Numerical Algorithms
https://doi.org/10.1007/s11075-021-01228-0

ORIGINAL PAPER

A family of optimal weighted conjugate-gradient-type methods for strictly convex quadratic minimization

Harry Oviedo[1] · **Roberto Andreani**[2] · **Marcos Raydan**[3]

Received: 25 February 2021 / Accepted: 25 October 2021 / Published online: 25 November 2021

Abstract

We introduce a family of weighted conjugate-gradient-type methods, for strictly convex quadratic functions, whose parameters are determined by a minimization model based on a convex combination of the objective function and its gradient norm. This family includes the classical linear conjugate gradient method and the recently published delayed weighted gradient method as the extreme cases of the convex combination. The inner cases produce a merit function that offers a compromise between function-value reduction and stationarity which is convenient for real applications. We show that each one of the infinitely many members of the family exhibits q-linear convergence to the unique solution. Moreover, each one of them enjoys finite termination and an optimality property related to the combined merit function. In particular, we prove that if the $n \times n$ Hessian of the quadratic function has $p < n$ different eigenvalues, then each member of the family obtains the unique global minimizer in exactly p iterations. Numerical results are presented that demonstrate that the proposed family is promising and exhibits a fast convergence behavior which motivates the use of preconditioning strategies, as well as its extension to the numerical solution of general unconstrained optimization problems.

✉ Marcos Raydan
m.raydan@fct.unl.pt

Harry Oviedo
harry.leon@fgv.br

Roberto Andreani
andreani@unicamp.br

[1] Escola de Matemática Aplicada, Fundação Getulio Vargas (FGV/EMAp), Rio de Janeiro, RJ, Brazil

[2] Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, Distrito Barão Geraldo, 13083-859 Campinas SP, Brazil

[3] Centro de Matemática e Aplicações (CMA), FCT, UNL, 2829–516 Caparica, Portugal

Keywords Gradient methods · Conjugate gradient methods · Strictly convex quadratics · Unconstrained optimization · Moreau envelope

1 Introduction

We are interested in solving the following convex quadratic minimization problem

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}x^\top Ax - x^\top b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix and $b \in \mathbb{R}^n$ is a given vector. Solving (1) is equivalent to finding the unique solution of the linear system of equations $Ax = b$. Many real-life applications require to solve large-scale linear systems of equations whose very large size makes iterative methods the best choice due to their simplicity and low computational cost. In addition, problem (1) is a simple setting to design effective methods for more general unconstrained optimization problems.

One of the fundamental iterative methods for solving (1) is the gradient method, which generates a sequence of iterates using the following recursive formula

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad \text{for } k \geq 0, \tag{2}$$

where $\alpha_k > 0$ is the step-size. Different ways of choosing $\alpha_k > 0$ lead to different gradient methods. The classical gradient method to solve (1), originally proposed by Cauchy [6], computes the step-size in (2) as

$$\alpha_k^{SD} := \arg\min_{\alpha>0} f(x_k - \alpha\nabla f(x_k)) = \frac{||\nabla f(x_k)||_2^2}{\nabla f(x_k)^\top A\nabla f(x_k)}. \tag{3}$$

The method given by Eqs. (2)–(3) is called the Cauchy method or the steepest descent (SD) method. Another classical example of step-size selection, associated with the gradient method (2), is the one that minimizes the gradient 2-norm at x_k, given by

$$\alpha_k^{MG} := \arg\min_{\alpha>0} ||\nabla f(x_k - \alpha\nabla f(x_k))||_2 = \frac{\nabla f(x_k)^\top A\nabla f(x_k)}{||A\nabla f(x_k)||_2^2}, \tag{4}$$

which is called the minimal gradient (MG) step-size.

The SD and the MG gradient-type methods are very inexpensive and intuitive, but they both suffer from a slow rate of convergence towards the unique solution of (1). In the last few decades, a wide variety of step-size rules have emerged to improve the efficiency of gradient-type methods, while preserving their simplicity and low memory requirements; see, e.g., [2, 5, 7, 8, 10–17, 19, 21, 25, 33, 34]. However, the-method-of-choice to solve problem (1) is the classical conjugate gradient (CG) method proposed by Hestenes and Stiefel [18]. The main reason for it to remain as one of the best low-cost options for solving (1) is its outstanding practical behavior that relies on its A-orthogonality and optimality properties on an underlying Krylov subspace. As a consequence, at every k, the CG method generates the iterate x_k as the

minimizer of the objective function $f(\cdot)$ on the k-dimensional already explored subspace. For a review on the CG method for strictly convex quadratics and its optimality properties, we refer the reader to [22, 26, 32].

Recently, in [27], a combination of a smoothing technique with a one-step delayed gradient scheme was developed as an enriched gradient-type method for solving (1). The so-called delayed weighted gradient method (DWGM) shows in practice a quite similar convergence behavior to the one observed in the CG method. Later, in [1], it was established that indeed the DWGM method has also some key A-orthogonality and some optimality properties, including the finite termination in at most p iterations, where p is the number of distinct eigenvalues of the matrix A. The main difference with the CG method is that, instead of minimizing the objective function $f(\cdot)$ on the entire explored subspace, the DWGM method minimizes the 2-norm of the gradient vector on the same subspace. A first attempt to extend the DWGM method was recently presented by Oviedo et al. in [28]. In [28], the authors combine the ideas of the general hybrid methods, introduced in [3, 4], with the DWGM method, to obtain the so-called hybrid gradient method (HGM). Unfortunately, the convergence analysis in [28] requires a strong hypothesis on the smallest eigenvalue of the Hessian matrix A.

As a generalization of the DWGM and HGM methods, in this work, we propose a family of low-cost methods that, depending on a real parameter $\mu \in [0, 1]$, goes from the CG method ($\mu = 0$) to the DWGM method ($\mu = 1$), keeping for all the infinitely many members of the family some key orthogonality and optimality properties on a convenient Krylov subspace. The internal cases, i.e., when $\mu \in (0, 1)$, produce iterates that are optimal for a properly chosen merit function that offers a compromise between function-value reduction and gradient-norm reduction (i.e., stationarity) which is convenient for real applications and also for possible extensions to the general unconstrained minimization framework. Each member of the proposed family computes the iterates by a two-step process. At the first step, a prediction of the new iterate is obtained by performing a gradient-type method with a step-size selected as the argument that minimizes the merit function. Then, the new iterate is computed by minimizing the merit function but now over the line that connects the prediction and the penultimate iterate. Under mild assumptions we prove some standard global convergence properties of our proposal. Moreover, for any member of the family, similar orthogonality properties and some optimal properties that hold for the CG and the DWGM methods are established. Finally, we benchmark our procedure over a set of sparse problems involving real data and large dimension, and compare it with the classical conjugate gradient method and the DWGM method.

The remainder of this paper is organized as follows. In Section 2, we introduce the new first-order algorithm to deal with problem (1). Section 3 is devoted to the global convergence analysis of our proposal. In Section 4, additional orthogonality properties are obtained, including finite termination and the minimization property of all members of the family on the already explored affine subspace. Then, in Section 5, several numerical tests are performed to assess the behavior of our procedure for solving real large-scale systems of equations. Finally, conclusions and perspectives are drawn in Section 6.

2 Derivation of the new family of methods

In this section, we derive a new family of first-order iterative methods for problem (1). First, given a fixed parameter $\mu \in [0, 1]$, we introduce a new merit function

$$F_\mu(x) := (1-\mu)E(x) + \mu||\nabla f(x)||_2^2, \tag{5}$$

where $E(x) := \frac{1}{2}(x - x^*)^\top A(x - x^*) = f(x) + \frac{1}{2}b^\top x^*$, and x^* denotes the unique solution of (1). Hence, $F_\mu(x)$ is essentially a convex combination of the objective function and its gradient norm. Additionally, observe that $x^* = A^{-1}b$ is the unique minimizer of $F_\mu(\cdot)$, which implies that minimizing $f(\cdot)$ is equivalent to minimizing (5).

Based on the development of the DWGM in [27], we propose to minimize the merit function (5) on the linear variety $S_k := x_k + \text{span}\{\nabla f(x_k),\ x_k - x_{k-1}\}$ by a two-step iteration; see also [22, pp. 254-256] for a similar approach that reproduces the classical CG method for convex quadratics. For that, we compute first a prediction z_k of x_{k+1} by performing a gradient method step (see (2))

$$z_k = x_k - \alpha_k \nabla f(x_k),$$

with the following optimal step-size

$$\alpha_k = \arg\min_{\alpha>0} F_\mu(\, x_k - \alpha \nabla f(x_k)\,) \tag{6}$$

$$= \frac{\nabla f(x_k)^\top W_\mu \nabla f(x_k)}{\nabla f(x_k)^\top W_\mu A \nabla f(x_k)} \tag{7}$$

$$= \alpha_k^{MG} \left(\frac{(1-\mu)\alpha_k^{SD} + 2\mu}{(1-\mu)\alpha_k^{MG} + 2\mu} \right), \tag{8}$$

where we have conveniently introduced the symmetric and positive definite matrix

$$W_\mu = (1-\mu)I + 2\mu A.$$

Then, we correct this prediction using an over-relaxation scheme with an optimal weight, that is

$$x_{k+1} = \beta_k z_k + (1-\beta_k)x_{k-1}, \tag{9}$$

where we select the weight β_k in (9), by minimizing the merit function $F_\mu(x_{k+1})$, i.e.,

$$\beta_k = \arg\min_{\beta} F_\mu(\beta z_k + (1-\beta)x_{k-1}) \tag{10}$$

$$= -\frac{\nabla f(x_{k-1})^\top ((1-\mu)s_k + 2\mu y_k)}{y_k^\top ((1-\mu)s_k + 2\mu y_k)} \tag{11}$$

$$= -\frac{\nabla f(x_{k-1})^\top W_\mu s_k}{y_k^\top W_\mu s_k}, \tag{12}$$

where $s_k := z_k - x_{k-1}$ and $y_k := \nabla f(z_k) - \nabla f(x_{k-1}) = As_k$. Notice that the new iterate can also be written as

$$x_{k+1} = x_k - \beta_k \alpha_k \nabla f(x_k) + (\beta_k - 1)(x_k - x_{k-1}). \tag{13}$$

From (13), we note that this two-step approach can also be seen as an optimal gradient method with momentum, and therefore the update formula used by our procedure is very similar to the one used by the CG method. Now we describe the obtained generalized DWGM (GDWGM) algorithm in detail.

Algorithm 1 Generalized delayed weighted gradient method (GDWGM).

Require: $A \in \mathbb{R}^{n\times n}$, $\mu \in [0,1]$, $W_\mu = (1-\mu)I + 2\mu A$, $b, x_0 \in \mathbb{R}^n$, $x_{-1} = x_0$, $g_0 = \nabla f(x_0)$, $g_{-1} = g_0$, $0 < \epsilon \ll 1$, $k = 0$.

while $||g_k||_2 > \epsilon$ **do**

$w_k = Ag_k$,

$\alpha_k = \frac{g_k^\top W_\mu g_k}{g_k^\top W_\mu w_k}$,

$z_k = x_k - \alpha_k g_k$,

$r_k = g_k - \alpha_k w_k$,

$s_k = z_k - x_{k-1}$,

$y_k = r_k - g_{k-1}$,

$\beta_k = -\frac{g_{k-1}^\top W_\mu s_k}{y_k^\top W_\mu s_k}$,

$x_{k+1} = x_{k-1} + \beta_k s_k$,

$g_{k+1} = g_{k-1} + \beta_k y_k$,

$k = k + 1$.

end while

Let us observe that if we fix $\mu = 1$, then Algorithm 1 reduces to the DWGM scheme developed in [27]. For the other extreme, it will be established, at the end of Section 4, that if we fix $\mu = 0$ then Algorithm 1 is equivalent to the CG method for solving (1). It is also worth mentioning that for the implementation of Algorithm 1 it is neither necessary nor recommendable for numerical reasons to explicitly build the matrix W_μ. In fact, note that this matrix is only used to update the step-size α_k and the parameter β_k. Thus, in practice, it is numerically convenient to use the formulas (8) and (11) to compute α_k and β_k, respectively. However, we present Algorithm 1 using W_μ in order to simplify the theoretical analysis of the proposed family of methods.

