Hybrid Modified Accelerated Gradient Method for Optimization Processes

Milena J. Petrović *, Ana Vučetić and Tanja Jovanović Spasojević

Faculty of Sciences and Mathematics, University of Pristina in Kosovska Mitrovica, Lole Ribara 29, 38220 Kosovska Mitrovica, Serbia; ana.vucetic@pr.ac.rs (A.V.); tanja.jovanovic@pr.ac.rs (T.J.S.)

* Correspondence: milena.petrovic@pr.ac.rs

Abstract: This research reveals a hybrid variant of the modified accelerated gradient method. We prove that derived iteration is linearly convergent on the set of uniformly convex functions. Performance profiles of the introduced hybrid method were numerically compared with its non-hybrid version. The analyzed characteristics were CPU time, the number of iterations and the number of function evaluations. The results of the numerical experiments show a better performance in favor of the derived hybrid accelerated model compared with its forerunner.

Keywords: gradient descent; line search; accelerated gradient methods; quasi-Newton method; convergence rate

MSC: 90C30; 90C06; 49M37; 65K99; 47H09; 47H10

1. Accelerated Gradient Optimization Schemes

To find a proper way to define the efficient optimization model for solving a general unconstrained minimization problem

$$\min f(x), \ x \in \mathbb{R}^n,$$

where $f$ is a uniformly convex and twice continuously differentiable objective function, many authors start from the common iterative rule

$$x_{k+1} = x_k + t_k d_k.$$  \hfill (2)

In (2), $x_{k+1}$ represents the next iterative point, $x_k$ represents the current one, $t_k$ is the value of the iterative step length and $d_k$ is the iterative search direction. Evidently, variables $t_k$ and $d_k$ directly impact the performance characteristics of a certain method. The iterative step-size parameter is usually calculated via some inexact line search procedures. For the search direction vector, it is expected to fulfill the descending condition:

$$g_k^T d_k < 0,$$  \hfill (3)

where $g_k^T$ represents the gradient of the objective function $f$ at the iterative point $x_k$. One of the crucial characteristics of an optimization method is the way of defining the search vector, i.e., direction vector. Although the direction vector $d_k$ may be constructed differently, the descent condition (3) generally ensures a fast convergence to local minima. Historically, Cauchy was the first one who suggested the oldest gradient descent GD method (4). This iteration is defined when the negative of the gradient direction is put in (3), i.e., $d_k = -g_k$:

$$x_{k+1} = x_k - t_k g_k.$$

**Note:** The citation and metadata at the bottom of the page are placeholders and not part of the natural text. The text is a complete and unabridged version of the original document.
Further on, various gradient descent directions were defined and, accordingly, many efficient gradient descent minimization methods. We mention here, for example, only a few significant schemes and highlight its gradient descent vector directions.

In the second-order convergent Newton method, the search direction is defined as the solution to the system of nonlinear equations \( G_k \mathbf{d} = -\mathbf{g}_k \) with respect to \( \mathbf{d} \), where \( G_k := \nabla^2 f(x_k) \) presents the Hessian of the objective function. Therewith, the Newton method with line search procedure is expressed as follows:

\[
x_{k+1} = x_k - t_k G_k^{-1} \mathbf{g}_k.
\]  

(5)

In (5) \( G_k^{-1} \) is the Hessian of the function \( f \) at the point \( x_k \), and the gradient descent search direction is defined as \( -G_k^{-1} \mathbf{g}_k \).

An expansion of the iteration (2) can be expressed as

\[
x_{k+1} = x_k - t_k \gamma_k^{-1} \mathbf{g}_k.
\]  

(6)

The additional parameter \( \gamma_k \) in (6) is the acceleration factor of the given optimization model.

Iterations (6) are classified as accelerated gradient schemes in [1]. This iterative type arrives from Newton’s method with the line search (5) with the search direction vector generated as \( -\gamma_k^{-1} \mathbf{g}_k \).

The Hessian of the objective function reveals much regarding the function’s properties. Still, the computational effort of calculating the function’s Hessian could be quite demanding. Consequently, the needed computational time and the number of iterations grow. These all can affect the general performance metrics of the analyzed method. For that reason, the accelerated gradient methods (6) present a satisfying substitution of the classical Newton method (5). In the contemporary literature, many authors present significant accelerated methods of the type (6). Many of these iterations show high convergence properties and strong effective aspects.

Chronologically, the authors in [1] introduced the class of accelerated gradient method and presented the SM method, defined as (6), with the accelerated parameter

\[
\gamma_{k+1}^{SM} = 2\gamma_k \frac{f(x_{k+1}) - f(x_k) + t_k \| \mathbf{g}_k \|^2}{t_k^2 \| \mathbf{g}_k \|^2}.
\]

In [1], the domination of the SM iteration among the classical gradient method \( x_{k+1} = x_k - t_k \mathbf{g}_k \) and the accelerated gradient method introduced in [2] is confirmed.

