A Minimal-Gradient Subspace Method for Unconstrained Optimization

Oscar Dalmau[a,*], Hugo de la Cruz Cansino[b]

[a] *Center for Research in Mathematics (CIMAT), Jalisco S/N, Col. Valenciana, Guanajuato, 36023, Guanajuato, Mexico*
[b] *FGV EMAp, Escola de Matemática Aplicada, Fundação Getulio Vargas, Praia de Botafogo, 190, Rio de Janeiro, 22250-900, Rio de Janeiro, Brazil*

Abstract

We propose a minimal-gradient subspace method for unconstrained optimization. For strictly convex quadratics, conjugate gradient can be interpreted as exact minimization over a two-dimensional affine subspace. We use the same reduced subspace in the nonlinear case, but compute a trial step by minimizing a local model of the next gradient norm. For SPD quadratics, every such subspace containing the gradient yields a uniform contraction of the gradient norm. In the nonlinear case, the trial step minimizes the norm of a linearized gradient model built from a Hessian approximation, but need not decrease the true objective. This step is accepted only when it satisfies a sufficient-decrease test; otherwise, the method uses a fallback along the negative-gradient direction. Experiments on synthetic SPD quadratics, real SPD matrices, and smooth PyCUTEst problems show a cost–robustness trade-off. The two- and three-dimensional quadratic variants have essentially the same iteration behavior. On PyCUTEst, the proposed method attains the highest success count, whereas L-BFGS requires fewer median gradient evaluations and less CPU time.

Keywords: nonlinear optimization, conjugate gradient, minimal gradient, subspace methods, global convergence, quadratic optimization

1. Introduction

The conjugate-gradient method is one of the main algorithms for minimizing strictly convex quadratic functions and for solving symmetric positive definite linear systems [1, 2]. In its classical presentation, the method is described through mutually conjugate directions and exact line searches. Conjugate gradient can also be presented from a variational viewpoint, i.e., the next iterate can be obtained by minimizing the quadratic function over a small affine subspace generated by the current gradient and the previous displacement [3]. The present work uses this subspace interpretation as the starting point to propose a method for nonlinear optimization.

*Corresponding author.
Email addresses: dalmau@cimat.mx (Oscar Dalmau), hugo.delacruz@fgv.br (Hugo de la Cruz Cansino)

The variational formulation is useful because it separates the role of the search space from the particular criterion used to choose the step. Classical steepest descent with exact line search may be viewed as function minimization over the one-dimensional gradient space [4]. The minimal-gradient stepsize instead chooses the step by minimizing the norm of the gradient along the same line [5]. In this paper, we combine these two ideas: we keep the short-memory subspace structure suggested by conjugate gradient, but select the trial step by a minimal-gradient criterion.

The methods considered here belong to the broader class of first-order methods that exploit short memory and low-dimensional subspaces. Relevant references include nonlinear conjugate-gradient methods and their globally convergent variants [6, 7, 8, 9, 10, 11, 12], as well as limited-memory and subspace methods [13, 14, 15]. Unlike these approaches, the method considered here does not obtain its trial direction from a nonlinear conjugate-gradient recurrence or by applying a quasi-Newton inverse directly to the gradient. It computes a small-subspace step that locally reduces the gradient norm and enforces descent through a safeguard.

The main subspace is generated by the current gradient and the previous displacement, as in the variational interpretation of conjugate gradient on SPD quadratics. We also consider an enriched space containing the previous gradient difference. In the quadratic case, this additional direction carries fixed-curvature information associated with the previous displacement and provides a direct test of whether the reduced space already captures the relevant behavior.

For function minimization on SPD quadratics, the enriched subspace does not change the conjugate-gradient step along CG trajectories: the reduced and enriched variants produce the same iterate as CG. This identity does not transfer directly to the minimal-gradient criterion, whose optimality condition is different. The contrast motivates the comparison between fixed-curvature quadratic problems and nonlinear problems with variable curvature.

For nonlinear optimization, minimal-gradient subspace steps are not necessarily descent steps for the objective function. Therefore, to guarantee descent, we use a safeguarded model-based fallback framework. At each iteration, a reduced minimal-gradient trial step is computed from a linearized gradient model over the subspace generated by the current gradient and the previous displacement. This trial step is accepted only if it produces sufficient decrease in the objective. Otherwise, the method invokes a safeguarded model-based fallback along the negative-gradient direction.

The main contributions are the following.

(i) We use the variational interpretation of conjugate gradient to motivate a reduced memory space as the basic subspace for minimal-gradient steps.

(ii) We show that adding the gradient-difference direction is redundant for the function-minimizing subspace method on SPD quadratics, but that this redundancy does not carry over directly to the minimal-gradient criterion.

(iii) We safeguard the nonlinear minimal-gradient step with a safeguarded model-based fallback along the negative-gradient direction and prove global convergence under standard assumptions.

(iv) We test the resulting methods on synthetic SPD quadratics, real SPD matrices, and smooth PyCUTEst problems. On the nonlinear benchmark, the proposed method solves more instances than the tested alternatives, although L-BFGS remains more efficient in median gradient evaluations and CPU time.

The paper is organized as follows. Section 2 develops the quadratic subspace framework and its gradient-norm contraction property. Section 3 introduces the nonlinear minimal-gradient step and the safeguarded fallback framework. Section 4 proves global convergence of the safeguarded method. Section 5 reports the numerical experiments, and Section 6 presents the conclusions.

2. Quadratic subspace framework

The conjugate-gradient method, in its usual presentation, is motivated geometrically by constructing mutually conjugate directions. For the quadratic function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{b}^T\mathbf{x}, \qquad \mathbf{Q} = \mathbf{Q}^T \succ 0, \tag{1}$$

one computes directions of the form

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1},$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, satisfying

$$\mathbf{p}_i^T \mathbf{Q} \mathbf{p}_j = 0, \qquad i \neq j.$$

The iterates are updated by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \tag{2}$$

where α_k is computed by exact line search along $\mathbf{p}_k$ [2].

Conjugate gradient can also be obtained from minimization in a small affine subspace [3]. Let

$$\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}.$$

Then the conjugate-gradient iterates for (1) can be characterized by

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathbf{x}_k + \operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}\}}{\arg\min} f(\mathbf{x}). \tag{3}$$

Note that under the previous formulation, the algorithm does not explicitly impose $\mathbf{Q}$-conjugacy. It only minimizes the objective function in the affine subspace generated by the current gradient and the previous displacement. Orthogonality and conjugacy arise from the optimality conditions. This variational viewpoint identifies $\operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}\}$ as a structurally meaningful memory subspace.

The one-dimensional analogue is the classical steepest-descent method with exact step size [4],

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k^F \mathbf{g}_k, \tag{4}$$

where α_k^F is chosen by minimizing f along the line $\mathbf{x}(\alpha) = \mathbf{x}_k - \alpha \mathbf{g}_k$, namely

$$\alpha_k^F = \arg\min_{\alpha} f(\mathbf{x}_k - \alpha \mathbf{g}_k). \tag{5}$$

Equivalently,

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x} \in \mathbf{x}_k + \mathrm{span}\{\mathbf{g}_k\}} f(\mathbf{x}). \tag{6}$$

Another one-dimensional choice is the minimal-gradient stepsize, obtained by minimizing the gradient norm along the same line [5]:

$$\alpha_k^{MG} = \arg\min_{\alpha} \|\nabla f(\mathbf{x}_k - \alpha \mathbf{g}_k)\|. \tag{7}$$

Thus, the same one-dimensional search space may be combined with two different criteria: minimization of the objective function or minimization of the gradient norm.

The subspace formulation (3) can be generalized to higher-dimensional subspaces. Let $\mathbf{D}_k \in \mathbb{R}^{n \times m}$ be a subspace matrix and consider iterations of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{D}_k \mathbf{z}_k, \qquad \mathbf{z}_k \in \mathbb{R}^m. \tag{8}$$

The sign convention is chosen so that, when $\mathbf{D}_k = [\mathbf{g}_k]$ and $\mathbf{z}_k = \alpha_k > 0$, the step moves along the steepest-descent direction.

