A Majorize-Minimize strategy for subspace optimization applied to image restoration

Emilie Chouzenoux, Jérôme Idier, Saïd Moussaoui

▶ **To cite this version:**

Emilie Chouzenoux, Jérôme Idier, Saïd Moussaoui. A Majorize-Minimize strategy for subspace optimization applied to image restoration. 2010. hal-00516585

HAL Id: hal-00516585
https://hal.science/hal-00516585v1

Preprint submitted on 10 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Majorize-Minimize Strategy for Subspace Optimization Applied to Image Restoration

Emilie Chouzenoux, Jérôme Idier and Saïd Moussaoui

Abstract

This paper proposes accelerated subspace optimization methods in the context of image restoration. Subspace optimization methods belong to the class of iterative descent algorithms for unconstrained optimization. At each iteration of such methods, a stepsize vector allowing the best combination of several search directions is computed through a multi-dimensional search. It is usually obtained by an inner iterative second-order method ruled by a stopping criterion that guarantees the convergence of the outer algorithm. As an alternative, we propose an original multi-dimensional search strategy based on the majorize-minimize principle. It leads to a closed-form stepsize formula that ensures the convergence of the subspace algorithm whatever the number of inner iterations. The practical efficiency of the proposed scheme is illustrated in the context of edge-preserving image restoration.

Index Terms

Subspace optimization, memory gradient, conjugate gradient, quadratic majorization, stepsize strategy, image restoration.

I. Introduction

This work addresses a wide class of problems where an input image $\boldsymbol{x}^o \in \mathbb{R}^N$ is estimated from degraded data $\boldsymbol{y} \in \mathbb{R}^T$. A typical model of image degradation is

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}^o + \boldsymbol{\epsilon}$$

where $\boldsymbol{H}$ is a linear operator, described as a $T \times N$ matrix, that models the image degradation process, and $\boldsymbol{\epsilon}$ is an additive noise vector. This simple formalism covers many real situations such as deblurring, denoising, inverse-Radon transform in tomography and signal interpolation.

E. Chouzenoux, J. Idier and S. Moussaoui are with IRCCyN (CNRS UMR 6597), Ecole Centrale Nantes, France. E-mail: {emilie.chouzenoux, jerome.idier, said.moussaoui}@irccyn.ec-nantes.fr.

Two main strategies emerge in the literature for the restoration of $\boldsymbol{x}^o$ [1]. The first one uses an *analysis-based* approach, solving the following problem [2, 3]:

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \left(F(\boldsymbol{x}) = \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|^2 + \lambda\Psi(\boldsymbol{x})\right). \tag{1}$$

In section V, we will consider an image deconvolution problem that calls for the minimization of this criterion form.

The second one employs a *synthesis-based* approach, looking for a decomposition $\boldsymbol{z}$ of the image in some dictionary $\boldsymbol{K} \in \mathbb{R}^{T\times R}$ [4, 5]:

$$\min_{\boldsymbol{z}\in\mathbb{R}^R} \left(F(\boldsymbol{z}) = \|\boldsymbol{H}\boldsymbol{K}\boldsymbol{z} - \boldsymbol{y}\|^2 + \lambda\Psi(\boldsymbol{z})\right). \tag{2}$$

This method is applied to a set of image reconstruction problems [6] in section IV.

In both cases, the penalization term Ψ, whose weight is set through the regularization parameter λ, aims at guaranteeing the robustness of the solution to the observation noise and at favorizing its fidelity to *a priori* assumptions [7].

From the mathematical point a view, problems (1) and (2) share a common structure. In this paper, we will focus on the resolution of the first problem (1), but we will also provide numerical results regarding the second one. On the other hand, we restrict ourselves to regularization terms of the form

$$\Psi(\boldsymbol{x}) = \sum_{c=1}^{C} \psi(\|\boldsymbol{V}_c\boldsymbol{x} - \boldsymbol{\omega}_c\|)$$

where $\boldsymbol{V}_c \in \mathbb{R}^{P\times N}$, $\boldsymbol{\omega}_c \in \mathbb{R}^P$ for $c = 1, ..., C$ and $\|.\|$ stands for the Euclidian norm. In the analysis-based approach, $\boldsymbol{V}_c$ is typically a linear operator yielding either the differences between neighboring pixels (*e.g.,* in the Markovian regularization approach), or the local spatial gradient vector (*e.g.,* in the total variation framework), or wavelet decomposition coefficients in some recent works such as [1]. In the synthesis-based approach, $\boldsymbol{V}_c$ usually identifies with the identity matrix.

The strategy used for solving the penalized least squares (PLS) optimization problem (1) strongly depends on the objective function properties (differentiability, convexity). Moreover, these mathematical properties contribute to the quality of the reconstructed image. In that respect, we particularly focus on differentiable, coercive, edge-preserving functions ψ, *e.g.,* ℓ_p norm with $1 < p < 2$, Huber, hyperbolic, or Geman and McClure functions [8–10], since they give rise to locally smooth images [11–13]. In contrast, some restoration methods rely on non differentiable regularizing functions to introduce priors such as sparsity of the decomposition coefficients [5] and piecewise constant patterns in the images [14]. As emphasized in [6], the non differentiable penalization term can be replaced by a smoothed version

without altering the reconstruction quality. Moreover, the use of a smoother penalty can reduce the staircase effect that appears in the case of total variation regularization [15].

In the case of large scale non linear optimization problems as encountered in image restoration, direct resolution is impossible. Instead, iterative optimization algorithms are used to solve (1). Starting from an initial guess $\boldsymbol{x}_0$, they generate a sequence of updated estimates $(\boldsymbol{x}_k)$ until sufficient accuracy is obtained. A fundamental update strategy is to produce a decrease of the objective function at each iteration: from the current value $\boldsymbol{x}_k$, $\boldsymbol{x}_{k+1}$ is obtained according to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k, \tag{3}$$

where $\alpha_k > 0$ is the *stepsize* and $\boldsymbol{d}_k$ is a *descent direction i.e.,* a vector such that $\boldsymbol{g}_k^T \boldsymbol{d}_k < 0$, where $\boldsymbol{g}_k = \nabla F(\boldsymbol{x}_k)$ denotes the gradient of F at $\boldsymbol{x}_k$. The determination of α_k is called the *line search*. It is usually obtained by partially minimizing the scalar function $f(\alpha) = F(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$ until the fulfillment of some sufficient conditions related to the overall algorithm convergence [16].

In the context of the minimization of PLS criteria, the determination of the descent direction $\boldsymbol{d}_k$ is customarily addressed using a half-quadratic (HQ) approach that exploits the PLS structure [11, 12, 17, 18]. A constant stepsize is then used while $\boldsymbol{d}_k$ results from the minimization of a quadratic majorizing approximation of the criterion [13], either resulting from Geman and Reynolds (GR) or from Geman and Yang (GY) constructions [2, 3].

Another effective approach for solving (1) is to consider subspace acceleration [6, 19]. As emphasized in [20], some descent algorithms (3) have a specific subspace feature: they produce search directions spanned in a low dimension subspace. For example,

- the nonlinear conjugate gradient (NLCG) method [21] uses a search direction in a two-dimensional (2D) space spanned by the opposite gradient and the previous direction.
- the L-BFGS quasi-Newton method [22] generates updates in a subspace of size $2m + 1$, where m is the limited memory parameter.

Subspace acceleration consists in relying on iterations more explicitly aimed at solving the optimization problem within such low dimension subspaces [23–27]. The acceleration is obtained by defining $\boldsymbol{x}_{k+1}$ as the approximate minimizer of the criterion over the subspace spanned by a set of M directions

$$\boldsymbol{D}_k = [\boldsymbol{d}_k^1, \ldots, \boldsymbol{d}_k^M]$$

with $1 \leq M \ll N$. More precisely, the iterates are given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{s}_k \tag{4}$$

where $\boldsymbol{s}_k$ is a multi-dimensional stepsize that aims at partially minimizing

$$f(\boldsymbol{s}) = F(\boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{s}). \tag{5}$$

The prototype scheme (4) defines an *iterative subspace optimization* algorithm that can be viewed as an extension of (3) to a search subspace of dimension larger than one. The subspace algorithm has been shown to outperforms standard descent algorithms, such as NLCG and L-BFGS, in terms of computational cost and iteration number before convergence, over a set of PLS minimization problems [6, 19].

The implementation of subspace algorithms requires a strategy to determine the stepsize $\boldsymbol{s}_k$ that guarantees the convergence of the recurrence (4). However, it is difficult to design a practical multi-dimensional stepsize search algorithm gathering suitable convergence properties and low computational time [26, 28]. Recently, GY and GR HQ approximations have led to an efficient majorization-minimization (MM) line search strategy for the computation of α_k when $\boldsymbol{d}_k$ is the NLCG direction [29] (see also [30] for a general reference on MM algorithms). In this paper, we generalize this strategy to define the multi-dimensional stepsize $\boldsymbol{s}_k$ in (4). We prove the mathematical convergence of the resulting subspace algorithm under mild conditions on $\boldsymbol{D}_k$. We illustrate its efficiency on four image restoration problems.

The rest of the paper is organized as follows: Section II gives an overview of existing subspace constructions and multi-dimensional search procedures. In Section III, we introduce the proposed HQ/MM strategy for the stepsize calculation and we establish general convergence properties for the overall subspace algorithm. Finally, Sections IV and V give some illustrations and a discussion of the algorithm performances by means of a set of experiments in image restoration.

