined by (5), satisfies the condition of Newton's direction, i. e.,

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = d_{k+1} = -g_{k+1} + \beta_k^{RMILHS} d_k. \tag{10}$$

Replacing β_k^{RMILHS} by its formula (4) and multiplying (10) by $s_k^T \nabla^2 f(x_{k+1})$, where $s_k = x_{k+1} - x_k$, we obtain

$$-s_k^T g_{k+1} = -s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1-\theta_k)\beta_k^{RMIL} s_k^T \nabla^2 f(x_{k+1}) d_k + \theta_k \beta_k^{HS} s_k^T \nabla^2 f(x_{k+1}) d_k. \tag{11}$$

Using the secant condition $\nabla^2 f(x_{k+1}) s_k = y_k$, (11) becomes:

$$-s_k^T g_{k+1} = -y_k^T g_{k+1} + (1-\theta_k)\beta_k^{RMIL} y_k^T d_k + \theta_k \beta_k^{HS} y_k^T d_k. \tag{12}$$

From (12) and with some simple algebraic manipulations, we define θ_k as follows:

$$\theta_k = \begin{cases} \theta_k^{NT} & \text{if} \quad 0 < \theta_k^{NT} < 1, \\ 0 & \text{if} \quad \theta_k^{NT} < 0, \\ 1 & \text{if} \quad \theta_k^{NT} > 1, \end{cases} \tag{13}$$

where

$$\theta_k^{NT} = \frac{-s_k^T g_{k+1} + y_k^T g_{k+1} - \beta_k^{RMIL} y_k^T d_k}{(\beta_k^{RMIL} - \beta_k^{HS}) y_k^T d_k}. \tag{14}$$

2.1. Algorithm RMILHS

The algorithm corresponding to our parameter β_k is given as follows:

Begin algorithm
Step 0: Given a starting point x_0 and a parameter $\varepsilon > 0$.
Step 1: Set $k = 0$ and compute $d_0 = -g_0$.
Step 2: If $\|g_k\| \leq \varepsilon$, **Stop**; else go to **Step 3**.
Step 3: Find the stepsize $\alpha_k \in]0, 1]$ (using strong Wolfe conditions (6) and (7)).
Step 4: Compute $x_{k+1} = x_k + \alpha_k d_k$.
Step 5: Compute $g_{k+1} = \nabla f(x_{k+1})$, $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$.
Step 6: Compute $\theta_k = \theta_k^{NT}$ (using (14) and (13)).
Step 7: Compute $\beta_k = \beta_k^{RMILHS} = (1-\theta_k)\beta_k^{RMIL} + \theta_k \beta_k^{HS}$ (using (8)).
Step 8: If $\mid g_{k+1}^T g_k \mid \geq 0.2\|g_{k+1}\|$ (restart criterion of Powell [23])
then set $d_{k+1} = -g_{k+1}$ and $\alpha_{k+1} = 1$,
else compute $d_{k+1} = -g_{k+1} + \beta_k d_k$.
Step 9: Let $k = k + 1$ and go to **Step 2.**
End algorithm.

2.2. Convergence Analysis

Throughout this section, we make the following assumptions:

(i) The level set $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded, which means that there exists a constant $M < \infty$, such that

$$\|x\| \leq M, \text{ for all } x \in \mathcal{L}.$$

(ii) In a neighborhood $\mathcal{N}$ of $\mathcal{L}$, the function f is continuously differentiable and its gradient $\nabla f(x)$ is Lipschitz continuous, it means that exists $0 < L < \infty$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \text{ for all } x, y \in \mathcal{N}. \tag{15}$$

Under these assumptions, there exists a constant $\mu \geq 0$, where

$$\|\nabla f(x)\| \leq \mu, \text{ for all } x \in \mathcal{L}. \tag{16}$$

To establish the sufficient descent condition, we introduce the following theorem.

Theorem 2.1. Let the sequences $\{g_k\}$ and $\{d_k\}$ be generated by the $RMILHS$ algorithm. Then, the search direction satisfies the sufficient descent condition:

$$g_k^T d_k^{RMILHS} \leq -c\|g_k\|^2, \text{ for all } k. \tag{17}$$

P r o o f. From the $RMILHS$ algorithm, we know that if the restart criterion of Powell holds, then $d_k = -g_k$ and (17) holds.

So, we assume that Powell criterion doesn't hold. Then, we have

$$\mid g_{k+1}^T g_k \mid < 0.2\|g_{k+1}\|^2. \tag{18}$$

The following proof is by induction.

If $k = 0$, then $g_0^T d_0 = -\|g_0\|^2$, so (17) holds.