For a given subspace matrix $\mathbf{D}_k$, we distinguish between two reduced problems. The function-minimizing step is obtained from

$$\mathbf{z}_k^F \in \arg\min_{\mathbf{z} \in \mathbb{R}^m} f(\mathbf{x}_k - \mathbf{D}_k \mathbf{z}), \tag{9}$$

whereas the minimal-gradient step is obtained from

$$\mathbf{z}_k^{MG} \in \arg\min_{\mathbf{z} \in \mathbb{R}^m} \|\nabla f(\mathbf{x}_k - \mathbf{D}_k \mathbf{z})\|. \tag{10}$$

The corresponding iterates are

$$\mathbf{x}_{k+1}^F = \mathbf{x}_k - \mathbf{D}_k \mathbf{z}_k^F, \qquad \mathbf{x}_{k+1}^{MG} = \mathbf{x}_k - \mathbf{D}_k \mathbf{z}_k^{MG}.$$

The subspace considered in this work is the reduced memory space generated by the current gradient and the previous displacement. We can naturally enrich this gs subspace using the gradient difference

$$\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}.$$

For the quadratic function (1), this vector satisfies

$$\mathbf{y}_{k-1} = \mathbf{Q}\mathbf{s}_{k-1},$$

so it carries curvature information associated with the previous displacement.

We use the notation

$$\mathbf{D}_k^g = [\mathbf{g}_k], \qquad \mathbf{D}_k^{gs} = [\mathbf{g}_k, \mathbf{s}_{k-1}], \qquad \mathbf{D}_k^{gsy} = [\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}].$$

The corresponding function-minimizing variants are denoted by F_g, F_{gs}, and F_{gsy}, respectively. The corresponding minimal-gradient variants are denoted by MG_g, MG_{gs}, and MG_{gsy}, respectively. The order in the subscript indicates the columns used in the subspace; the mathematical object is the span of those columns.

2.1. Function-minimizing subspace steps

The function-minimization subproblem is

$$\mathbf{z}_k^F = \arg\min_{\mathbf{z}\in\mathbb{R}^m} f(\mathbf{x}_k - \mathbf{D}_k\mathbf{z}). \tag{11}$$

Since

$$\nabla f(\mathbf{x}_k - \mathbf{D}_k\mathbf{z}) = \mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z},$$

the first-order optimality condition for (11) is

$$\mathbf{D}_k^T\left(\mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z}_k^F\right) = 0.$$

Equivalently,

$$\mathbf{D}_k^T\mathbf{Q}\mathbf{D}_k\mathbf{z}_k^F = \mathbf{D}_k^T\mathbf{g}_k. \tag{12}$$

When $\mathbf{D}_k^T\mathbf{Q}\mathbf{D}_k$ is nonsingular,

$$\mathbf{z}_k^F = \left(\mathbf{D}_k^T\mathbf{Q}\mathbf{D}_k\right)^{-1}\mathbf{D}_k^T\mathbf{g}_k. \tag{13}$$

The new gradient satisfies

$$\mathbf{D}_k^T\mathbf{g}_{k+1} = 0. \tag{14}$$

Thus, the F-criterion makes the new gradient orthogonal to the subspace used in the step.

The one-dimensional case. If $\mathbf{D}_k = [\mathbf{g}_k]$, then (11) gives exact steepest descent [4]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\mathbf{g}_k, \qquad \alpha_k = \frac{\mathbf{g}_k^T\mathbf{g}_k}{\mathbf{g}_k^T\mathbf{Q}\mathbf{g}_k}.$$

We denote this method by F_g.

The classical gs subspace. Let

$$\mathbf{D}_k^{gs} = [\mathbf{g}_k, \mathbf{s}_{k-1}], \qquad \mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}.$$

The method

$$F_{gs}: \qquad \mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathbf{x}_k+\mathrm{span}\{\mathbf{g}_k,\mathbf{s}_{k-1}\}} f(\mathbf{x}) \tag{15}$$

is the two-dimensional subspace formulation highlighted in Bertsekas' exercise on conjugate gradient [3]. In exact arithmetic, for the strictly convex quadratic problem, F_{gs} generates the conjugate-gradient iterates [1, 2].

Equation (15) identifies $\mathrm{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}\}$ as a classical memory subspace, which we use below for minimal-gradient iterations.

The subspace formulation also makes clear why conjugacy appears. Let

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k.$$

From the optimality conditions of F_{gs},

$$\mathbf{g}_{k+1}^T\mathbf{g}_k = 0, \qquad \mathbf{g}_{k+1}^T\mathbf{s}_{k-1} = 0.$$

From the previous exact minimization step,

$$\mathbf{g}_k^T \mathbf{s}_{k-1} = 0.$$

Since

$$\mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{Q}\left(\mathbf{x}_{k+1} - \mathbf{x}_k\right) = \mathbf{Q}\mathbf{s}_k,$$

we obtain

$$\mathbf{s}_k^T \mathbf{Q}\mathbf{s}_{k-1} = \left(\mathbf{g}_{k+1} - \mathbf{g}_k\right)^T \mathbf{s}_{k-1} = \mathbf{g}_{k+1}^T \mathbf{s}_{k-1} - \mathbf{g}_k^T \mathbf{s}_{k-1} = 0.$$

Then, the consecutive displacements are $\mathbf{Q}$-conjugate:

$$\mathbf{s}_k^T \mathbf{Q}\mathbf{s}_{k-1} = 0.$$

Therefore, the subspace minimization yields conjugacy without imposing it explicitly.

Enriched gsy subspace. The enriched subspace considered here is

$$\operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}\},$$

where

$$\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}.$$

For the quadratic problem,

$$\mathbf{y}_{k-1} = \mathbf{Q}\left(\mathbf{x}_k - \mathbf{x}_{k-1}\right) = \mathbf{Q}\mathbf{s}_{k-1}.$$

Define

$$\mathbf{D}_k^{gsy} = \left[\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}\right].$$

The corresponding function-minimizing step is

$$F_{gsy}: \qquad \mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathbf{x}_k + \operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}\}}{\arg\min} f(\mathbf{x}). \tag{16}$$

Along the conjugate-gradient sequence, the additional direction is redundant for the F-criterion, as stated next.

Proposition 1. *Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{b}^T\mathbf{x}$, with $\mathbf{Q} = \mathbf{Q}^T \succ 0$. Suppose that the iterates are generated by F_{gs}, equivalently by conjugate gradient, in exact arithmetic and without breakdown. Then F_{gsy}, F_{gs}, and CG generate the same next iterate.*

Proof. Let $\mathbf{x}_{k+1}^{gs}$ be the point generated by F_{gs}, and let

$$\mathbf{g}_{k+1} = \mathbf{Q}\mathbf{x}_{k+1}^{gs} - \mathbf{b}.$$

By the optimality conditions for F_{gs},

$$\mathbf{g}_{k+1}^T \mathbf{g}_k = 0, \qquad \mathbf{g}_{k+1}^T \mathbf{s}_{k-1} = 0.$$

To show that the same point minimizes f over the enriched subspace, it remains to show that

$$\mathbf{g}_{k+1}^T \mathbf{y}_{k-1} = 0.$$

Using

$$\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1},$$

we get

$$\mathbf{g}_{k+1}^T \mathbf{y}_{k-1} = \mathbf{g}_{k+1}^T \mathbf{g}_k - \mathbf{g}_{k+1}^T \mathbf{g}_{k-1}.$$

Along the conjugate-gradient sequence, gradients are mutually orthogonal. Therefore,

$$\mathbf{g}_{k+1}^T \mathbf{g}_k = 0, \qquad \mathbf{g}_{k+1}^T \mathbf{g}_{k-1} = 0,$$

and hence

$$\mathbf{g}_{k+1}^T \mathbf{y}_{k-1} = 0.$$

Thus,

$$\mathbf{g}_{k+1} \perp \operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}\}.$$

This is the first-order optimality condition for minimizing the strictly convex quadratic f over the enriched affine subspace. Since $\mathbf{Q} \succ 0$, the minimizer over the subspace is unique. Hence the enriched subspace step coincides with the gs step, and both coincide with conjugate gradient. □

This proposition establishes a redundancy for function minimization, however, adding $\mathbf{y}_{k-1}$ changes the minimal-gradient reduced problem, so MG_{gs} and MG_{gsy} are treated as distinct methods below.

