HAL open science

Optimization Methods for Signal Processing

Emilie Chouzenoux, Jean-Christophe Pesquet

▶ **To cite this version:**

Emilie Chouzenoux, Jean-Christophe Pesquet. Optimization Methods for Signal Processing. C.Jutten, L.T. Duarte, S. Moussaoui. Source Separation in Physical-Chemical Sensing, Chapter 2, IEEE Press; JOHN WILEY & SONS, 2023, 9781119137276. hal-04250055

HAL Id: hal-04250055

https://inria.hal.science/hal-04250055v1

Submitted on 12 Dec 2023

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.

Distributed under a Creative Commons Attribution 4.0 International License

Contents

Notations *iii*

1 Optimization Methods for Signal Processing *1*
E. Chouzenoux and J.-C. Pesquet
1.1 Introduction to optimization problems *2*
1.1.1 Problem formulation *2*
1.1.2 Theoretical background *3*
1.1.3 Examples in the context of source separation *5*
1.1.4 Chapter outline *7*
1.2 Majorization-Minimization approaches *8*
1.2.1 Majorization-Minimization principle *8*
1.2.2 Majorization techniques *9*
1.2.3 MM quadratic methods *14*
1.2.4 Variable metric forward-backward algorithm *20*
1.2.5 Block-coordinate MM algorithms *23*
1.3 Primal-dual methods *28*
1.3.1 Lagrange duality *28*
1.3.2 Alternating direction method of multipliers *29*
1.3.3 Primal-dual proximal algorithms *31*
1.3.4 Primal-dual interior point algorithm *32*
1.4 Illustration in the context of DOSY NMR signal restoration *38*
1.4.1 Quadratic penalization *39*
1.4.2 Sparsity prior in the signal domain *40*
1.4.3 Sparsity prior in a transformed domain *41*
1.4.4 Sparsity prior and range constraints *42*
1.4.5 Concluding remarks *43*
1.5 Conclusion *44*

Notations

$\boldsymbol{x}$	column vector of components x_p, $1 \leq p \leq P$
s, x, y	sources, observations, separator outputs
R	number of sources
P	number of sensors
T	number of observed samples
$\star$	convolution
$\boldsymbol{A}$	matrix with components A_{ij}
$\boldsymbol{A}$, $\boldsymbol{B}$	mixing and separation matrices
$\boldsymbol{G}$, $\boldsymbol{W}$, $\boldsymbol{Q}$	global, whitening, and separating unitary matrices
$\check{g}$	Fourier transform of g
$\widehat{\boldsymbol{s}}$	estimate of quantity $\boldsymbol{s}$
$p_{\boldsymbol{x}}$	probability density of $\boldsymbol{x}$
ψ	joint score function
φ_i	marginal score function of source s_i
$\mathbb{E}\,\boldsymbol{x}$, $\mathbb{E}\{\boldsymbol{x}\}$	mathematical expectation of $\boldsymbol{x}$
$I\{\boldsymbol{y}\}$ or $I(p_{\boldsymbol{y}})$	mutual information of $\boldsymbol{y}$
$K\{\boldsymbol{x};\boldsymbol{y}\}$ or $K(p_{\boldsymbol{x}};p_{\boldsymbol{y}})$	Kullback divergence between $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{y}}$
$H\boldsymbol{x}$ or $H(p_{\boldsymbol{x}})$	Shannon entropy $\boldsymbol{x}$
$\mathcal{L}$	likelihood
$\mathcal{A}$, $\mathcal{B}$	mixing, and separating (non linear) operators
$\mathrm{cum}\{x_1, \ldots, x_P\}$	joint cumulant of variables $\{x_1, \ldots, x_P\}$
$\mathrm{cum}_R\{y\}$	marginal cumulant of order R of variable y

$\boldsymbol{Q}^{\mathsf{T}}$	transposition
$\boldsymbol{Q}^{H}$	conjugate transposition
$\boldsymbol{Q}^{*}$	complex conjugation
$\boldsymbol{Q}^{\dagger}$	pseudo-inverse
Υ	contrast function
$\mathbb{R}$	real field
$\mathbb{C}$	complex field
$\widehat{\boldsymbol{A}}$	estimator of mixing matrix
$\mathrm{rank}\{\boldsymbol{A}\}$	rank of matrix $\boldsymbol{A}$
$\mathrm{krank}\{\boldsymbol{A}\}$	Kruskal's k-rank of matrix $\boldsymbol{A}$
$\mathrm{diag}\{\boldsymbol{A}\}$	vector whose components are the diagonal of matrix $\boldsymbol{A}$
$\mathrm{Diag}\{\boldsymbol{a}\}$	diagonal matrix whose entries are those of vector $\boldsymbol{a}$
$\mathrm{trace}\{\boldsymbol{A}\}$	trace of matrix $\boldsymbol{A}$
$\det \boldsymbol{A}$	determinant of matrix $\boldsymbol{A}$
$\check{s}(\nu)$	Fourier transform of process $s(t)$
$\boxtimes$	Kronecker product between matrices
$\odot$	Khatri-Rao (column-wise Kronecker) product between matrices
$\boxdot$	Hadamard (entry-wise) product between arrays
$\otimes$	tensor product
$\bullet_j$	contraction over index j
$\mathcal{S}$, $\mathcal{G}$	sets
$\boldsymbol{\mathcal{T}}$	tensor or tensor array (representing multilinear operators)
$\mathcal{T}_{ijk}$	entries of a tensor array
$\mathrm{vec}(\cdot)$	vectorization of an array
$\mathrm{Unvec}_n(\cdot)$	unvectorization into a matrix with n rows
$\boldsymbol{T}^{(d)}$	dth mode unfolding of a tensor array into a matrix (also called matricization or flattening)
$\mathrm{aff}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p\}$	affine hull of the set $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p\}$
$\mathrm{conv}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p\}$	convex hull of the set $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p\}$
$\mathrm{vol}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p\}$	volume of $\mathrm{conv}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_p\}$