Next, for $k > 0$ we have

$$d_{k+1} = -g_{k+1} + \beta_k^{RMILHS} d_k,$$

which can be written as

$$\begin{aligned} d_{k+1} &= -(\theta_k g_{k+1} + (1-\theta_k)g_{k+1}) + ((1-\theta_k)\beta_k^{RMIL} + \theta_k \beta_k^{HS})d_k \\ &= \theta_k(-g_{k+1} + \beta_k^{HS} d_k) + (1-\theta_k)(g_{k+1} + \beta_k^{RMIL} d_k). \end{aligned}$$

It follows that:

$$d_{k+1} = \theta_k d_{k+1}^{HS} + (1-\theta_k) d_{k+1}^{RMIL}. \tag{19}$$

Multiplying (19) by g_{k+1}^T from the left side, we get

$$g_{k+1}^T d_{k+1} = \theta_k g_{k+1}^T d_{k+1}^{HS} + (1-\theta_k) g_{k+1}^T d_{k+1}^{RMIL}. \tag{20}$$

Firstly, if $\theta_k = 0$, d_k coincides with descent direction of Rivaie et al., $d_{k+1}^{RMIL} = -g_{k+1} + \beta_k^{RMIL} d_k$, where they proved in [25] that

$$g_{k+1}^T d_{k+1}^{RMIL} \leq -c_1\|g_{k+1}\|^2, \text{ for all } k. \tag{21}$$

Now, if $\theta_k = 1$, d_k coincides with descent direction of Hesteness-Steifel, $d_{k+1}^{HS} = -g_{k+1} + \beta_k^{HS} d_k$, where Djordjevic proved in [10] that

$$g_{k+1}^T d_{k+1}^{HS} \leq -c_2\|g_{k+1}\|^2, \tag{22}$$

with

$$c_2 = \frac{1 - 2.2\sigma}{1 - \sigma}, \text{ and } \sigma < \frac{5}{11}.$$

Finally, if $0 < \theta_k < 1$, then there exist scalars λ_1 and λ_2, such that

$$0 < \lambda_1 \leq \theta_k \leq \lambda_2 < 1. \tag{23}$$

From the formula (20) and using (23), we conclude

$$g_{k+1}^T d_{k+1} \leq \lambda_1 g_{k+1}^T d_{k+1}^{HS} + (1 - \lambda_2) g_{k+1}^T d_{k+1}^{RMIL}. \tag{24}$$

Let $c = \lambda_1 c_2 + (1 - \lambda_2) c_1$, then from (21), (22) and (24) we finally get

$$g_{k+1}^T d_{k+1} \leq -c \|g_{k+1}\|^2.$$

□

Lemma 2.2. Suppose that assumptions (i) and (ii) hold. Consider common iterate (2), where d_k is a descent direction (5) and α_k is determined by the strong Wolfe line search (6) and (7). Then, the Zoutendjik condition

$$\sum_{k \geq 0} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty, \tag{25}$$

holds.

P r o o f . The proof follows directly from [29]. □

Theorem 2.3. Consider the $RMILHS$ conjugate gradient method and suppose that assumptions (i), (ii) and (17) hold. Then either $g_k = 0$ for some k, or

$$\lim_{k \to \infty} \inf \|g_k\| = 0. \tag{26}$$

P r o o f . Suppose that $g_k \neq 0$, for all k. Then we have to prove (26).

Suppose, on the contrary, that (26) doesn't hold. Then there exists a constant $t > 0$, such that

$$\|g_k\| \geq t, \text{ for all } k. \tag{27}$$

Let D be the diameter of the level set $\mathcal{L}$.

From the formula of β_k^{RMILHS} we get

$$\left|\beta_k^{RMILHS}\right| \leq \left|\beta_k^{RMIL}\right| + \left|\beta_k^{HS}\right| = \frac{\left|g_{k+1}^T y_k\right|}{\|d_k\|^2} + \left|\frac{g_{k+1}^T y_k}{d_k^T y_k}\right|. \tag{28}$$

Further, using the second strong Wolfe condition (7), we get

$$y_k^T d_k = g_{k+1}^T d_k - g_k^T d_k \geq (\sigma - 1) g_k^T d_k = -(1 - \sigma) g_k^T d_k > 0, \tag{29}$$

which gives

$$\frac{1}{y_k^T d_k} \leq \frac{1}{-(1-\sigma) g_k^T d_k}. \tag{30}$$

The direction d_k satisfies (17), so it holds:

$$-g_k^T d_k \geq c\|g_k\|^2,$$

which implies using (27) that

$$\frac{-1}{g_k^T d_k} \leq \frac{1}{c\|g_k\|^2} \leq \frac{1}{ct^2}. \tag{31}$$

So, from (31) the formula (30) satisfies

$$\frac{1}{y_k^T d_k} \leq \frac{1}{-(1-\sigma) g_k^T d_k} \leq \frac{1}{(1-\sigma)ct^2}. \tag{32}$$