Inspired by the results presented in [1], the authors in [3,4] studied the variants of the accelerated gradient methods with double directions and double step sizes. From this research, the accelerated double direction and accelerated double step size iterations, so called the ADD (7) and the ADSS (8) iterations methods, arose:

\[
x_{k+1} = x_k + t_k^2 \mathbf{d}_k - t_k \gamma_k^{ADD-1} \mathbf{g}_k.
\]  

(7)

\[
x_{k+1} = x_k - t_k \gamma_k^{ADSS-1} \mathbf{g}_k - p_k \mathbf{g}_k = x_k - \left( t_k \gamma_k^{ADSS-1} + p_k \right) \mathbf{g}_k.
\]  

(8)

In (7) and (8), \( t_k \) and \( p_k \) are two step sizes obtained through the Backtracking algorithms, while in (7), the vector direction \( \mathbf{d}_k \) is calculated via the specially defined procedure. The variables \( \gamma_k^{ADD-1} \) and \( \gamma_k^{ADSS-1} \) are adequately defined acceleration parameters for the ADD and the ADSS, respectively. In [3], the reduction of the number of iterations is achieved in favor of the ADD scheme when compared with the SM method. With that, an important study presented in [3] illustrates the comparison between the accelerated and non-accelerated version of the ADD method and unequivocally confirms the better performance characteristics of the accelerated variant. In [4], the dominance of the proposed accelerated double step size method regarding the analyzed metrics, in comparison with
the SM model from [1], is numerically recorded. The excellent performance features of the ADSS scheme were given further motivation in generating the transformed accelerated double step size (TDASS) method, which presents a specific variant of the ADSS iteration where two step length parameters fulfill a special condition $t_k + p_k = 1$

$$x_{k+1} = x_k - [t_k(\gamma_k^{TDASS-1} - 1) + 1]g_k.$$  (9)

Taking the accelerated properties that we mentioned above, as well as Khan’s hybridization technique, described in the Section 2, we aim to define the efficient gradient descent method for solving large-scale optimization problems. Our intention is also to apply the developed model in solving a system of nonlinear equations in some further studies. An efficient system for solving the nonlinear equations

$$F(x) = 0, \ x \in \mathbb{R}^n,$$  (10)

where $F$ is the mapping $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $F(x) = (F_1(x), \ldots, F_n(x))$, and $F_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is the $i$th component of $F$, is one of the most important applications of an optimization method. Under the assumption that the mapping of $F$ is continuously differentiable, the general optimization problem (10) is equivalent to the minimization problem of the objective function $f$ in (1), where

$$f(x) = \frac{1}{2} \|F(x)\|^2 = \frac{1}{2} \sum_{i=1}^{n} (F_i(x))^2.$$  (11)

Using the common technique, which is based on the equivalence of the relations (10) and (1), when function $f$ in (1) is defined as (11), it is possible to apply the method that we introduce in this paper to solve various systems of nonlinear equations.

The application of a certain optimization method in solving a system of nonlinear equations is not the only interest of the optimization community. Optimization processes are generally of the multimodal kind. To solve these problems, various global and local techniques are required [5-7]. Our developed model may be considered as a loop of one complex optimization multimodal process. In the sphere of the global–local strategy, we may investigate the modes to apply the method that is introduced within this paper.

2. Hybridization of the Modified Accelerated Gradient Method

In [8], author proposed a three-term method

$$\begin{cases} 
  x_1 = x \in \mathbb{R}, \\
  x_{k+1} = Ty_k, \\
  y_k = (1 - \alpha_k)x_k + \alpha_k T x_k, \quad k \in \mathbb{N}
\end{cases}$$  (12)

where $\alpha_k \in (0,1)$. This iterative rule was a motivation for developing many hybrid gradient optimization models. The main idea of this hybridization approach resulted in the hybrid SM method, shortly denoted as HSM method. Based on the same Khan’s three-term iteration, the hybrid double direction method, the hybrid double step-sized and the hybrid transformed accelerated models are developed. In [9], authors proposed a whole class of hybrid iterative processes guided by the same Khan’s principle. Many of mentioned hybrid methods are later involved in solving systems equations.

In this section, we present the hybridization of the modified accelerated method, signed as modADS and presented in [10]:

$$x_{k+1} = x_k - \left(t_k \gamma^{modADS-1} + t_k^2\right)g_k \equiv x_k - t_k \gamma^{modADS-1}g_k - t_k^2g_k.$$  (13)

In (13) step length parameter $t_k$ is obtained via the Backtracking algorithm:

1. Objective function $f(x)$, the direction $d_k$ of the search at the point $x_k$ and numbers $0 < \sigma < 0.5$ and $\beta \in (0,1)$ are required;
We assume that the accelerated parameter arises directly from the expression (22), since the denominator

\( \gamma_{k+1} \) for \( \alpha \) because, under this assumption, the second-order necessary and the second order-sufficient conditions are fulfilled. The positiveness of accelerated factor arises directly from the expression (22), since the denominator

\[ \alpha^2 \omega_k^2 \|g_k\|^2 > 0 \]
and the numerator of the expression (22) can be estimated as

\[
f(x_{k+1}) - f(x_k) + a\omega_k\|g_k\|^2 \approx -a\omega_k\|g_k\|^2 + \frac{1}{2}a^2\omega_k^2\gamma_{k+1}\|g_k\|^2 + a\omega_k\|g_k\|^2
\]

\[
= \frac{1}{2}a^2\omega_k^2\|g_k\|^2 > 0.
\]

Still, it may happen that some iterative value of the accelerated parameter (22) is a negative one. In that case, we set \( \gamma_{k+1}^{\text{HmodADS}} = 1 \) and calculate the next iteration as

\[
x_{k+2} = x_{k+1} - \alpha \left( t_{k+1} + t_{k+1}^2 \right) g_{k+1}.
\]

The fact that parameters \( a, t_k \subset (0, 1) \) confirms that the iteration (24) is a gradient descent scheme. Based on the previous analysis, in the following lemma, we prove that the iteration (18) is an accelerated gradient descent process.