In the rest of this paper, we will denote by $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ the eigenvalues of A, where we assume that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0.$$

In our analysis, we will use the Kantorovich inequality (see, e.g., [22]) applied to the symmetric and positive definite matrix A, i.e.,

$$\frac{(x^\top x)^2}{(x^\top Ax)(x^\top A^{-1}x)} \geq \frac{4\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2},$$

for all nonzero vectors $x \in \mathbb{R}^n$.

Remark 1 Note that Algorithm 1 is essentially an exact two-step line search method applied to the minimization of $F_\mu(\cdot)$ over $\mathbb{R}^n$. In fact, since $\nabla F_\mu(x) = W_\mu \nabla f(x)$

for all $x \in \mathbb{R}^n$, then in the first phase of the method the search direction $d_k = -g_k$ satisfies

$$\nabla F_\mu(x_k)^\top d_k = -(1-\mu)||g_k||_2^2 - 2\mu g_k^\top A g_k < 0, \quad \forall k \in \mathbb{N},$$

and hence d_k is a descent direction of $F_\mu(\cdot)$ at x_k. In the second phase, observe that using Step 6 in Algorithm 1 and rearranging (9) we have

$$x_{k+1} = z_k + (\beta_k - 1)s_k. \tag{14}$$

Later on we will establish that $g_{k+1}^\top W_\mu s_k = 0$ for all k (see Lemma 2), and also that $\beta_k > 1$ for all k (see Lemma 4), which combined with Steps 5, 7, and 8 in Algorithm 1 yields

$$\begin{aligned}
\nabla F_\mu(z_k)^\top s_k &= ((1-\mu)\nabla f(z_k) + 2\mu A \nabla f(z_k))^\top s_k \\
&= r_k^\top W_\mu s_k = (y_k + g_{k-1})^\top W_\mu s_k \\
&= y_k^\top W_\mu s_k + g_{k-1}^\top W_\mu s_k \\
&= y_k^\top W_\mu s_k - \beta_k\, y_k^\top W_\mu s_k = (1-\beta_k)\, s_k^\top A W_\mu s_k.
\end{aligned}$$

Now, since $AW_\mu = W_\mu A$ is symmetric and positive definite (it is the sum of two positive definite matrices), then $\nabla F_\mu(z_k)^\top s_k = (1-\beta_k)\, s_k^\top A W_\mu s_k < 0$, which implies that s_k is a descent direction for the merit function $F_\mu(\cdot)$ at z_k. Therefore, Algorithm 1 can be seen as an optimization procedure that performs two optimal line searches per iteration to minimize the merit function (5).

Remark 2 There exists an interesting connection between the merit function (5) and the Moreau envelope of the quadratic objective function (1). For a given function $f : \mathbb{R}^n \to \mathbb{R}$ and a given parameter $r \geq 0$, the Moreau envelope $e_r f : \mathbb{R}^n \to \mathbb{R}$ (also known as the Moreau-Yosida regularization) is defined by

$$e_r f(x) = \inf_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{r}{2}\|x-y\|_2^2 \right\}. \tag{15}$$

The function $e_r f(x)$ was originally introduced by J. J. Moreau in the mid-1960s [23, 24], and it has been extensively studied in the optimization literature, since it offers a variety of smoothing and regularization properties for different scenarios; see, e.g., [20, 29, 31].

Since (1) can be written as $f(x) = \frac{1}{2}(x-x^*)^\top A(x-x^*) - \frac{1}{2}b^\top x^*$, it suffices to assume, without any loss of generality, that $f(x) = \frac{1}{2}x^\top Ax$. In this case, $x^* = 0$, $\nabla f(x) = Ax$, and

$$f(x) = \frac{1}{2}\nabla f(x)^\top A^{-1} \nabla f(x). \tag{16}$$

Let us now consider the function $L(x,y) = f(y) + \frac{r}{2}\|x-y\|_2^2$ which is jointly convex in x and y. Notice that, if we substitute $f(y)$ by its Taylor expansion, $L(x,y)$ can also be written as

$$L(x,y) = f(x) + \nabla^\top f(x)(y-x) + \frac{1}{2}(y-x)^\top A(y-x) + \frac{r}{2}\|x-y\|_2^2, \tag{17}$$

and it follows that $\nabla_y L(x, y) = \nabla f(x) + A(y - x) + r(y - x)$. If we force $\nabla_y L(x, y) = 0$, to identify the solution of the optimization problem involved in (15), we obtain that

$$(y - x) = -(A + rI)^{-1}\nabla f(x). \tag{18}$$

Combining (15), (17), and (18), we have that

$$\begin{aligned} e_r f(x) &= \inf_{y\in\mathbb{R}^n} \left\{ f(x) + \nabla^\top f(x)(y - x) + \frac{1}{2}(y - x)^\top (A + rI)(y - x) \right\} \\ &= f(x) - \nabla f(x)^\top (A + rI)^{-1}\nabla f(x) + \frac{1}{2}\nabla f(x)^\top (A + rI)^{-1}\nabla f(x) \\ &= f(x) - \frac{1}{2}\nabla f(x)^\top (A + rI)^{-1}\nabla f(x). \end{aligned}$$

To explore the link between (5) and (15), we choose a parameter $\mu \in (0, 1)$ and recalling (16) we get

$$\begin{aligned} e_r f(x) &= (1 - \mu) f(x) + \frac{\mu}{2}\nabla f(x)^\top A^{-1}\nabla f(x) - \frac{1}{2}\nabla f(x)^\top (A + rI)^{-1}\nabla f(x) \\ &= (1 - \mu) f(x) + \mu\nabla f(x)^\top R\, \nabla f(x), \end{aligned}$$

where $R = \frac{1}{2}A^{-1} - \frac{1}{2\mu}(A + rI)^{-1}$. The matrix R is clearly symmetric, and it is positive definite if and only if all its eigenvalues are positive. Since the matrices A and $(A+rI)^{-1}$ are diagonalized by the same orthogonal matrix, then the eigenvalues $\lambda(R)_i$, $1 \le i \le n$, of R are given by $\lambda(R)_i = \frac{1}{2\lambda_i} - \frac{1}{2\mu(\lambda_i+r)}$, where λ_i, $1 \le i \le n$, are the eigenvalues of A. Therefore, all the eigenvalues of R are positive, and as a consequence R has a symmetric and positive definite square root $R^{\frac{1}{2}}$, if $\mu > \lambda_i/(\lambda_i + r)$ for all i. For $r = 0$ in (15), $e_r f(x) = f(x^*) = 0$, and so we focus in the case $r > 0$. Moreover, for $r > 0$, the function $\psi(\lambda) = \lambda/(\lambda + r)$ is increasing ($\psi'(\lambda) = r/(\lambda + r)^2 > 0$) and hence it is enough that

$$\mu > \lambda_1/(\lambda_1 + r) \tag{19}$$

to guarantee that $R^{\frac{1}{2}}$ is well-defined. Notice that for each given $r \in (0, +\infty)$ we can find $\mu \in (0, 1)$ that satisfies (19), and vice versa. In that case,

$$e_r f(x) = (1 - \mu) f(x) + \mu\|R^{\frac{1}{2}}\nabla f(x)\|_2^2,$$

and we conclude that the merit function (5) can be seen as a simplified version of the Moreau envelope of the function (1), for which the weight matrix $R^{\frac{1}{2}}$ is the identity matrix. Therefore, (5) involves a certain type of regularization.

3 Convergence analysis

We now analyze Algorithm 1 by studying the asymptotic behavior of $F_\mu(x_k)$ and $||\nabla f(x_k)||_2$ when k goes to infinity. The proposition below and its proof provide some key properties.

Proposition 1 *Consider the iteration given by* (2)*, then*

a. $f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k), \quad \forall \alpha_k \in [0, 2\alpha_k^{SD}]$.

b. $||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2 \leq ||\nabla f(x_k)||_2, \quad \forall \alpha_k \in [0, 2\alpha_k^{MG}]$.

c. *If* $\alpha_k = \alpha_k^{SD}$ *in* (2) *then* $E(x_{k+1}) \leq C_1 E(x_k)$, *where* $C_1 = \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2$.

d. *If* $\alpha_k = \alpha_k^{MG}$ *in* (2) *then* $f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k)$ *and* $||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2^2 \leq C_1 ||\nabla f(x_k)||_2^2$.

Proof (*a*) Using (1) and the fact that $\nabla f(x_k) = Ax_k - b$, we get

$$f(x_k - \alpha_k \nabla f(x_k)) = f(x_k) - \alpha_k ||\nabla f(x_k)||_2^2 + \frac{\alpha_k^2}{2} \nabla f(x_k)^\top A \nabla f(x_k). \tag{20}$$

Using the definition of α_k^{SD} in (3) and (20), we obtain that $f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k)$ for all $\alpha_k \in [0, 2\alpha_k^{SD}]$.
(*b*) By simple algebraic manipulations, we have that

$$||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2^2 = ||\nabla f(x_k)||_2^2 - 2\alpha_k \nabla f(x_k)^\top A \nabla f(x_k) + \alpha_k^2 ||A \nabla f(x_k)||_2^2. \tag{21}$$

Once again, recalling the definition of α_k^{MG} given by (4), we obtain from (21) that

$$||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2 \leq ||\nabla f(x_k)||_2 \quad \text{for all} \quad \alpha_k \in [0, 2\alpha_k^{MG}].$$

(*c*) The proof of this part appears in detail in [22, Section 7.6].

(*d*) The first inequality is a direct consequence of (*a*) and the fact that $\alpha_k^{MG} \leq \alpha_k^{SD}$, which follows from

$$(\nabla f(x_k)^\top A \nabla f(x_k))^2 \leq \|\nabla f(x_k)\|_2^2 \, \|A \nabla f(x_k)\|_2^2 \quad \text{(Cauchy-Schwarz inequality)} .$$

Therefore, it suffices to prove that $||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2 \leq C_1 ||\nabla f(x_k)||_2$. Indeed, from (21) and using that $\alpha_k = \alpha_k^{MG}$, we have

$$||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2^2 = (1 - \hat{c}) ||\nabla f(x_k)||_2^2, \tag{22}$$

where $\hat{c} = \frac{(\nabla f(x_k)^\top A \nabla f(x_k))^2}{||\nabla f(x_k)||_2^2 ||A \nabla f(x_k)||_2^2}$. By applying the Kantorovich inequality to $\hat{c}$ in (22), we arrive at

$$||\nabla f(x_k - \alpha_k \nabla f(x_k))||_2 \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right) ||\nabla f(x_k)||_2, \tag{23}$$

which completes the proof. □

Lemma 1 *Let* $\mu \in [0, 1]$ *and* $\{x_k\}$ *be a sequence generated by Algorithm 1. Then,* $\{F_\mu(x_k)\}$ *is a convergent sequence.*

Proof It follows from the construction of Algorithm 1, Proposition 1, and the minimization properties (6) and (10) that

$$\begin{aligned} F_\mu(x_{k+1}) &\leq F_\mu(z_k) \\ &\leq F_\mu(x_k - \alpha_k^{MG} g_k) \\ &\leq (1-\mu)E(x_k - \alpha_k^{MG} g_k) + \mu C_1 ||g_k||_2^2 \\ &\leq (1-\mu)E(x_k) + \mu C_1 ||g_k||_2^2 \\ &< (1-\mu)E(x_k) + \mu ||g_k||_2^2 = F_\mu(x_k). \end{aligned} \tag{24}$$

Thus, $\{F_\mu(x_k)\}$ is a monotonically decreasing sequence. Moreover, $F_\mu(x_k) \geq 0$ for all $k \in \mathbb{N}$; therefore, we conclude that $\{F_\mu(x_k)\}$ is a convergent sequence. □

From Lemma 1, we see that the sequence $\{x_k\}$ generated by Algorithm 1 converges to the unique solution of (1) whenever $F_\mu(x_k)$ goes to zero, since both sequences $\{E(x_k)\}$ and $||g_k||_2$ are non-negative. The following theorem establishes the global convergence of Algorithm 1.