2.2. Quadratic minimal-gradient subspace steps

For a quadratic function, the minimal-gradient subproblem in the subspace $\mathbf{D}_k$ is

$$\mathbf{z}_k^{MG} = \arg\min_{\mathbf{z}\in\mathbb{R}^m} \|\mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z}\|. \tag{17}$$

The normal equations are

$$\mathbf{D}_k^T \mathbf{Q}^2 \mathbf{D}_k \mathbf{z}_k^{MG} = \mathbf{D}_k^T \mathbf{Q}\mathbf{g}_k.$$

When $\mathbf{D}_k^T \mathbf{Q}^2 \mathbf{D}_k$ is nonsingular,

$$\mathbf{z}_k^{MG} = \left(\mathbf{D}_k^T \mathbf{Q}^2 \mathbf{D}_k\right)^{-1} \mathbf{D}_k^T \mathbf{Q}\mathbf{g}_k. \tag{18}$$

The new gradient satisfies

$$\left(\mathbf{Q}\mathbf{D}_k\right)^T \mathbf{g}_{k+1} = 0. \tag{19}$$

Thus, the function-minimizing and minimal-gradient criteria impose different orthogonality conditions:

$$F: \qquad \mathbf{g}_{k+1} \perp \operatorname{range}\left(\mathbf{D}_k\right),$$

whereas

$$MG: \qquad \mathbf{g}_{k+1} \perp \operatorname{range}\left(\mathbf{Q}\mathbf{D}_k\right).$$

For a fixed subspace, the two steps optimize different quantities. The function-minimizing step satisfies

$$f(\mathbf{x}_k - \mathbf{D}_k\mathbf{z}_k^F) \le f(\mathbf{x}_k - \mathbf{D}_k\mathbf{z}_k^{MG}),$$

while the minimal-gradient step satisfies

$$\|\mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z}_k^{MG}\| \le \|\mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z}_k^F\|.$$

Neither criterion dominates the other in both objective value and gradient norm.

Reduced and enriched subspaces. The reduced quadratic minimal-gradient method uses

$$\mathbf{D}_k^{gs} = [\mathbf{g}_k, \mathbf{s}_{k-1}]$$

in (17). Thus,

$$\mathbf{z}_k^{MG,gs} = \arg\min_{\mathbf{z}\in\mathbb{R}^2} \|\mathbf{g}_k - \mathbf{Q}\,[\mathbf{g}_k, \mathbf{s}_{k-1}]\,\mathbf{z}\|\,, \qquad \mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{g}_k, \mathbf{s}_{k-1}]\,\mathbf{z}_k^{MG,gs}.$$

We call this method MG_{gs}.

The enriched quadratic method uses

$$\mathbf{D}_k^{gsy} = [\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}]\,,$$

and solves

$$\mathbf{z}_k^{MG,gsy} = \arg\min_{\mathbf{z}\in\mathbb{R}^3} \|\mathbf{g}_k - \mathbf{Q}\,[\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}]\,\mathbf{z}\|\,.$$

Because

$$\operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}\} \subseteq \operatorname{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}\},$$

the enriched step cannot give a larger locally minimized gradient norm at the same iterate:

$$\|\mathbf{g}_{k+1}^{MG,gsy}\| \le \|\mathbf{g}_{k+1}^{MG,gs}\|.$$

The following proposition extends the contraction argument of Oviedo et al. [16] from the specific three-term AMGM subspace to any subspace containing the current gradient, including the reduced MG_{gs} subspace.

Proposition 2. *Let f be given by (1), and suppose that $\mathbf{D}_k$ contains $\mathbf{g}_k$ as a column. If*

$$\mathbf{z}_k^{MG} \in \arg\min_{\mathbf{z}} \|\mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z}\|, \qquad \mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{D}_k\mathbf{z}_k^{MG},$$

then

$$\|\mathbf{g}_{k+1}\| \le q_{\mathbf{Q}}\|\mathbf{g}_k\|, \qquad q_{\mathbf{Q}} = \frac{\lambda_{\max}(\mathbf{Q}) - \lambda_{\min}(\mathbf{Q})}{\lambda_{\max}(\mathbf{Q}) + \lambda_{\min}(\mathbf{Q})} < 1. \tag{20}$$

Consequently, repeated quadratic minimal-gradient steps converge globally and R-linearly in the gradient norm. In particular, the bound applies to MG_g, MG_{gs}, and MG_{gsy}.

Proof. Since the subspace contains $\mathbf{g}_k$, its least-squares residual is no larger than the residual obtained by restricting the step to the gradient direction. Hence, with

$$\alpha_* = \frac{2}{\lambda_{\min}(\mathbf{Q}) + \lambda_{\max}(\mathbf{Q})},$$

we have

$$\|\mathbf{g}_{k+1}\| = \min_{\mathbf{z}} \|\mathbf{g}_k - \mathbf{Q}\mathbf{D}_k\mathbf{z}\| \le \|(\mathbf{I} - \alpha_*\mathbf{Q})\mathbf{g}_k\| \le \|\mathbf{I} - \alpha_*\mathbf{Q}\|_2\|\mathbf{g}_k\|.$$

Because $\mathbf{Q}$ is symmetric positive definite,

$$\|\mathbf{I} - \alpha_*\mathbf{Q}\|_2 = \max_{\lambda\in[\lambda_{\min}(\mathbf{Q}),\lambda_{\max}(\mathbf{Q})]} |1 - \alpha_*\lambda| = q_{\mathbf{Q}},$$

which proves (20). Iterating the bound gives

$$\|\mathbf{g}_k\| \le q_{\mathbf{Q}}^k\|\mathbf{g}_0\|, \qquad k \ge 0.$$

Since $q_{\mathbf{Q}} < 1$, it follows that $\|\mathbf{g}_k\| \to 0$. □

The contraction factor in Proposition 2 is a uniform baseline inherited from the gradient direction. The additional subspace directions can improve the realized reduction, but are not needed to guarantee it.

This local property must be balanced against cost and implementation. For MG_{gs}, the normal equations require

$$\mathbf{QD}_k^{gs} = [\mathbf{Qg}_k, \mathbf{Qs}_{k-1}] = [\mathbf{Qg}_k, \mathbf{y}_{k-1}],$$

so only the column $\mathbf{Qg}_k$ requires a new matrix-vector product if $\mathbf{y}_{k-1}$ is already available from the gradient difference. A direct implementation of MG_{gsy} forms

$$\mathbf{QD}_k^{gsy} = [\mathbf{Qg}_k, \mathbf{Qs}_{k-1}, \mathbf{Qy}_{k-1}] = [\mathbf{Qg}_k, \mathbf{y}_{k-1}, \mathbf{Qy}_{k-1}],$$

and therefore requires the additional image direction

$$\mathbf{Qy}_{k-1} = \mathbf{Q}^2\mathbf{s}_{k-1}.$$

This explains why the direct enriched implementation has a larger matrix-vector-product count.

However, in the quadratic setting, this extra matrixvector product can be avoided. If $\mathbf{w}_k = \mathbf{Qg}_k$, then

$$\mathbf{Qy}_{k-1} = \mathbf{Q}(\mathbf{g}_k - \mathbf{g}_{k-1}) = \mathbf{w}_k - \mathbf{w}_{k-1}.$$

The AMGM variant reported in the experiments uses this recurrence and an internal rank or conditioning rule that may reduce the working subspace from three dimensions to two or one. When no reduction is triggered, AMGM is algebraically equivalent in exact arithmetic to the full three-dimensional step, but its implementation differs from direct MG_{gsy}.

The distinction between F and MG is essential. The identity

$$F_{gsy} = F_{gs} = \mathrm{CG}$$

is exact along the conjugate-gradient sequence, but it is a statement about objective minimization. For the minimal-gradient criterion, adding $\mathbf{y}_{k-1}$ changes the reduced problem because the orthogonality condition involves $\mathbf{QD}_k$. The quadratic experiments in Sections 5.1 and 5.2 show that, on the tested SPD problems, MG_{gs} and MG_{gsy} have nearly identical iteration counts and final gradient reductions, while MG_{gs} uses substantially fewer matrix-vector products.

3. Nonlinear minimal-gradient subspace methods

We now move from quadratic functions to general smooth nonlinear optimization:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{21}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. We write

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k), \qquad \mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}, \qquad \mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}.$$

The nonlinear methods studied in this paper use one of the two small subspaces

$$\mathbf{D}_k^{gs} = [\mathbf{g}_k, \mathbf{s}_{k-1}], \qquad \mathbf{D}_k^{gsy} = [\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}]. \tag{22}$$

The first is the reduced subspace motivated by the classical conjugate-gradient characterization. The second is an enriched secant subspace that includes the most recent gradient difference [16].