$\mathrm{dom}\, f$	domain of function f
$\mathrm{prox}_f^{\boldsymbol{A}}(\boldsymbol{x})$	proximity operator of function f within the metric induced by $\boldsymbol{A}$ computed at $\boldsymbol{x}$
ι_C	indicator function of set C
$\mathrm{P_C}$	projector on set C
$\mathrm{epi}\, f$	epigraph of function f
$\underset{C}{\mathrm{argmin}} f$	minimum argument of f over set C
$\underset{C}{\mathrm{argmax}}\, f$	maximum argument of f over set C
$\sup\limits_C f$	supremum of f over set C
$\inf\limits_C f$	infimum of f over set C
∇f	gradient of f
$\nabla^2 f$	Hessian of f
$\partial f(\boldsymbol{u})$	subdifferential set of f at $\boldsymbol{u}$
f^*	conjugate of function f
${}^{\gamma} f$	Moreau envelope of f of parameter γ
$\square$	infimum-convolution
$\|\cdot\|_{\mathrm{F}}$	Frobenius norm
$\|\|\|\cdot\|\|\|$	spectral norm

1
Optimization Methods for Signal Processing

E. Chouzenoux and J.-C. Pesquet

Abstract

In a wide range of problems arising in source separation, large scale optimization problems need to be solved. The objective of this chapter is to introduce the theoretical background which makes it possible to develop efficient algorithms to successfully address these problems. This chapter will be mainly focused on nonlinear optimization tools for dealing with convex and nonconvex problems. Proximal tools, parallel splitting techniques, Majorization-Minimization strategies and alternating minimization approaches will be presented. Illustrations of these methods on various problems will be provided.

Keywords

Optimization, Convex analysis, Proximal algorithms, Splitting techniques, Primal-dual algorithms, Majorization-Minimization, Half-Quadratic strategies, Block coordinate minimization, Interior point algorithms.

Affiliations

E. Chouzenoux, and J.-C. Pesquet are with OPIS of Inria Saclay (Palaiseau, France) and CVN at CentraleSupélec, Université Paris-Saclay (Saclay, France).

1.1 Introduction to optimization problems

1.1.1 Problem formulation

In this chapter, we are interested in general problems involving the minimization of a cost function f with respect to a vector $\boldsymbol{u} \in \mathbb{R}^N$ with components $(u_n)_{1\leq n\leq N}$ representing variables to estimate. In the context of source separation, the unknown vector can often be block-decomposed into $J \geq 1$ blocks having various roles and properties. More specifically, we will decompose $\boldsymbol{u}$ as $(\boldsymbol{u}^{(j)})_{1\leq j\leq J} \in \mathbb{R}^{N_1} \times \ldots \times \mathbb{R}^{N_J}$, where, for every $j \in \{1, \ldots, J\}$, $\boldsymbol{u}^{(j)} = (u_n)_{n\in\mathbb{J}_j} \in \mathbb{R}^{N_j}$ and $\mathbb{J}_j$ is the index set of the components corresponding to the j-th block. With this notation, the optimization problem reads

$$\underset{\boldsymbol{u}^{(1)}\in\mathbb{R}^{N_1},\ldots,\boldsymbol{u}^{(J)}\in\mathbb{R}^{N_J}}{\text{minimize}} f(\boldsymbol{u}). \tag{1.1}$$

Function f is usually expressed as a composite objective function consisting of a sum of terms: a data fidelity term describing the existing relations existing between the observed data and the target variables, and penalization (or regularization) terms modeling the available prior information on the sought variables. Various examples of possible choices for these functions in the context of source separation will be given in Section 1.1.3. In particular, one may want to introduce some constraints on the domain of each block, namely that, for every $j \in \{1, \ldots, J\}$, $\boldsymbol{u}^{(j)}$ belongs to some set $C_j \subset \mathbb{R}^{N_j}$. Problem (1.1) then becomes a constrained optimization problem

$$\underset{\boldsymbol{u}^{(1)}\in C_1,\ldots,\boldsymbol{u}^{(J)}\in C_J}{\text{minimize}} f(\boldsymbol{u}). \tag{1.2}$$