Theorem 1 *Let $\mu \in [0, 1]$ and $\{x_k\}$ be the sequence generated by Algorithm 1. Then, the sequence $\{F_\mu(x_k)\}$ converges to zero q-linearly with convergence factor $\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2$.*

Proof First, observe that the step-size α_k in Step 3 of Algorithm 1 can be written as

$$\alpha_k = \frac{g_k^\top W_\mu g_k}{g_k^\top W_\mu A g_k}. \tag{25}$$

In addition, since $Ax^* = b$, note that

$$F_\mu(x) = \frac{1}{2} x^\top A W_\mu x - x^\top W_\mu b + \frac{1}{2} b^\top W_\mu A^{-1} b \tag{26}$$

$$= \frac{1}{2}(x - x^*)^\top A W_\mu (x - x^*). \tag{27}$$

From the construction of Algorithm 1, Eqs. (26) and (27), and the minimization properties (6) and (10), we have

$$\begin{aligned} F_\mu(x_{k+1}) &\leq F_\mu(z_k), \\ &= F_\mu(x_k - \alpha_k g_k) \\ &= F_\mu(x_k) - \alpha_k g_k^\top W_\mu g_k + \frac{\alpha_k^2}{2} g_k^\top A W_\mu g_k \\ &= F_\mu(x_k) - \frac{\alpha_k}{2} g_k^\top W_\mu g_k \\ &= \left(1 - \frac{\alpha_k}{2} \frac{g_k^\top W_\mu g_k}{F_\mu(x_k)}\right) F_\mu(x_k) \\ &= (1 - \tilde{c}_k) F_\mu(x_k), \end{aligned} \tag{28}$$

where $\tilde{c}_k = \frac{\alpha_k}{2} \frac{g_k^\top W_\mu g_k}{F_\mu(x_k)}$.

On the other hand, in view of Eqs. (25) and (27), we obtain

$$\tilde{c}_k = \frac{(g_k^\top W_\mu g_k)^2}{(g_k^\top W_\mu A g_k)(g_k^\top W_\mu A^{-1} g_k)}. \tag{29}$$

Since W_μ is symmetric and positive definite then it has a symmetric and positive definite square root $W_\mu^{1/2}$, and so $W_\mu A = W_\mu^{1/2} A W_\mu^{1/2}$ and $W_\mu A^{-1} = W_\mu^{1/2} A^{-1} W_\mu^{1/2}$, and it follows that

$$\tilde{c}_k = \frac{(p_k^\top p_k)^2}{(p_k^\top A p_k)(p_k^\top A^{-1} p_k)}, \tag{30}$$

where $p_k = W_\mu^{1/2} \nabla f(x_k)$. Now, applying the Kantorovich inequality in (30), we arrive at

$$\tilde{c}_k \geq \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}. \tag{31}$$

Merging Eqs. (28) and (31), we obtain

$$F_\mu(x_{k+1}) \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right)^2 F_\mu(x_k). \tag{32}$$

It follows immediately that $\{F_\mu(x_k)\}$ converges to zero q-linearly with convergence factor $\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2$. □

Remark 3 From Theorem 1, we have that $\lim_{k\to\infty} F_\mu(x_k) = 0$, which is the sum of two positive sequences so they both converge to zero, i.e., $\lim_{k\to\infty} E(x_k) = 0$ and $\lim_{k\to\infty} ||g_k||_2 = 0$. Therefore, from this fact and the positive definiteness of A, we conclude that the sequence $\{x_k\}$ tends to the unique global minimizer of $f(\cdot)$ when k goes to infinity.

4 Finite termination and optimality properties

In this section, we establish some key W_μ-orthogonality properties that add understanding to the fast practical behavior of the GDWGM family of methods, including the finite termination. Most of these results can be viewed as generalizations of the A-orthogonality results established for the DWGM method in [1], i.e., for the specific case $\mu = 1$. In here, the structure and the ordering of the presentation of the theoretical results follow the same pattern used in [1]. However, moving from the fixed case of $\mu = 1$ to handling infinitely many cases ($\mu \in [0, 1]$) at once, the required mathematical arguments as well as the specific details differ from the ones used in [1].

Lemma 2 *Let us consider Algorithm 1. Then, we have*

a. $\beta_0 = 1$, *and hence* $x_1 = z_0$, $g_1 = r_0$.
The following equalities hold for all $k \geq 0$,
b. $r_k^\top W_\mu g_k = 0$.

c. $r_k^\top g_k = \left(\frac{2\mu}{(1-\mu)\alpha_k+2\mu}\right) r_k^\top r_k$.
d. $g_{k+1}^\top W_\mu s_k = 0$.

Proof (*a*) By Steps 4, 6, and 7 in Algorithm 1, we have that $s_0 = z_0 - x_0 = -\alpha_0 g_0$ and $y_0 = r_0 - g_0 = -\alpha_0 A g_0$. Hence,

$$\beta_0 = -\frac{g_0^\top W_\mu s_0}{y_0^\top W_\mu s_0} = -\frac{-\alpha_0 g_0^\top W_\mu g_0}{-\alpha_0(-\alpha_0 g_0^\top A W_\mu g_0)} = \frac{1}{\alpha_0}\left(\frac{g_0^\top W_\mu g_0}{g_0^\top A W_\mu g_0}\right) = 1.$$

Therefore, we obtain $x_1 = x_{-1} + \beta_0 s_0 = x_0 + s_0 = x_0 - \alpha_0 g_0 = z_0$, and

$$g_1 = g_{-1} + \beta_0 y_0 = g_0 + y_0 = g_0 + r_0 - g_0 = r_0.$$

(*b*) Using (*a*) and the definition of α_k, we get

$$r_k^\top W_\mu g_k = (g_k - \alpha_k A g_k)^\top W_\mu g_k = g_k^\top W_\mu g_k - \alpha_k g_k^\top A W_\mu g_k = 0.$$

(*c*) Let us define $\widehat{\mu} = 2\mu/((1-\mu)\alpha_k + 2\mu)$. Since $(1-\mu)\alpha_k + 2\mu \neq 0$, for all $\mu \in [0, 1]$, then it follows from (*b*) that

$$\begin{aligned}
r_k^\top g_k &= \left(\frac{(1-\mu)\alpha_k + 2\mu}{(1-\mu)\alpha_k + 2\mu}\right) r_k^\top g_k = \widehat{\mu}\, r_k^\top (r_k + \alpha_k w_k) + \left(\frac{(1-\mu)\alpha_k}{(1-\mu)\alpha_k + 2\mu}\right) r_k^\top g_k \\
&= \widehat{\mu}\, r_k^\top r_k + \widehat{\mu}\, r_k^\top (\alpha_k w_k) + \left(\frac{(1-\mu)\alpha_k}{(1-\mu)\alpha_k + 2\mu}\right) r_k^\top g_k \\
&= \widehat{\mu}\, r_k^\top r_k + \left(\frac{\alpha_k}{(1-\mu)\alpha_k + 2\mu}\right) r_k^\top (2\mu A + (1-\mu) I) g_k \\
&= \widehat{\mu}\, r_k^\top r_k + \left(\frac{\alpha_k}{(1-\mu)\alpha_k + 2\mu}\right) r_k^\top W_\mu g_k \\
&= \widehat{\mu}\, r_k^\top r_k = \left(\frac{2\mu}{(1-\mu)\alpha_k + 2\mu}\right) r_k^\top r_k.
\end{aligned}$$

(*d*) Again, by construction of Algorithm 1, we have

$$\begin{aligned}
g_{k+1}^\top W_\mu s_k &= (g_{k-1} + \beta_k y_k)^\top W_\mu s_k \\
&= g_{k-1}^\top W_\mu s_k + \beta_k y_k^\top W_\mu s_k \\
&= g_{k-1}^\top W_\mu s_k + \left(\frac{-g_{k-1}^\top W_\mu s_k}{y_k^\top W_\mu s_k}\right) y_k^\top W_\mu s_k = 0,
\end{aligned}$$

which proves the lemma. □

Lemma 3 *Let* $\{g_k\}$ *be the sequence of gradient vectors generated by Algorithm 1. Then for all* $k \geq 1$

$$g_k^\top W_\mu g_{k-1} = 0. \tag{33}$$

Proof The proof is by induction. For the case $k = 1$, we have from Lemma 2 that $g_1^\top W_\mu g_0 = r_0^\top W_\mu g_0 = 0$. Let us now assume the inductive hypothesis on k, namely,

that $g_k^\top W_\mu g_{k-1} = 0$ for all $k = p \geq 2$. Now, we consider the next iteration, $k = p + 1$. By applying Lemma 2 and the inductive hypothesis, we find that

$$\begin{aligned} g_{p+1}^\top W_\mu g_p &= (g_{p-1} + \beta_p y_p)^\top W_\mu g_p \\ &= (g_{p-1} + \beta_p (r_p - g_{p-1}))^\top W_\mu g_p \\ &= (1 - \beta_p) g_{p-1} W_\mu g_p + \beta_p r_p^\top W_\mu g_p = 0. \end{aligned} \tag{34}$$

Therefore, we have shown that $g_k^\top W_\mu g_{k-1} = 0$, for all $k \geq 1$, and the proof is complete. □

Lemma 4 *In Algorithm 1, the following statements hold for all* $k \geq 1$

a. $g_k^\top W_\mu (z_{k-1} - x_{k-1}) = g_{k-1}^\top W_\mu (z_k - x_k) = 0.$
b. $y_k = A s_k.$
c. $y_k^\top W_\mu s_k = (x_k - x_{k-1})^\top W_\mu A (x_k - x_{k-1}) - \frac{(g_k^\top W_\mu g_k)^2}{g_k^\top W_\mu A g_k}.$
d. $g_{k+1}^\top W_\mu x_{k+1} = g_{k+1}^\top W_\mu x_k = g_{k+1}^\top W_\mu x_{k-1}.$
e. $g_{k-1}^\top W_\mu (x_k - x_{k-1}) = -(g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}).$
f. $\beta_k > 1.$

Proof (*a*) From Step 4 in Algorithm 1 and Lemma 3, we have

$$g_k^\top W_\mu (z_{k-1} - x_{k-1}) = -\alpha_{k-1} (g_k^\top W_\mu g_{k-1}) = 0,$$

and also

$$g_{k-1}^\top W_\mu (z_k - x_k) = -\alpha_k (g_{k-1}^\top W_\mu g_k) = 0.$$

(*b*) Using several steps in Algorithm 1 it follows that

$$y_k = r_k - g_{k-1} = g_k - g_{k-1} - \alpha_k A g_k = A(x_k - x_{k-1} - \alpha_k g_k) = A(z_k - x_{k-1}) = A s_k.$$