**Proposition 1.** Iteration (18) is an accelerated gradient descent method.

**Proof.** We first prove that the iteration (18) fulfills the gradient descent condition (3). According to the Backtracking algorithm, iterative step length parameter \( t_k \in (0, 1) \). With that, we assume the positiveness of the acceleration parameter \( \gamma_{k}^{\text{HmodADS}} \). From these two presumptions, we have

\[
\omega_k = t_k \gamma_k^{-1} + t_k^2 > 0,
\]

which, considering that \( a \in (1, 2) \), implies

\[
a\omega_k > 0.
\]

This leads to the final conclusion:

\[
-a\omega_k g_k^T g_k < 0.
\]

Estimation (25) confirms that (18) is the gradient descent process, i.e., the HmodADS iteration fulfills the gradient descent condition (3) for the search direction \(-a\omega_k g_k^T g_k\).

Since the iteration (18) includes the acceleration parameter \( \gamma_{k}^{\text{HmodADS}} > 0 \), derived from the second order Taylor’s expansion, the HmodADS method belongs to the class of accelerated gradient methods established in [1].

Now, we list the algorithmic steps of the HmodADS method:

1. Set \( k = 0 \), compute \( f(x_0), g_0 = g(x_0) \) and take \( \gamma_0 = 1 \);
2. If \( \|g_k\| < \epsilon \), then go to Step 8; else, continue to step 3;
3. Apply Backtracking algorithm to calculate the iterative step length \( a_k \);
4. Compute \( x_{k+1} \) using (18);
5. Determine the acceleration parameter \( \gamma_{k+1} \) using (22);
6. If \( \gamma_{k+1} < 0 \), then take \( \gamma_{k+1} = 1 \);
7. Set \( k := k + 1 \), then go to Step 2;
8. Return \( x_{k+1} \) and \( f(x_{k+1}) \).

3. Convergence of the HmodADS Method on the Set of the Uniform Convex Functions

In this section, we will prove the convergence of the proposed hybrid modADS scheme. We will first analyze a set of uniformly convex functions and, afterward, a subset of strictly convex quadratic functions. Primarily, we will start with the well-known preposition and lemma whose proofs can be found in [11,12].

**Proposition 2 ([11,12]).** Let function \( f : \mathbb{R}^n \to \mathbb{R} \) be twice continuously differentiable and uniformly convex on \( \mathbb{R}^n \). Then, the next two statements are true:
1. The function \( f \) has a lower bound on \( L_0 = \{ x \in \mathbb{R}^n | f(x) \leq f(x_0) \} \), where \( x_0 \in \mathbb{R}^n \) is available;
2. The gradient \( g \) is Lipschitz-continuous in an open convex set \( B \) which contains \( L_0 \); i.e., there exists \( L > 0 \) such that

\[
\| g(x) - g(y) \| \leq \| x - y \|, \forall x, y \in B. \tag{26}
\]

**Lemma 1 ([11,12])**. Under the assumptions of Proposition 1, there exist the numbers \( m, M \in \mathbb{R} \) satisfying

\[
0 < m \leq 1 < M,
\]

such that \( f(x) \) has an unique minimizer \( x^* \) and fulfills the next inequalities:

\[
m\|y\|^2 \leq y^T \nabla f(x)y \leq M\|y\|^2, \quad \forall x, y \in \mathbb{R}^n; \tag{28}
\]

\[
\frac{1}{2}m\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2}M\|x - x^*\|^2, \quad \forall x \in \mathbb{R}^n; \tag{29}
\]

\[
m\|x - y\|^2 \leq (g(x) - g(y))^T (x - y) \leq M\|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n. \tag{30}
\]

The following Lemma will reveal the iterative difference of the objective function in two successive points.

**Lemma 2.** Let function \( f \) be a twice continuously differentiable and uniformly convex function defined on \( \mathbb{R}^n \), and let the sequence \( \{ x_k \} \) be defined by the (15). Then, the following holds:

\[
f(x_k) - f(x_{k+1}) \geq \mu \| g_k \|^2, \tag{31}
\]

where

\[
\mu = \min \left\{ \frac{\sigma \alpha}{M}, \frac{\sigma (1 - \sigma)}{L} \beta \right\}, \tag{32}
\]

where \( L > 0 \) is the Lipschitz constant from Proposition 1, and \( M \in \mathbb{R} \) is defined in Lemma 1.

**Proof.** Within the iteration formula of the HmodADS method

\[
x_{k+1} = x_k - \alpha (t_k \gamma_k^{-1} + t_k^2) g_k = x_k - a t_k \gamma_k^{-1} \ g_k - a t_k^2 \ g_k.
\]

we observe two vector directions:

1. \( d_k = -a \gamma_k^{-1} \ g_k; \)
2. \( d_k = -a \ g_k. \)

We analyze the next two cases which refer to the value of the iterative step size: \( t_k < 1 \) and \( t_k = 1 \).