II. Subspace optimization methods

The first subspace optimization algorithm is the memory gradient method, proposed in the late 1960's by Miele and Cantrell [23]. It corresponds to

$$\boldsymbol{D}_k = [-\boldsymbol{g}_k, \boldsymbol{d}_{k-1}]$$

and the stepsize $\boldsymbol{s}_k$ results from the exact minimization of $f(\boldsymbol{s})$. When F is quadratic, it is equivalent to the nonlinear conjugate gradient algorithm [31].

More recently, several other subspace algorithms have been proposed. Some of them are briefly reviewed in this section. We first focus on the subspace construction, and then we describe several existing stepsize strategies.

A. Subspace construction

Choosing subspaces $\boldsymbol{D}_k$ of dimensions larger than one may allow faster convergence in terms of iteration number. However, it requires a multi-dimensional stepsize strategy, which can be substantially more complex (and computationaly costly) than the usual line search. Therefore, the choice of the subspace must achieve a tradeoff between the iteration number to reach convergence and the cost per iteration. Let us review some existing iterative subspace optimization algorithms and their associated set of directions. For the sake of compactness, their main features are summarized in Tab. I. Two families of algorithms are distinguished.

1) Memory gradient algorithms: In the first seven algorithms, $\boldsymbol{D}_k$ mainly gathers successive gradient and direction vectors.

The third one, introduced in [32] as supermemory descent (SMD) method, generalizes SMG by replacing the steepest descent direction by any direction $\boldsymbol{p}_k$ non orthogonal to $\boldsymbol{g}_k$ *i.e.,* $\boldsymbol{g}_k^T \boldsymbol{p}_k \neq 0$. PCD-SESOP and SSF-SESOP algorithms from [6, 19] identify with SMD algorithm, when $\boldsymbol{p}_k$ equals respectively the parallel coordinate descent (PCD) direction and the separable surrogate functional (SSF) direction, both described in [19].

Although the fourth algorithm was introduced in [33–35] as a supermemory gradient method, we rather refer to it as a *gradient subspace* (GS) algorithm in order to make the distinction with the supermemory gradient (SMG) algorithm introduced in [24].

The orthogonal subspace (ORTH) algorithm was introduced in [36] with the aim to obtain a first order algorithm with an optimal worst case convergence rate. The ORTH subspace corresponds to the opposite gradient augmented with the two so-called Nemirovski directions, $\boldsymbol{x}_k - \boldsymbol{x}_0$ and $\sum_{i=0}^{k} w_i \boldsymbol{g}_i$, where w_i are pre-specified, recursively defined weights:

$$w_i = \begin{cases} 1 & \text{if } i = 0, \\ \frac{1}{2} + \sqrt{\frac{1}{4} + w_{i-1}^2} & \text{otherwise.} \end{cases} \tag{6}$$

In [26], the Nemirovski subspace is augmented with previous directions, leading to the SESOP algorithm whose efficiency over ORTH is illustrated on a set of image reconstruction problems. Moreover, experimental tests showed that the use of Nemirovski directions in SESOP does not improve practical convergence speed. Therefore, in their recent paper [6], Zibulevsky *et al.* do not use these additionnal vectors so that their modified SESOP algorithm actually reduces to the SMG algorithm from [24].

2) Newton type subspace algorithms: The last two algorithms introduce additional directions of the Newton type.

Acronym	Algorithm	Set of directions $\boldsymbol{D}_k$	Subspace size
MG	Memory gradient [23, 31]	$\left[-\boldsymbol{g}_k, \boldsymbol{d}_{k-1}\right]$	2
SMG	Supermemory gradient [24]	$\left[-\boldsymbol{g}_k, \boldsymbol{d}_{k-1}, \ldots, \boldsymbol{d}_{k-m}\right]$	$m+1$
SMD	Supermemory descent [32]	$\left[\boldsymbol{p}_k, \boldsymbol{d}_{k-1}, \ldots, \boldsymbol{d}_{k-m}\right]$	$m+1$
GS	Gradient subspace [33, 34, 37]	$\left[-\boldsymbol{g}_k, -\boldsymbol{g}_{k-1}, \ldots, -\boldsymbol{g}_{k-m}\right]$	$m+1$
ORTH	Orthogonal subspace [36]	$\left[-\boldsymbol{g}_k, \boldsymbol{x}_k - \boldsymbol{x}_0, \sum_{i=0}^{k} w_i \boldsymbol{g}_i\right]$	3
SESOP	Sequential Subspace Optimization [26]	$\left[-\boldsymbol{g}_k, \boldsymbol{x}_k - \boldsymbol{x}_0, \sum_{i=0}^{k} w_i \boldsymbol{g}_i, \boldsymbol{d}_{k-1}, \ldots, \boldsymbol{d}_{k-m}\right]$	$m+3$
QNS	Quasi-Newton subspace [20, 25, 38]	$\left[-\boldsymbol{g}_k, \boldsymbol{\delta}_{k-1}, \ldots, \boldsymbol{\delta}_{k-m}, \boldsymbol{d}_{k-1}, \ldots, \boldsymbol{d}_{k-m}\right]$	$2m+1$
SESOP-TN	Truncated Newton subspace [27]	$\left[\boldsymbol{d}_k^{\ell}, \boldsymbol{Q}_k(\boldsymbol{d}_k^{\ell}), \boldsymbol{d}_k^{\ell} - \boldsymbol{d}_k^{\ell-1}, \boldsymbol{d}_{k-1}, \ldots, \boldsymbol{d}_{k-m}\right]$	$m+3$

TABLE I

SET OF DIRECTIONS CORRESPONDING TO THE MAIN EXISTING ITERATIVE SUBSPACE ALGORITHMS. THE WEIGHTS w_i AND THE VECTORS $\boldsymbol{\delta}_i$ ARE DEFINED BY (6) AND (7), RESPECTIVELY. $\boldsymbol{Q}_k$ IS DEFINED BY (8), AND $\boldsymbol{d}_k^{\ell}$ IS THE ℓTH OUTPUT OF A CG ALGORITHM TO SOLVE $\boldsymbol{Q}_k(\boldsymbol{d}) = \boldsymbol{0}$.

In the Quasi-Newton subspace (QNS) algorithm proposed in [25], $\boldsymbol{D}_k$ is augmented with

$$\boldsymbol{\delta}_{k-i} = \boldsymbol{g}_{k-i+1} - \boldsymbol{g}_{k-i},\ i = 1, \ldots, m. \tag{7}$$

This proposal is reminiscent from the L-BFGS algorithm [22], since the latter produces directions in the space spanned by the resulting set $\boldsymbol{D}_k$.

SESOP-TN has been proposed in [27] to solve the problem of sensitivity to an early break of conjugate gradient (CG) iterations in the truncated Newton (TN) algorithm. Let $\boldsymbol{d}_k^{\ell}$ denote the current value of $\boldsymbol{d}$ after ℓ iterations of CG to solve the Gauss-Newton system $\boldsymbol{Q}_k(\boldsymbol{d}) = \boldsymbol{0}$, where

$$\boldsymbol{Q}_k(\boldsymbol{d}) = \nabla^2 F(\boldsymbol{x}_k)\, \boldsymbol{d} + \boldsymbol{g}_k. \tag{8}$$

In the standard TN algorithm, $\boldsymbol{d}_k^{\ell}$ defines the search direction [39]. In SESOP-TN, it is only the first component of $\boldsymbol{D}_k$, while the second and third components of $\boldsymbol{D}_k$ also result from the CG iterations.

Finally, to accelerate optimization algorithms, a common practice is to use a preconditioning matrix. The principle is to introduce a linear transform on the original variables, so that the new variables have a Hessian matrix with more clustered eigenvalues. Preconditioned versions of subspace algorithms are easily defined by using $\boldsymbol{P}_k \boldsymbol{g}_k$ instead of $\boldsymbol{g}_k$ in the previous direction sets [26].

B. Stepsize strategies

The aim of the multi-dimensional stepsize search is to determine $\boldsymbol{s}_k$ that ensures a sufficient decrease of function f defined by (5) in order to guarantee the convergence of recurrence (4). In the scalar case, typical line search procedures generate a series of stepsize values until the fulfillment of sufficient convergence conditions such as Armijo, Wolfe and Goldstein [40]. An extension of these conditions to the multi-dimensional case can easily be obtained (*e.g.,* the multi-dimensional Goldstein rule in [28]). However, it is difficult to design practical multi-dimensional stepsize search algorithms allowing to check these conditions [28].

Instead, in several subspace algorithms, the stepsize results from an iterative descent algorithm applied to function f, stopped before convergence. In SESOP and SESOP-TN, the minimization is performed by a Newton method. However, unless the minimizer is found exactly, the resulting subspace algorithms are not proved to converge. In the QNS and GS algorithms, the stepsize results from a trust region recurrence on f. It is shown to ensure the convergence of the iterates under mild conditions on $\boldsymbol{D}_k$ [25, 34, 35]. However, except when the quadratic approximation of the criterion in the trust region is separable [34], the trust region search requires to solve a non-trivial constrained quadratic programming problem at each inner iteration.