It is important to note that Problem (1.2) actually is an instance of Problem (1.1). Indeed, it suffices to replace the cost function f in (1.1) by the sum $f + \iota_C$, where ι_C is the indicator function of set C:

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad \iota_C(\boldsymbol{u}) = \begin{cases} 0 & \text{if } \boldsymbol{u} \in C \\ +\infty & \text{elsewhere,} \end{cases} \tag{1.3}$$

and C gathers all the constraints on the block variables, i.e. C is the Cartesian product $C_1 \times \ldots \times C_J$ of the J constraint spaces. In a nutshell, allowing a cost function value to be equal to $+\infty$ over some set is a convenient way for constraining its minimizer to be out of this set.

With the exception of very specific cases (typically, fully quadratic problems), solving Problem (1.1) requires the design of an algorithm that builds a sequence of iterates $(\boldsymbol{u}_k)_{k\in\mathbb{N}}$ converging to a minimizer $\widehat{\boldsymbol{u}}$ of f. There is in the literature a bunch of optimization methods, which differ by their convergence properties, their

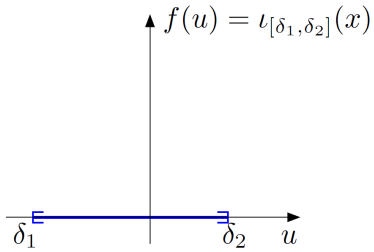

Figure 1.1 Indicator function of the interval $[\delta_1, \delta_2]$

computational cost, their memory requirements, their convergence speed, and their sensitivity to numerical errors. The goal of this chapter is to help the reader to bring answers to the following questions:

- How to exploit the mathematical properties of each term involved in f? How to handle constraints efficiently ? How to deal with non differentiable terms in f ? Which convergence result can be expected if f is non convex?
- How to reduce the memory requirements of an optimization algorithm? How to avoid large-size matrix inversion?
- What are the benefits of block alternating strategies? What are their convergence guaranties?
- How to accelerate the convergence speed of a first-order (gradient-like) optimization method?

1.1.2 Theoretical background

1.1.2.1 Convex functions

Throughout this chapter, all the functions will be assumed to take their values in $\mathbb{R}$, possibly extended to $\mathbb{R} \cup \{+\infty\}$. The domain $\mathrm{dom}\, f$ of a function f from $\mathbb{R}^N$ to $\mathbb{R} \cup \{+\infty\}$ is the subset of $\mathbb{R}^N$ over which this function is finite-valued. When this domain is nonempty, it is said that the function is *proper*. In optimization, one is able to handle discontinuous functions, but most of the time, the involved functions must be assumed to be *lower-semicontinuous* (lsc). A function f from $\mathbb{R}^N$ to $\mathbb{R} \cup \{+\infty\}$ is lower-semicontinuous if its epigraph

$$\mathrm{epi}(f) = \{(\boldsymbol{u}, \zeta) \in \mathbb{R}^N \times \mathbb{R} \mid f(\boldsymbol{u}) \leq \zeta\},$$

which basically represents the area above the graph of the function, is a closed set. It is then easy to check that continuous functions are lower-semicontinuous. In the rest of this chapter, we will assume, without recalling it explicitly, that all the considered functions are proper and lower-semicontinuous.

Another desirable property of a cost function is its convexity. Recall that a set C is convex if any line segment linking two arbitrary points of C is included in C. A

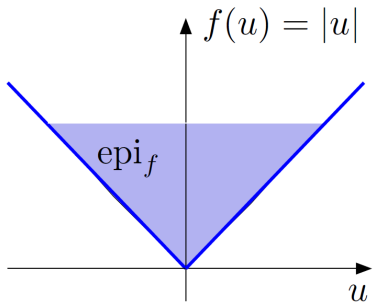

Figure 1.2 Epigraph of the absolute value function.

function f from $\mathbb{R}^N$ to $\mathbb{R} \cup \{+\infty\}$ is convex if its epigraph is convex, which is also equivalent to the following inequality:

$$\big(\forall(\boldsymbol{u},\boldsymbol{v}) \in (\mathbb{R}^N)^2\big)\big(\forall\lambda \in]0,1[\big) \quad f\big(\lambda\boldsymbol{u} + (1-\lambda)\boldsymbol{v}\big) \leq \lambda f(\boldsymbol{u}) + (1-\lambda) f(\boldsymbol{v}).$$

In addition, the function is strictly convex if the above inequality is strict whenever $(\boldsymbol{u},\boldsymbol{v}) \in (\mathrm{dom}\, f)^2$ and $\boldsymbol{u} \neq \boldsymbol{v}$. An important property of a convex function is that any of its local minimizers i.e., minimizer on a local neighborhood in its domain, is a (global) minimizer over $\mathbb{R}^N$. If the function is strictly convex, such a minimizer, whenever it exists, is unique. These properties are important since many optimization algorithms are only guaranteed to converge to a local minimizer, which may lead to suboptimal performance in practice, when the associated cost function is non convex.