In the quadratic case, $\mathbf{y}_{k-1} = \mathbf{Q}\mathbf{s}_{k-1}$ for a fixed matrix $\mathbf{Q}$. In the nonlinear case, by applying the fundamental theorem of calculus to $t \mapsto \nabla f(\mathbf{x}_{k-1} + t\mathbf{s}_{k-1})$, we obtain

$$\mathbf{y}_{k-1} = \int_0^1 \nabla^2 f(\mathbf{x}_{k-1} + t\mathbf{s}_{k-1})\, \mathbf{s}_{k-1}\, dt,$$

whenever f is twice continuously differentiable along the segment. Thus, $\mathbf{y}_{k-1}$ contains local secant information associated with variable curvature. This is the main reason why the enriched subspace can behave differently in nonlinear problems than in fixed-curvature SPD quadratics.

Exact nonlinear minimal-gradient step. For a chosen subspace matrix $\mathbf{D}_k \in \{\mathbf{D}_k^{gs}, \mathbf{D}_k^{gsy}\}$, the exact nonlinear minimal-gradient step is defined by

$$\mathbf{z}_k^{NMG} \in \arg\min_{\mathbf{z}\in\mathbb{R}^m} \|\nabla f(\mathbf{x}_k - \mathbf{D}_k\mathbf{z})\|, \tag{23}$$

where m is the number of columns of $\mathbf{D}_k$. The associated trial step is

$$\mathbf{d}_k^{NMG} = -\mathbf{D}_k\mathbf{z}_k^{NMG}. \tag{24}$$

Thus, the trial point is

$$\mathbf{x}_k + \mathbf{d}_k^{NMG} = \mathbf{x}_k - \mathbf{D}_k\mathbf{z}_k^{NMG}.$$

The step minimizes the nonlinear gradient norm over $\mathbf{x}_k + \operatorname{span}(\mathbf{D}_k)$. Unlike the quadratic case, (23) is generally a nonlinear least-squares problem in two or three variables. The implementation below instead uses a linearized gradient model.

Linearized nonlinear minimal-gradient step. A lower-cost alternative uses a symmetric Hessian approximation $\mathbf{B}_k$ and defines

$$\mathbf{W}_k = \mathbf{B}_k\mathbf{D}_k.$$

Since $\mathbf{B}_k$ approximates $\nabla^2 f(\mathbf{x}_k)$, the matrix $\mathbf{W}_k$ approximates $\nabla^2 f(\mathbf{x}_k)\mathbf{D}_k$. The resulting local linear model of the gradient is

$$\nabla f(\mathbf{x}_k - \mathbf{D}_k\mathbf{z}) \approx \mathbf{g}_k - \mathbf{W}_k\mathbf{z}.$$

The corresponding linearized minimal-gradient subproblem is

$$\mathbf{z}_k^{LMG} \in \arg\min_{\mathbf{z}\in\mathbb{R}^m} \|\mathbf{g}_k - \mathbf{W}_k\mathbf{z}\|. \tag{25}$$

When $\mathbf{W}_k^T\mathbf{W}_k$ is nonsingular, one may use

$$\mathbf{z}_k^{LMG} = \left(\mathbf{W}_k^T\mathbf{W}_k\right)^{-1}\mathbf{W}_k^T\mathbf{g}_k, \tag{26}$$

or otherwise solve the small least-squares problem with a rank-revealing procedure. The corresponding trial step is

$$\mathbf{d}_k^{LMG} = -\mathbf{D}_k\mathbf{z}_k^{LMG}.$$

With this definition, the step can be interpreted as the minimal-gradient step for the local quadratic model later used in the fallback, restricted to $\text{span}(\mathbf{D}_k)$. Thus, the quadratic minimal-gradient formulas of Section 2.2 are used locally, with $\mathbf{QD}_k$ replaced by the model matrix $\mathbf{W}_k$.

For $\mathbf{D}_k^{gs} = [\mathbf{g}_k, \mathbf{s}_{k-1}]$, the model matrix requires the actions

$$\mathbf{B}_k\mathbf{g}_k, \qquad \mathbf{B}_k\mathbf{s}_{k-1}.$$

Because $\mathbf{B}_k$ approximates the Hessian, the second action has the natural secant target

$$\mathbf{B}_k\mathbf{s}_{k-1} \approx \mathbf{y}_{k-1}.$$

These model actions may be obtained from exact Hessian-vector products, finite differences of gradients, a quasi-Newton approximation, or a limited-memory secant model [17, 18, 2].

For $\mathbf{D}_k^{gsy}$, the model also requires the action $\mathbf{B}_k\mathbf{y}_{k-1}$. This additional column may provide curvature information beyond the reduced subspace, but it also increases the cost of the trial step.

A minimal-gradient step reduces a model of the gradient norm, not necessarily the objective function. A computed trial step

$$\mathbf{d}_k^{trial} = -\mathbf{D}_k\mathbf{z}_k^{trial}$$

may fail to satisfy

$$\mathbf{g}_k^T\mathbf{d}_k^{trial} < 0.$$

It may therefore fail to be a descent step. We accept it only when it satisfies the actual-decrease condition

$$f\left(\mathbf{x}_k + \mathbf{d}_k^{trial}\right) \leq f(\mathbf{x}_k) - \eta\|\mathbf{g}_k\|^2. \tag{27}$$

If the test holds, the next iterate is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k^{trial}. \tag{28}$$

If the test fails, the method invokes a safeguarded model-based fallback along the negative-gradient direction. The underlying model candidate is computed from the local quadratic model

$$m_k(\mathbf{d}) = f(\mathbf{x}_k) + \mathbf{g}_k^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T\mathbf{B}_k\mathbf{d},$$

where $\mathbf{B}_k$ is the curvature matrix used to build the local gradient model. The quadratic-model candidate is defined by

$$\widehat{\alpha}_k^F \in \arg\min_{\alpha\geq 0} m_k\left(-\alpha\mathbf{g}_k\right), \tag{29}$$

whenever this one-dimensional model minimization is well defined. The safeguarded fallback is required to return an accepted point $\mathbf{x}_{k+1}$ satisfying

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \eta_F \|\mathbf{g}_k\|^2, \qquad \eta_F > 0. \tag{30}$$

Here, η_F is independent of k. We denote this safeguarded model-based fallback by F_g^{model}. The main nonlinear method studied in the experiments is therefore

$$MG_{gs} \to F_g^{model},$$

where the trial step is the linearized minimal-gradient step in $\text{span}\{\mathbf{g}_k, \mathbf{s}_{k-1}\}$. If that trial step is rejected, the safeguarded model-based fallback is invoked. Details of the particular safeguard implementation used in the numerical experiments are provided in Section 5.

Algorithm 1 Nonlinear MG subspace method

Require: Initial point $\mathbf{x}_0$, subspace type $\mathcal{S} \in \{gs, gsy\}$, trial sufficient-decrease parameter $\eta > 0$, and fallback decrease parameter $\eta_F > 0$.

Compute $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$.

Choose an initial stepsize $\alpha_0 > 0$ and set

$$\mathbf{x}_1 = \mathbf{x}_0 - \alpha_0 \mathbf{g}_0.$$

for $k = 1, 2, \ldots$ **do**

Compute

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k), \qquad \mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}, \qquad \mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}.$$

Form the selected subspace matrix

$$\mathbf{D}_k = \begin{cases} [\mathbf{g}_k, \mathbf{s}_{k-1}], & \mathcal{S} = gs, \\ [\mathbf{g}_k, \mathbf{s}_{k-1}, \mathbf{y}_{k-1}], & \mathcal{S} = gsy. \end{cases}$$

Build the local model matrix $\mathbf{B}_k$, set $\mathbf{W}_k = \mathbf{B}_k \mathbf{D}_k$, and compute the linearized minimal-gradient trial step

$$\mathbf{z}_k^{LMG} \in \arg\min_{\mathbf{z} \in \mathbb{R}^m} \|\mathbf{g}_k - \mathbf{W}_k \mathbf{z}\|, \qquad \mathbf{d}_k^{trial} = -\mathbf{D}_k \mathbf{z}_k^{LMG}.$$

if $\mathbf{d}_k^{trial}$ satisfies (27) **then**

Accept the trial step and set

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k^{trial}.$$

else

Compute the safeguarded model-based fallback F_g^{model} along $-\mathbf{g}_k$, returning an accepted point $\mathbf{x}_{k+1}$ satisfying (30).

end if

end for

4. Global convergence

The convergence argument depends only on the decrease achieved by the accepted step [3, 2]. Accepted minimal-gradient trial steps satisfy (27), whereas the safeguarded model-based fallback satisfies (30). The argument is therefore independent of the particular mechanism used to enforce the safeguard.