In the particular case of modern SMG algorithms [41–44], $\boldsymbol{s}_k$ is computed in two steps. First, a descent direction is constructed by combining the vectors $\boldsymbol{d}_k^i$ with some predefined weights. Then a scalar stepsize is calculated through an iterative line search. This strategy leads to the recurrence

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \left(-\beta_k^0 \boldsymbol{g}_k + \sum_{i=1}^{m} \beta_k^i \boldsymbol{d}_{k-i} \right).$$

Different expressions for the weights β_k^i have been proposed. To our knowledge, their extension to the preconditioned version of SMG or to other subspaces is an open issue. Moreover, since the computation of (α_k, β_k^i) does not aim at minimizing f in the SMG subspace, the resulting schemes are not true subspace algorithms.

In the next section, we propose an original strategy to define the multi-dimensional stepsize $\boldsymbol{s}_k$ in (4). The proposed stepsize search is proved to ensure the convergence of the whole algorithm, under low assumptions on the subspace, and to require low computationnal cost.

III. Proposed multi-dimensional stepsize strategy

A. GR and GY majorizing approximations

Let us first introduce Geman & Yang [3] and Geman & Reynolds [2] matrices $\boldsymbol{A}_{\mathrm{GY}}$ and $\boldsymbol{A}_{\mathrm{GR}}$, which play a central role in the multi-dimensional stepsize strategy proposed in this paper:

$$\boldsymbol{A}^{a}_{\mathrm{GY}} = 2\boldsymbol{H}^T\boldsymbol{H} + \frac{\lambda}{a}\boldsymbol{V}\boldsymbol{V}^T, \tag{9}$$

$$\boldsymbol{A}_{\mathrm{GR}}(\boldsymbol{x}) = 2\boldsymbol{H}^T\boldsymbol{H} + \lambda\boldsymbol{V}^T\mathrm{Diag}\left\{\boldsymbol{b}(\boldsymbol{x})\right\}\boldsymbol{V}, \tag{10}$$

where $\boldsymbol{V}^T = \left[\boldsymbol{V}_1^T|...|\boldsymbol{V}_C^T\right]$, $a > 0$ is a free parameter, and $\boldsymbol{b}(\boldsymbol{x})$ is a $CP \times 1$ vector with entries

$$b_{cp}(\boldsymbol{x}) = \frac{\dot{\psi}(\|\boldsymbol{V}_c\boldsymbol{x} - \boldsymbol{\omega}_c\|)}{\|\boldsymbol{V}_c\boldsymbol{x} - \boldsymbol{\omega}_c\|}.$$

Both GY and GR matrices allow the construction of majorizing approximation for F. More precisely, let us introduce the following second order approximation of F in the neighborhood of $\boldsymbol{x}_k$

$$Q(\boldsymbol{x}, \boldsymbol{x}_k) = F(\boldsymbol{x}_k) + \nabla F(\boldsymbol{x}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^T\boldsymbol{A}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k). \tag{11}$$

Let us also introduce the following assumptions on the function ψ:

(H1) ψ is $\mathcal{C}^1$ and coercive,
$\dot{\psi}$ is L-Lipschitz.

(H2) ψ is $\mathcal{C}^1$, even and coercive,
$\psi(\sqrt{.})$ is concave on $\mathbb{R}^+$,
$0 < \dot{\psi}(t)/t < \infty, \quad \forall t \in \mathbb{R}$.

Then, the following lemma holds.

Lemma 1. *[13]*
Let F defined by (1) *and $\boldsymbol{x}_k \in \mathbb{R}^N$. If Assumption H1 holds and $\boldsymbol{A} = \boldsymbol{A}^{a}_{\mathrm{GY}}$ with $a \in (0, 1/L)$ (resp. Assumption H2 holds and $\boldsymbol{A} = \boldsymbol{A}_{\mathrm{GR}}$), then for all $\boldsymbol{x}$,* (11) *is a* tangent majorant *for F at $\boldsymbol{x}_k$ i.e., for all $\boldsymbol{x} \in \mathbb{R}^n$,*

$$\begin{cases} Q(\boldsymbol{x}, \boldsymbol{x}_k) \geq F(\boldsymbol{x}), \\ Q(\boldsymbol{x}_k, \boldsymbol{x}_k) = F(\boldsymbol{x}_k). \end{cases} \tag{12}$$

The majorizing property (12) ensures that the MM recurrence

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} Q(\boldsymbol{x}, \boldsymbol{x}_k) \tag{13}$$

produces a nonincreasing sequence $(F(\boldsymbol{x}_k))$ that converges to a stationnary point of F [30, 45]. Half-quadratic algorithms [2, 3] are based on the relaxed form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \theta(\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_k). \tag{14}$$

where $\hat{\boldsymbol{x}}_{k+1}$ is obtained by (13). The convergence properties of recurrence (14) are analysed in [12, 13, 46].

B. Majorize-Minimize line search

In [29], $\boldsymbol{x}_{k+1}$ is defined as (3) where $\boldsymbol{d}_k$ is the NLCG direction and the stepsize value α_k results from $J \geq 1$ successive minimizations of quadratic tangent majorant functions for the scalar function $f(\alpha) = F(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$, expressed as

$$q(\alpha, \alpha_k^j) = f(\alpha_k^j) + (\alpha - \alpha_k^j)\dot{f}(\alpha_k^j) + \frac{1}{2} b_k^j (\alpha - \alpha_k^j)^2$$

at α_k^j. The scalar parameter b_k^j is defined as

$$b_k^j = \boldsymbol{d}_k^T \boldsymbol{A}(\boldsymbol{x}_k + \alpha_k^j \boldsymbol{d}_k)\boldsymbol{d}_k.$$

where $\boldsymbol{A}(.)$ is either the GY or the GR matrix, respectively defined by (9) and (10). The stepsize values are produced by the relaxed MM recurrence

$$\begin{cases} \alpha_k^0 & = 0 \\ \alpha_k^{j+1} & = \alpha_k^j - \theta \dot{f}(\alpha_k^j)/b_k^j, \quad j = 0, \ldots, J-1 \end{cases} \tag{15}$$

and the stepsize α_k corresponds to the last value α_k^J. The distinctive feature of the MM line search is to yield the convergence of standard descent algorithms without any stopping condition whatever the recurrence length J and relaxation parameter $\theta \in (0, 2)$ [29]. Here, we propose to extend this strategy to the determination of the multi-dimensional stepsize $\boldsymbol{s}_k$, and we prove the convergence of the resulting family of subspace algorithms.

C. MM multi-dimensional search

Let us define the $M \times M$ symmetric positive definite (SPD) matrix

$$B_k^j = \boldsymbol{D}_k^T \boldsymbol{A}_k^j \boldsymbol{D}_k$$

with $\boldsymbol{A}_k^j \triangleq \boldsymbol{A}(\boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{s}_k^j)$ and $\boldsymbol{A}$ is either the GY matrix or the GR matrix. According to Lemma 1,

$$q(\boldsymbol{s}, \boldsymbol{s}_k^j) = f(\boldsymbol{s}_k^j) + \nabla f(\boldsymbol{s}_k^j)^T(\boldsymbol{s} - \boldsymbol{s}_k^j) + \frac{1}{2}(\boldsymbol{s} - \boldsymbol{s}_k^j)^T \boldsymbol{B}_k^j (\boldsymbol{s} - \boldsymbol{s}_k^j) \tag{16}$$

is quadratic tangent majorant for $f(\boldsymbol{s})$ at $\boldsymbol{s}_k^j$. Then, let us define the MM multi-dimensional stepsize by $\boldsymbol{s}_k = \boldsymbol{s}_k^J$, with

$$\begin{cases} \boldsymbol{s}_k^0 &= \mathbf{0}, \\ \hat{\boldsymbol{s}}_k^{j+1} &= \arg\min_{\boldsymbol{s}} q(\boldsymbol{s}, \boldsymbol{s}_k^j), \quad j = 0, \ldots, J-1. \\ \boldsymbol{s}_k^{j+1} &= \boldsymbol{s}_k^j + \theta(\hat{\boldsymbol{s}}_k^{j+1} - \boldsymbol{s}_k^j) \end{cases} \tag{17}$$

Given (16), we obtain an explicit stepsize formula

$$\boldsymbol{s}_k^{j+1} = \boldsymbol{s}_k^j - \theta\,(\boldsymbol{B}_k^j)^{-1} \nabla f(\boldsymbol{s}_k^j).$$

Moreover, according to [13], the update rule (17) produces monotonically decreasing values $(f(\boldsymbol{s}_k^j))$ if $\theta \in (0, 2)$. Let us emphasize that this stepsize procedure identifies with the HQ/MM iteration (14) when $\mathrm{span}(\boldsymbol{D}_k) = \mathbb{R}^N$, and to the HQ/MM line search (15) when $\boldsymbol{D}_k = \boldsymbol{d}_k$.

D. Convergence analysis

This section establishes the convergence of the iterative subspace algorithm (4) when $\boldsymbol{s}_k$ is chosen according to the MM strategy (17).