- For \( d_k = -a \gamma_k^{-1} \ g_k \) and \( t_k < 1 \), using the estimation \( t_k > -\frac{\beta (1 - \sigma)}{L} \frac{g_k^T d_k}{\| d_k \|^2} \) proven in [13], we have

\[
l_k > -\frac{\beta (1 - \sigma)}{L} \frac{g_k^T d_k}{\| d_k \|^2} = -\frac{\beta (1 - \sigma)}{L} \frac{g_k^T (-a \gamma_k^{-1} \ g_k)}{\| -a \gamma_k^{-1} \ g_k \|^2} = -\frac{\beta (1 - \sigma)}{L} \frac{\gamma_k}{\alpha} \tag{33}
\]

Considering the exit condition of Backtracking Algorithm 1, the next inequality is valid:

\[
f(x_k) - f(x_{k+1}) \geq -\sigma t_k g_k^T d_k > -a \frac{\beta (1 - \sigma)}{L} \frac{\gamma_k}{\alpha} \ g_k^T (-a \gamma_k^{-1} \ g_k) = -a \frac{\beta (1 - \sigma)}{L} \| g_k \|^2.
\]

- In order to consider the situation where \( t_k = 1 \), we have to pay attention to the fact that \( y_k < M \). This follows from the fact that \( y_k \) is an approximation of the Hessian and from inequality (28) in Lemma 1. Knowing this, it is easy to derive the following:

\[
f(x_k) - f(x_{k+1}) \geq -\sigma g_k^T d_k = -a \| g_k \|^2 = \frac{M}{\alpha} \| g_k \|^2.
\]
In the second case, where \( d_k = -\alpha g_k \), the proof is conducted in the same way as in the first case, by replacing \( d_k = -\alpha g_k^{-1} g_k \) with \( d_k = -\alpha g_k \). 

In the following Theorem 1, we prove that the HmodADS model converges at least linearly.

**Theorem 1.** The sequence \( \{x_k\} \) generated by (15) and applied in the uniformly convex and twice differentiable objective function \( f \), converges at least linearly to its solution \( x^* \) and

\[
\lim_{k \to \infty} \|g_k\| = 0.
\]

**Proof.** From Lemma 1, we know that the objective function \( f \), when applied in the HmodADS process, is bounded below and decreases, so it is evident that

\[
\lim_{k \to \infty} (f(x_k) - f(x_{k+1})) = 0.
\]

Applying estimation (31) from Lemma 2, we obtain

\[
\lim_{k \to \infty} \|g_k\| = 0.
\]

Now, we have to prove that the sequence \( \{x_k\} \) defined by (15) converges to \( x^* \). In other words, we have to prove

\[
\lim_{k \to \infty} \|x_k - x^*\| = 0. \quad (34)
\]

We use the following substitution \( x^* \equiv y \) in (30), which implies

\[
m\|x - x^*\|^2 \leq (g(x) - g(x^*))^T (x - x^*) \leq M\|x - x^*\|^2, \quad \forall x, y \in \mathbb{R}^n.
\]

Regarding the Mean Value Theorem and the Cauchy–Schwartz inequality, further on, we obtain

\[
m\|x - x^*\|^2 \leq \|g_k\| \leq M\|x - x^*\|^2.
\]

This estimation and the inequality (31) leads to

\[
\mu \|g_k\|^2 \geq \mu m^2\|x - x^*\|^2 \geq 2\mu \frac{m^2}{M} (f(x_k) - f(x^*)) \to_{k \to \infty} 0.
\]

which proves the relation (34).

To complete this proof, at the end, we show that the HmodADS process is linearly convergent under the assumption \( M \geq \sqrt{2} \), where \( M \) is a number defined by (27) in Lemma 1. To do this, we practically need to prove that

\[
\rho = \sqrt{2\mu \frac{m^2}{M}} < 1.
\]

We know from (32) in Lemma 2 that there are two values of the variable \( \mu \): \( \mu = \frac{\sigma \alpha}{M} \) and \( \mu = \frac{\sigma (1-\sigma) \beta}{L} \):

- For \( \mu = \frac{\sigma \alpha}{M} \), we have

\[
\rho^2 = 2\mu \frac{m^2}{M} = 2 \frac{\sigma \alpha}{M} \frac{m^2}{M} < 2 \frac{\sigma^2 m^2}{M^2} < 2 \frac{1}{2} \frac{1}{\sqrt{2}^2} < 1.
\]

- For \( \mu = \frac{\sigma (1-\sigma) \beta}{L} \), we have

\[
\rho^2 = 2\mu \frac{m^2}{M} = 2 \frac{\sigma (1-\sigma) m^2}{LM} < \frac{m^2}{LM} < 1.
\]
Since \( m \leq L \), the previous inequality is valid, which completes this proof. \( \square \)

4. Convergence of the HmodADS Method on the Set of the Strictly Convex Quadratics

Let us show now that the iteration HmodADS is convergent regarding the set of strictly convex quadratic functions.

\[
f(x) = \frac{1}{2}x^TAx - b^Tx.
\]

In (35), it is assumed that \( A \) is a real \( n \times n \) symmetric positive definite matrix and that the vector \( b \in \mathbb{R}^n \) is given. The smallest and the largest eigenvalues of the matrix \( A \), respectively, are denoted by \( \lambda_1 \) and \( \lambda_n \). However, before we reveal the theorem about the convergence of the function (35), we have to prove the following lemma.