Assumption 1. *The function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and bounded below.*

Lemma 1. *Suppose that ∇f is L-Lipschitz continuous on an open set $\mathcal{U}$. Then, for any points $\mathbf{x}, \mathbf{y} \in \mathcal{U}$ whose connecting segment is contained in $\mathcal{U}$,*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2.$$

The acceptance condition (27) is formulated directly in terms of a sufficient decrease in the objective function, rather than through conditions on the direction and the step size. For comparison, more classical angle and size conditions would also guarantee (27). In particular, suppose that

$$\mathbf{g}_k^T \mathbf{d}_k^{\text{trial}} \leq -\sigma \|\mathbf{g}_k\| \, \|\mathbf{d}_k^{\text{trial}}\|, \qquad m\|\mathbf{g}_k\| \leq \|\mathbf{d}_k^{\text{trial}}\| \leq M\|\mathbf{g}_k\|.$$

Assume also that the segment joining $\mathbf{x}_k$ and $\mathbf{x}_k + \mathbf{d}_k^{\text{trial}}$ is contained in $\mathcal{U}$. Then, by Lemma 1,

$$f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{d}_k^{\text{trial}}) \geq \left(\sigma m - \frac{L}{2}M^2\right) \|\mathbf{g}_k\|^2.$$

Hence, (27) is guaranteed whenever

$$\sigma m - \frac{L}{2}M^2 \geq \eta > 0.$$

Theorem 1. *Suppose Assumption 1 holds and the safeguarded model-based fallback produces an accepted point satisfying (30), with $\eta_F > 0$ independent of k. Then any sequence generated by the safeguarded nonlinear minimal-gradient subspace method satisfies*

$$\lim_{k \to \infty} \|\nabla f(\mathbf{x}_k)\| = 0.$$

Proof. At every iteration, either the minimal-gradient trial step is accepted through (27), or the safeguarded model-based fallback is accepted through (30).

In the first case,

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \eta \, \|\mathbf{g}_k\|^2.$$

In the second case,

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \eta_F \, \|\mathbf{g}_k\|^2.$$

Therefore, with

$$\gamma = \min\{\eta, \eta_F\} > 0,$$

we have, for every $k \geq 1$,

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \gamma \, \|\mathbf{g}_k\|^2.$$

Since f is bounded below,

$$\sum_{k=1}^{\infty} \|\mathbf{g}_k\|^2 \leq \frac{f(\mathbf{x}_1) - \inf f}{\gamma} < \infty.$$

Hence,

$$\|\mathbf{g}_k\| \to 0,$$

which proves the result. □

5. Numerical experiments

This section reports three experiments. The first two use the SPD quadratic model introduced in Section 2. In both cases, the right-hand side is generated as

$$\mathbf{b} = \mathbf{Q}\mathbf{x}^*,$$

so that the exact solution $\mathbf{x}^*$ and the optimal value $f^* = f(\mathbf{x}^*)$ are known. For these quadratic problems we report the relative gradient norm and the relative objective error

$$E_g(k) = \frac{\|\mathbf{g}_k\|}{\|\mathbf{g}_0\|}, \qquad E_f(k) = \frac{f(\mathbf{x}_k) - f^*}{\max\{1, |f(\mathbf{x}_0) - f^*|\}}.$$

The stopping criteria for the quadratic experiments are

$$\frac{\|\mathbf{g}_k\|}{\|\mathbf{g}_0\|} \leq \tau_g,$$

with $\tau_g = 10^{-8}$ and the maximum number of iterations, which is set to 5000.

The third experiment uses smooth unconstrained PyCUTEst problems. In this experiment we also include the safeguarded nonlinear implementation described in Sections 3 and 4. For the nonlinear problems, the stopping criterion is based on the relative gradient norm, with the tolerance and evaluation limits specified below.

The reported quantities emphasize both performance and cost: iteration counts, CPU time and the final relative gradient norm. Performance profiles summarize aggregate behavior [19].

5.1. Synthetic SPD quadratic experiments

We first test the quadratic framework on synthetic SPD instances. For each dimension n and condition number κ, let

$$t_i = \frac{i-1}{n-1}, \qquad i = 1, \ldots, n.$$

The linear, geometric, and exponentially biased spectra are defined by

$$\lambda_i^{\text{lin}} = 1 + (\kappa - 1)t_i, \tag{31}$$

$$\lambda_i^{\text{geo}} = \kappa^{t_i}, \tag{32}$$

$$\lambda_i^{\text{exp}} = 1 + (\kappa - 1)\frac{\exp(at_i) - 1}{\exp(a) - 1}, \qquad a = 6. \tag{33}$$

For the clustered random spectrum, we use $m = 5$ geometrically spaced cluster centers

$$c_j = \kappa^{(j-1)/(m-1)}, \qquad j = 1, \ldots, m.$$

The eigenvalues are first generated as

$$\widetilde{\lambda}_i = c_{j_i} \exp(\sigma_\lambda \xi_i), \qquad \xi_i \sim \mathcal{N}(0,1), \qquad \sigma_\lambda = 0.15,$$

where the indices $j_i \in \{1, \ldots, m\}$ are assigned in a balanced manner and then randomly permuted. After sorting $\widetilde{\lambda}_{(1)} \le \cdots \le \widetilde{\lambda}_{(n)}$, the eigenvalues are rescaled as

$$\lambda_i = 1 + (\kappa - 1)\frac{\widetilde{\lambda}_{(i)} - \widetilde{\lambda}_{(1)}}{\widetilde{\lambda}_{(n)} - \widetilde{\lambda}_{(1)}}.$$

Thus, $\lambda_{\min} = 1$, $\lambda_{\max} = \kappa$, and the resulting matrix has condition number κ.

For each spectrum, an orthogonal matrix $\mathbf{U}$ is obtained from the QR factorization of a Gaussian random matrix, and the SPD matrix is formed as

$$\mathbf{Q} = \mathbf{U}\,\mathrm{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{U}^T.$$

The exact solution is generated randomly and the right-hand side is set to

$$\mathbf{b} = \mathbf{Q}\mathbf{x}^*.$$

The benchmark contains four spectra; three dimensions, $n \in \{100, 500, 1000\}$; three condition numbers, $\kappa \in \{10^3, 10^4, 10^5\}$; and 30 random seeds per configuration. This gives $4 \times 3 \times 3 \times 30 = 1080$ quadratic problems.

Table 1 shows the results. The one-dimensional methods F_g and MG_g are not competitive on this benchmark; they solve only 11.1% and 15.4% of the runs, respectively. On the other hand, all two-dimensional and three-dimensional subspace methods converge on all 1080 instances. As expected, F_{gs} reproduces the behavior of CG very closely: the median iteration counts are 218.5 for F_{gs} and 218.0 for CG. The enriched function-minimizing method F_{gsy} gives essentially the same iteration count, but its median number of matrix-vector products is about twice as large.

The main direct comparison is MG_{gs} versus MG_{gsy}. Both methods have a global median of 206 iterations and the same final relative gradient at the reported precision. Direct MG_{gsy}, however, requires a median of 412 matrix-vector products, compared with 207 for MG_{gs}. Table 2 shows the same pattern in every spectrum and condition-number group: the median iteration ratio is one, the matrix-vector-product ratio is approximately two, and the time ratio ranges from 1.43 to 1.47.

AMGM uses $\mathbf{Q}\mathbf{y}_{k-1} = \mathbf{Q}\mathbf{g}_k - \mathbf{Q}\mathbf{g}_{k-1}$ and therefore avoids the extra matrix-vector product of direct MG_{gsy}. It also applies an internal $3D \to 2D \to 1D$ rank/conditioning safeguard. Thus, its matrix-vector-product count is close to that of MG_{gs}, but it is not the same implementation as direct MG_{gsy}.

Figure 1 provides a comparison in terms of matrix-vector products and CPU time. In the matrix-vector-product profile, CG, F_{gs}, MG_{gs}, and AMGM form the leading group. The profiles of the direct enriched methods, F_{gsy} and MG_{gsy}, are shifted to the right because

Table 1: Global summary for the synthetic SPD quadratic experiment. Success is measured by $\|\mathbf{g}_k\|/\|\mathbf{g}_0\| \leq 10^{-8}$. Medians are over the 1080 runs of each method. AMGM denotes the recurrence-based adaptive enriched variant described in the text.