1.1.2.2 Differentiability and subdifferentiability

The subdifferential of a function f from $\mathbb{R}^N$ to $\mathbb{R} \cup \{+\infty\}$, at $\boldsymbol{u} \in \mathbb{R}^N$, is defined as

$$\partial f(\boldsymbol{u}) = \{\boldsymbol{t} \in \mathbb{R}^N \mid (\forall \boldsymbol{v} \in \mathbb{R}^N) \;\; f(\boldsymbol{v}) \geq f(\boldsymbol{u}) + \boldsymbol{t}^{\mathsf{T}}(\boldsymbol{v}-\boldsymbol{u})\}. \tag{1.4}$$

Any vector $\boldsymbol{t} \in \partial f(\boldsymbol{u})$ is called a subgradient of f at $\boldsymbol{u} \in \mathbb{R}^N$. The above definition actually corresponds to the Moreau subdifferential, but other definitions are possible [1], which may be more relevant in the non convex case. It is worth noting that the affine term $f(\boldsymbol{u}) + \boldsymbol{t}^{\mathsf{T}}(\boldsymbol{v}-\boldsymbol{u})$, $\boldsymbol{t} \in \partial f(\boldsymbol{u})$ actually correspond to a supporting hyperplane crossing the graph of f at $\boldsymbol{u}$, thus generalizing the notion of tangent plane to non necessarily differentiable functions. In particular, a function may have several sugradients at a given point $\boldsymbol{u} \in \mathbb{R}^N$ or none (in which case, $\partial f(\boldsymbol{u})$ is empty). But, if f is convex and continuous at $\boldsymbol{u} \in \mathbb{R}^N$, then $\partial f(\boldsymbol{u})$ is nonempty. Furthermore, if f is differentiable at $\boldsymbol{u} \in \mathbb{R}^N$ and it is convex, its only subgradient at $\boldsymbol{u} \in \mathbb{R}^N$ is its gradient $\nabla f(\boldsymbol{u})$.

A key property of the Moreau subdifferential is that it allows the characterization of solutions to an optimization problem. Indeed, according to Fermat's rule, $\widehat{\boldsymbol{u}}$ is a (global) minimizer of f if and only if $\mathbf{0}$ is a subgradient of f at $\widehat{\boldsymbol{u}}$. However, if f

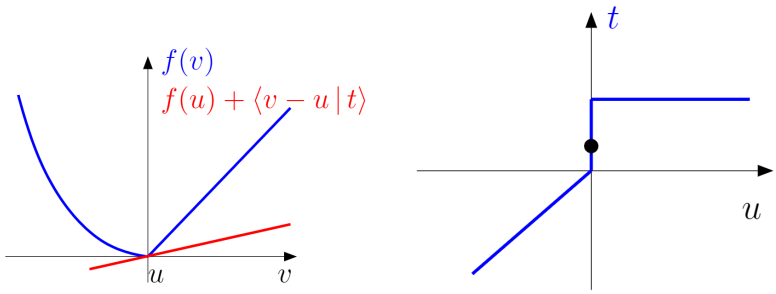

Figure 1.3 Function f and a supporting line of it at $u = 0$ (left). Graph of the subdifferential (right).

is differentiable at $\widehat{\boldsymbol{u}}$ and non convex, the condition $\nabla f(\widehat{\boldsymbol{u}}) = \mathbf{0}$ only constitutes a necessary condition for $\widehat{\boldsymbol{u}}$ to be a minimizer of f.

1.1.3 Examples in the context of source separation

1.1.3.1 Non-negative matrix factorization

The problem in non-negative matrix factorization is to decompose a given , possibly noisy, data matrix $\boldsymbol{X} \in \mathbb{R}^{P \times T}$ into a product $\boldsymbol{AS}$ where $\boldsymbol{A} \in \mathbb{R}^{P \times R}$ and $\boldsymbol{S} \in \mathbb{R}^{R \times T}$ are constrained to have non-negative entries. A common strategy to solve this algebraic problem is to define $\boldsymbol{A}$ and $\boldsymbol{S}$ as the solutions of a constrained optimization problem of the form:

$$\underset{\boldsymbol{A} \geq 0, \boldsymbol{S} \geq 0}{\text{minimize}} \; \Phi(\boldsymbol{AS}, \boldsymbol{X}) + \Theta_{\mathrm{a}}(\boldsymbol{A}) + \Theta_{\mathrm{s}}(\boldsymbol{S}), \tag{1.5}$$

where Φ measures the discrepancy between the data $\boldsymbol{X}$ and the sought product $\boldsymbol{AS}$ (see Table 1.1 for some examples), and Θ_{a} (resp. Θ_{s}) introduces some a priori information on the unknown matrices.