**Lemma 3.** For the strictly convex quadratic function \( f \) given by the expression (35) which involves the symmetric positive definite matrix \( A \rightarrow \mathbb{R}^{n \times n} \) and the gradient descent method (15), the following holds:

\[
\frac{1}{4\lambda_n} \leq \omega_{k+1} \leq \frac{1}{\lambda_1} + 1,
\]

under the assumption that \( \lambda_1 \) and \( \lambda_n \) are, respectively, the smallest and the largest eigenvalues of \( A \) and \( \omega_{k+1} \) is defined by (17).

**Proof.** Taking \( f \), defined by (35), we have

\[
f(x_{k+1}) - f(x_k) = \frac{1}{2}x_{k+1}^TAx_{k+1} - b^Tx_{k+1} - \frac{1}{2}x_k^TAx_k + b^Tx_k.
\]

After the replacement \( x_{k+1} = x_k - \alpha \omega_k g_k \), we obtain

\[
f(x_{k+1}) - f(x_k) = \frac{1}{2}(x_k - \alpha \omega_k g_k)^TA(x_k - \alpha \omega_k g_k) - b^T(x_k - \alpha \omega_k g_k) - \frac{1}{2}x_k^TAx_k + b^Tx_k =
\]

\[
= \frac{1}{2}x_k^TAx_k - \frac{1}{2}\alpha \omega_k x_k^TAg_k - \frac{1}{2}\alpha \omega_k g_k^TAx_k + \frac{1}{2}\alpha^2 \omega_k^2 g_k^TAg_k
\]

\[
- b^Tx_k + \alpha \omega_k b^Tg_k - \frac{1}{2}x_k^TAx_k + b^Tx_k =
\]

\[
= - \frac{1}{2}\alpha \omega_k x_k^TAg_k - \frac{1}{2}\alpha \omega_k g_k^TAx_k + \frac{1}{2}\alpha^2 \omega_k^2 g_k^TAg_k + \alpha \omega_k b^Tg_k.
\]

Since the matrix \( A \) is symmetric, we can use the property \( b^Tg_k = g_k^Tb \). We can also use the fact that the gradient of the function (35) is \( g_k = Ax_k - b \), which implies

\[
f(x_{k+1}) - f(x_k) = -\frac{1}{2}\alpha \omega_k x_k^TAg_k - \frac{1}{2}\alpha \omega_k g_k^TAx_k + \frac{1}{2}\alpha^2 \omega_k^2 g_k^TAg_k + \alpha \omega_k b^Tg_k
\]

\[
= \alpha \omega_k \left( -\frac{1}{2}x_k^TAx_k + \frac{1}{2}x_k^TAx_k - \frac{1}{2}g_k^TAx_k + \frac{1}{2}g_k^TAx_k + \frac{1}{2}g_k^Tb + b^Tg_k \right)
\]

\[
= \alpha \omega_k \left( -\frac{1}{2}g_k^TAx_k + \frac{1}{2}g_k^TAx_k + \frac{1}{2}g_k^Tb \right)
\]

\[
= \alpha \omega_k \left( -g_k^TAx_k + b^Tg_k \right)
\]

\[
= \alpha \omega_k \left( -g_k^Tg_k + \frac{1}{2}\alpha \omega_k g_k^TAg_k \right).
\]
Putting this result into (22), \( \gamma_{k+1}^{\text{modADS}} \) becomes

\[
\gamma_{k+1} = 2 \frac{\left[ f(x_{k+1}) - f(x_k) \right] + \alpha \omega_k \| g_k \|^2}{\alpha^2 \omega_k^2 \| g_k \|^2} = \frac{\alpha \omega_k \left( -g_k^T g_k + \frac{1}{2} \alpha \omega_k^2 A_g g_k \right) + \alpha \omega_k \| g_k \|^2}{\alpha^2 \omega_k^2 \| g_k \|^2} = \frac{\alpha \omega_k^2 A_g g_k}{\alpha^2 \omega_k^2 \| g_k \|^2} = \frac{g_k^T A_g g_k}{\| g_k \|^2}.
\]

The last equality confirms that \( \gamma_{k+1} \) is the Rayleigh quotient of the real symmetric matrix \( A \) at the vector \( g_k \), and, therefore, the following inequalities hold:

\[
\lambda_1 \leq \gamma_{k+1} \leq \lambda_n, \quad k \in \mathbb{N}.
\]

Since \( \omega_{k+1} \equiv \left( t_{k+1} \gamma_{k+1}^{-1} + t_{k+1}^2 \right) \) and \( 0 \leq t_{k+1} \leq 1 \to 0 \leq t_{k+1}^2 \leq 1 \), the following estimations are valid:

\[
\omega_{k+1} = \frac{t_{k+1} \gamma_{k+1}^{-1} + t_{k+1}^2}{\gamma_{k+1}} \leq \frac{1}{\lambda_1} + 1.
\]