Method	Success (%)	Median iter.	Median matvecs	Median time (s)	Median rel. grad.
F_g	11.1	5000	5001	12.8	9.33×10^{-6}
MG_g	15.4	5000	5001	12.5	6.39×10^{-6}
CG	100.0	218.0	219.0	0.594	8.80×10^{-9}
F_{gs}	100.0	218.5	219.5	0.684	8.80×10^{-9}
F_{gsy}	100.0	219.0	438.0	1.04	8.87×10^{-9}
MG_{gs}	100.0	206.0	207.0	0.668	9.52×10^{-9}
MG_{gsy}	100.0	206.0	412.0	0.968	9.52×10^{-9}
AMGM	100.0	206.0	207.0	0.651	9.51×10^{-9}

Table 2: Direct comparison MG_{gsy}/MG_{gs} on the synthetic SPD problems. Ratios are medians over paired successful runs. Values above one favor the reduced MG_{gs} method.

Spectrum	κ	Paired runs	Iter. ratio	Matvec ratio	Time ratio
clustered_random	10^3	90	1	1.99	1.46
clustered_random	10^4	90	1	1.99	1.46
clustered_random	10^5	90	1	1.99	1.44
exp_biased	10^3	90	1	1.99	1.47
exp_biased	10^4	90	1	2.00	1.46
exp_biased	10^5	90	1	2.00	1.45
geometric	10^3	90	1	1.99	1.45
geometric	10^4	90	1	2.00	1.47
geometric	10^5	90	1	2.00	1.46
linear	10^3	90	1	1.98	1.43
linear	10^4	90	1	1.99	1.44
linear	10^5	90	1	1.99	1.44

computing the additional image $\mathbf{Q}\mathbf{y}_{k-1}$ increases the number of matrix-vector products. In particular, MG_{gsy} achieves the same success rate and essentially the same iteration behavior as MG_{gs}, but at a substantially larger matrix-vector cost. The CPU-time profile shows the same general pattern, although the separation is less pronounced. Finally, the profiles of F_g and MG_g remain below 0.2, consistently with their low success rates in Table 1.

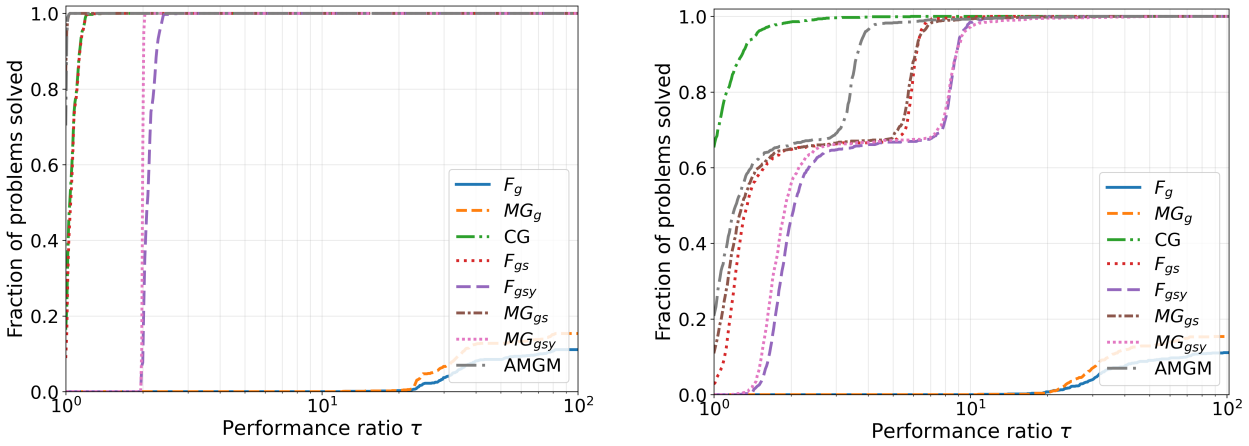


Figure 1: Performance profiles for the synthetic SPD experiment. The matrix-vector-product profile is the primary cost measure for the quadratic tests.

5.2. Real SPD quadratic experiments

For the second quadratic experiment, we use symmetric positive definite matrices selected from the SuiteSparse Matrix Collection [20]. We consider three dimension ranges:

$$B_{500} : 250 \le n \le 750, \qquad B_{1000} : 800 \le n \le 2000, \qquad B_{5000} : 3000 \le n \le 7000.$$

The matrices were required to be real, square, and symmetric positive definite. We sought 20 matrices per bucket and obtained 20 matrices in B_{500}, 19 in B_{1000}, and 20 in B_{5000}, for a total of 59 matrices. The complete list of matrices is provided with the experimental code.

For each matrix, we generate

$$\mathbf{x}^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

using seed zero and set

$$\mathbf{b} = \mathbf{Q}\mathbf{x}^*,$$

so that the exact minimizer is known. The initial point is $\mathbf{x}_0 = \mathbf{0}$. Table 3 summarizes the dimensions and sparsity levels of the selected matrices.

The real SPD benchmark is substantially harder than the synthetic one. Table 4 summarizes the results over the 59 matrix instances. CG solves 66.1% of the instances, and F_{gs} has the same success rate and a comparable median iteration count, consistently with the classical subspace interpretation. The direct F_{gsy} variant has a slightly lower success rate and larger median iteration and matrix-vector-product counts. Since the methods do not necessarily succeed on the same subsets, the effect of enriching the gs subspace is examined separately below.

The reduced MG_{gs} method has the highest success rate among the methods considered: 88.1%, compared with 86.4% for direct MG_{gsy}, 79.7% for AMGM, and 66.1% for CG. Its median iteration and matrix-vector-product counts are lower than those of CG, whereas its median CPU time is higher because of the subspace-algebra overhead in the current implementation. Since the methods succeed on different subsets, these global medians should not be interpreted as a direct problem-by-problem comparison.

Table 3: Summary of the real SPD matrix buckets.

Bucket	Matrices	$n_{\min}$	n_{med}	$n_{\max}$	$\text{nnz}_{\min}$	nnz_{med}	$\text{nnz}_{\max}$
B_{500}	20	289	444	726	1377	3966	34518
B_{1000}	19	817	1224	2000	4054	22189	63454
B_{5000}	20	3134	4752	5489	18316	138863	290378

Table 4: Summary of the real SPD quadratic experiment. Success is measured by $\|\mathbf{g}_k\|/\|\mathbf{g}_0\| \le 10^{-8}$. Success percentages are computed over the 59 matrix instances, and medians are computed from the available records. AMGM denotes the recurrence-based adaptive enriched variant described in the text.

Method	Success (%)	Median iter.	Median matvecs	Median time (s)
F_g	8.5	5000	5001	10.1
MG_g	8.5	5000	5001	22.0
CG	66.1	2051	2052	5.09
F_{gs}	66.1	2118	2119	10.8
F_{gsy}	64.4	2777	5554	19.7
MG_{gs}	88.1	1925	1926	12.8
MG_{gsy}	86.4	2087	4174	20.6
AMGM	79.7	2309	2310	8.82

AMGM uses the recurrence

$$\mathbf{Q}\mathbf{y}_{k-1} = \mathbf{Q}\mathbf{g}_k - \mathbf{Q}\mathbf{g}_{k-1}$$

and may reduce the working subspace. It is therefore reported as a recurrence-based adaptive variant rather than as direct MG_{gsy}.

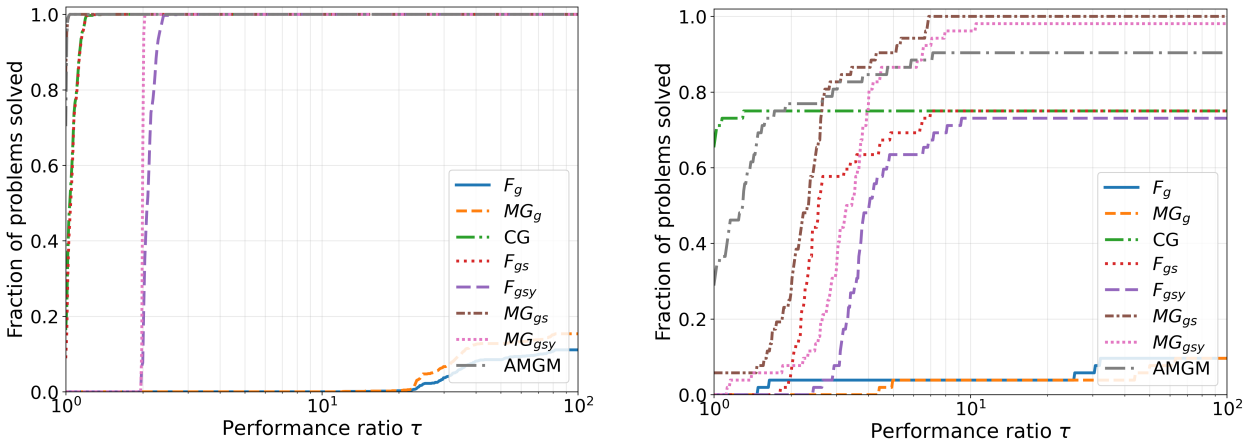
To isolate the effect of adding the gradient-difference direction, Table 5 compares the gsy and gs variants on the matrix instances for which both corresponding methods succeeded. For both the function-minimizing and minimal-gradient variants, the median iteration ratio is one at the reported precision in all three buckets. This does not contradict the different global medians in Table 4, since those medians are computed over different sets of runs. The matrix-vector-product ratio is approximately two, and the time ratio is also above one, although the time overhead is smaller for the largest matrices.