We introduce the following assumption, which is a necessary condition to ensure that the penalization term $\Psi(\boldsymbol{x})$ regularizes the problem of estimating $\boldsymbol{x}$ from $\boldsymbol{y}$ in a proper way

(H3) $\boldsymbol{H}$ and $\boldsymbol{V}$ are such that

$$\ker(\boldsymbol{H}^T\boldsymbol{H}) \cap \ker(\boldsymbol{V}^T\boldsymbol{V}) = \{\mathbf{0}\}.$$

Lemma 2. *[13]*

Let F be defined by (1)*, where $\boldsymbol{H}$ and $\boldsymbol{V}$ satisfy Assumption H3. If Assumption H1 or H2 holds, F is continuously differentiable and bounded below. Moreover, if for all k, j, $\boldsymbol{A} = \boldsymbol{A}_{\mathrm{GY}}^a$ with $0 < a < 1/L$ (resp., $\boldsymbol{A} = \boldsymbol{A}_{\mathrm{GR}}$), then $(\boldsymbol{A}_k^j)$ has a* positive bounded spectrum*, i.e., there exists $\nu_1 \in \mathbb{R}$ such that*

$$0 < \boldsymbol{v}^T \boldsymbol{A}_k^j \boldsymbol{v} \leq \nu_1 \|\boldsymbol{v}\|^2,\ \forall k,\ j \in \mathbb{N}, \forall \boldsymbol{v} \in \mathbb{R}^N.$$

Let us also assume that the set of directions $\boldsymbol{D}_k$ fulfills the following condition:

(H4) for all $k \geq 0$, the matrix of directions $\boldsymbol{D}_k$ is of size $N \times M$ with $1 \leq M \leq N$ and the first subspace direction $\boldsymbol{d}_k^1$ fulfills

$$\boldsymbol{g}_k^T \boldsymbol{d}_k^1 \leq -\gamma_0 \|\boldsymbol{g}_k\|^2, \tag{18}$$

$$\|\boldsymbol{d}_k^1\| \leq \gamma_1 \|\boldsymbol{g}_k\|, \tag{19}$$

with $\gamma_0, \gamma_1 > 0$.

Then, the convergence of the MM subspace scheme holds according to the following theorem.

Theorem 1. *Let F defined by* (1)*, where $\boldsymbol{H}$ and $\boldsymbol{V}$ satisfy Assumption H3. Let $\boldsymbol{x}_k$ defined by* (4)-(17) *where $\boldsymbol{D}_k$ satisfies Assumption H4, $J \geq 1$, $\theta \in (0, 2)$ and $\boldsymbol{B}_k^j = \boldsymbol{D}_k^T \boldsymbol{A}_{\mathrm{GY}}^a \boldsymbol{D}_k$ with $0 < a < 1/L$ (resp., $\boldsymbol{B}_k^j = \boldsymbol{D}_k^T \boldsymbol{A}_{\mathrm{GR}}(\boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{s}_k^j) \boldsymbol{D}_k$). If Assumption H1 (resp., Assumption H2) holds, then*

$$F(\boldsymbol{x}_{k+1}) \leq F(\boldsymbol{x}_k). \tag{20}$$

Moreover, we have convergence in the following sense:

$$\lim_{k \to \infty} \|\boldsymbol{g}_k\| = 0.$$

Proof: See Appendix A. ■

Remark 1. *Assumption H4 is fulfilled by a large family of descent directions. In particular, the following results hold.*

- *Let $(\boldsymbol{P}_k)$ be a series of SPD matrices with eigenvalues that are bounded below and above, respectively by γ_1 and $\gamma_0 > 0$. Then, according to [16, Sec. 1.2], Assumption H4 holds if $\boldsymbol{d}_k^1 = -\boldsymbol{P}_k \boldsymbol{g}_k$.*
- *According to [47], Assumption H4 also holds if $\boldsymbol{d}_k^1$ results from any fixed positive number of CG iterations on the linear system $\boldsymbol{M}_k \boldsymbol{d} = -\boldsymbol{g}_k$, provided that $(\boldsymbol{M}_k)$ is a matrix series with a positive bounded spectrum.*
- *Finally, Lemma 3 in Appendix B ensures that Assumption H4 holds if $\boldsymbol{d}_k^1$ is the PCD direction, provided that F is strongly convex and has a Lipschitz gradient.*

Remark 2. *For a preconditioned NLCG algorithm with a variable preconditioner $\boldsymbol{P}_k$, the generated iterates belong to the subspace spanned by $-\boldsymbol{P}_k \boldsymbol{g}_k$ and $\boldsymbol{d}_{k-1}$. Whereas the convergence of the PNLCG scheme with a variable preconditioner is still an open problem [21, 48], the preconditioned MG algorithm using $\boldsymbol{D}_k = [-\boldsymbol{P}_k \boldsymbol{g}_k, \boldsymbol{d}_{k-1}]$ and the proposed MM stepsize is guaranteed to converge for bounded SPD matrices $\boldsymbol{P}_k$, according to Theorem 1.*

E. Implementation issues

In the proposed MM multi-dimensional search, the main computational burden originates from the need to multiply the spanning directions with linear operators $\boldsymbol{H}$ and $\boldsymbol{V}$, in order to compute $\nabla f(\boldsymbol{s}_k^j)$

Acronym	Recursive form of $\boldsymbol{D}_k$	$\boldsymbol{N}_k$	$\boldsymbol{W}_k$
MG	$[-\boldsymbol{g}_k, \boldsymbol{D}_{k-1}\boldsymbol{s}_{k-1}]$	$-\boldsymbol{g}_k$	$\boldsymbol{s}_{k-1}$
SMG	$[-\boldsymbol{g}_k, \boldsymbol{D}_{k-1}\boldsymbol{s}_{k-1}, \boldsymbol{D}_{k-1}(2:m)]$	$-\boldsymbol{g}_k$	$[\boldsymbol{s}_{k-1}, \boldsymbol{I}_{2:m}]$
GS	$[-\boldsymbol{g}_k, \boldsymbol{D}_{k-1}(1:m)]$	$-\boldsymbol{g}_k$	$\boldsymbol{I}_{1:m}$
ORTH	$[-\boldsymbol{g}_k, \boldsymbol{x}_k - \boldsymbol{x}_0, \omega_k\boldsymbol{g}_k + \boldsymbol{D}_{k-1}(3)]$	$[-\boldsymbol{g}_k, \boldsymbol{x}_k - \boldsymbol{x}_0, \omega_k\boldsymbol{g}_k]$	$\boldsymbol{I}_3$
QNS	$[-\boldsymbol{g}_k, \boldsymbol{g}_k + \boldsymbol{D}_{k-1}(1), \boldsymbol{D}_{k-1}(2:m), \boldsymbol{D}_{k-1}\boldsymbol{s}_{k-1}, \boldsymbol{D}_{k-1}(m+2:2m)]$	$[-\boldsymbol{g}_k, \boldsymbol{g}_k]$	$[\boldsymbol{I}_1, \boldsymbol{I}_{2:m}, \boldsymbol{s}_{k-1}, \boldsymbol{I}_{m+2:2m}]$
SESOP-TN	$[\boldsymbol{d}_k^\ell, \boldsymbol{Q}_k(\boldsymbol{d}_k^\ell), \boldsymbol{d}_k^\ell - \boldsymbol{d}_k^{\ell-1}, \boldsymbol{D}_{k-1}(4:m+2)]$	$[\boldsymbol{d}_k^\ell, \boldsymbol{Q}_k(\boldsymbol{d}_k^\ell), \boldsymbol{d}_k^\ell - \boldsymbol{d}_k^{\ell-1}]$	$\boldsymbol{I}_{4:m+2}$

TABLE II

RECURSIVE MEMORY FEATURE AND DECOMPOSITION (21) OF SEVERAL ITERATIVE SUBSPACE ALGORITHMS. HERE, $\boldsymbol{D}(i:j)$ DENOTES THE SUBMATRIX OF $\boldsymbol{D}$ MADE OF COLUMNS i TO j, AND $\boldsymbol{I}_{i:j}$ DENOTES THE MATRIX SUCH THAT $\boldsymbol{D}\,\boldsymbol{I}_{i:j} = \boldsymbol{D}(i:j)$.

and $\boldsymbol{B}_k^j$. When the problem is large scale, these products become expensive and may counterbalance the efficiency obtained when using a subset of larger dimension. In this section, we give a strategy to reduce the computational cost of the product $\boldsymbol{M}_k \triangleq \boldsymbol{\Delta}\boldsymbol{D}_k$ when $\boldsymbol{\Delta} = \boldsymbol{H}$ or $\boldsymbol{V}$. This generalizes the strategy proposed in [26, Sec. 3] for the computation of $\nabla f(\boldsymbol{s})$ and $\nabla^2 f(\boldsymbol{s})$ during the Newton search of the SESOP algorithm.

For all subspace algorithms, the set $\boldsymbol{D}_k$ can be expressed as the sum of a new matrix and a weighted version of the previous set:

$$\boldsymbol{D}_k = [\boldsymbol{N}_k|\boldsymbol{0}] + [\boldsymbol{0}|\boldsymbol{D}_{k-1}\boldsymbol{W}_k]\,. \tag{21}$$

The obtained expressions for $\boldsymbol{N}_k$ and $\boldsymbol{W}_k$ are given in Tab. II. According to (21), $\boldsymbol{M}_k$ can be obtained by the recurrence

$$\boldsymbol{M}_k = [\boldsymbol{\Delta}\boldsymbol{N}_k|\boldsymbol{0}] + [\boldsymbol{0}|\boldsymbol{M}_{k-1}\boldsymbol{W}_k]\,.$$

Assuming that $\boldsymbol{M}_k$ is stored at each iteration, the computationnal burden reduces to the product $\boldsymbol{\Delta}\boldsymbol{N}_k$. This strategy is efficient as far as $\boldsymbol{N}_k$ has a small number of columns. Moreover, the cost of the latter product does not depend on the subspace dimension, by contrast with the direct computation of $\boldsymbol{M}_k$.