Name	Expression	Reference
Euclidean	$\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{AS}\|_{\mathrm{F}}^2$	[2]
Kullback-Leibler	$\sum_{p=1}^{P} \sum_{t=1}^{T} (\boldsymbol{AS})_{p,t} - X_{p,t} \log\left((\boldsymbol{AS})_{p,t}\right)$	[2]
Itakura-Saito	$\sum_{p=1}^{P} \sum_{t=1}^{T} X_{p,t}/(\boldsymbol{AS})_{p,t} + \log\left((\boldsymbol{AS})_{p,t}\right)$	[3]
β-divergence	$\sum_{p=1}^{P} \sum_{t=1}^{T} \beta^{-1}(\boldsymbol{AS})_{p,t}^{\beta} - (\beta-1)^{-1} X_{p,t} (\boldsymbol{AS})_{p,t}^{\beta-1}$	[4]

Table 1.1 Examples of discrepancy measures in NMF (with $\beta \in \mathbb{R} \setminus \{0, 1\}$).

1.1.3.2 Independent component analysis

The problem of ICA consists in retrieving unknown statistically independent components of a vector $\boldsymbol{s}(t)$ which are mixed by an unknown linear operator $\boldsymbol{A}$, leading

to an observed vector $\boldsymbol{x}(t)$, at time $t = 1, \ldots, T$. If $\boldsymbol{A}$ is an $R \times R$ invertible matrix, and $\boldsymbol{B} = \boldsymbol{A}^{-1}$ denotes the separating matrix, the likelihood of $\boldsymbol{B}$ is

$$\mathcal{L}(\boldsymbol{B}) = \prod_{t=1}^{T} p_{\boldsymbol{s}(t)}\big(\boldsymbol{B}\boldsymbol{x}(t)\big)|\det \boldsymbol{B}|, \tag{1.6}$$

where $p_{\boldsymbol{s}(t)}$ is the probability density function of the source vector at time t. For simplicity, let us now assume that the vectors $\big(\boldsymbol{s}(t)\big)_{1\leq t\leq T}$ are independent and identically distributed. The maximum likelihood estimator of $\boldsymbol{B}$ is then obtained by minimizing the neg-log-likelihood of $\boldsymbol{B}$, that is

$$f(\boldsymbol{B}) = \sum_{t=1}^{T}\sum_{r=1}^{R} \varphi_r\big(\boldsymbol{B}, \boldsymbol{x}(t)\big) - T\log|\det \boldsymbol{B}|, \tag{1.7}$$

where, for every $r = 1, \ldots, R$, $\varphi_r\big(\boldsymbol{B}, \cdot\big) = -\log p_{s_r}\big([\boldsymbol{B}\cdot]_r\big)$ and p_{s_r} is the probability density function of the r-th component of the source vector. Note that, even if these probability density functions are log-concave, f is a non-convex function due to the last logarithmic term. In addition, several global minimizers may exist, especially when some of the source component probability distributions are equal (permutation ambiguity) or even (sign ambiguity). When the source distributions are unknown, a criterion similar to (1.7) can be minimized where some approximations to the probability distributions of the sources are employed [5].

A commonly used technique in ICA is to perform a prewhitening of the observations which aims at computing a matrix $\boldsymbol{W}$ such that, for every $t = 1, \ldots, T$, $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$ has an identity correlation matrix $\mathbb{E}\{\boldsymbol{y}(t)\boldsymbol{y}(t)^{\mathsf{T}}\}$. We have then $\boldsymbol{s}(t) = \boldsymbol{Q}\boldsymbol{y}(t)$ where $\boldsymbol{Q}$ is a matrix which remains to be estimated. If the sources are assumed to be zero-mean with unit variance, then this matrix is orthogonal and the neg-log-likelihood reduces to

$$\boldsymbol{Q} \mapsto \sum_{t=1}^{T}\sum_{r=1}^{R} \varphi_r\big(\boldsymbol{Q}, \boldsymbol{y}(t)\big) \tag{1.8}$$

This simplification comes at the expense of the orthogonality constraint which needs to be imposed on the sought matrix. The resulting cost function to be minimized on the space of $R \times R$ matrices reads

$$f(\boldsymbol{Q}) = \sum_{t=1}^{T}\sum_{r=1}^{R} \varphi_r\big(\boldsymbol{Q}, \boldsymbol{y}(t)\big) + \iota_{\mathcal{O}}(\boldsymbol{Q}), \tag{1.9}$$

where $\mathcal{O}$ is the non-convex closed set of orthogonal matrices.

1.1.3.3 Tensor decomposition

Source separation problems can often be recast either directly, or after some preprocessing (e.g. by computing higher-order cumulants of the observations), under a tensorial form [6]. One important problem in such tensor analysis is the Canonical

Polyadic Decomposition (CPD) which consists in decomposing a tensor $\boldsymbol{\mathcal{T}}$ of order D in a sum of R one-rank tensors, i.e.