To prove the right-hand side of inequality, we use the relation (33) from Lemma 2 where \( \beta \in (\sigma, 1), \sigma \in (0, \frac{1}{2}) \) and \( \alpha \in (1, 2):

\[
t_k > \frac{\beta (1 - \sigma)}{L} \frac{\gamma_k}{\alpha} \Rightarrow \frac{t_k}{\gamma_k} > \frac{\beta (1 - \sigma)}{L \alpha},
\]

\[
\omega_k = t_k \gamma_k^{-1} + t_k^2 > \frac{t_k}{\alpha} > \frac{\beta (1 - \sigma)}{2 \alpha L} > \frac{\sigma}{4} > \frac{1}{4 \lambda_n}.
\]

The estimation of the Lipschitz constant \( L \) via the largest eigenvalue \( \lambda_n \) is certainly valid since the matrix \( A \) is symmetric and \( g_k = A x_k - b \). From these two facts, we conclude that

\[
\| g(x) - g(y) \| = \| A x - A y \| = \| A (x - y) \| \leq \| A \| \| x - y \| = \lambda_n \| x - y \|,
\]

which completes the proof. \( \square \)

**Theorem 2.** Assume that \( \lambda_n < \frac{2 \lambda_1}{1 + \lambda_1} \) holds for the smallest and largest values of the positive definite matrix \( A \in \mathbb{R}^{n \times n} \) and let \( f \) be the strictly convex quadratic function given by (35). Assume \( \{v_1, v_2, ..., v_n\} \) are the orthonormal eigenvectors of matrix \( A \), and suppose that \( \{x_k\} \) is the sequence of values constructed via the HmodADS method, for the gradients of convex quadratic functions defined by (35). Then, the following holds:

\[
g_k = \sum_{i=1}^{n} d_i^k v_i, \quad (36)
\]

where

\[
(d_i^{k+1})^2 \leq \delta^2 (d_i^k)^2, \quad \delta = \max \left\{ 1 - \frac{\alpha \lambda_1}{4 \lambda_n}, \alpha \left( \frac{1}{\lambda_1} + 1 \right) \lambda_n - 1 \right\},
\]

for some real parameters \( d_1^k, d_2^k, ..., d_n^k \). With that,

\[
\lim_{k \to \infty} \| g_k \| = 0.
\]

**Proof.** Applying the iteration (18) in the strictly convex function (35), we get
\[ g_{k+1} = Ax_{k+1} - b = A(x_k - \alpha \omega_k g_k) - b = Ax_k - b - \alpha \omega_k Ag_k = g_k - \alpha \omega_k Ag_k = (I - \alpha \omega_k A)g_k, \]

since \( g_k = Ax_k - b \). Taking (36), we obtain that

\[ g_{k+1} = \sum_{i=1}^{n} d_{i_{k+1}}^2 v_i = \sum_{i=1}^{n} (I - \alpha \omega_k A)d_i^2 v_i. \]

In order to finish the proof, we have to show that \( |1 - \alpha \omega_k \lambda | \leq \delta. \)

\[ |1 - \alpha \omega_k \lambda_i | \leq \delta \Leftrightarrow \begin{cases} 1 - \alpha \omega_k \lambda_i, & \alpha \omega_k \lambda_i \leq 1 \\ \alpha \omega_k \lambda_i - 1, & \alpha \omega_k \lambda_i > 1, \end{cases} \]

So, we analyze two cases:

1. \( \alpha \omega_k \lambda_i \leq 1 \Rightarrow 1 \geq \alpha \omega_k \lambda_i \geq \frac{\lambda_i}{\lambda + \epsilon} \Rightarrow -\alpha \omega_k \lambda_i \leq - \frac{\lambda_i}{\lambda + \epsilon} \Rightarrow 1 - \alpha \omega_k \lambda_i \leq 1 - \frac{\lambda_i}{\lambda + \epsilon} \leq \delta; \)

2. \( \alpha \omega_k \lambda_i > 1 \Rightarrow 1 < \alpha \omega_k \lambda_i \leq a \left( \frac{1}{\lambda_i} - 1 \right) \lambda_i \Rightarrow \alpha \omega_k \lambda_i - 1 \leq a \left( \frac{1}{\lambda_i} - 1 \right) \lambda_i - 1 < \delta. \)

From (36), we have that the measure of the gradient norm square is \( \|g_k\|^2 = \sum_{i=1}^{n} (d_i^k)^2, \)

and since the parameter \( \delta \in (0, 1) \), we derive the final conclusion that \( \lim_{k \to \infty} \|g_k\| = 0. \)

5. Results

In this section, we will see numerical results that will confirm the faster convergence of the HmodADS method compared with the original modADS method. In [10], the superiority of the modADS method over the ADD, ADSS, TADSS, GD and AGD methods was shown, and here we show that HmodADS is even better than the original modADS method. For the purpose of this numerical testing, we used 25 different test functions from [2]. We chose 10 values for the number of parameters for each test function: 100; 500; 1000; 3000; 5000; 10,000; 15,000; 20,000; 25,000 and 30,000. The stopping criteria for both of these methods are

\[ \|g_k\| \leq 10^{-6} \quad \text{and} \quad \frac{|f(x_{k+1}) - f(x_k)|}{1 + |f(x_k)|}. \]

The backtracking parameters are \( \sigma = 0.0001 \) and \( \beta = 0.8. \)

In Table 1, we compare the needed number of iterations for both methods.