Now, using (15) and (16), we get

$$\left|g_{k+1}^T y_k\right| \leq \|g_{k+1}\| \, \|g_{k+1} - g_k\| \leq \mu L \|x_{k+1} - x_k\| \leq \mu L \|s_k\| \leq \mu L D. \tag{33}$$

Then, from (32) and (33), we have

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k} \leq \frac{\mu L D}{(1-\sigma)ct^2}. \tag{34}$$

On the one hand and using (15), we have

$$y_k^T d_k \leq \|y_k\| \, \|d_k\| \leq L \, \|s_k\| \, \|d_k\| = L\alpha_k \|d_k\|^2,$$

then

$$\|d_k\|^2 \geq \frac{1}{L\alpha_k} y_k^T d_k. \tag{35}$$

So, using (29) and (17) the formula (35) becomes

$$\|d_k\|^2 \geq -\frac{1}{L\alpha_k}(1-\sigma) g_k^T d_k \geq \frac{1}{L\alpha_k}(1-\sigma)c\|g_k\|^2.$$

Using (27), we get

$$\frac{1}{\|d_k\|^2} \leq \frac{L\alpha_k}{(1-\sigma)c\|g_k\|^2} \leq \frac{L\alpha_k}{(1-\sigma)ct^2}. \tag{36}$$

From (33), (36) and the formula of β_k^{RMIL}, we obtain

$$\left|\beta_k^{RMIL}\right| = \frac{\left|g_{k+1}^T y_k\right|}{\|d_k\|^2} \leq \frac{\mu L^2 D}{(1-\sigma)ct^2}\alpha_k. \tag{37}$$

Now, from (28), (34) and (37) we get

$$|\, \beta_k^{RMILHS} \,| \leq \frac{\mu L D}{(1-\sigma)ct^2}(L\alpha_k + 1). \tag{38}$$

Next, we are going to prove as in [9] that there exists $\alpha_* > 0$, such that

$$\alpha_k \geq \alpha_* > 0, \text{ for all } k.$$

Suppose, on the contrary, that there isn't an α_*, such that $\alpha_k \geq \alpha_* > 0$. Then there exists an infinite subsequence $\alpha_k = \gamma^{j_k}$, $k \in K_1$ such that

$$\lim_{k \in K_1} \alpha_k = 0.$$

Then

$$\lim_{k \in K_1} \gamma^{j_k - 1} = 0, \tag{39}$$

i. e.,

$$\lim_{k \in K_1} j_k - 1 = \infty.$$

Now, from (6) we get

$$f(x_k + \gamma^{j_k} d_k) - f(x_k) \leq \delta \gamma^{j_k} g_k^T d_k, \tag{40}$$

$$f(x_k + \gamma^{j_k - 1} d_k) - f(x_k) > \delta \gamma^{j_k - 1} g_k^T d_k, \tag{41}$$

where $\delta < 1$. From (41), we have

$$\frac{f(x_k + \gamma^{j_k - 1} d_k) - f(x_k)}{\gamma^{j_k - 1}} > \delta g_k^T d_k. \tag{42}$$

Using (39) and passing to the limit of (42), we conclude that

$$g_k^T d_k \geq \delta g_k^T d_k. \tag{43}$$

But, our method satisfies the sufficient descent, so $g_k^T d_k \leq 0$.

Also, $0 < \delta < 1$, so, the relation (43), is correct only if $g_k^T d_k = 0$. Then, from the second strong Wolfe condition (7), we get that $g_{k+1}^T d_k = 0$, which correspond to the exact line search. So, we have a contradiction.

Now,we can write

$$\|d_{k+1}\| \leq \|g_{k+1}\| + \mid \beta_k^{RMILHS} \mid \|d_k\|. \tag{44}$$

As $s_k = \alpha_k d_k$, we write $d_k = \dfrac{s_k}{\alpha_k}$. So, from (16) and (38), we have

$$\|d_{k+1}\| \leq \mu + \frac{\mu L D}{(1-\sigma) c t^2} (L \alpha_k + 1) \frac{\|s_k\|}{\alpha_k}.$$

Since $\alpha_k \geq \alpha_*$ and $\|s_k\| \leq D$, it follows that

$$\|d_{k+1}\| \leq \mu + \frac{\mu L D^2 (L \alpha_* + 1)}{\alpha_* (1-\sigma) c t^2} = Cste, \tag{45}$$

which gives

$$\sum_{k\geq 1}\frac{1}{\|d_k\|^2}=\infty. \tag{46}$$

On the other hand, from (17), (27) and from the Zoutendijk condition (25), it results that

$$c^2t^4\sum_{k\geq 0}\frac{1}{\|d_k\|^2}\leq\sum_{k\geq 0}\frac{c^2\|g_k\|^4}{\|d_k\|^2}\leq\sum_{k\geq 0}\frac{(g_k^Td_k)^2}{\|d_k\|^2}<\infty,$$

which contradicts (46). Therefore, (27) doesn't hold.