IV. APPLICATION TO THE SET OF IMAGE PROCESSING PROBLEMS FROM [6]

In this section, we consider three image processing problems, namely image deblurring, tomography and compressive sensing, generated with M. Zibulevsky's code available at http://iew3.technion.ac.il/~mcib. For all problems, the synthesis-based approach is used for the reconstruction. The image is assumed to be well described as $\boldsymbol{x}^o = \boldsymbol{K}\boldsymbol{z}^o$ with a known dictionary $\boldsymbol{K}$ and a sparse vector $\boldsymbol{z}^o$. The restored image

is then defined as $\boldsymbol{x}^* = \boldsymbol{K}\boldsymbol{z}^*$ where $\boldsymbol{z}^*$ minimizes the PLS criterion

$$F(\boldsymbol{z}) = \|\boldsymbol{H}\boldsymbol{K}\boldsymbol{z} - \boldsymbol{y}\|^2 + \lambda \sum_{i=1}^{N} \psi(z_i),$$

with ψ the logarithmic smooth version of the ℓ_1 norm

$$\psi(u) = |u| - \delta \log(1 + |u|/\delta)$$

that aims at sparsifying the solution.

In [6], several subspace algorithms are compared in order to minimize F. In all cases, the multi-dimensional stepsize results from a fixed number of Newton iterations. The aim of this section is to test the convergence speed of the algorithms when the Newton procedure is replaced by the proposed MM stepsize strategy.

A. Subspace algorithm settings

SESOP [26] and PCD-SESOP [19] direction sets are considered here. The latter uses SMD vectors with $\boldsymbol{p}_k$ defined as the PCD direction

$$\boldsymbol{p}_{i,k} = \arg\min_{\alpha} F(\boldsymbol{x}_k + \alpha \boldsymbol{e}_i),\ i = 1, ..., N, \tag{22}$$

where $\boldsymbol{e}_i$ stands for the ith elementary unit vector. Following [6], the memory parameter is tuned to $m = 7$ (*i.e.*, $M = 8$). Moreover, the Nemirovski directions are discarded, so that SESOP identifies with the SMG subspace.

Let us define SESOP-MM and PCD-SESOP-MM algorithms by associating SESOP and PCD-SESOP subspaces with the multi-dimensional MM stepsize strategy (17). The latter is fully specified by $\boldsymbol{A}_k^j$, J and θ. For all k, j, we define $\boldsymbol{A}_k^j = \boldsymbol{A}_{\mathrm{GR}}(\boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{s}_k^j)$ where $\boldsymbol{A}_{\mathrm{GR}}(.)$ is given by (10), and $J = \theta = 1$. Function ψ is strictly convex and fulfills both Assumptions H1 and H2. Therefore, Lemma 1 applies. Matrix $\boldsymbol{V}$ identifies with the identity matrix, so Assumption H3 holds and Lemma 2 applies. Moreover, according to Lemma 3, Assumption H4 holds and Theorem 1 ensures the convergence of SESOP-MM and PCD-SESOP-MM schemes.

MM versions of SESOP and PCD-SESOP are compared to the original algorithms from [6], where the inner minimization uses Newton iterations with backtracking line search, until the tight stopping criterion

$$\|\nabla f(\boldsymbol{s})\| < 10^{-10}$$

is met, or seven Newton updates are achieved.

For each test problem, the results were plotted as functions of either iteration numbers, or of computational times in seconds, on an Intel Pentium 4 PC (3.2 GHz CPU and 3 GB RAM).

B. Results and discussion

1) Choice between subspace strategies: According to Figs. 1, 2 and 3, the PCD-SESOP subspace leads to the best results in terms of objective function decrease per iteration, while the SESOP subspace leads to the largest decrease of the gradient norm, independently from the stepsize strategy. Moreover, when considering the computational time, it appears that SESOP and PCD-SESOP algorithms have quite similar performances.

2) Choice between stepsize strategies: The impact of the stepsize strategy is the central issue in this paper. According to a visual comparison between thin and thick plots in Figs. 1, 2 and 3, the MM stepsize strategy always leads to significantly faster algorithms compared to the original versions based on Newton search, mainly because of a reduced computational time per iteration.

Moreover, let us emphasize that the theoretical convergence of SESOP-MM and PCD-SESOP-MM is ensured according to Theorem 1. In contrast, unless the Newton search reaches the exact minimizer of $f(\boldsymbol{s})$, the convergence of SESOP and PCD-SESOP is not guaranteed theoretically.

V. Application to Edge-Preserving Image Restoration

The problem considered here is the restoration of the well-known images `boat`, `lena` and `peppers` of size $N = 512 \times 512$. These images are firstly convolved with a Gaussian point spread function of standard deviation 2.24 and of size 17×17. Secondly, a white Gaussian noise is added with a variance

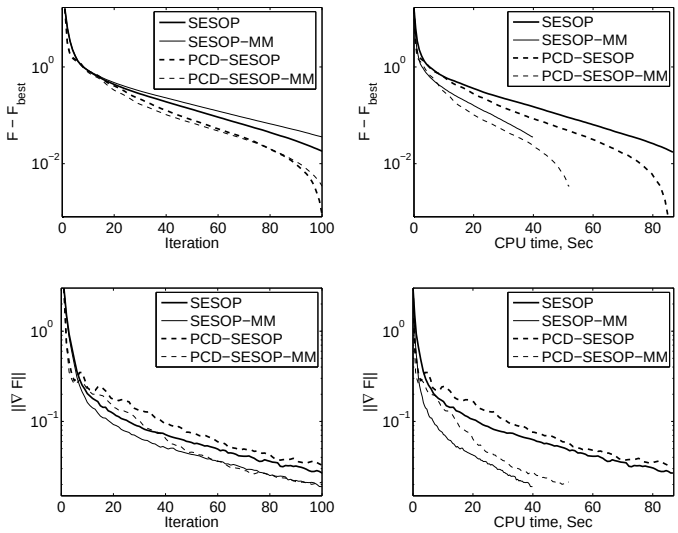

Fig. 1. Deblurring problem taken from [6] (128×128 pixels): The objective function and the gradient norm value as a function of iteration number (left) and CPU time in seconds (right) for the four tested algorithms.

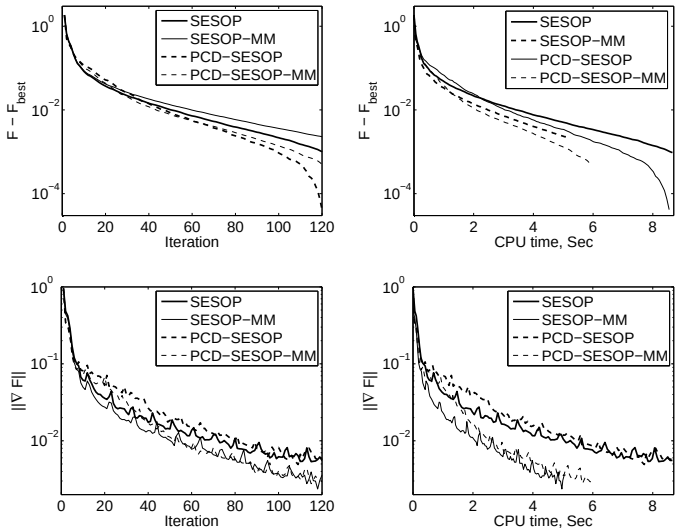

Fig. 2. Tomography problem taken from [6] (32×32 pixels): The objective function and the gradient norm value as a function of iteration number (left) and CPU time in seconds (right) for the four tested algorithms.

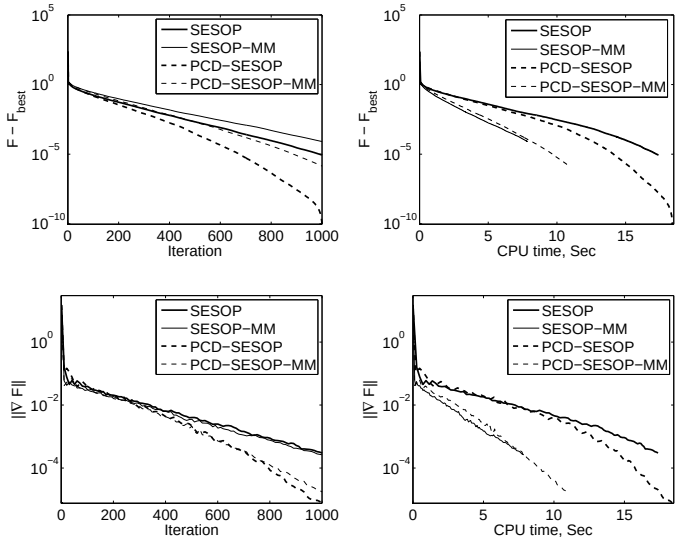

Fig. 3. Compressed sensing problem taken from [6] (64×64 pixels): The objective function and the gradient norm value as a function of iteration number (left) and CPU time in seconds (right) for the four tested algorithms.

adjusted to get a signal-to-noise ratio (SNR) of 40 dB. The following analysis-based PLS criterion is considered

$$F(\boldsymbol{x}) = \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|^2 + \lambda \sum_c \sqrt{\delta^2 + [\boldsymbol{V}\boldsymbol{x}]_c^2}$$

Fig. 4. Noisy, blurred `peppers` image, 40 dB (left) and restored image (right).

where $\boldsymbol{V}$ is the first-order difference matrix. This criterion depends on the parameters λ and δ. They are assessed to maximize the peak signal to noise ratio (PSNR) between each image $\boldsymbol{x}^o$ and its reconstruction version $\boldsymbol{x}$. Tab. III gives the resulting values of PSNR and relative mean square error (RMSE), defined by

$$\mathrm{PSNR}(\boldsymbol{x}, \boldsymbol{x}^o) = 20 \log_{10} \left(\frac{\max_i(x_i)}{\sqrt{1/N \sum_i (x_i - x_i^o)^2}} \right)$$

and

$$\mathrm{RMSE}(\boldsymbol{x}, \boldsymbol{x}^o) = \frac{\|\boldsymbol{x} - \boldsymbol{x}^o\|^2}{\|\boldsymbol{x}\|^2}.$$

The purpose of this section is to test the convergence speed of the multi-dimensional MM stepsize strategy (17) for different subspace constructions. Furthermore, these performances are compared with standard iterative descent algorithms associated with the MM line search described in Subsection III-B.