(*c*) From Steps 3, 4, 5, and 7 in Algorithm 1 and Lemmas 2 and 3, we obtain

$$\begin{aligned} y_k^\top W_\mu s_k &= (r_k - g_{k-1})^\top W_\mu (z_k - x_{k-1}) \\ &= (g_k - g_{k-1} - \alpha_k w_k)^\top W_\mu (x_k - x_{k-1} - \alpha_k g_k) \\ &= (g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}) - \alpha_k w_k^\top W_\mu (x_k - x_{k-1}) \\ &\quad -\alpha_k (g_k - g_{k-1})^\top W_\mu g_k + \alpha_k^2 w_k^\top W_\mu g_k \\ &= (g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}) - \alpha_k w_k^\top W_\mu (x_k - x_{k-1}) \\ &= (g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}) - \alpha_k g_k^\top W_\mu (g_k - g_{k-1}) \\ &= (g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}) - \alpha_k g_k^\top W_\mu g_k \\ &= (x_k - x_{k-1})^\top W_\mu A (x_k - x_{k-1}) - \frac{(g_k^\top W_\mu g_k)^2}{g_k^\top W_\mu A g_k}. \end{aligned} \tag{35}$$

(*d*) Using Steps 4 and 6 in Algorithm 1, Lemma 2, and Lemma 3, we get

$$\begin{aligned} g_{k+1}^\top W_\mu (x_k - x_{k-1}) &= g_{k+1}^\top W_\mu (s_k + \alpha_k g_k) \\ &= g_{k+1}^\top W_\mu s_k + \alpha_k g_{k+1}^\top W_\mu g_k = 0. \end{aligned}$$

Hence, $g_{k+1}^\top W_\mu x_k = g_{k+1}^\top W_\mu x_{k-1}$. Now, it follows from Step 9 in Algorithm 1 and Lemma 2 that

$$\begin{aligned} g_{k+1}^\top W_\mu (x_{k+1} - x_k) &= g_{k+1}^\top W_\mu (x_{k-1} - x_k + \beta_k s_k) \\ &= g_{k+1}^\top W_\mu (x_{k-1} - x_k) + \beta_k g_{k+1}^\top W_\mu s_k \\ &= g_{k+1}^\top W_\mu (x_{k-1} - x_k) = 0, \end{aligned}$$

which yields

$$g_{k+1}^\top W_\mu x_{k+1} = g_{k+1}^\top W_\mu x_k = g_{k+1}^\top W_\mu x_{k-1}.$$

(e) By using the previous item, it follows that

$$\begin{aligned} 0 &= g_k^\top W_\mu (x_k - x_{k-1}) \\ &= (g_k - g_{k-1} + g_{k-1})^\top W_\mu (x_k - x_{k-1}) \\ &= (g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}) + g_{k-1}^\top W_\mu (x_k - x_{k-1}), \end{aligned}$$

which implies that

$$g_{k-1}^\top W_\mu (x_k - x_{k-1}) = -(g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}). \tag{36}$$

(f) First, let us note that we can rewrite the parameter β_k as follows

$$\begin{aligned} \beta_k &= \frac{g_{k-1}^\top W_\mu (x_{k-1} - z_k)}{y_k^\top W_\mu s_k} \\ &= \frac{g_{k-1}^\top W_\mu (x_{k-1} - x_k + \alpha_k g_k)}{y_k^\top W_\mu s_k} \\ &= \frac{g_{k-1}^\top W_\mu (x_{k-1} - x_k)}{y_k^\top W_\mu s_k}. \end{aligned} \tag{37}$$

Now combining (35), (36), and (37), we have

$$\beta_k = \frac{(g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1})}{(g_k - g_{k-1})^\top W_\mu (x_k - x_{k-1}) - \alpha_k \, (g_k^\top W_\mu g_k)}, \tag{38}$$

or equivalently,

$$\beta_k = \frac{(x_k - x_{k-1})^\top W_\mu A (x_k - x_{k-1})}{(x_k - x_{k-1})^\top W_\mu A (x_k - x_{k-1}) - \alpha_k \, (g_k^\top W_\mu g_k)}. \tag{39}$$

Using (b) and (c), and the fact that AW_μ is a symmetric positive definite matrix, we conclude that the denominator in (39) satisfies

$$(x_k - x_{k-1})^\top W_\mu A (x_k - x_{k-1}) - \alpha_k \, (g_k^\top W_\mu g_k) = y_k^\top W_\mu s_k = s_k^T A W_\mu s_k > 0,$$

which means that

$$(x_k - x_{k-1})^\top W_\mu A (x_k - x_{k-1}) > \alpha_k \, (g_k^\top W_\mu g_k) > 0.$$

Therefore, we conclude that the numerator of β_k in (39) is strictly bigger than the denominator and they are both positive. Hence, $\beta_k > 1$ for all $k \geq 1$, and the result is established. □

4.1 Finite termination

Our next result plays a fundamental role to establish the finite termination of the GDWGM family of methods.

Theorem 2 *Algorithm 1 generates the sequences* $\{g_k\}$ *and* $\{r_k\}$ *such that*

a. *For* $k \geq 2$, $g_k^\top W_\mu g_j = 0$, *for all* $-1 \leq j \leq k-2$.

b. *For* $k \geq 2$, $r_k^\top W_\mu g_j = 0$, *for all* $-1 \leq j \leq k-2$.

Proof The proof is by induction. Concerning (a), since $g_{-1} = g_0$, $x_1 = z_0$, and $\alpha_0 > 0$, using (d) in Lemma 4 and step 4 of Algorithm 1, we have

$$\begin{aligned} g_2^\top W_\mu g_{-1} &= g_2^\top W_\mu g_0 \\ &= \frac{1}{\alpha_0} g_2^\top W_\mu (x_0 - z_0) \qquad (40) \\ &= \frac{1}{\alpha_0} g_2^\top W_\mu (x_0 - x_1) = 0, \qquad (41) \end{aligned}$$

and the result is obtained for $k = 2$. Concerning (b), since $\alpha_0 > 0$, $g_{-1} = g_0$, using (a) in Lemma 2, Lemma 3, steps 4 and 5 of GDWGM, (41) and the fact that $AW_\mu = W_\mu A$, we obtain

$$\begin{aligned} r_2^\top W_\mu g_0 &= \frac{1}{\alpha_0} r_2^\top W_\mu (x_0 - z_0) \\ &= \frac{1}{\alpha_0} r_2^\top W_\mu (x_0 - x_1) = -\frac{1}{\alpha_0} r_2^\top W_\mu (x_1 - x_0) + 0 \\ &= -\frac{1}{\alpha_0} r_2^\top W_\mu (x_1 - x_0) + \frac{1}{\alpha_0} g_2^\top W_\mu (x_1 - x_0) \\ &= \frac{1}{\alpha_0} (g_2 - r_2)^\top W_\mu (x_1 - x_0) \\ &= \frac{\alpha_2}{\alpha_0} g_2^\top A W_\mu (x_1 - x_0) = \frac{\alpha_2}{\alpha_0} g_2^\top W_\mu (g_1 - g_0) \\ &= \frac{\alpha_2}{\alpha_0} [g_2^\top W_\mu g_1 - g_2^\top W_\mu g_0] = 0. \end{aligned}$$

In addition, since $r_2^\top W_\mu g_{-1} = r_2^\top W_\mu g_0$, then the result is established for $k = 2$.

Let us now assume, by induction on k, that (a) and (b) hold up to $k = \hat{k} \geq 3$, and consider the next iteration. Hence, we need to show that $g_{\hat{k}+1}^\top W_\mu g_j = 0$, and also that $r_{\hat{k}+1}^\top W_\mu g_j = 0$, for all $-1 \leq j \leq \hat{k} - 1$.

For $-1 \leq j \leq \hat{k} - 2$, using Lemma 3, Steps 7 and 10 in Algorithm 1, and the inductive hypothesis associated with (a) and (b), we have that

$$\begin{aligned} g_{\hat{k}+1}^\top W_\mu g_j &= (g_{\hat{k}-1} + \beta_{\hat{k}}(r_{\hat{k}} - g_{\hat{k}-1}))^\top W_\mu g_j \\ &= (1 - \beta_{\hat{k}})\, g_{\hat{k}-1}^\top W_\mu g_j + \beta_{\hat{k}} r_{\hat{k}}^\top W_\mu g_j = 0. \end{aligned}$$

For $j = \hat{k} - 1$, using step 4, adding and subtracting $x_{\hat{k}-2}$, and then using the fact that $z_{\hat{k}-1} - x_{\hat{k}-2} = (x_{\hat{k}} - x_{\hat{k}-2})/\beta_{\hat{k}-1}$ (obtained from Steps 6 and 9 in Algorithm 1), we get

$$\begin{aligned} g_{\hat{k}+1}^\top W_\mu g_{\hat{k}-1} &= -\frac{1}{\alpha_{\hat{k}-1}}\, g_{\hat{k}+1}^\top W_\mu (z_{\hat{k}-1} - x_{\hat{k}-1}) \\ &= -\frac{1}{\alpha_{\hat{k}-1}}\, g_{\hat{k}+1}^\top W_\mu (z_{\hat{k}-1} - x_{\hat{k}-2} + x_{\hat{k}-2} - x_{\hat{k}-1}) \\ &= -\frac{1}{\alpha_{\hat{k}-1}}\, g_{\hat{k}+1}^\top W_\mu \left(\frac{1}{\beta_{\hat{k}-1}} (x_{\hat{k}} - x_{\hat{k}-2}) + x_{\hat{k}-2} - x_{\hat{k}-1} \right) \\ &= -\frac{1}{\alpha_{\hat{k}-1}} \left[\frac{1}{\beta_{\hat{k}-1}} g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}} - x_{\hat{k}-2}) + g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-2} - x_{\hat{k}-1}) \right]. \end{aligned}$$

Now, adding and subtracting $g_{\hat{k}+1}^\top W_\mu x_{\hat{k}-1}$, and using (d) in Lemma 4, we arrive at

$$\begin{aligned} g_{\hat{k}+1}^\top W_\mu g_{\hat{k}-1} &= -\frac{1}{\alpha_{\hat{k}-1}} \left[\frac{g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}} - x_{\hat{k}-1}) + g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2})}{\beta_{\hat{k}-1}} + g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-2} - x_{\hat{k}-1}) \right] \\ &= -\frac{1}{\alpha_{\hat{k}-1}} \left[\frac{1}{\beta_{\hat{k}-1}} g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) + g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-2} - x_{\hat{k}-1}) \right] \\ &= \gamma_{\hat{k}}\, g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}), \end{aligned} \tag{42}$$

where $\gamma_{\hat{k}} = (\beta_{\hat{k}-1} - 1)/(\alpha_{\hat{k}-1}\beta_{\hat{k}-1})$ is well-defined since $\beta_k > 1$ for all k and

$$\alpha_k = \frac{g_k^\top W_\mu g_k}{g_k^\top W_\mu A g_k} = \frac{(W_\mu^{1/2} g_k)^\top (W_\mu^{1/2} g_k)}{(W_\mu^{1/2} g_k)^\top A (W_\mu^{1/2} g_k)},$$

is an inverse Rayleigh-quotient of A, i.e., $\alpha_k \geq \frac{1}{\lambda_1} > 0$ for all k.