Then, $\lim\limits_{k\to\infty}\inf\|g_k\|=0$. This completes the proof. □

3. NUMERICAL EXPERIMENTS

In this section, we will study the effectiveness of our algorithm ($RMILHS$) on two parts. In this first part, we consider an unconstrained optimization problem of the form (1). To show the effectiveness and to measure the convergence behavior of our algorithm ($RMILHS$), we applied it on a set of test functions taken from [1] in comparing with some existing methods, namely $HSFR$ method [10], $LSCD$ method [9] and $LSDY$ method [28]. We take several dimensions, from $n = 10$ to $n = 10000$ and we considered the precision $\varepsilon = 10^{-6}$.

We used Matlab language on HP laptop with Intel(R) Core(TM) i5-6300U CPU @ 2.40 GHz processor and 8GB RAM memory and Windows 8.1.

We applied here the iterations number, CPU time and the gradient evaluations. The performance profile of Dolan and Moré [11] offers a systematic means to evaluate and compare the performance of a set of solvers S on a set P of problems. Assuming n_p problems and n_s solvers, for each problem p and solver s, they define $t_{p,s}$ the computing time required to solve problem p by solver s. A need of reference point for comparisons. They evaluate the performance of solver s on problem p against the optimal performance achieved by any solver on the same problem using the performance ratio

$$r_{p,s}=\frac{t_{p,s}}{\min\{t_{p,s}:s\in S\}}.$$

They suppose that a parameter $r_M\geq r_{p,s}$ for all chosen p, s, and $r_{p,s}=r_M$ if and only if solver s does not solve problem p. Define

$$\rho_s(\tau)=\frac{1}{n_p}size\{p\in P:\log_2 r_{p,s}\leq\tau\},$$

where $\rho_s(\tau)$ is the probability for solver $s\in S$ that the performance ratio $r_{p,s}$ is within a factor $\tau\in\mathbb{R}$ of the best possible ratio. The function ρ_s is the (cumulative) distribution function for the performance ratio. The value of $\rho_s(0)$ is the probability that the solver will win over the rest of the solvers. Table 1 represents several test functions that are used in this experiments for different dimensions and different choices of the initial points.

Test function	Dimension	Initial point
Raydan 1	$10, 100, 500, 1000, 2000$	$(1, \ldots, 1)$
Raydan 2	$10, 100, 500, 1000, 3000$	$(4, \ldots, 4)$
Diagonal 4	$10, 100, 500, 1000, 10000$	$(1, \ldots, 1)$
Extended Woods	$10, 100, 500, 1000, 10000$	$(2, \ldots, 2)$
HIMMELBG	$10, 100, 500, 1000, 10000$	$(1.5, \ldots, 1.5)$
Extended Block-Diagonal 1	$10, 100, 500, 1000, 10000$	$(1, \ldots, 1)$
Prod1	$10, 100, 500, 1000, 10000$	$(1, \ldots, 1)$
Extended Maratos	$10, 100, 500, 1000, 10000$	$(0.1, \ldots, 0.1)$
Pertubed Quadratic	$10, 100, 500, 1000, 10000$	$(0.5, \ldots, 0.5)$
Extended White and Holst	$10, 100, 500, 1000, 10000$	$(1.5, \ldots, 1.5)$
Diagonal 2	$10, 100, 500, 1000, 10000$	$(1, \frac{1}{2}, \ldots, \frac{1}{n})$
POWER	$10, 100, 500, 1000, 10000$	$(1, \ldots, 1)$
Hager	$10, 100, 500, 1000, 5000$	$(1, \ldots, 1)$
DENSCHNA	$10, 100, 500, 1000, 10000$	$(1, \ldots, 1)$
HIMMELBC	$10, 100, 500, 1000, 10000$	$(1, \ldots, 1)$
Extended TET	$10, 100, 500, 1000, 2000$	$(0.1, \ldots, 0.1)$
Extended Cliff	$10, 100, 500, 1000, 5000$	$(1, \ldots, 1)$
Tridia	$10, 100, 500, 1000, 10000$	$(\frac{1}{n}, \ldots, \frac{1}{n})$
Diagonal 5	$10, 100, 500, 1000, 3000$	$(1.1, \ldots, 1.1)$
ARWHEAD	$10, 100, 500$	$(1, \ldots, 1)$
QUARTC	$10, 100, 500, 1000, 10000$	$(2, \ldots, 2)$

Tab. 1. The test functions and their dimensions with the initial points.

In the second part, we applied our algorithm to image restoration problems with the use of the two-phase scheme.

As known, images can get corrupted due to various factors including noise during acquisition. In this part, we are going to deal with restorations of an image corrupted by impulse noise problems. Impulse noise is one of the most common noise models, where only a portion of the pixels is contaminated by the noise and the information on the true values of these pixels is completely lost. Chan et al. [5] have applied the two-phase scheme to restore a corrupted image. Several researchers have used these two phases to make conjugate gradient algorithms capable of restoring images corrupted by impulse noise, even though the noise ratio is high or even reaches 90% [16, 18, 19].