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \boldsymbol{a}^{(r,1)} \otimes \cdots \otimes \boldsymbol{a}^{(r,d)}, \tag{1.10}$$

where R is the tensor rank and, for every $r = 1, \ldots, R$ and $d = 1, \ldots, D$, $\boldsymbol{a}^{(r,d)}$ is a P_d-dimensional vector. More generally, one may be interested in approximating a tensor $\boldsymbol{\mathcal{T}}$ by a sum corresponding to the right-hand side summation in (1.10) where R is then less than or equal to the tensor rank. Then, it may be useful to introduce constraints on the sought vectors $(\boldsymbol{a}^{(r,d)})_{1\leq r\leq R, 1\leq d\leq D}$, which are related to properties of the underlying physical model. For example, the components of these vectors can sometimes be assumed to belong to $\mathbb{R}_+$. Due to possible symmetry properties, some of these vectors can also be constrained to be equal. Finally, to limit indeterminacies, some of these vectors can be normalized by setting their norms to 1. All these constraints, mean that $\boldsymbol{a} = (\boldsymbol{a}^{(r,d)})_{1\leq r\leq R, 1\leq d\leq D}$ has to belong to a given constraint set C. The resulting constrained CPD problem can therefore be reformulated as the minimization of

$$f(\boldsymbol{a}) = \Phi\Big(\boldsymbol{\mathcal{T}} - \sum_{r=1}^{R} \boldsymbol{a}^{(r,1)} \otimes \cdots \otimes \boldsymbol{a}^{(r,d)}\Big) + \iota_C(\boldsymbol{a}), \tag{1.11}$$

where Φ is a given distance measure. The latter function is often chosen equal to the squared Frobenius norm. Because of the multilinearity of model (1.10), even if Φ is a convex function and C is a convex set, the resulting optimization problem is non convex.

1.1.4 Chapter outline

In Section 1.2, we will introduce the Majorization-Minimization (MM) principle which will be our main guideline for all the approaches presented in this chapter. General recipes for building a good surrogate majorant to a given cost function will be discussed. Then, we will focus on quadratic MM methods, which will be shown to cover a wide range of existing algorithms for minimizing both smooth functions (e.g. MM subspace algorithms) and non smooth ones (e.g. the Variable Metric Forward-Backward algorithm). In the last part of this section, we will present block-coordinate approaches which now play a prominent role in the solution of large-scale optimization problems. Section 1.3 will be dedicated to primal-dual algorithms which have gained much popularity in the last years. These methods will be introduced from the viewpoint of Lagrange duality. Both proximal approaches and interior point methods will be described. We conclude this chapter by presenting in Section 1.4 an example of signal restoration, arising in spectroscopy, and various optimization approaches to resolve it.

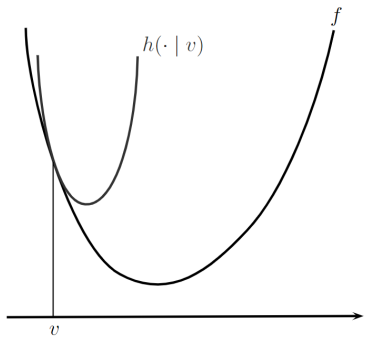

Figure 1.4 Graph of function f and a majorant function $h(\cdot \mid v)$ of f at v

1.2 Majorization-Minimization approaches

Majorization-Minimization (MM) [7], also known as optimization transfer [8], auxiliary function method [9], quadratic approximation [10], surrogate minimization [11], successive upper-bound minimization [12], is at the core of many optimization algorithms for large scale signal/image processing. In particular, it appears that most of the optimization methods that are useful to solve source separation problems (gradient descent schemes, NMF algorithms, proximal methods, ...) can be rewritten under this generic framework, which motivates us for making a detailed focus on MM algorithms, along with their convergence guarantees, and their applicability on specific examples inspired from the context of source separation.

1.2.1 Majorization-Minimization principle

The main idea behind MM methods lies in the concept of *majorant function*, defined below.

DEFINITION 1.1 *Let* $f : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$. *Let* $\boldsymbol{v} \in \mathbb{R}^N$. $h(\cdot|\boldsymbol{v}) : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ *is a majorant function of* f *at* $\boldsymbol{v}$ *if*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \qquad f(\boldsymbol{u}) \leq h(\boldsymbol{u}|\boldsymbol{v}), \qquad \textit{and} \qquad f(\boldsymbol{v}) = h(\boldsymbol{v}|\boldsymbol{v}).$$

The MM algorithm consists of solving the minimization problem

$$\underset{\boldsymbol{u} \in \mathbb{R}^N}{\text{minimize}}\ f(\boldsymbol{u}), \tag{1.12}$$

by alternating between two steps:

1) **M**ajorize the criterion at current iterate with a *majorant function*,
2) **M**inimize the majorant function to define the next iterate.