By using Steps 5, 7, and 10 in Algorithm 1, Lemma 3, Lemma 4, the inductive hypothesis, and the fact that $AW_\mu = W_\mu A$, we obtain

$$\begin{aligned} g_{\hat{k}+1}^\top W_\mu g_{\hat{k}-1} &= \gamma_{\hat{k}}\, g_{\hat{k}+1}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= \gamma_{\hat{k}}\, (g_{\hat{k}-1} + \beta_{\hat{k}} (r_{\hat{k}} - g_{\hat{k}-1}))^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= \gamma_{\hat{k}} (1 - \beta_{\hat{k}})\, g_{\hat{k}-1}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) + \gamma_{\hat{k}} \beta_{\hat{k}}\, r_{\hat{k}}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= \gamma_{\hat{k}} \beta_{\hat{k}}\, r_{\hat{k}}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= \gamma_{\hat{k}} \beta_{\hat{k}}\, (g_{\hat{k}} - \alpha_{\hat{k}} A g_{\hat{k}})^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= \gamma_{\hat{k}} \beta_{\hat{k}}\, g_{\hat{k}}^\top W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) - \gamma_{\hat{k}} \beta_{\hat{k}} \alpha_{\hat{k}}\, g_{\hat{k}}^\top A W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= -\gamma_{\hat{k}} \beta_{\hat{k}} \alpha_{\hat{k}}\, g_{\hat{k}}^\top A W_\mu (x_{\hat{k}-1} - x_{\hat{k}-2}) \\ &= -\gamma_{\hat{k}} \beta_{\hat{k}} \alpha_{\hat{k}}\, g_{\hat{k}}^\top W_\mu (g_{\hat{k}-1} - g_{\hat{k}-2}) \\ &= -\gamma_{\hat{k}} \beta_{\hat{k}} \alpha_{\hat{k}}\, [g_{\hat{k}}^\top W_\mu g_{\hat{k}-1} - g_{\hat{k}}^\top W_\mu g_{\hat{k}-2}] = 0. \end{aligned}$$

Therefore, (a) is established for all $k \geq 2$ and for $-1 \leq j \leq k-2$.

On the other hand, concerning (b), for $-1 \leq j \leq \hat{k}-1$, using Steps 7 and 10 in Algorithm 1, Lemma 3, item (a) which has now been established, and that $\beta_k > 1$ for all k, we obtain

$$\begin{aligned} r_{\hat{k}+1}^{\top} W_\mu g_j &= \frac{1}{\beta_{\hat{k}+1}} \left(g_{\hat{k}+2} + (\beta_{\hat{k}+1} - 1) g_{\hat{k}}\right)^{\top} W_\mu g_j \\ &= \frac{1}{\beta_{\hat{k}+1}} g_{\hat{k}+2} W_\mu g_j + \frac{\beta_{\hat{k}+1} - 1}{\beta_{\hat{k}+1}} g_{\hat{k}}^{\top} W_\mu g_j = 0, \end{aligned}$$

and (b) is also established, which completes the proof. □

In summary, combining Lemma 3 with item (a) from Theorem 2, it follows that for all k, g_k is W_μ-orthogonal to all previous gradient vectors, i.e., for all $k \geq 1$

$$g_k^{\top} W_\mu g_j = 0, \quad \text{for all} \quad j \leq k-1. \tag{43}$$

Theorem 3 *For any initial guess $x_0 \in \mathbb{R}^n$, Algorithm 1 generates the iterates x_k, $k \geq 1$, such that $x_n = A^{-1}b$.*

Proof Using (43), we have that the first n gradient vectors g_k $(0 \leq k \leq n-1)$ generated by Algorithm 1 form a W_μ-orthogonal set, which implies that they form a linearly independent set of n vectors in $\mathbb{R}^n$. As a consequence, to satisfy (43), the next gradient vector $g_n \in \mathbb{R}^n$ must be zero. Thus, $x_n = A^{-1}b$. □

Concerning the finite termination of Algorithm 1, as it has been already established for the extreme cases: $\mu = 0$ (CG, see, e.g., [26, 32]) and $\mu = 1$ (DWGM, see [1]), all the infinitely many members of the GDWGM family actually terminate in at most $p \leq n$ iterations where p is the number of distinct eigenvalues of the matrix A. To establish this fundamental result, we first need to show that for all $k \geq 1$ the vector g_k generated by GDWGM belongs to the Krylov subspace

$$\mathcal{K}_{k+1}(A, g_0) := span\{g_0, Ag_0, A^2 g_0, \ldots, A^k g_0\}.$$

Lemma 5 *In Algorithm GDWGM, for all $k \geq 1$, $g_k \in \mathcal{K}_{k+1}(A, g_0)$.*

Proof The proof is identical to the proof of Lemma 7 in [1]. □

Theorem 4 *If A has only $p < n$ distinct eigenvalue, then for any initial guess $x_0 \in \mathbb{R}^n$ Algorithm 1 generates the iterates x_k, $k \geq 1$ such that $x_p = A^{-1}b$.*

Proof The proof is identical to the proof of Theorem 9 in [1]. □

4.2 Minimization of $F_\mu(\cdot)$ on the explored affine subspace

We now focus our attention on the minimization property of the map $F_\mu(\cdot)$, at each iteration, on the already explored affine subspace. For that, we need to establish for any $\mu \in [0, 1]$ the W_μ-orthogonality of the current gradient g_k with all the

previously explored search directions, which are given by the vectors $(x_j - x_{j-1})$ for $1 \leq j \leq k$.

Theorem 5 *For any $\mu \in [0, 1]$, Algorithm 1 generates the sequences $\{g_k\}$ and $\{x_k\}$ such that for $k \geq 2$*

$$g_k^\top W_\mu (x_j - x_{j-1}) = 0, \quad \textit{for } \ 1 \leq j \leq k. \tag{44}$$

Proof The proof is by induction. For $k = 2$, using (d) in Lemma 4, we have that

$$g_2^\top W_\mu (x_2 - x_1) = g_2^\top W_\mu (x_1 - x_1) = 0.$$

Let us now assume that (44) holds up to $k = p$, and let us consider the next iteration. When $j = p$, using again (d) in Lemma 4, we obtain

$$g_{p+1}^\top W_\mu (x_{p+1} - x_p) = g_{p+1}^\top W_\mu (x_p - x_{p-1}) = 0.$$

It remains to consider $1 \leq j \leq p - 1$. Using the inductive hypothesis, Steps 7 and 10 in Algorithm 1, and Theorem 2, we have that

$$\begin{aligned}
g_{p+1}^\top W_\mu (x_j - x_{j-1}) &= (g_{p-1} + \beta_p y_p)^\top W_\mu (x_j - x_{j-1}) \\
&= -((\beta_p - 1) g_{p-1} - \beta_p r_p)^\top W_\mu (x_j - x_{j-1}) \\
&= \beta_p r_p^\top W_\mu (x_j - x_{j-1}) \\
&= \beta_p (g_p - \alpha_p A g_p)^\top W_\mu (x_j - x_{j-1}) \\
&= -\beta_p \alpha_p (A g_p)^\top W_\mu (x_j - x_{j-1}) \\
&= -\beta_p \alpha_p g_p^\top W_\mu A (x_j - x_{j-1}) \\
&= -\beta_p \alpha_p g_p^\top W_\mu (g_j - g_{j-1}) = 0.
\end{aligned}$$

□

Let us notice that, at iteration k, the explored affine subspace V_k is given by

$$V_k = \left\{ x \in \mathbb{R}^n \ \mid \ x = x_0 + \sum_{j=1}^{k} \eta_j (x_j - x_{j-1}) \ \text{ and } \ \eta = (\eta_1, \eta_2 \ldots, \eta_k) \in \mathbb{R}^k \right\}. \tag{45}$$

Corollary 1 *For all $k \geq 1$, the iterate x_k generated by Algorithm 1 is the argument that minimizes the merit function $F_\mu(\cdot)$ on V_k.*

Proof Let us consider the following convex minimization problem

$$\min F_\mu(x) \quad \text{subject to} \quad x \in V_k \subseteq \mathbb{R}^n. \tag{46}$$

Clearly, the constrained problem (46) is equivalent to the following unconstrained minimization problem

$$\min_{\eta \in \mathbb{R}^k} G_\mu(\eta) = F_\mu(\mathbf{x}(\eta)), \tag{47}$$

where $\mathbf{x} : \mathbb{R}^k \to \mathbb{R}^n$ is a linear function defined by $\mathbf{x}(\eta) := x_0 + \sum_{j=1}^{k} \eta_j (x_j - x_{j-1})$, and η_j denotes the j-th entry of η. Now, observe that the cost function $G_\mu(\cdot)$ is clearly

a strictly convex function in $\mathbb{R}^k$. This fact implies that (47) has a unique solution, say $\eta^* \in \mathbb{R}^k$. Then, η^* must satisfy the first-order necessary optimality conditions

$$\frac{\partial G_\mu(\eta)}{\partial \eta_j} = \nabla f(\mathbf{x}(\eta))^\top W_\mu(x_j - x_{j-1}) = 0, \quad \text{for } 1 \leq j \leq k, \tag{48}$$

which are also sufficient due to the convexity of problem (47).

Hence, it follows that $\nabla f(\mathbf{x}(\eta^*))$ is W_μ-orthogonal to the subspace generated by the set $\{x_k - x_{k-1}, \ldots, x_1 - x_0\}$. In view of Theorem 5, we have that g_k is also W_μ-orthogonal to the subspace generated by the set $\{x_k - x_{k-1}, \ldots, x_1 - x_0\}$. Moreover, note that selecting $\eta = (1, 1, \ldots 1) \in \mathbb{R}^k$ we obtain that $x_k = \mathbf{x}(\eta) \in V_k$ and also $\nabla f(\mathbf{x}(\eta)) = g_k$. Therefore, by the uniqueness of the solution of (47), we find that $g_k = \nabla f(\mathbf{x}(\eta^*))$. Then, applying the equivalence of the minimization problems (46) and (47), we have that the iterate x_k, generated by Algorithm 1, can be written as

$$x_k = \mathbf{x}(\eta^*) = x_0 + \sum_{j=1}^{k} \eta_j^*(x_j - x_{j-1}),$$

which completes the proof. □

Remark 4 The subspace generated by the vector set $\{x_k - x_{k-1}, \ldots, x_1 - x_0\}$, which appears in (45), coincides with the Krylov subspace $\mathcal{K}_k(A, g_0)$. Indeed, since both subspaces have the same dimension, it suffices to show that

$$(x_j - x_{j-1}) \in \mathcal{K}_j(A, g_0) \quad \text{for each } j \geq 1. \tag{49}$$

For $j = 1$, using Lemma 2 and Step 4 in Algorithm 1, we know that

$$x_1 - x_0 = -\alpha_0 g_0 \in \mathcal{K}_1(A, g_0).$$

Let us now assume, by induction on j, that (49) holds up to $j = k$, and consider the next iteration. From (13), we have that

$$x_{k+1} - x_k = (\beta_k - 1)(x_k - x_{k-1}) - \beta_k \alpha_k g_k.$$

Using now the inductive hypothesis (49), we get that $(x_k - x_{k-1}) \in \mathcal{K}_k(A, g_0)$, and using Lemma 5, we obtain that $g_k \in \mathcal{K}_{k+1}(A, g_0)$. Hence, $x_{k+1} - x_k \in \mathcal{K}_{k+1}(A, g_0)$. From the fact that the two mentioned subspaces are identical, combined with Corollary 1, we conclude that for all $k \geq 1$ the iterate x_k generated by Algorithm 1 is the argument that minimizes the merit function $F_\mu(\cdot)$ on the affine subspace $x_0 + \mathcal{K}_k(A, g_0)$.