The two-phase scheme applied can be briefly described as follows. In the first phase, we use adaptive median filter to detect noisy pixels. In the second phase the noise from the corrupted pixels is removed by solving the following smooth problem (47) proposed by Cai et al.[4]. We used our proposed $RMILHS$ algorithm to solve the problem (47), in comparison with $HSFR$, $LSCD$ and $LSFR$ methods.

Let X be an image of size M-by-N and $\mathcal{A} = \{1, 2, \ldots, M\} \times \{1, 2, \ldots, N\}$ be the index set of the image X. We minimize $F_\alpha(u)$ where

$$F_\alpha(u) = \sum_{(i,j)\in\mathcal{B}} \left(2 \sum_{(m,n)\in\mathcal{V}_{i,j}\setminus\mathcal{B}} \varphi_\alpha(u_{i,j} - y_{m,n}) + \sum_{(m,n)\in\mathcal{V}_{i,j}\cap\mathcal{B}} \varphi_\alpha(u_{i,j} - u_{m,n}) \right), \quad (47)$$

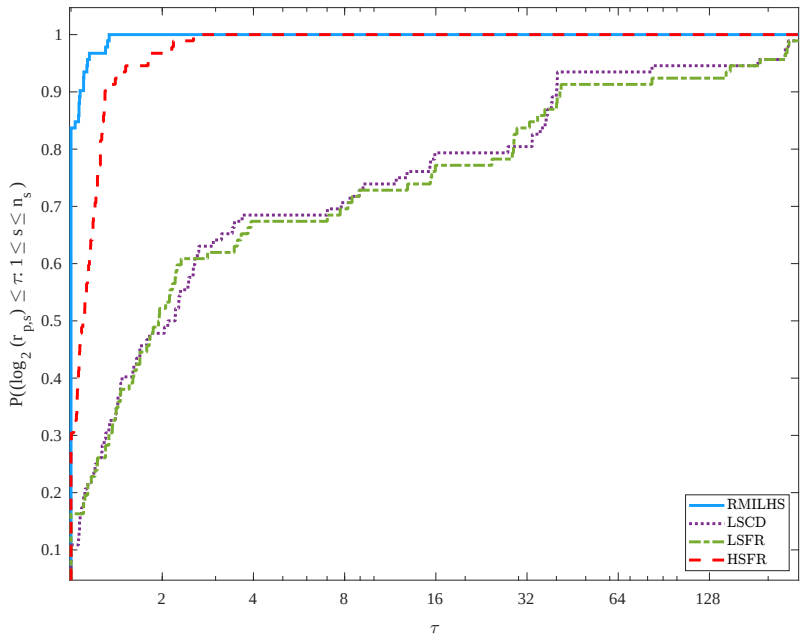

Fig. 1. Performance profile based on the iteration number.

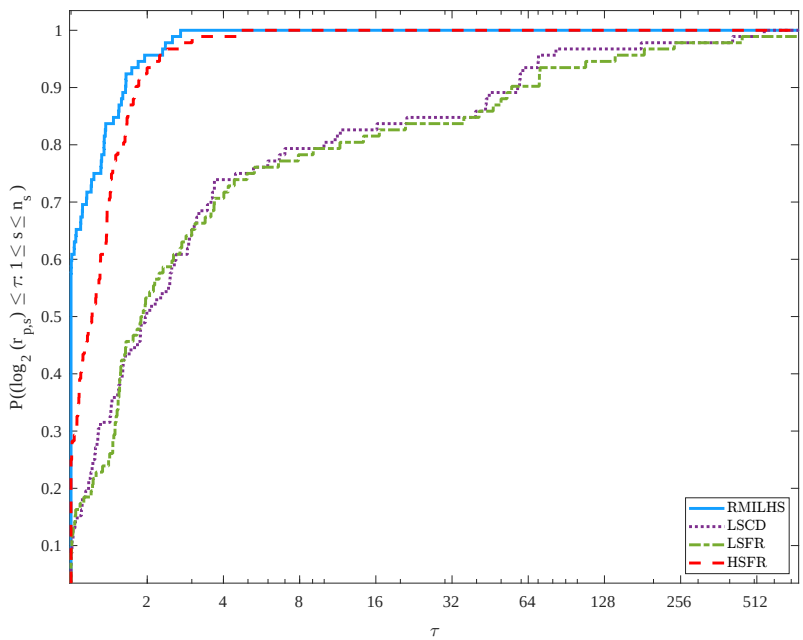

Fig. 2. Performance profile based on the CPU time.