From Table 1, it can be clearly seen that, in most cases, the number of function iterations is smaller with the HmodADS methods than with the modADS. More precisely, even 12 functions have a smaller number of iterations when the HmodADS method is used, which is almost half. The number of functions in which the modADS and the HmodADS methods have the same number of iterations is 11, and only in the case of two test functions does the modADS method outperform its hybrid version. This clearly indicates that the HmodADS is superior to the modADS, at least when it comes to the number of iterations. We further compare these two methods regarding the number of function evolutions metric.

For the same set of test functions and the same number of variables, the results from Table 2 confirm the better performance of the HmodADS compared with the modADS when it comes to function evaluation. The dominance of the HmodADS in relation to the modADS method is noticeable in 15 test functions, 7 functions have the same value in both methods, and the modADS method has an advantage only in 3 test functions. It is evident that the number of cases where the modADS outperforms the HmodADS is incomparably smaller than the number of the opposite cases. This leads us to the conclusion that, even with this indicator relating to the function evolutions, the HmodADS method has an advantage over the modADS.

Regarding the CPU time, both methods give very low outcomes. For all numbers of the parameters and for each test function, the CPU time values for both models are close to zero. For that reason, we additionally test a chosen set of test functions for five larger numbers of parameters: 50,000, 100,000, 150,000, 200,000 and 250,000. These results are displayed in Table 3. As can be seen, the HmodADS is faster than the modADS in seven
cases, while the opposite situation was recorded in five cases. Both models give the same CPU time in the case of 13 test functions.

Table 1. Number of iterations for modADS and HmodADS.

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Function</th>
<th>modADS</th>
<th>HmodADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Extended Penalty</td>
<td>51</td>
<td>50</td>
</tr>
<tr>
<td>2.</td>
<td>Perturbed Quadratic</td>
<td>451</td>
<td>445</td>
</tr>
<tr>
<td>3.</td>
<td>Raydan-1</td>
<td>45</td>
<td>41</td>
</tr>
<tr>
<td>4.</td>
<td>Diagonal 1</td>
<td>53</td>
<td>68</td>
</tr>
<tr>
<td>5.</td>
<td>Diagonal 3</td>
<td>51</td>
<td>51</td>
</tr>
<tr>
<td>6.</td>
<td>Generalized Tridiagonal-1</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>7.</td>
<td>Extended Tridiagonal-1</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>8.</td>
<td>Extended Three Expon. Terms</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>9.</td>
<td>Diagonal 4</td>
<td>783</td>
<td>770</td>
</tr>
<tr>
<td>10.</td>
<td>Extended Himmelblau</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>11.</td>
<td>Quadratic Diago. Perturbed</td>
<td>446</td>
<td>442</td>
</tr>
<tr>
<td>12.</td>
<td>Quadratic QF1</td>
<td>490</td>
<td>484</td>
</tr>
<tr>
<td>13.</td>
<td>Exten. Quadr. Penalty QP1</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>15.</td>
<td>Quadratic QF2</td>
<td>61</td>
<td>60</td>
</tr>
<tr>
<td>16.</td>
<td>Extended EP1</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>17.</td>
<td>Extended Tridiagonal-2</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>18.</td>
<td>Arwhead</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>19.</td>
<td>Almost Perturbed Quadratic</td>
<td>450</td>
<td>445</td>
</tr>
<tr>
<td>20.</td>
<td>Engval1</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>21.</td>
<td>Quarte</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>22.</td>
<td>Generalized Quartic</td>
<td>72</td>
<td>70</td>
</tr>
<tr>
<td>23.</td>
<td>Diagonal 7</td>
<td>2202</td>
<td>2096</td>
</tr>
<tr>
<td>24.</td>
<td>Diagonal 8</td>
<td>2216</td>
<td>2112</td>
</tr>
<tr>
<td>25.</td>
<td>Diagonal 9</td>
<td>44</td>
<td>51</td>
</tr>
</tbody>
</table>

Table 2. Number of function evaluations for modADS and HmodADS.