We are now ready to show that the iterate x_k generated by the CG method for solving (1) coincides at each k with the k-th iterate generated by Algorithm 1 when $\mu = 0$, as long as both methods start at the same initial point x_0. For that, let us first recall a couple of key properties of the CG method for solving (1): at each iteration k, the subspace generated by all the already explored directions (say $span\{d_0, d_1, \ldots, d_{k-1}\}$) coincides with the Krylov subspace $\mathcal{K}_k(A, g_0)$, and also that the iterate x_k is the minimizer of the strictly convex quadratic $E(x) = f(x) + \frac{1}{2} b^\top x^*$ on the entire explored affine subspace, i.e., on the affine subspace given by $x_0 + \mathcal{K}_k(A, g_0)$; see, e.g., [22, 26, 32].

On the other hand, when $\mu = 0$ in Algorithm 1, the merit function $F_\mu(\cdot)$ reduces to the strictly convex quadratic function $E(\cdot)$, and based on Remark 4, we conclude that both methods generate iterates that minimize the same function $E(\cdot)$ on the same affine subspace. Since $E(\cdot)$ is a strictly convex function then it has a unique global minimizer on that affine subspace, which imply that if we start the CG method and Algorithm 1 with $\mu = 0$ from the same initial guess x_0, then they produce the same iterates for all k.

5 Numerical results

In order to give further insight into the GDWGM family of methods, we present the results of some numerical experiments. We test our algorithm on some well-known real large-scale strictly convex quadratic problems. All experiments have been performed on an intel(R) CORE(TM) i7-4770, CPU 3.40 GHz with 500GB HD and 16GB RAM. The algorithm was coded in Matlab (version 2017b) with double precision. The running times are always given in CPU seconds. The implementation of our algorithm is available in http://www.optimization-online.org/DB_HTML/2020/09/8039.html.

We analyze the numerical behavior of the GDWGM algorithm to approximate the solution of randomly generated dense linear systems of equations, and also of some sparse linear systems of equations with real data. In our numerical tests, we run all the algorithms up to $K = 150000$ iterations and stop them at iteration $k < K$ if $||\nabla f(x_k)||_2 < \epsilon ||\nabla f(x_0)||_2$. For comparison purposes, we compare the numerical performance of our GDWGM family of methods with the classical conjugate gradient method (CG) and the recently published delayed weighted gradient method (DWGM). For the GDWGM family that depends on the parameter μ, we present the numerical results associated with the best value of μ taken exhaustively in the following set $\Omega = \{0, 0.05, 0.1, 0.15, \ldots, 0.95, 1\}$.

The first set of test problems includes randomly generated dense positive definite matrices assembled as $A = QDQ^\top$, where

$$Q = \left(I - 2\frac{v_1 v_1^\top}{||v_1||_2^2}\right)\left(I - 2\frac{v_2 v_2^\top}{||v_2||_2^2}\right)\left(I - 2\frac{v_3 v_3^\top}{||v_3||_2^2}\right),$$

where $v_1, v_2, v_3 \in \mathbb{R}^{n\times n}$ are random vectors, $D = diag(d_1, \ldots, d_n)$ is a diagonal matrix where $d_1 = 1e\text{-}5$, $d_2, d_3, \ldots d_{n/5}$ are distributed in $[1, 100]$, and the rest of the d_i's numbers follow a uniform distribution in the interval $[\frac{\kappa(A)}{2}, \kappa(A)]$, where $\kappa(A) = \lambda_1/\lambda_n$. This random and dense experiment design was originally proposed in [16], and has also been employed in [8, 19, 34]. In particular, we set $\kappa(A) = 10^4$ and use three values for the tolerance $\epsilon \in \{1e\text{-}8, 1e\text{-}10, 1e\text{-}12\}$. The vector $b \in \mathbb{R}^n$ and the initial point $x_0 \in \mathbb{R}^n$ were generated with the following Matlab commands:

$$b = 20 * rand(n, 1) - 10; \quad x_0 = zeros(n, 1).$$

For each pair (n, ϵ), we randomly generate 100 independent simulations and, in Table 1, we report the average number of iterations (IT), the average number of total

Table 1 Numerical results for randomly generated dense problems

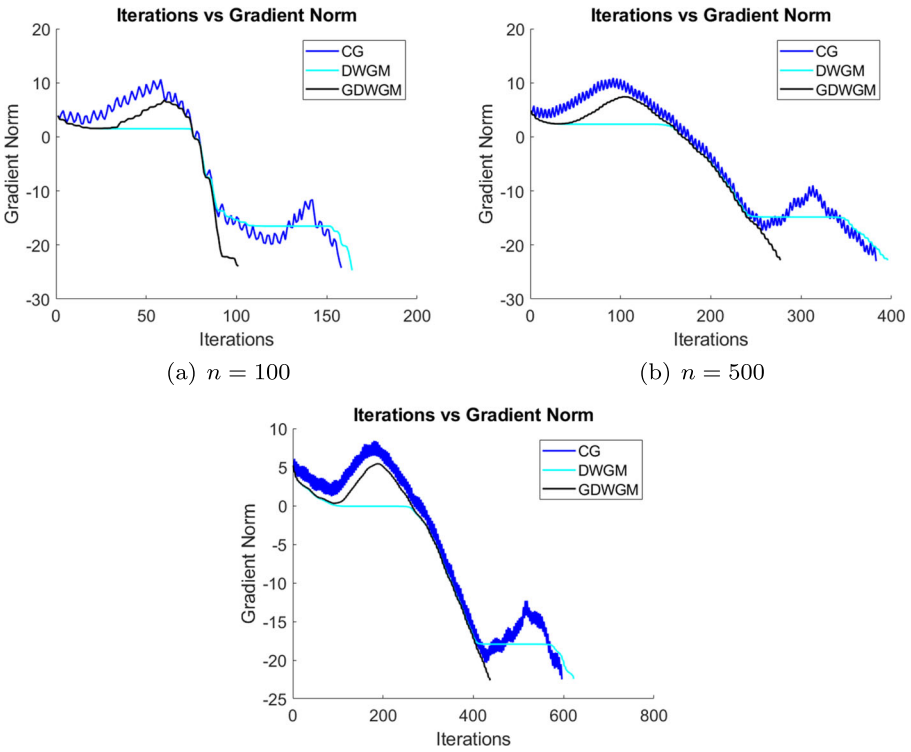
n	ϵ	CG			DWGM			GDWGM			
		IT	CT	Res	IT	CT	Res	IT	CT	Res	μ
100	1e-8	97	0.0014	5.86e-09	108	0.0021	7.55e-09	91	0.0018	4.68e-09	0.145
	1e-10	140	0.0018	5.61e-11	164	0.0030	6.45e-11	98	0.0018	6.56e-11	0.212
	1e-12	167	0.0022	5.08e-13	181	0.0032	7.07e-13	117	0.0020	8.19e-13	0.43
500	1e-8	283	0.0080	7.12e-09	282	0.0096	8.22e-09	280	0.0089	8.22e-09	0.124
	1e-10	328	0.0099	7.41e-11	378	0.0127	8.41e-11	302	0.0097	8.20e-11	0.136
	1e-12	440	0.0136	7.07e-13	456	0.0163	8.17e-13	324	0.0111	8.82e-13	0.322
1000	1e-8	380	0.0397	7.79e-09	377	0.0434	8.72e-09	374	0.0412	8.84e-09	0.137
	1e-10	415	0.0433	7.93e-11	451	0.0511	9.24e-11	403	0.0451	8.94e-11	0.163
	1e-12	569	0.0601	8.11e-13	588	0.0662	8.79e-13	438	0.0472	9.15e-13	0.224

computational time (CT), and the average number of the residual (Res) defined by $Res(\hat{x}) = ||\nabla f(\hat{x})||_2/||\nabla f(x_0)||_2$, where $\hat{x}$ is the estimated solution achieved by each method. In addition, for our GDWGM method, we report the average value of μ. From Table 1, we notice that GDWGM outperforms the DWGM and CG methods for intermediate values of the parameter μ, that is for $\mu \in (0, 1)$. For these types of experiments, the selection of an appropriate $\mu \in (0, 1)$ shows its potential as can be seen in Fig. 1. In particular, we can observe in Fig. 1 that, for some specific values of μ, our approach requires less iterations than the CG and DWGM methods to achieve convergence. We also note that the best possible $\mu \in (0, 1)$ is different for each pair (n, ϵ), and so it is problem dependent.

In our second experiment, we consider the application of Algorithm 1 to approximate the solution of 40 sparse symmetric and positive definite linear system of equations $Ax = b$, where the matrices $A \in \mathbb{R}^{n\times n}$ are taken from the SuiteSparse Matrix Collection [9][1]; meanwhile, the vector $b \in \mathbb{R}^n$ and the initial point $x_0 \in \mathbb{R}^n$ are generated by the following Matlab commands: $b = A * \texttt{ones}(n, 1)$ and $x_0 = \texttt{zeros}(n, 1)$, respectively. This particular design of experiments was also considered in [5]. In this experiment, we use $\epsilon =$ 1e-6.

The numerical results concerning this experiment are summarized in Table 2. In this table, "Fres" denotes the final residual objective value, i.e., $\text{Fres} = |f(\hat{x}) - f(x^*)|$, where $\hat{x}$ denotes the approximated solution obtained by each method and $x^* = A^{-1}b$; "CT"and "IT" denote the total computing time in seconds and the number of iterations, respectively. We also report, in the last column of Table 2, the value of $\mu \in \Omega$, for which GDWGM reaches the required precision in the gradient norm in the fewest possible number of iterations. As shown in Table 2, in most cases, both the DWGM and GDWGM methods reach the desired gradient-norm accuracy, in fewer iterations than the CG method. However, the CG method obtains a lower value of

[1]The SuiteSparse Matrix Collection tool–box is available in https://sparse.tamu.edu/.



(a) $n = 100$ (b) $n = 500$

(c) $n = 1000$

Fig. 1 Convergence history of the three algorithms for the randomly generated dense matrices, when $n = 100, 500, 1000$ and $\epsilon = 1e\text{-}12$. For GDWGM, we use **a** $\mu = 0.2$, **b** $\mu = 0.3$, and **c** $\mu = 0.1$. The y-axis represents the logarithm of the relative gradient norm, that is $\log(||\nabla f(x_k)||_2/||\nabla f(x_0)||_2)$

the residual *Fres* than the other methods. This means that DWGM and its generalization reduce the gradient norm faster than CG, while CG approaches the optimal value $f(x^*)$ faster than the other two methods. Furthermore, from Table 2, we see that in most cases, it was possible to find a value of $\mu \in (0, 1)$, for which GDWGM converges in fewer iterations than the DWGM and the CG methods.

In Fig. 2, we illustrate the behavior of Algorithm 1 by varying μ, for the matrices "1138_bus" and "cfd1." These figures show that Algorithm 1 can converge to the solution of the system of linear equations in a different number of iterations for different values of μ. Additionally, for these two special sparse matrices, we note that values close to $\mu = 1$, in Algorithm 1, achieve convergence in less number of iterations.