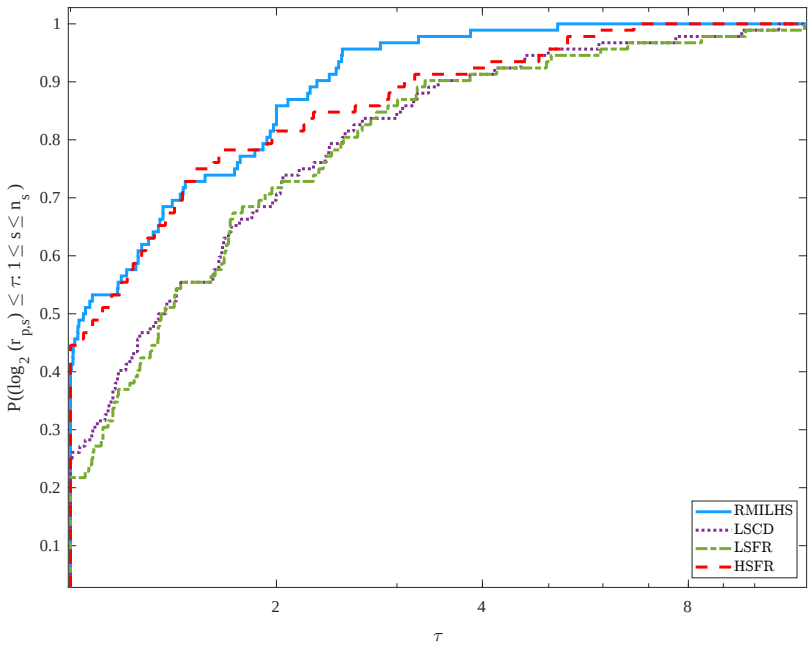

Fig. 3. Performance profile based on gradient evaluations.

in which $\mathcal{B} \subset \mathcal{A}$, the set of indices of the noise pixels detected from the first phase and η its number of elements. $\mathcal{V}_{i,j} = \{(i, j-1), (i, j+1), (i-1, j), (i+1, j)\}$ is the set of the four closest neighbors for the pixel at pixel location, for all $(i, j) \in \mathcal{A}$, and $y_{i,j}$ be the observed pixel value of the image at pixel location (i, j). Also $\varphi_\alpha(t)$ is an edge-preserving functional which is chosen as $\varphi_\alpha(t) = \sqrt{t^2 + \alpha}$, in our tests we set $\alpha = 100$. We use the peak signal to noise ratio (PSNR), defined by

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} \left(x^r_{i,j} - x^*_{i,j}\right)^2},$$

where $x^r_{i,j}$ and $x^*_{i,j}$ denote the pixel values of the restored image and the original one, respectively. We tested Lena (512×512), Man (512×512) and Pepper (512×512). We also set the stopping criteria as

$$Itr > 300 \text{ or } \frac{|F_\alpha(u_k) - F_\alpha(u_{k-1})|}{F_\alpha(u_k)} \leq 10^{-4}.$$

The noise levels of the salt-and-pepper noise used are as follows: 30%, 50%, 70%, and 90%.

In the following figures (Fig. 4 and Fig. 5), we present the most significant results of the noisy images which correspond to 70% and 90%.

Fig. 4. The original images, the noisy images with 70% salt-and-paper noise and the restored images by RMILHS , HSFR, LSCD and LSFR.

Fig. 5. The original images, the noisy images with 90% salt-and-paper noise and the restored images by RMILHS , HSFR, LSCD and LSFR.

Images	Methods	RMILHS			HSFR			LSCD			LSFR		
	Ratio	ITER	CPU	PSNR	ITER	CPU	PSNR	ITER	CPU	PSNR	ITER	CPU	PSNR
Lena	30%	21	11.34	38.07	21	11.09	38.07	19	10.42	38.06	19	10.41	38.06
	50%	24	18.23	35.53	24	18.18	35.53	22	16.82	35.50	22	16.79	35.50
	70%	29	27.02	32.22	29	27.10	32.22	26	23.00	32.21	26	22.87	32.21
	90%	34	44.07	27.21	34	43.88	27.21	44	57.73	27.07	44	57.73	27.07
Man	30%	22	16.25	34.46	22	14.64	34.46	19	10.43	34.43	19	10.49	34.43
	50%	23	18.66	32.02	23	18.49	32.02	22	16.88	32.00	22	16.87	32.00
	70%	29	29.53	29.04	29	27.53	29.04	27	24.79	29.06	27	22.57	29.06
	90%	39	49.07	25.02	39	49.58	25.02	49	58.29	24.96	49	58.49	24.96
Pepppers	30%	23	16.28	34.08	23	16.25	34.08	19	13.11	34.15	19	12.96	34.15
	50%	25	19.64	32.37	25	19.92	32.37	23	20.28	32.33	23	18.32	32.33
	70%	30	32.79	29.93	30	31.83	29.93	29	33.16	29.91	29	31.12	29.91
	90%	37	51.58	25.96	37	51.70	25.96	51	70.91	25.77	51	71.58	25.77

Tab. 2. Numerical results for image restoration problem.