For the gsy variants, the median contribution of the $\mathbf{y}_{k-1}$ component is very low, i.e., of order 10^{-14}. Thus, on the instances solved by both methods, the additional direction produces no systematic iteration advantage. It increases the computational cost and does not improve the success rate relative to the corresponding gs variant.

Figure 2 complements the results in Tables 4 and 5. In the matrix-vector-product profile, MG_{gs} attains the largest overall coverage and remains competitive with the best methods. The direct enriched variants are shifted to the right, consistently with the additional matrix-vector product required at each iteration. The final height of each profile also reflects its success rate: MG_{gs}, MG_{gsy}, and AMGM solve a larger fraction of the matrices than CG and F_{gs}, whereas the one-dimensional methods F_g and MG_g solve only a small fraction. The CPU-time profile gives a less favorable picture for MG_{gs}: although it retains the highest success rate, CG and AMGM are more competitive in runtime. This confirms that the

Table 5: Direct comparison of the enriched and reduced subspaces on real SPD quadratics. "Common successes" denotes the number of matrices solved by both methods in the indicated comparison. For each such matrix, the ratios are computed as enriched/reduced; the table reports their medians.

Bucket	Methods compared	Common successes	Iter. ratio	Matvec ratio	Time ratio
B_{500}	F_{gsy}/F_{gs}	19	1	1.99	1.56
B_{500}	MG_{gsy}/MG_{gs}	19	1	2.00	1.44
B_{1000}	F_{gsy}/F_{gs}	13	1	2.00	1.46
B_{1000}	MG_{gsy}/MG_{gs}	19	1	2.00	1.58
B_{5000}	F_{gsy}/F_{gs}	6	1	2.01	1.23
B_{5000}	MG_{gsy}/MG_{gs}	13	1	2.00	1.13



matrix-vector products CPU time

Figure 2: Performance profiles for the real SPD experiment. The reduced MG_{gs} method is competitive in matrix-vector products, whereas CPU time still reflects implementation overhead.

main advantage of the reduced method in the present implementation lies in robustness and matrix-vector-product efficiency rather than CPU time.

5.3. Nonlinear PyCUTEst experiments

The nonlinear experiment evaluates the reduced minimal-gradient subspace method on unconstrained CUTEst problems accessed through PyCUTEst [21, 22]. In particular, we test

$$MG_{gs} \to F_g^{\text{model}},$$

using the globalization framework described in Section 3 and Algorithm 1. The linearized MG_{gs} trial step is first tested directly for sufficient decrease. If this test fails, the safeguarded model-based F_g fallback is used. In the numerical experiments, the negative-gradient safeguard candidate is computed from the one-dimensional quadratic model, using its exact minimizer. The candidate is accepted only if it satisfies the prescribed sufficient-decrease condition in the true objective. Armijo backtracking is retained as a final safety mechanism, but it was not invoked in the reported experiments.

Problems were selected before any optimization method was run. We queried the local PyCUTEst installation for unconstrained problems and retained instances having actual

Table 6: Dimension distribution of the stratified PyCUTEst benchmark. The strata were formed after sorting the validated unconstrained instances by their actual dimension. Because the strata are rank-based, adjacent strata may share boundary dimensions.

Stratum	Instances	$n_{\min}$	n_{med}	$n_{\max}$
S_1, small	30	22	82	102
S_2, medium-low	30	134	1000	3000
S_3, medium-high	30	3000	5000	5000
S_4, large	30	5000	10000	123200

dimension $n \geq 20$, no constraints, finite objective and gradient values at the initial point, and initial gradient norm larger than 10^{-12}. The validated instances were divided into four rank-based strata of approximately equal size. From each stratum, 30 instances were sampled uniformly using seed 20260625, giving a total of 120 problems. Table 6 summarizes their dimension distribution.

We compare the proposed method with standard first-order and quasi-Newton baselines. These include gradient descent with Armijo backtracking GD; the one-dimensional functional method F_g; $BB1$ with Armijo backtracking [23, 24]; Fletcher–Reeves nonlinear conjugate gradient $NCG\text{-}FR$ [6]; Polak–Ribière–Polyak nonlinear conjugate gradient with restart $NCG\text{-}PRP+$ [7]; and L-BFGS [17, 18, 2].

We also include the variants $MG_{gs} \to GD$ and $MG_{gsy} \to GD$. In these variants, a rejected minimal-gradient trial step is replaced by a gradient-descent direction, and Armijo backtracking is used to determine the step length.

Finally, we report the Armijo-globalized variant $MG_{gs} \to F_g^{\text{Armijo}}$. It constructs the same linearized MG_{gs} trial direction, but applies angle and size safeguards followed by Armijo backtracking. If the trial direction is rejected or the line search fails, the method invokes the functional F_g fallback. Thus, this variant differs from $MG_{gs} \to GD$, whose fallback is a gradient-descent step with Armijo backtracking.

All methods use the same stopping criterion,

$$\frac{\|\mathbf{g}_k\|}{\max\{1, \|\mathbf{g}_0\|\}} \leq 10^{-6},$$

with maximum budgets of 5000 iterations.

Table 7 summarizes the results. We include the numbers of gradient and function evaluations, denoted by N_g and N_f. The proposed $MG_{gs} \to F_g^{\text{model}}$ variant solves 91 of the 120 instances, the highest success count among the evaluated methods. It improves on the gradient-descent fallback $MG_{gs} \to GD$, which solves 84 instances, and on the Armijo-globalized functional-fallback variant $MG_{gs} \to F_g^{\text{Armijo}}$, which solves 89 instances. The two functional-fallback variants have 89 common successful instances. The model-fallback variant solves two additional instances, whereas the Armijo-globalized variant solves no additional instance.

L-BFGS is the most efficient method in terms of median gradient evaluations and CPU time. The F_g method requires substantially more function evaluations because of its one-dimensional searches. The safeguarded subspace variants require more gradient evaluations than L-BFGS, but attain higher success counts.

Table 7: Aggregate results on the 120 nonlinear PyCUTEst instances. Medians are computed over all instances. The proposed variant is $MG_{gs} \to F_g^{\text{model}}$.

Method	Success	Iter.	N_g	N_f	Time (s)	Rel. grad.
GD	55/120	804.0	805.0	4598.5	1.239	3.35×10^{-6}
F_g	59/120	312.0	313.0	7779.5	2.652	1.12×10^{-6}
$BB1$	80/120	174.0	175.0	378.5	0.257	9.63×10^{-7}
$NCG\text{-}FR$	62/120	170.5	171.5	1949.5	0.678	9.98×10^{-7}
$NCG\text{-}PRP+$	72/120	144.0	145.0	959.0	0.307	9.84×10^{-7}
L-BFGS	88/120	50.5	51.5	67.0	0.148	8.93×10^{-7}
$MG_{gs} \to GD$	84/120	57.0	114.0	82.5	0.154	9.79×10^{-7}
$MG_{gsy} \to GD$	81/120	50.0	149.0	84.0	0.244	9.64×10^{-7}
$MG_{gs} \to F_g^{\text{Armijo}}$	89/120	47.0	94.0	106.5	0.217	9.63×10^{-7}
$MG_{gs} \to F_g^{\text{model}}$	91/120	54.5	109.0	98.5	0.187	9.61×10^{-7}