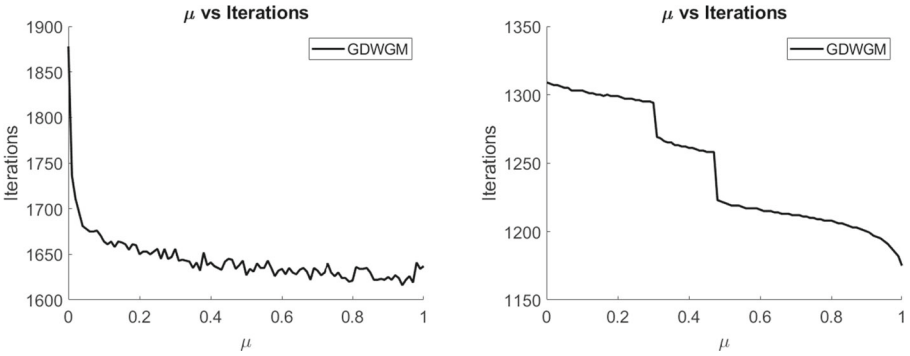
On the other hand, in Figs. 3 and 4, we plot the convergence history of CG, DWGM, and GDWGM, from the same initial point, considering the following three measures: $|f(x_k) - f(x^*)|$, $||\nabla f(x_k)||_2$, and $|F_\mu(x_k) - F_\mu(x^*)|$, for the instances "apache1" with $\mu = 0.15$, and "1138bus" with $\mu = 0.3$, respectively. From these figures, we can see that CG is superior to the other methods minimizing the objective

Table 2 Numerical results for solving linear systems from the SuiteSparse Matrix Collection

Name	n	CG			DWGM			GDWGM			
		IT	CT	Fres	IT	CT	Fres	IT	CT	Fres	μ
1138_bus	1138	1752	0.02	4.65e-8	1637	0.0281	3.81e-6	1621	0.0244	2.66e-6	0.8
2cubes_sphere	101492	16831	41.6334	1.62e+5	1888	5.6172	1.60e+7	1886	5.5244	1.60e+7	0.05
af_0_k101	503625	3009	61.4318	4.17e+2	1101	27.3089	4.07e+3	1101	27.8893	4.00e+3	0.25
af_shell3	504855	1450	28.4402	3.30e-3	948	23.328	7.43e-1	944	23.205	7.28e-1	0.55
af_shell7	504855	1441	29.1214	3.40e-3	947	21.0272	7.43e-1	941	22.4374	7.38e-1	0.75
apache1	80800	1490	1.202	2.51e-10	1458	1.6442	1.11e-9	1454	1.5736	6.33e-10	0.15
bcsstk09	1083	178	0.0055	2.38e-2	168	0.0068	4.41e-1	168	0.0059	4.41e-1	0.05
bcsstk10	1086	1690	0.0348	1.40e-1	1026	0.0267	2.02e+1	1024	0.0232	2.02e+1	0.4
bcsstk11	1473	1636	0.0526	1.77e+2	698	0.0266	1.39e+4	697	0.0304	1.39e+4	0.45
bcsstk13	2003	10542	0.7083	3.06e+5	2239	0.1631	2.19e+7	2212	0.1591	2.19e+7	0.95
bcsstk14	1806	3096	0.1545	1.23e+3	1212	0.0711	4.67e+5	1209	0.0667	4.67e+5	0.95
bcsstk16	4884	228	0.062	3.79e+1	207	0.0568	4.47e+1	206	0.0637	4.52e+1	0.95
bcsstk17	10974	9573	3.8372	2.75e+2	4063	2.0702	3.38e+4	4051	2.033	3.38e+4	0.6
bcsstk21	3600	4173	0.164	1.40e+1	744	0.0445	5.65e+3	743	0.045	5.67e+3	0.1
bcsstk23	3134	780	0.0432	4.34e+10	533	0.0407	5.75e+11	529	0.0392	5.79e+11	0.15
bcsstk24	3562	993	0.1337	7.71e+7	555	0.0837	7.21e+8	550	0.0776	7.21e+8	0.55
bcsstk25	15439	1864	0.6184	3.57e+8	857	0.3397	2.27e+9	841	0.331	2.29e+9	0.55
bcsstk27	1224	575	0.0291	4.58e-4	477	0.0253	1.98e-2	476	0.0201	1.98e-2	0.6
bcsstk28	4410	13208	1.9773	2.50e-4	11504	2.1991	3.55e+1	11371	2.1131	3.43e+1	0.4
bcsstk36	23052	14006	21.5351	2.97e+1	3178	5.113	1.43e+3	3163	6.1965	1.42e+3	0.3
bcsstk38	8032	1126	0.343	2.10e+5	328	0.1133	6.71e+6	323	0.1128	6.72e+6	0.45

Table 2 (continued)

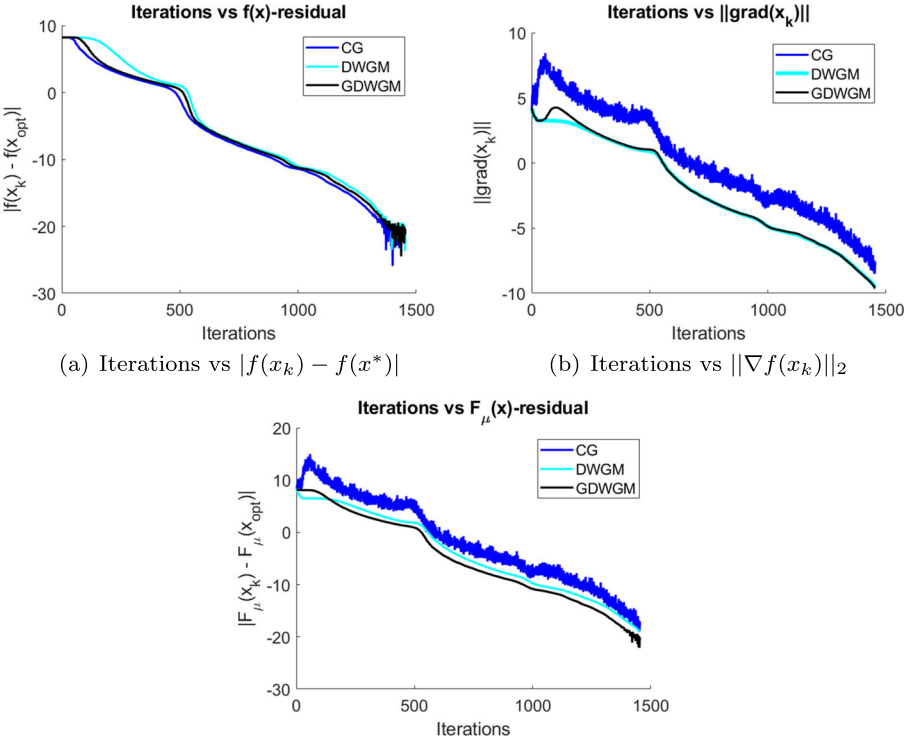
Name	n	CG			DWGM			GDWGM			
		IT	CT	Fres	IT	CT	Fres	IT	CT	Fres	μ
bcsstm08	1074	38	0.0024	9.86e-1	42	0.002	1.15e+0	41	0.0044	1.04e+0	0.1
bcsstm11	1473	20	0.0023	9.49e-6	21	0.0018	9.53e-6	20	0.004	9.49e-6	0
bcsstm23	3134	947	0.0201	1.23e+0	684	0.0283	3.08e+1	678	0.0259	2.92e+1	0.1
bcsstm24	3562	692	0.0167	3.23e-1	442	0.0181	3.09e+0	439	0.0201	3.07e+0	0.75
cfd1	70656	1307	2.9433	1.85e-8	1175	3.5296	3.37e-6	1175	3.6709	3.37e-6	1
ex15	6867	956	0.0919	1.45e+2	746	0.0857	1.29e+3	743	0.0858	1.28e+3	0.2
msc04515	4515	3517	0.2737	8.15e-1	2598	0.2444	1.76e+3	2596	0.2597	1.76e+3	0.85
msc23052	25052	14127	21.1266	1.17e+2	3160	5.3699	5.70e+3	3157	5.0011	5.70e+3	0.75
nasa4704	4704	10979	0.9319	1.52e-2	7426	0.7643	1.27e+1	7418	0.7583	1.21e+1	0.4
nasasrb	54870	15354	44.0807	1.41e+2	3477	11.0308	1.65e+5	3475	11.2238	1.65e+5	0.1
sts4098	4098	5811	0.4112	1.88e+1	1756	0.1559	1.41e+3	1747	0.1561	1.41e+3	0.5
bmw7st_1	141347	90	0.704	2.47e+10	72	0.6242	4.40e+10	70	0.5982	4.38e+10	0.35
cbuckle	13681	1820	1.2605	5.30e-3	1493	1.1764	1.32e-1	1491	1.3583	1.23e-1	0.95
ct20stif	52329	2084	5.5123	7.19e+5	1082	3.3689	1.11e+7	1077	3.3958	1.11e+7	0.1
ex13	2568	723	0.0451	1.56e+0	698	0.0481	1.11e+1	696	0.0479	1.12e+1	0.45
gyro	17361	10568	10.7319	1.10e+3	3136	3.5315	2.16e+5	3127	3.5223	2.16e+5	0.2
LFAT5000	19994	103883	15.198	7.31e+5	4970	1.076	8.47e+7	4952	1.0496	8.47e+7	0.75
nasa1824	1824	3255	0.1019	2.00e-3	2399	0.0848	3.66e-1	2390	0.0844	3.59e-1	0.55
nasa2910	2910	5208	0.6128	1.90e-3	3743	0.494	6.68e-1	3719	0.4914	6.68e-1	0.75



(a) matrix 1138_bus (b) matrix cfd1

Fig. 2 Behavior of Algorithm 1 varying μ for the matrices **a** "1138_bus" and **b** "cfd1"

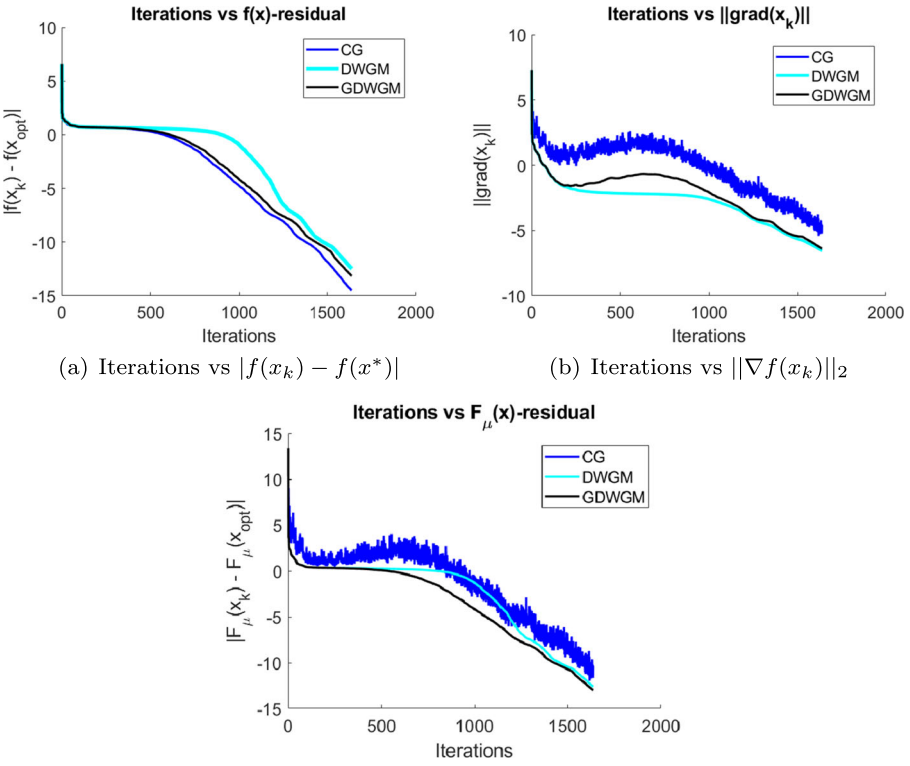
function, DWGM is the best method optimizing the gradient norm, while GDWGM is superior to the rest of the methods reducing the merit function $F_\mu(\cdot)$, which agrees with the theoretical result established in Corollary 1. In addition, we observe that