The obtained results of the Table 2 are converted into a bar chart so that the subtle differences can be seen better.

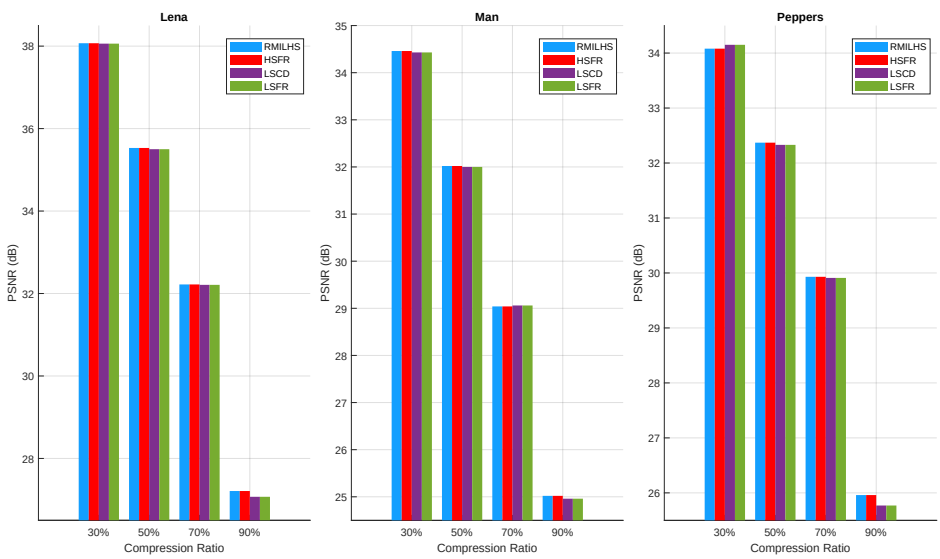

Fig. 6. PSNR Comparison for different methods and images.

3.1. Commentaries

Concerning the first part, based on the performance profiles depicted in Figures 1, 2 and 3 for the number of iterations, CPU time and gradient evaluations, respectively, we can say that our new approach employing β_k^{RMILHS} is more efficient than the other considered conjugate gradient methods.

For the second part, the numerical results cited in Table 2, Figures 4, 5 and 6 show the superiority of the LSCD and LSFR approaches compared to our RMILHS approach. We observe either an equality or a slight increase in the number of iterations or computation time in RMILHS compared to LSCD, LSFR, and HSFR. On the other hand, our approach RMILHS and also HSFR offer a better PSNR compared to LSCD and LSFR.

4. CONCLUSION

We have proposed a new parameter β_k, which is the convex combination of β_k^{RMIL} and β_k^{HS}. We have provided proof of the global convergence of the algorithm under the strong Wolfe line search. The performance of the proposed new method is significantly better than other existing conjugate gradient methods, namely HSFR, LSCD and LSFR, for unconstrained optimization problems. On the other hand, applied to restoration image problems, our method is competitive with the considered methods and it is even better when the image's noise is higher.

5. ACKNOWLEDGMENTS

The authors would like to thank the anonymous referee for his useful comments and suggestions, which helped to improve the presentation of this paper.

(Received April 1, 2024)

REFERENCES

[1] N. Andrei: An unconstrained optimization test functions collection. Adv. Model. Optim. *10* (2008), 147–161. DOI:10.1002/adem.200890003

[2] N. Andrei: Nonlinear conjugate gradient methods for unconstrained optimization. Springer Optimization and its Applications, Romania 2020.

[3] S. Ben Hanachi, B. Sellami, and M. Belloufi: New iterative conjugate gradient method for nonlinear unconstrained optimization. RAIRO – Oper. Res. *56* (2022), 2315–2327. DOI:10.1051/ro/2022109

[4] J. F. Cai, R. Chan, and B. Morini: Minimization of an edge-preserving regularization functional by conjugate gradient type methods. Image Processing Based on Partial Differential Equations. Mathematics and Visualization. Springer, Berlin, Heidelberg 2007.

[5] R. H. Chan, C. W. Ho, M. Nikolova: Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE Trans. Image Process. *14* (2005), 10, 1479-1485. DOI:10.1109/TIP.2005.852196

[6] Y. H. Dai and Y. Yuan: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. *10* (1999), 1, 177–182. DOI:10.1137/S1052623497318992

[7] Y. H. Dai and Y. Yuan: An efficient hybrid conjugate gradient method for unconstrained optimization. Ann. Oper. Res. *103* (2001), 33–47.

[8] S. Delladji, M. Belloufi, and B. Sellami: New hybrid conjugate gradient method as a convex combination of FR and BA methods. J. Inform. Optim. Sci. *42* (2021), 3, 591–602.