	`boat`	`lena`	`peppers`
λ	0.2	0.2	0.2
δ	13	13	8
PSNR	28.4	30.8	31.6
RMSE	$5 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$	$2 \cdot 10^{-3}$

TABLE III

VALUES OF HYPERPARAMETERS λ, δ AND RECONSTRUCTION QUALITY IN TERMS OF PSNR AND RMSE.

A. Subspace algorithm settings

The MM stepsize search is used with the Geman & Reynolds HQ matrix and $\theta = 1$. Since the hyperbolic function ψ is a strictly convex function that fulfills both Assumptions H1 and H2, Lemma 1 applies. Furthermore, Assumption H3 holds [29] so Lemma 2 applies.

Our study deals with the preconditioned form of the following direction sets: SMG, GS, QNS and SESOP-TN. The preconditioner $\boldsymbol{P}$ is a SPD matrix based on the 2D Cosine Transform. Thus, Assumption H4 holds and Theorem 1 ensures the convergence of the proposed scheme for all $J \geq 1$. Moreover, the implementation strategy described in Subsection III-E will be used.

For each subspace, we first consider the reconstruction of `peppers`, illustrated in Fig. 4, allowing us to discuss the tuning of the memory parameter m, related to the size of the subspace M as described in Tab. I, and the performances of the MM search. The latter is again compared with the Newton search from [6].

Then, we compare the subspace algorithms with iterative descent methods in association with the MM scalar line search.

The global stopping rule $\|\boldsymbol{g}_k\|/\sqrt{N} < 10^{-4}$ is considered. For each tested scheme, the performance results are displayed under the form K/T where K is the number of global iterations and T is the global minimization time in seconds.

B. Gradient and memory gradient subspaces

The aim of this section is to analyze the performances of SMG and GS algorithms.

SMG(m)		1	2	5	10
Newton		76/578	75/630	76/701	74/886
MM (J)	1	**67/119**	68/125	67/140	67/163
	2	66/141	66/147	67/172	67/206
	5	74/211	72/225	71/255	72/323
	10	76/297	74/319	73/394	74/508

TABLE IV

RECONSTRUCTION OF `peppers`: COMPARISON BETWEEN MM AND NEWTON STRATEGIES FOR THE MULTI-DIMENSIONAL SEARCH IN SMG ALGORITHM, IN TERMS OF ITERATION NUMBER AND TIME BEFORE CONVERGENCE (IN SECONDS).

1) Influence of tuning parameters: According to Tables IV-V, the algorithms perform better when the stepsize is obtained with the MM search. Furthermore, it appears that $J = 1$ leads to the best results in

GS(m)		1	5	10	15
Newton		458/3110	150/1304	96/1050	81/1044
MM (J)	1	315/534	128/258	76/180	**67/175**
	2	316/656	134/342	86/257	70/232
	5	317/856	137/481	91/400	78/386
	10	317/1200	137/709	92/619	78/598

TABLE V

RECONSTRUCTION OF `peppers`: COMPARISON BETWEEN MM AND NEWTON STRATEGIES FOR THE MULTI-DIMENSIONAL SEARCH IN GS ALGORITHM.

terms of computation time which indicates that the best strategy corresponds to a rough minimization of $f(\boldsymbol{s})$. Such a conclusion meets that of [29].

The effect of the memory size m differs according to the subspace construction. For the SMG algorithm, an increase of the size of the memory m does not accelerate the convergence. On the contrary, it appears that the number of iterations for GS decreases when more gradients are saved and the best tradeoff is obtained with $m = 15$.

2) Comparison with conjugate gradient algorithms: Let us compare the MG algorithm (*i.e.*, SMG with $m = 1$) with the NLCG algorithm making use of the MM line search strategy proposed in [29]. The latter is based on the following descent recurrence:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k(-\boldsymbol{g}_k + \beta_k \boldsymbol{d}_{k-1})$$

where β_k is the conjugacy parameter. Tab. VI summarizes the performances of NLCG for five different conjugacy strategies described in [21]. The stepsize α_k in NLCG results from J iterations of (15) with $\boldsymbol{A} = \boldsymbol{A}_{\mathrm{GR}}$ and $\theta = 1$. According to Tab. VI, the convergence speed of the conjugate gradient method is very sensitive to the conjugacy strategy. The last line of Tab. VI reproduces the first column of Tab. IV. The five tested NLCG methods are outperformed by the MG subspace algorithm with $J = 1$, both in terms of iteration number and computational time.

The two other cases `lena` and `boat` lead to the same conclusion, as reported in Tab. VII.

C. Quasi-Newton subspace

Dealing with the QNS algorithm, the best results were observed with $J = 1$ iteration of the MM stepsize strategy and the memory parameter $m = 1$. For this setting, the `peppers` image is restored

J	1	2	5	10
NLCG-FR	$145/270$	$137/279$	$143/379$	$143/515$
NLCG-DY	$234/447$	$159/338$	$144/387$	$143/516$
NLCG-PRP	$77/137$	$69/139$	$75/202$	$77/273$
NLCG-HS	$68/122$	$67/134$	$75/191$	$77/289$
NLCG-LS	$82/149$	$67/135$	$74/190$	$76/266$
MG	$\mathbf{67/119}$	$66/141$	$74/211$	$76/297$

TABLE VI

RECONSTRUCTION OF `peppers`: COMPARISON BETWEEN MG AND NLCG FOR DIFFERENT CONJUGACY STRATEGIES. IN ALL CASES, THE STEPSIZE RESULTS FROM J ITERATIONS OF THE MM RECURRENCE.

	`boat`	`lena`	`peppers`
NLCG-FR	$77/141$	$98/179$	$145/270$
NLCG-DY	$86/161$	$127/240$	$234/447$
NLCG-PRP	$40/74$	$55/99$	$77/137$
NLCG-HS	$39/71$	$50/93$	$68/122$
NLCG-LS	$42/81$	$57/103$	$82/149$
MG	$\mathbf{37/67}$	$\mathbf{47/85}$	$\mathbf{67/119}$

TABLE VII

COMPARISON BETWEEN MG AND NLCG ALGORITHMS. IN ALL CASES, THE NUMBER OF MM SUBITERATIONS IS SET TO $J = 1$.

after 68 iterations, which takes $124\,$s. As a comparison, when the Newton search is used and $m = 1$, the QNS algorithm requires 75 iterations that take more than $1000\,$s.

Let us now compare the QNS algorithm with the standard L-BFGS algorithm from [22]. Both algorithms require the tuning of the memory size m. Fig. 5 illustrates the performances of the two algorithms. In both cases, the stepsize results from 1 iteration of MM recurrence. Contrary to L-BFGS, QNS is not sensitive to the size of the memory m. Moreover, according to Tab. VIII, the QNS algorithm outperforms the standard L-BFGS algorithm with its best memory setting for the three restoration problems.

D. Truncated Newton subspace

Now, let us focus on the second order subspace method SESOP-TN. The first component of $\boldsymbol{D}_k^\ell$, $\boldsymbol{d}_k^\ell$, is computed by applying ℓ iterations of the preconditioned CG method to the Newton equations. Akin

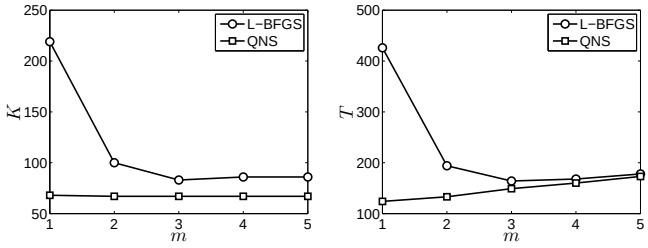

Fig. 5. Reconstruction of `peppers`: Influence of memory m for algorithms L-BFGS and QNS in terms of iteration number K and computation time T in seconds. In all cases, the number of MM subiterations is set to $J = 1$.