(a) Iterations vs $|f(x_k) - f(x^*)|$ (b) Iterations vs $||\nabla f(x_k)||_2$

(c) Iterations vs $|F_\mu(x_k) - F_\mu(x^*)|$

Fig. 3 Convergence history of the three considered algorithms for the matrix "apache1." For GDWGM, we use $\mu = 0.15$. In all cases, the y-axis is in logarithmic scale. We report iterations versus: **a** $|f(x_k) - f(x^*)|$, **b** $||\nabla f(x_k)||_2$, and **c** $|F_\mu(x_k) - F_\mu(x^*)|$



(a) Iterations vs $|f(x_k) - f(x^*)|$ (b) Iterations vs $||\nabla f(x_k)||_2$

(c) Iterations vs $|F_\mu(x_k) - F_\mu(x^*)|$

Fig. 4 Convergence history of all the algorithms using $\mu = 0.3$ for the matrix "1138_bus." The y-axis is in logarithmic scale. We report iterations versus: **a** $|f(x_k) - f(x^*)|$, **b** $||\nabla f(x_k)||_2$, and **c** $|F_\mu(x_k) - F_\mu(x^*)|$

the CG method shows an oscillatory pattern in terms of reducing $||\nabla f(x_k)||_2$ and $|F_\mu(x_k) - F_\mu(x^*)|$, and presents a smooth decrease in terms of reducing $f(\cdot)$, while DWGM and GDWGM reduce in a smooth way the three considered measures. The connection between GDWGM and the Moreau-Yosida regularization, described in Remark 2, is one way of accounting for the observed smooth behavior.

6 Concluding remarks and perspectives

We have proposed and analyzed a family of optimal first-order methods for the minimization of strictly convex quadratic functions. Similar to the CG method, each member of the family has certain orthogonality properties. Specifically, we proved that the gradient vector at the current iteration is W_μ-orthogonal to all the previous gradient vectors, which implies directly the finite termination of the method for all $\mu \in [0, 1]$. Moreover, we demonstrated that if the matrix $A \in \mathbb{R}^{n\times n}$ has only $p < n$ distinct eigenvalues, then the proposed algorithm obtains the desired solution in exactly p iterations. In addition, we show that any member of the family constructs

a sequence of points $\{x_k\}$, such that x_k verifies an optimality condition related to the problem of minimizing the merit function $F_\mu(\cdot)$ over the linear manifold generated by all the explored previous search directions. We also establish that the sequence $\{F_\mu(x_k)\}$ converges to zero q-linearly when k tends to infinity for all $\mu \in [0, 1]$, which implies that the sequence $\{x_k\}$ converges to the unique global minimizer of $f(\cdot)$. Finally, we have tested our procedure on a variety of dense and sparse large-scale symmetric positive definite linear systems of equations, in order to illustrate its performance.

The attractiveness of the proposed family is based mainly on its strong global convergence properties similar to the mathematical magic that the conjugate gradient method has for the minimization of quadratic cost functions, and its simplicity characterized by low storage requirements and a very low computational cost per iteration. These good features make each member of this family a very nice candidate to tackle the solution of large-scale positive definite linear systems of equations. Another fundamental feature of this novel approach is that it provides a collection of optimal methods that allows the user to choose a suitable weight $\mu \in (0, 1)$, in order to favor the reduction of $f(\cdot)$, or to promote the decrease of gradient norm towards stationarity, according to his practical requirements. This special characteristic is very important since generally, in several practical problems, it is only necessary to obtain an approximation of the solution $x^* = A^{-1}b$ with low precision.

The theoretical result stated in Corollary 1 roughly suggests that each member of the proposed family is as good as any other method of the family, since all the methods satisfy an analogous optimality condition. However, observe that Algorithm 1 with $\mu = 1$ (DWGM) has the advantage that it minimizes the gradient norm, which is precisely the usual stopping rule for iterative algorithms in the general nonlinear optimization field. This peculiarity can lead the DWGM method to achieve the solution in fewer iterations than the rest of the choices. On the other hand, the best method in terms of computational complexity is obtained when $\mu = 0$ (CG method), since CG is the method that requires to compute the fewest number of inner products per iteration. In this scenario, the selections $\mu \in (0, 1)$ in Algorithm 1 generate the worst methods, in terms of the amount of floating-point operations needed per iteration. Nevertheless, as shown in our preliminary numerical experiments, for some specific intermediate values of the parameter $\mu \in (0, 1)$, the corresponding generalized method is able to converge to stationary points faster than the CG and DWGM methods for the minimization of large-scale strictly convex quadratic problems with a dense or sparse Hessian matrix.

Finally, it remains to investigate topics concerning extensions of the proposed algorithm for general unconstrained optimization problems as well as box-constrained optimization problems. A possible extension can be derived by incorporating the new scheme within the framework of the trust-region methods (by following the ideas of Steihaug's method [30]), while another possible generalization is suggested by Remark 1, whose extension can be obtained by performing a couple of inexact line searches per iteration. For all these possible extensions, the connection with the Moreau envelope described in Remark 2 could be helpful. These ideas will be investigated and analyzed in future researches.

Acknowledgements The authors are very grateful to two anonymous referees whose constructive remarks have improved the quality of the paper. In particular, one referee suggested considering dense matrices in Section 5, and the connection to the Moreau envelope developed in Remark 2 was inspired by a comment from the other referee.

Funding The first author was financially supported by FGV (Fundação Getulio Vargas) through the excellence post–doctoral fellowship program. The second author was financially supported by FAPESP (Projects 2013/05475-7 and 2017/18308-2) and CNPq (Project 301888/2017-5). The third author was financially supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UIDB/MAT/00297/2020 (Centro de Matemática e Aplicações).

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Andreani, R., Raydan, M.: Properties of the delayed weighted gradient method. Comput. Optim. Appl. **78**, 167–180 (2021)
2. Barzilai, J., Borwein, J.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (1988)
3. Brezinski, C.: Hybrid methods for solving systems of equations. NATO ASI Ser. C Math. Phys. Sci.-Adv. Study Inst. **508**, 271–290 (1998)
4. Brezinski, C., Redivo-Zaglia, M.: Hybrid procedures for solving linear systems. Numer. Math. **67**, 1–19 (1994)
5. Burdakov, O., Dai, Y.-H., Huang, N.: Stabilized Barzilai-Borwein method. J. Comput. Math. **37**, 916–936 (2019)
6. Cauchy, A.: Méthode générale pour la résolution des systemes d'équations simultanées. Comptes Rendus Sci. Paris **25**, 536–538 (1847)
7. Dai, Y.-H., Fletcher, R.: On the asymptotic behaviour of some new gradient methods. Math. Program. Ser. A **13**, 541–559 (2005)
8. Dai, Y.-H., Huang, Y., Liu, X.: A family of spectral gradient methods for optimization. Comput. Optim. Appl. **74**, 43–65 (2019)
9. Davis, T.-A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**, 1–25 (2011)
10. De Asmundis, R., Di Serafino, D., Riccio, F., Toraldo, G.: On spectral properties of steepest descent methods. IMA J. Numer. Anal. **33**(4), 1416–1435 (2013)
11. De Asmundis, R., Di Serafino, D., Hager, W., Toraldo, G., Zhang, H.: An efficient gradient method using the Yuan steplength. Comput. Optim. Appl. **59**, 541–563 (2014)
12. Di Serafino, D., Ruggiero, V., Toraldo, G., Zanni, L.: On the steplength selection in gradient methods for unconstrained optimization. Appl. Math. Comput. **318**, 176–195 (2018)
13. Fletcher, R.: On the Barzilai-Borwein method. In: Qi, L., Teo, K., Yang, X. (eds.) Optimization and Control with Applications, vol. 96, pp. 235–256. Series in Applied Optimization, Kluwer (2005)
14. Fletcher, R.: A limited memory steepest descent method. Math. Program. Ser. A **135**, 513–436 (2012)
15. Frassoldati, G., Zanni, L., Zanghirati, G.: New adaptive stepsize selections in gradient methods. J. Ind. Manag. Optim. **4**(2), 299–312 (2008)
16. Friedlander, A., Martínez, J.M., Molina, B., Raydan, M.: Gradient method with retards and generalizations. SIAM J. Numer. Anal. **36**(1), 275–289 (1999)
17. Gonzaga, C., Schneider, R.M.: On the steepest descent algorithm for quadratic functions. Comput. Optim. Appl. **63**(2), 523–542 (2016)
18. Hestenes, M.R., Stiefel, E.: Method of conjugate gradient for solving linear system. J. Res. Nat. Bur. Stand. **49**, 409–436 (1952)
19. Huang, Y., Dai, Y.-H., Liu, X.-W., Zhang, H.: Gradient methods exploiting spectral properties. Optim. Meth. Soft. **35**(4), 681–705 (2020)

20. Lemaréchal, C., Sagastizábal, C.: Practical aspects of the Moreau–Yosida regularization: theoretical preliminaries. SIAM J. Optim. **7**(2), 367–385 (1997)
21. Liu, Z., Liu, H., Dong, X.: An efficient gradient method with approximate optimal stepsize for the strictly convex quadratic minimization problem. Optimization **67**(3), 427–440 (2018)
22. Luenberger, D. Introduction to Linear and Nonlinear Programming, 2nd ed. Addison Wesley, Amsterdam (1984)
23. Moreau, J.-J.: Propriétés des applications "prox". C. R. Acad. Sci. Paris **256**, 1069–1071 (1963)
24. Moreau, J.-J.: Proximité et dualité dans un espace Hilbertien. Bull. Soc. Math. France **22**, 5–35 (1965)
25. Nocedal, J., Sartenaer, A., Zhu, C.: On the behavior of the gradient norm in the steepest descent method. Comput. Optim. Appl. **22**, 5–35 (2002)
26. Nocedal, J., Wright, S. Numerical Optimization, 2nd ed. Springer, New York (2006)
27. Oviedo-Leon, H.F.: A delayed weighted gradient method for strictly convex quadratic minimization. Comput. Optim. Appl. **74**, 729–746 (2019)
28. Oviedo, H., Dalmau, O., Herrera, R.: A hybrid gradient method for strictly convex quadratic programming, to appear in Numerical Linear Algebra with Applications, 28(4), https://doi.org/10.1002/nla.2360 (2020)
29. Planiden, C., Wang, X.: Proximal Mappings and Moreau Envelopes of Single-Variable Convex Piecewise Cubic Functions and Multivariable Gauge Functions. In: Hosseini, S., Mordukhovich, B., Uschmajew, A. (eds.) Nonsmooth Optimization and Its Applications, International Series of Numerical Mathematics, vol. 170, pp. 89–130. Springer Nature, Birkhäuser (2019)
30. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal. **20**, 626–637 (1983)
31. Stella, L., Themelis, A., Patrinos, P.: Forward–backward quasi-Newton methods for nonsmooth optimization problems. Comput. Optim. Appl. **67**, 443–487 (2017)
32. Trefethen, L.N., Bau, D.: Numerical Linear Algebra. SIAM, Philadelphia (1997)
33. Yuan, Y.: A new stepsize for the steepest descent method. J. Comput. Math. **24**, 149–156 (2006)
34. Zhou, B., Gao, L., Dai, Y.-H.: Gradient methods with adaptive step-sizes. Comput. Optim. Appl. **35**, 69–86 (2006)