[9] S. Djordjevic: New hybrid conjugate gradient method as a convex combination of LS and FR methods. Acta Math. Scientia *39B* (2019), 1, 214–228. DOI:10.1007/s10473-019-0117-6

[10] S. Djordjevic: New hybrid conjugate gradient method as a convex combination of HS and FR methods. J. Appl. Math. Comput. *2* (2018), 9, 366–378.

[11] E. D. Dolan and J. J. Moré: Benchmarking optimization software with performance profiles. Math. Program. *91* (2002), 201–213. DOI:10.1007/s101070100263

[12] R. Fletcher: Practical Methods of Optimization. Unconstrained Optimization. Wiley, New York 1987.

[13] R. Fletcher and C. M. Reeves: Function minimization by conjugate gradients. Comput. J. *7* (1964), 2, 149–154. DOI:10.1093/comjnl/7.2.149

[14] W. W. Hager and H. Zhang: A survey of nonlinear conjugate gradient methods. Pacific J. Optim. *2* (2006), 35–58.

[15] M. R. Hestenes and E. Steifel: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. *49* (1952), 6, 409–436. DOI:10.6028/jres.049.044

[16] X. Jiang, W. Liao, J. Yin, and J. Jian: A new family of hybrid three-term conjugate gradient methods with applications in image restoration. Numer. Algor. *91* (2022), 161–191. DOI:10.1007/s11075-022-01258-2

[17] Y. Liu and C. Storey: Efficient generalized conjugate gradient algorithm. Part 1: Theory. J. Optim. Theory Appl. *69* (1991), 1, 129–137. DOI:10.1007/BF00940464

[18] G. Ma, H. Lin, W. Jin, and D. Han: Two modified conjugate gradient methods for unconstrained optimization with applications in image restoration problems. J. Appl. Math. Comput. *68* (2022), 4733–4758. DOI:10.1007/s12190-022-01725-y

[19] M. Malik, I. M. Sulaiman, A. B. Abubakar, G. Ardaneswari, and A. Sukono: A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection. AIMS Math. *8* (2023), 1–28. DOI:10.1155/2023/8851478

[20] P. Mtagulwa and P. Kaelo: A convergent modified HS-DY hybrid conjugate gradient method for unconstrained optimization problems. J. Inform. Optim. Sci. *40* (2019), 1, 97–113.

[21] E. Polak and G. Ribiere: Note sur la convergence des méthodes de directions conjuguées. Rev. Française Inform. Recherche Opertionelle *16* (1969), 35–43.

[22] B. T. Polyak: The conjugate gradient method in extreme problems. U.S.S.R. Comput. Math. Phys. *9* (1969), 94–112.

[23] M. J. D. Powell: Restart procedures of the conjugate gradient method. Math. Program. *2* (1977), 241–254.

[24] M. Rivaie, M. Mustafa, W. J. Leong, and M. Ismail: A new class of nonlinear conjugate gradient coefficients with global convergence properties. Appl. Math. Comput. *218* (2012), 22, 11323–11332. DOI:10.1016/j.amc.2012.05.030

[25] M. Rivaie, M. Mustafa, and A. Abdelrhaman: A new class of non linear conjugate gradient coefficients with exact and inexact line searches. Appl. Math. Comput. *268* (2015), 1152–1163. DOI:10.1016/j.amc.2015.07.019

[26] B. Sellami and Y. Chaib: A new family of globally convergent conjugate gradient methods. Ann. Oper. Res. Springer *241* (2016), 497–513. DOI:10.1007/s10479-016-2120-9

[27] B. Sellami and Y. Chaib: New conjugate gradient method for unconstrained optimization. RAIRO Oper. Res. *50* (2016), 1013–1026. DOI:10.1051/ro/2015064

[28] X. Yang, Z. Luo, and X. Dai: A global convergence of LS-CD hybrid conjugate gradient method. Adv. Numer. Anal. *2013* (2013), 5 pp.

[29] G. Zoutendijk: Nonlinear programming, computational methods. In: Integer and Nonlinear Programming (J. Abadie, ed.), 1970, pp. 37–86.

Youcef Elhamam Hemici, Laboratory of Fundamental and Numerical Mathematics, Department of Mathematics, Faculty of Sciences, Setif1 University-Ferhat Abbas, 19000. Algeria.
e-mail: youcefelhamam.hemici@univ-setif.dz

Samia Khelladi, Laboratory of Fundamental and Numerical Mathematics, Department of Mathematics, Faculty of Sciences, Setif1 University-Ferhat Abbas, 19000. Algeria.
e-mail: samia.boukaroura@univ-setif.dz

Djamel Benterki, Laboratory of Fundamental and Numerical Mathematics, Department of Mathematics, Faculty of Sciences, Setif1 University-Ferhat Abbas, 19000. Algeria.
e-mail: djbenterki@univ-setif.dz