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Function</th>
<th>modADS</th>
<th>HmodADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Extended Penalty</td>
<td>1177</td>
<td>1174</td>
</tr>
<tr>
<td>2.</td>
<td>Perturbed Quadratic</td>
<td>1256</td>
<td>1244</td>
</tr>
<tr>
<td>3.</td>
<td>Raydan-1</td>
<td>337</td>
<td>505</td>
</tr>
<tr>
<td>4.</td>
<td>Diagonal 1</td>
<td>2307</td>
<td>1096</td>
</tr>
<tr>
<td>5.</td>
<td>Diagonal 3</td>
<td>2254</td>
<td>836</td>
</tr>
<tr>
<td>6.</td>
<td>Generalized Tridiagonal-1</td>
<td>455</td>
<td>462</td>
</tr>
<tr>
<td>7.</td>
<td>Extended Tridiagonal-1</td>
<td>270</td>
<td>270</td>
</tr>
<tr>
<td>8.</td>
<td>Extended Three Expon. Terms</td>
<td>220</td>
<td>230</td>
</tr>
<tr>
<td>9.</td>
<td>Diagonal 4</td>
<td>1756</td>
<td>1730</td>
</tr>
<tr>
<td>10.</td>
<td>Extended Himmelblau</td>
<td>340</td>
<td>330</td>
</tr>
<tr>
<td>11.</td>
<td>Quadratic Diago. Perturbed</td>
<td>1259</td>
<td>1251</td>
</tr>
<tr>
<td>12.</td>
<td>Quadratic QF1</td>
<td>1302</td>
<td>1290</td>
</tr>
<tr>
<td>15.</td>
<td>Quadratic QF2</td>
<td>579</td>
<td>561</td>
</tr>
<tr>
<td>16.</td>
<td>Extended EP1</td>
<td>350</td>
<td>350</td>
</tr>
<tr>
<td>17.</td>
<td>Extended Tridiagonal-2</td>
<td>320</td>
<td>280</td>
</tr>
<tr>
<td>18.</td>
<td>Arwhead</td>
<td>580</td>
<td>580</td>
</tr>
<tr>
<td>19.</td>
<td>Almost Perturbed Quadratic</td>
<td>1252</td>
<td>1242</td>
</tr>
<tr>
<td>20.</td>
<td>Engval1</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>21.</td>
<td>Quarte</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>22.</td>
<td>Generalized Quartic</td>
<td>440</td>
<td>433</td>
</tr>
<tr>
<td>23.</td>
<td>Diagonal 7</td>
<td>4424</td>
<td>4212</td>
</tr>
<tr>
<td>24.</td>
<td>Diagonal 8</td>
<td>4482</td>
<td>4274</td>
</tr>
<tr>
<td>25.</td>
<td>Diagonal 9</td>
<td>1902</td>
<td>1537</td>
</tr>
</tbody>
</table>
Table 3. CPU metric for modADS and HmodADS.

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Function</th>
<th>modADS</th>
<th>HmodADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Extended Penalty</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>2.</td>
<td>Perturbed Quadratic</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>Raydan-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.</td>
<td>Diagonal 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5.</td>
<td>Diagonal 3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.</td>
<td>Generalized Tridiagonal-1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>7.</td>
<td>Extended Tridiagonal-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8.</td>
<td>Extended Three Expon. Terms</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9.</td>
<td>Diagonal 4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>10.</td>
<td>Extended Himmelblau</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11.</td>
<td>Quadratic Diag. Perturbed</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>12.</td>
<td>Quadratic QF1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>13.</td>
<td>Ext. Quatr. Penalty QP1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>14.</td>
<td>Ext. Quadr. Penalty QP2</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>15.</td>
<td>Quadratic QF2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>16.</td>
<td>Extended EP1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17.</td>
<td>Extended Tridiagonal-2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18.</td>
<td>Arwhead</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>19.</td>
<td>Almost Perturbed Quadratic</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>20.</td>
<td>Engval1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>21.</td>
<td>Quarte</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22.</td>
<td>Generalized Quartic</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>23.</td>
<td>Diagonal 7</td>
<td>19</td>
<td>16</td>
</tr>
<tr>
<td>24.</td>
<td>Diagonal 8</td>
<td>18</td>
<td>15</td>
</tr>
<tr>
<td>25.</td>
<td>Diagonal 9</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

6. Conclusions

In this paper, we present a hybrid version of an improved modification of the accelerated double direction and double step size method for solving unconstrained optimization problems. The HmodADS method is derived from the modADS method using the hybrid three-term method, introduced in [8]. We showed that HmodADS is an accelerated gradient descent method and, with a properly defined acceleration parameter, we proved the linear convergence rate of HmodADS for uniformly convex functions and for strictly convex quadratic functions. The derived model is numerically tested and compared with its forerunner, the modADS method. Noticeable improvement in favor of the HmodADS process regarding the number of iterations and number of function evaluations is observed.

From all exposed, we conclude that the specified hybrid version of the modADS method is significantly better than its forerunner and that it can be used for solving many unconstrained optimization tasks.

The obtained results confirm, once again, that Khan’s hybridization rule can improve the performance profiles of the accelerated optimization scheme. Like many other optimization models, the derived hybrid method can be applied in solving nonlinear equation systems.

**Author Contributions:** Conceptualization, M.J.P.; methodology, M.J.P. and A.V.; software, A.V. and T.J.S.; validation, M.J.P. and T.J.S.; formal analysis, M.J.P., A.V. and T.J.S.; investigation, M.J.P. and A.V.; resources, T.J.S.; data curation, A.V. and T.J.S.; writing—original draft preparation, M.J.P. and A.V.; writing—review and editing, M.J.P. and T.J.S.; visualization, A.V. and T.J.S.; supervision, M.J.P.; project administration, A.V.; funding acquisition, M.J.P. and T.J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Ministry of Education and Science of Republic of Serbia, Grant No. 174025.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.
Data Availability Statement: Data will be provided on request to authors.

Acknowledgments: Milena J. Petrović acknowledges support from the Project supported by Ministry of Education and Science of Republic of Serbia, Grant No. 174025.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.