A simple instance of MM algorithm for minimizing function f over $\mathbb{R}^N$ reads:

$$\begin{array}{l}\text{For } k = 0, 1, \ldots \\ \left\lfloor \boldsymbol{u}_{k+1} \in \underset{\boldsymbol{u}\in\mathbb{R}^N}{\text{Argmin}} \; h(\boldsymbol{u}|\boldsymbol{u}_k) \right.\end{array} \tag{1.13}$$

where $h(\cdot|\boldsymbol{u}_k)$ is a majorant function for f at $\boldsymbol{u}_k$. By construction, we have the following inequalities:

$$(\forall k \in \mathbb{N}) \quad f(\boldsymbol{u}_{k+1}) \underset{\text{Maj.}}{\leq} h(\boldsymbol{u}_{k+1}|\boldsymbol{u}_k) \underset{\text{Min.}}{\leq} h(\boldsymbol{u}_k|\boldsymbol{u}_k) = f(\boldsymbol{u}_k).$$

We thus see that the sequence $(f(\boldsymbol{u}_k))_{k\in\mathbb{N}}$ is decreasing. The construction of an MM algorithm requires to define a strategy for building majorant functions, as well as a strategy for minimizing (or at least, decreasing) them, with the aim to guarantee the convergence of the sequence $(\boldsymbol{u}_k)_{k\in\mathbb{N}}$ to a solution to (1.12).

1.2.2 Majorization techniques

In this section, we recall some important properties from [8, 11] that may be useful to construct a suitable majorant function for a given criterion f.

PROPERTY 1.1 *Let $f_1 : \mathbb{R}^N \to \mathbb{R}\cup\{+\infty\}$, $f_2 : \mathbb{R}^N \to \mathbb{R}\cup\{+\infty\}$ and $\boldsymbol{v} \in \mathbb{R}^N$. Assume that $h_1(\cdot|\boldsymbol{v}) : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is a majorant function of f_1 at $\boldsymbol{v}$, and $h_2(\cdot|\boldsymbol{v}) : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is a majorant function of f_2 at $\boldsymbol{v}$. Then,*

(i) *$h_1(\cdot|\boldsymbol{v}) + h_2(\cdot|\boldsymbol{v})$ is a majorant function of $f_1 + f_2$ at $\boldsymbol{v}$.*
(ii) *If, for all $\boldsymbol{u} \in \mathbb{R}^N$, $f_1(\boldsymbol{u}) \geq 0$ and $f_2(\boldsymbol{u}) \geq 0$, then $h_1(\cdot|\boldsymbol{v})h_2(\cdot|\boldsymbol{v})$ is a majorant function of $f_1 f_2$ at $\boldsymbol{v}$.*
(iii) *If $\phi : \mathbb{R} \to \mathbb{R}\cup\{+\infty\}$ is an increasing function, then $\phi(h_1(\cdot|\boldsymbol{v}))$ is a majorant function of $\phi(f_1)$ at $\boldsymbol{v}$.*

The definition of the subdifferential in (1.4) yields the following useful property.

PROPERTY 1.2 *Let $f : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ and $\boldsymbol{v} \in \mathbb{R}^N$. If f is concave (i.e., $-f$ is convex) on $\mathbb{R}^N$, then a majorant function for f at $\boldsymbol{v}$ is*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f(\boldsymbol{v}) + \boldsymbol{t}^\mathsf{T}(\boldsymbol{u} - \boldsymbol{v}),$$

where $(-\boldsymbol{t}) \in \partial(-f)(\boldsymbol{v})$, where ∂f denotes the subdifferential of f.

Note that the above Property 1.2 is at the core of DC (difference of convex) programming algorithms for minimizing the difference of two convex functions [13]. At each iteration of such methods, the concave part of the criterion is majorized at the current point by a linear function while the convex part is majorized by a simpler convex (typically quadratic) function. The resulting majorant function is then minimized, according to an MM scheme.

Quadratic majorant functions can be derived, by making use of second-order properties of f.

PROPERTY 1.3 **(i)** *Assume that f is β-Lipschitz differentiable on $\mathbb{R}^N$, i.e. there exists $\beta > 0$ such that*

$$(\forall(\boldsymbol{u},\boldsymbol{v}) \in \mathbb{R}^N \times \mathbb{R}^N) \quad \|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{u})\| \leq \beta\|\boldsymbol{u}-\boldsymbol{v}\|.$$

Then a majorant function for f at $\boldsymbol{v}$ is

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f(\boldsymbol{v}) + \big(\nabla f(\boldsymbol{v})\big)^{\mathsf{T}}(\boldsymbol{u}-\boldsymbol{v}) + \frac{\mu}{2}\|\boldsymbol{u}-\boldsymbol{v}\|^2,$$

where $\mu \in [\beta, +\infty[$.

(ii) *If f is twice differentiable on $\mathbb{R}^N$ with Hessian $\nabla^2 f$, then a majorant function for f at $\boldsymbol{v}$ is*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f(\boldsymbol{v}) + \big(\nabla f(\boldsymbol{v})\big)^{\mathsf{T}}(\boldsymbol{u}-\boldsymbol{v}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_{\boldsymbol{A}}^2,$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is a positive semidefinite matrix such that, for every $\boldsymbol{u} \in \mathbb{R}^N$, $\boldsymbol{A} - \nabla^2 f(\boldsymbol{u})$ is positive semidefinite and $\|\boldsymbol{u}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u}}$ denotes the weighted Euclidean norm of $\boldsymbol{u}$ within the metric induced by $\boldsymbol{A}$.