	`boat`	`lena`	`peppers`
L-BFGS ($m = 3$)	$45/94$	$62/119$	$83/164$
QNS ($m = 1$)	$\mathbf{38/83}$	$\mathbf{48/107}$	$\mathbf{68/124}$

TABLE VIII

COMPARISON BETWEEN QNS AND L-BFGS ALGORITHMS FOR $J = 1$.

to the standard TN algorithm, ℓ is chosen according to the following convergence test

$$\|\boldsymbol{g}_k + \boldsymbol{H}_k \boldsymbol{d}_k^\ell\| / \|\boldsymbol{g}_k\| < \eta,$$

where $\eta > 0$ is a threshold parameter. Here, the setting $\eta = 0.5$ has been adopted since it leads to lowest computation time for the standard TN algorithm.

In Tables IX and X, the results are reported in the form K/T where K denotes the total number of CG steps.

According to Tab. IX, SESOP-TN-MM behaves differently from the previous algorithms. A quite large value of J is necessary to obtain the fastest version. In this example, the MM search is still more efficient than the Newton search, provided that we choose $J \geq 5$. Concerning the memory parameter, the best results are obtained for $m = 2$.

Finally, Tab. X summarizes the results for the three test images, in comparison with the standard TN (not fully standard, though, since the MM line search has been used). Our conclusion is that the subspace version of TN does not seem to bring a significant acceleration compared to the standard version. Again, this contrasts with the results obtained for the other tested subspace methods.

SESOP-TN(m)		0	1	2	5
Newton		159/436	155/427	128/382	151/423
MM (J)	1	415/870	410/864	482/979	387/840
	2	253/532	232/506	239/525	345/731
	5	158/380	132/316	143/359	139/351
	10	122/322	134/323	**119/301**	128/334
	15	114/320	134/365	117/337	127/389

TABLE IX

RECONSTRUCTION OF `peppers`: COMPARISON BETWEEN MM AND NEWTON STEPSIZE STRATEGIES IN SESOP-TN ALGORITHM.

	`boat`	`lena`	`peppers`
TN	65/192	**74/199**	137/322
SESOP-TN(2)	**55/180**	76/218	**119/301**

TABLE X

COMPARISON BETWEEN SESOP-TN AND TN ALGORITHMS FOR $\eta = 0.5$ AND $J = 10$.

VI. CONCLUSION

This paper explored the minimization of penalized least squares criteria in the context of image restoration, using the subspace algorithm approach. We pointed out that the existing strategies for computing the multi-dimensional stepsize suffer either from a lack of convergence results (*e.g.*, Newton search) or from a high computational cost (*e.g.*, trust region method). As an alternative, we proposed an original stepsize strategy based on a MM recurrence. The stepsize results from the minimization of a half-quadratic approximation over the subspace. Our method benefits from mathematical convergence results, whatever the number of MM iterations. Moreover, it can be implemented efficiently by taking advantage of the recursive structure of the subspace.

On practical restoration problems, the proposed search is significantly faster than the Newton minimization used in [6, 26, 27], in terms of computational time before convergence. Quite remarkably, the best performances have almost always been obtained when only one MM iteration was performed ($J = 1$), and when the size of the memory was reduced to one stored iterate ($m = 1$), which means that simplicity and efficiency meet in our context. In particular, the resulting algorithmic structure contains no nested

iterations.

Finally, among all the tested variants of subspace methods, the best results were obtained with the memory gradient subspace (*i.e.*, where the only stored vector is the previous direction), using a single MM iteration for the stepsize. The resulting algorithm can be viewed as a new form of preconditioned, nonlinear conjugate gradient algorithm, where the conjugacy parameter and the step-size are jointly given by a closed-form formula that amounts to solve a 2×2 linear system.

APPENDIX

A. Proof of Theorem 1

Let us introduce the scalar function

$$h(\alpha) \triangleq q([\alpha, 0, \ldots, 0]^T, \mathbf{0}),\ \forall \alpha \in \mathbb{R}. \tag{23}$$

According to the expression of $q(., \mathbf{0})$, h reads

$$h(\alpha) = f(\mathbf{0}) + \alpha \boldsymbol{g}_k^T \boldsymbol{d}_k^1 + \frac{1}{2}\alpha^2 \boldsymbol{d}_k^{1T} \boldsymbol{A}_k^0 \boldsymbol{d}_k^1. \tag{24}$$

Its minimizer $\hat{\alpha}_k$ is given by

$$\hat{\alpha}_k = -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k^1}{\boldsymbol{d}_k^{1T} \boldsymbol{A}_k^0 \boldsymbol{d}_k^1}. \tag{25}$$

Therefore,

$$h(\hat{\alpha}_k) = f(\mathbf{0}) + \frac{1}{2}\hat{\alpha}_k \boldsymbol{g}_k^T \boldsymbol{d}_k^1. \tag{26}$$

Moreover, according to the expression of $\hat{\boldsymbol{s}}_k^1$,

$$q(\hat{\boldsymbol{s}}_k^1, \mathbf{0}) = f(\mathbf{0}) + \frac{1}{2}\nabla f(\mathbf{0})^T \hat{\boldsymbol{s}}_k^1. \tag{27}$$

$\hat{\boldsymbol{s}}_k^1$ minimizes $q(\boldsymbol{s}, \mathbf{0})$ hence $q(\hat{\boldsymbol{s}}_k^1, \mathbf{0}) \leq h(\hat{\alpha}_k)$. Thus, using (26)-(27),

$$\hat{\alpha}_k \boldsymbol{g}_k^T \boldsymbol{d}_k^1 \geq \nabla f(\mathbf{0})^T \hat{\boldsymbol{s}}_k^1. \tag{28}$$

According to (24) and (25), the relaxed stepsize $\alpha_k = \theta \hat{\alpha}_k$ fulfills

$$h(\alpha_k) = f(\mathbf{0}) + \delta\, \hat{\alpha}_k \boldsymbol{g}_k^T \boldsymbol{d}_k^1, \tag{29}$$

where $\delta = \theta(1 - \theta/2)$. Moreover,

$$q(\boldsymbol{s}_k^1, \mathbf{0}) = f(\mathbf{0}) + \delta\, \nabla f(\mathbf{0})^T \hat{\boldsymbol{s}}_k^1. \tag{30}$$

Thus, using (28)-(29)-(30), we obtain $q(\boldsymbol{s}_k^1, \mathbf{0}) \leq h(\alpha_k)$ and

$$f(\mathbf{0}) - q(\boldsymbol{s}_k^1, \mathbf{0}) \geq -\delta \hat{\alpha}_k \boldsymbol{g}_k^T \boldsymbol{d}_k^1. \tag{31}$$

Furthermore, $q(\boldsymbol{s}_k^1, \mathbf{0}) \geq f(\boldsymbol{s}_k^1) \geq f(\boldsymbol{s}_k)$ according to Lemma 1 and [13, Prop.5]. Thus,

$$f(\mathbf{0}) - f(\boldsymbol{s}_k) \geq -\delta \hat{\alpha}_k \boldsymbol{g}_k^T \boldsymbol{d}_k^1 \tag{32}$$

According to Lemma 2,

$$\hat{\alpha}_k \geq -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k^1}{\nu_1 \|\boldsymbol{d}_k^1\|^2} \tag{33}$$

Hence, according to (32), (33) and Assumption H4,

$$f(\mathbf{0}) - f(\boldsymbol{s}_k) \geq \frac{\delta \gamma_0^2}{\nu_1 \gamma_1^2} \|\boldsymbol{g}_k\|^2 \tag{34}$$

which also reads

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_{k+1}) \geq \frac{\delta \gamma_0^2}{\nu_1 \gamma_1^2} \|\boldsymbol{g}_k\|^2 \tag{35}$$

Thus, (20) holds. Moreover, F is bounded below according to Lemma 2. Therefore, $\lim_{k\to\infty} F(\boldsymbol{x}_k)$ is finite. Thus,

$$\infty > \left(\frac{\delta \gamma_0^2}{\nu_1 \gamma_1^2}\right)^{-1} \left(F(\boldsymbol{x}_0) - \lim_{k\to\infty} F(\boldsymbol{x}_k)\right) \geq \sum_k \|\boldsymbol{g}_k\|^2,$$

and finally

$$\lim_{k\to\infty} \|\boldsymbol{g}_k\| = 0.$$

B. Relations between the PCD and the gradient directions

Lemma 3. *Let the PCD direction be defined by $\boldsymbol{p} = (p_i)$, with*

$$p_i = \arg\min_{\alpha} F(\boldsymbol{x} + \alpha \boldsymbol{e}_i),\ i = 1, ..., N,$$

where $\boldsymbol{e}_i$ stands for the ith elementary unit vector. If F is gradient Lipschitz and strongly convex on $\mathbb{R}^N$, then there exist $\gamma_0, \gamma_1 > 0$ such that $\boldsymbol{p}$ fulfills

$$\boldsymbol{g}^T \boldsymbol{p} \leq -\gamma_0 \|\boldsymbol{g}\|^2, \tag{36}$$

$$\|\boldsymbol{p}\| \leq \gamma_1 \|\boldsymbol{g}\|, \tag{37}$$

for all $\boldsymbol{x} \in \mathbb{R}^N$.