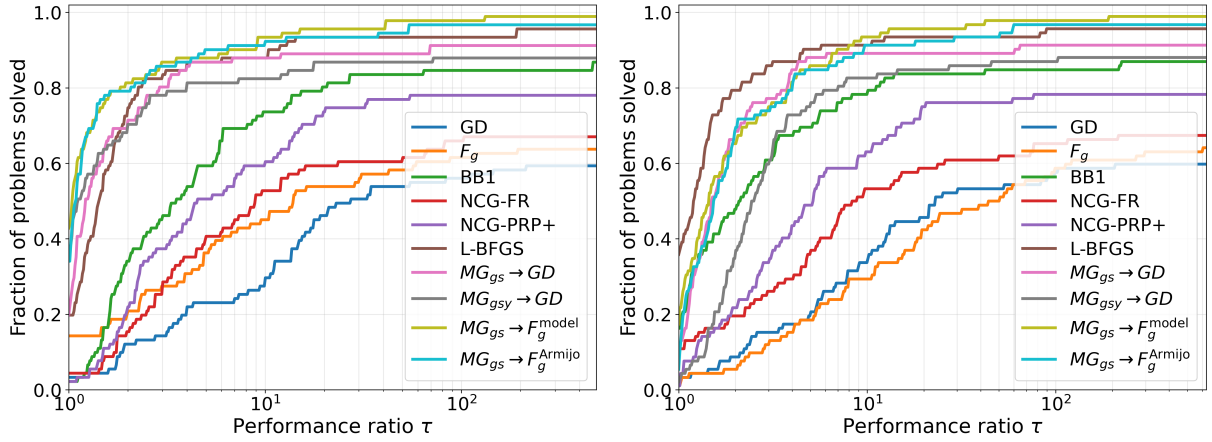


Figure 3: Performance profiles on the 120 nonlinear PyCUTEst instances, using iteration count and CPU time as performance measures.

Figure 3 illustrates the same trade-off. The performance profile for $MG_{gs} \to F_g^{\text{model}}$ reaches the largest final fraction of solved problems, whereas L-BFGS is the strongest method in terms of CPU time. The proposed method therefore provides greater robustness on this benchmark, although not the lowest median computational cost.

Table 8 reports the success counts by dimension stratum for the main methods. The model-fallback variant ties for the highest success count in the small, medium-low, and large strata, and attains the highest count in the medium-high stratum.

Overall, the safeguarded model-based fallback increases the success count of the reduced MG_{gs} method relative to both the gradient-descent fallback $MG_{gs} \to GD$ and the Armijo-fallback variant $MG_{gs} \to F_g^{\text{Armijo}}$. L-BFGS remains more efficient in median gradient evaluations and CPU time, whereas $MG_{gs} \to F_g^{\text{model}}$ provides the highest success count on the selected benchmark.

Table 8: Success counts by dimension stratum for the main nonlinear methods. Each stratum contains 30 PyCUTEst instances.

Method	S_1	S_2	S_3	S_4
L-BFGS	25/30	24/30	23/30	16/30
$MG_{gs} \to GD$	22/30	24/30	23/30	15/30
$MG_{gsy} \to GD$	18/30	22/30	25/30	16/30
$MG_{gs} \to F_g^{\text{Armijo}}$	25/30	24/30	24/30	16/30
$MG_{gs} \to F_g^{\text{model}}$	25/30	24/30	26/30	16/30

6. Conclusions

We studied safeguarded minimal-gradient methods using the subspace generated by the current gradient and the previous step. For function minimization on SPD quadratics, this subspace recovers conjugate gradient, and adding the gradient-difference direction does not change the iterate. For the minimal-gradient criterion, every subspace containing the current gradient inherits a uniform contraction of the gradient norm. The reduced and enriched variants show nearly identical iteration behavior on the tested SPD problems, but the direct enriched variant incurs a higher matrix-vector-product cost.

For nonlinear problems, the LMG least-squares step minimizes the norm of a linearized gradient model over the selected subspace. Because reduction in this model does not guarantee decrease in the true objective, a trial step is accepted only when it achieves sufficient actual decrease; otherwise, a safeguarded model-based fallback is used along the negative-gradient direction. The convergence analysis depends only on the decrease produced by the accepted steps. Under the stated assumptions, the proposed framework is globally convergent to first-order stationarity.

On the PyCUTEst benchmark, the reduced model-fallback variant achieves the highest success count among the tested methods, while L-BFGS remains more efficient in median gradient evaluations and CPU time. These results support the reduced subspace and indicate that, on the tested problems, the fallback strategy contributes more to robustness than adding the gradient-difference direction to the main subspace.

Acknowledgments

The authors thank Daylenis Dorta Enriquez and Jose Juan Quiroz Benitez for preliminary numerical explorations of related minimal-gradient subspaces.

References

[1] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49 (6) (1952) 409–436. doi:10.6028/jres.049.044.

[2] J. Nocedal, S. J. Wright, Numerical Optimization, 2nd Edition, Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, 2006. doi:10.1007/978-0-387-40065-5.

[3] D. P. Bertsekas, Nonlinear Programming, 2nd Edition, Athena Scientific, Belmont, MA, 1999.

[4] A.-L. Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences 25 (1847) 536–538.

[5] R. De Asmundis, D. di Serafino, W. W. Hager, G. Toraldo, H. Zhang, An efficient gradient method using the Yuan steplength, Computational Optimization and Applications 59 (3) (2014) 541–563. doi:10.1007/s10589-014-9669-5.

[6] R. Fletcher, C. M. Reeves, Function minimization by conjugate gradients, The Computer Journal 7 (2) (1964) 149–154. doi:10.1093/comjnl/7.2.149.

[7] E. Polak, G. Ribière, Note sur la convergence de méthodes de directions conjuguées, Revue française d'informatique et de recherche opérationnelle. Série rouge 3 (R1) (1969) 35–43.

[8] Y.-H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization 10 (1) (1999) 177–182. doi:10.1137/S1052623497318992.

[9] W. W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM Journal on Optimization 16 (1) (2005) 170–192. doi:10.1137/030601880.

[10] W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, Pacific Journal of Optimization 2 (1) (2006) 35–58.

[11] W. W. Hager, H. Zhang, Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent, ACM Transactions on Mathematical Software 32 (1) (2006) 113–137. doi:10.1145/1132973.1132979.

[12] S. Das Gupta, R. M. Freund, X. A. Sun, A. Taylor, Nonlinear conjugate gradient methods: Worst-case convergence rates via computer-assisted analyses, Mathematical Programming 213 (1–2) (2025) 1–49. doi:10.1007/s10107-024-02127-7.

[13] W. W. Hager, H. Zhang, The limited memory conjugate gradient method, SIAM Journal on Optimization 23 (4) (2013) 2150–2168. doi:10.1137/120898097.

[14] M. Li, H. Liu, Z. Liu, A new subspace minimization conjugate gradient method with nonmonotone line search for unconstrained optimization, Numerical Algorithms 79 (2018) 195–219. doi:10.1007/s11075-017-0434-6.

[15] Y. Yuan, A review on subspace methods for nonlinear optimization, in: Proceedings of the International Congress of Mathematicians, Seoul 2014, Vol. IV, Kyung Moon Sa Co., Ltd., Seoul, 2014, pp. 807–827.

[16] H. Oviedo, O. Dalmau, R. Herrera, An accelerated minimal gradient method with momentum for strictly convex quadratic optimization, BIT Numerical Mathematics 62 (2) (2022) 591–606. doi:10.1007/s10543-021-00886-9.

[17] J. Nocedal, Updating Quasi-Newton matrices with limited storage, Mathematics of Computation 35 (151) (1980) 773–782. doi:10.1090/S0025-5718-1980-0572855-7.

[18] D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Mathematical Programming 45 (1989) 503–528. doi:10.1007/BF01589116.

[19] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2) (2002) 201–213. doi:10.1007/s101070100263.

[20] T. A. Davis, Y. Hu, The University of Florida sparse matrix collection, ACM Transactions on Mathematical Software 38 (1) (2011) 1:1–1:25. doi:10.1145/2049662.2049663.

[21] N. I. M. Gould, D. Orban, P. L. Toint, CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization, Computational Optimization and Applications 60 (3) (2015) 545–557. doi:10.1007/s10589-014-9687-3.

[22] J. Fowkes, L. Roberts, Á. Bűrmen, PyCUTEst: An open source Python package of optimization test problems, Journal of Open Source Software 7 (78) (2022) 4377. doi:10.21105/joss.04377.

[23] J. Barzilai, J. M. Borwein, Two-point step size gradient methods, IMA Journal of Numerical Analysis 8 (1) (1988) 141–148. doi:10.1093/imanum/8.1.141.

[24] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM Journal on Optimization 7 (1) (1997) 26–33. doi:10.1137/S1052623494266365.