Proof: Let us introduce the scalar functions $f_i(\alpha) \triangleq F(\boldsymbol{x} + \alpha \boldsymbol{e}_i)$, so that

$$p_i = \arg\min_{\alpha} f_i(\alpha). \tag{38}$$

F is gradient Lipschitz, so there exists $L > 0$ such that for all i,

$$|\dot{f}_i(a) - \dot{f}_i(b)| \leqslant L|a - b|, \quad \forall a, b \in \mathbb{R}.$$

In particular, for $a = 0$ and $b = p_i$, we obtain

$$|p_i| \geqslant |\dot{f}_i(0)|/L,$$

given that $\dot{f}_i(p_i) = 0$ according to (38). According to the expression of f_i,

$$\boldsymbol{g}^T\boldsymbol{p} = \sum_{i=1}^{N} \dot{f}_i(0)p_i.$$

Moreover, p_i minimizes the convex function f_i on $\mathbb{R}$ so

$$p_i\dot{f}_i(0) \leqslant 0, \quad i = 1, ..., N. \tag{39}$$

Therefore,

$$\boldsymbol{g}^T\boldsymbol{p} = -\sum_{i=1}^{N} |\dot{f}_i(0)||p_i| \leqslant \frac{1}{L}\|\boldsymbol{g}\|^2. \tag{40}$$

F is strongly convex, so there exists $\nu > 0$ such that for all i,

$$(\dot{f}_i(a) - \dot{f}_i(b))(a - b) \geqslant \nu(a - b)^2, \quad \forall a, b \in \mathbb{R}.$$

In particular, $a = 0$ and $b = p_i$ give

$$-\dot{f}_i(0)p_i \geqslant \nu p_i^2, \quad i = 1, ..., N. \tag{41}$$

Using (39) we obtain

$$p_i^2 \leqslant \nu|\dot{f}_i(0)|^2/\nu^2, \quad i = 1, ..., N. \tag{42}$$

Therefore,

$$\|\boldsymbol{p}\|^2 = \sum_{i=1}^{N} p_i^2 \leqslant \frac{1}{\nu^2}\|\boldsymbol{g}\|^2 \tag{43}$$

Thus, (36)-(37) hold for $\gamma_0 = 1/L$ and $\gamma_1 = 1/\nu$. ■

REFERENCES

[1] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

[2] S. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 367–383, March 1992.

[3] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Processing*, vol. 4, no. 7, pp. 932–946, July 1995.

[4] A. Chambolle, R. A. De Vore, L. Nam-Yong, and B. Lucier, "Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 319–335, March 1998.

[5] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Processing*, vol. 16, no. 12, pp. 2980–2991, 2007.

[6] M. Zibulevsky and M. Elad, "$\ell_2 - \ell_1$ optimization in signal and image processing," *IEEE Signal Processing Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.

[7] G. Demoment, "Image reconstruction and restoration: Overview of common estimation structure and problems," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-37, no. 12, pp. 2024–2036, December 1989.

[8] P. J. Huber, *Robust Statistics*. New York, NY: John Wiley, 1981.

[9] S. Geman and D. McClure, "Statistical methods for tomographic image reconstruction," in *Proceedings of the 46th Session of the ICI, Bulletin of the ICI*, vol. 52, 1987, pp. 5–21.

[10] C. Bouman and K. D. Sauer, "A generalized gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Processing*, vol. 2, no. 3, pp. 296–310, July 1993.

[11] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Processing*, vol. 6, pp. 298–311, 1997.

[12] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, pp. 937–966, 2005.

[13] A. Allain, J. Idier, and Y. Goussard, "On global and local convergence of half-quadratic algorithms," *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1130–1142, 2006.

[14] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physical Review D*, vol. 60, pp. 259–268, 1992.

[15] M. Nikolova, "Weakly constrained minimization: application to the estimation of images and signals involving constant regions," *J. Math. Imag. Vision*, vol. 21, pp. 155–175, 2004.

[16] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[17] P. Ciuciu and J. Idier, "A half-quadratic block-coordinate descent method for spectral estimation," *Signal Processing*, vol. 82, no. 7, pp. 941–959, July 2002.

[18] M. Nikolova and M. K. Ng, "Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning," in *Proceedings of the International Conference on Image Processing*, 2001.

[19] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Appl. Comput. Harmon. Anal.*, vol. 23, pp. 346–367, 2006.

[20] Y. Yuan, "Subspace techniques for nonlinear optimization," in *Some Topics in Industrial and Applied Mathematics*, R. Jeltsh, T.-T. Li, and H. I. Sloan, Eds. Series on Concrete and Applicable Mathematics, 2007, vol. 8, pp. 206–218.

[21] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific J. Optim.*, vol. 2, no. 1, pp. 35–58, January 2006.

[22] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Prog.*, vol. 45, no. 3, pp. 503–528, 1989.

[23] A. Miele and J. W. Cantrell, "Study on a memory gradient method for the minimization of functions," *J. Optim. Theory Appl.*, vol. 3, no. 6, pp. 459–470, 1969.

[24] E. E. Cragg and A. V. Levy, "Study on a supermemory gradient method for the minimization of functions," *J. Optim. Theory Appl.*, vol. 4, no. 3, pp. 191–205, 1969.

[25] Z. Wang, Z. Wen, and Y. Yuan, "A subspace trust region method for large scale unconstrained optimization," in *Numerical linear algebra and optimization*, M. Science Press, Ed., 2004, pp. 264–274.

[26] G. Narkiss and M. Zibulevsky, “Sequential subspace optimization method for large-scale unconstrained problems,” Israel Institute of Technology, Technical Report 559, October 2005, http://iew3.technion.ac.il/~mcib/sesop_report_version301005.pdf.

[27] M. Zibulevsky, “SESOP-TN: Combining sequential subspace optimization with truncated Newton method,” Israel Institute of Technology,” Technical Report, September 2008, http://www.optimization-online.org/DB_FILE/2008/09/2098.pdf.

[28] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, “On iterated-subspace minimization methods for nonlinear optimization,” Rutherford Appleton Laboratory, Oxfordshire UK, Technical Report 94-069, May 1994, ftp://130.246.8.32/pub/reports/cgstRAL94069.ps.Z.

[29] C. Labat and J. Idier, “Convergence of conjugate gradient methods with a closed-form stepsize formula,” *J. Optim. Theory Appl.*, vol. 136, no. 1, pp. 43–60, January 2008.

[30] D. R. Hunter and K. L., “A tutorial on MM algorithms,” *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, February 2004.

[31] J. Cantrell, “Relation between the memory gradient method and the Fletcher-Reeves method,” *J. Optim. Theory Appl.*, vol. 4, no. 1, pp. 67–71, 1969.

[32] M. Wolfe and C. Viazminsky, “Supermemory descent methods for unconstrained minimization,” *J. Optim. Theory Appl.*, vol. 18, no. 4, pp. 455–468, 1976.

[33] Z.-J. Shi and J. Shen, “A new class of supermemory gradient methods,” *Appl. Math. and Comp.*, vol. 183, pp. 748–760, 2006.

[34] ——, “Convergence of supermemory gradient method,” *Appl. Math. and Comp.*, vol. 24, no. 1-2, pp. 367–376, 2007.

[35] Z.-J. Shi and Z. Xu, “The convergence of subspace trust region methods,” *J. Comput. Appl. Math.*, vol. 231, no. 1, pp. 365–377, 2009.

[36] A. Nemirovski, “Orth-method for smooth convex optimization,” *Izvestia AN SSSR,Transl.: Eng. Cybern. Soviet J. Comput. Syst. Sci.*, vol. 2, 1982.

[37] Z.-J. Shi and J. Shen, “A new super-memory gradient method with curve search rule,” *Appl. Math. and Comp.*, vol. 170, pp. 1–16, 2005.

[38] Z. Wang and Y. Yuan, “A subspace implementation of quasi-Newton trust region methods for unconstrained optimization,” *Numer. Math.*, vol. 104, pp. 241–269, 2006.

[39] S. G. Nash, “A survey of truncated-Newton methods,” *J. Comput. Appl. Math.*, vol. 124, pp. 45–59, 2000.

[40] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY: Springer-Verlag, 1999.

[41] Z.-J. Shi, “Convergence of line search methods for unconstrained optimization,” *Appl. Math. and Comp.*, vol. 157, pp. 393–405, 2004.

[42] Y. Narushima and Y. Hiroshi, “Global convergence of a memory gradient method for unconstrained optimization,” *Comput. Optim. and Appli.*, vol. 35, no. 3, pp. 325–346, 2006.

[43] Z. Yu, “Global convergence of a memory gradient method without line search,” *J. Appl. Math. and Comput.*, vol. 26, no. 1-2, pp. 545–553, February 2008.

[44] J. Liu, H. Liu, and Y. Zheng, “A new supermemory gradient method without line search for unconstrained optimization,” in *The Sixth International Symposium on Neural Networks*, S. Berlin, Ed., 2009, vol. 56, pp. 641–647.

[45] M. Jacobson and J. Fessler, “An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms,” *IEEE Trans. Image Processing*, vol. 16, no. 10, pp. 2411–2422, October 2007.

[46] J. Idier, “Convex half-quadratic criteria and interacting auxiliary variables for image restoration,” *IEEE Trans. Image Processing*, vol. 10, no. 7, pp. 1001–1009, July 2001.

[47] R. S. Dembo and T. Steihaug, “Truncated-Newton methods algorithms for large scale unconstrained optimization,” *Math. Prog.*, vol. 26, pp. 190–212, 1983.

[48] M. Al-Baali and R. Fletcher, “On the order of convergence of preconditioned nonlinear conjugate gradient methods,” *SIAM J. Sci. Comput.*, vol. 17, no. 3, pp. 658–665, 1996.