EXAMPLE 1.4 *Consider Problem (1.12), where f is β-Lipschitz differentiable on $\mathbb{R}^N$. Then, Property 1.3(i) holds, and, for this particular choice of majorizing function (by setting $\mu = \gamma^{-1}$), the MM algorithm (1.13) reads:*

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \gamma \in]0, \beta^{-1}[\\ \textit{For } k = 0, 1, \ldots \\ \lfloor \ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \gamma \nabla f(\boldsymbol{u}_k) \end{array} \tag{1.14}$$

The above algorithm identifies with the steepest descent algorithm *(or* gradient algorithm*) with fixed stepsize. Its convergence properties can be found in [14], for the over-relaxed case when $\gamma \in]0, 2\beta^{-1}[$. The advantage of this method is its simplicity, but this may come at the expense of a slow convergence rate. The convergence speed of the steepest descent algorithm can be much improved by introducing a variable stepsize associated to a linesearch strategy [14, Chap.3]. More sophisticated strategies, based on preconditioning, or subspace acceleration will be discussed in the next section.*

EXAMPLE 1.5 *Consider Problem (1.12), where f is a convex lsc function on $\mathbb{R}^N$. For all $\boldsymbol{v} \in \mathbb{R}^N$, a strongly convex majorant function of f can be derived by setting*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_{\boldsymbol{A}}^2,$$

with $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ a positive definite matrix. The minimizer of the later majorant function is unique, and defines the so-called proximity operator *of f at $\boldsymbol{v}$ within the metric induced by $\boldsymbol{A}$:*

$$\mathrm{prox}_f^{\boldsymbol{A}}(\boldsymbol{v}) = \underset{\boldsymbol{u}\in\mathbb{R}^N}{\mathrm{argmin}} f(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_{\boldsymbol{A}}^2 \tag{1.15}$$

with the simplified notation $\mathrm{prox}_f^{\boldsymbol{I}} \equiv \mathrm{prox}_f$. *The proximity operator benefits from many interesting properties [15]*[1]. *In particular, if* $\widehat{\boldsymbol{u}}$ *is a minimizer of* f, *we have the fixed-point relation:*

$$\widehat{\boldsymbol{u}} = \mathrm{prox}_f^{\boldsymbol{A}}(\widehat{\boldsymbol{u}}),$$

which is at the core of the convergence analysis of the proximal point algorithm *given below:*

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \textit{For } k = 0, 1, \ldots \\ \lfloor \ \boldsymbol{u}_{k+1} = \mathrm{prox}_f^{\boldsymbol{A}}(\boldsymbol{u}_k). \end{array} \tag{1.16}$$

Let $\boldsymbol{A} = \gamma^{-1}\boldsymbol{I}$, *with* $\gamma \in]0, +\infty[$. *According to the definition of the proximity operator, the main iteration* (1.16) *can be rewritten as:*

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \gamma \boldsymbol{t}_k, \qquad \textit{with} \qquad \boldsymbol{t}_k \in \partial f(\boldsymbol{u}_{k+1}), \tag{1.17}$$

which (1.17) *can be viewed as an implicit subgradient scheme. The main advantage of this implicit scheme is that it allows the stepsize* γ *to be constant along the iterations, whereas the step-size is required to decrease to zero in the case of the standard (explicit) subgradient algorithm [16].*

EXAMPLE 1.6 *Let us consider the resolution of Problem* (1.12) *where*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{z}\|_2^2 + \lambda \sum_{n=1}^{N} \log(\eta + |u_n|) \tag{1.18}$$

with $\boldsymbol{H} \in \mathbb{R}^{M \times N}$, $\boldsymbol{z} \in]0, +\infty[^M$, $(\lambda, \eta) \in]0, +\infty[^2$. *Function* $u \mapsto \log(\eta + u)$ *is concave on* $[0, +\infty[$, *so according to Property 1.2, for every* $\boldsymbol{v} \in \mathbb{R}^N$, *a tangent majorant function of* (1.18) *is*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = \frac{1}{2}\|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{z}\|_2^2 + \lambda \sum_{n=1}^{N} \frac{|u_n|}{\eta + |v_n|} + C(\boldsymbol{v}) \tag{1.19}$$

where $C(\boldsymbol{v})$ *is a constant such that* $h(\boldsymbol{v}|\boldsymbol{v}) = f(\boldsymbol{v})$. *The application of the MM algorithm to the majorizing approximation* (1.19) *leads to the so-called* iterative reweighted ℓ_1 *(IRL1) algorithm for sparse signal reconstruction [17, 18, 19, 20]. This name comes from the fact that function* (1.19) *reads as a least squares term, penalized by a weighted* ℓ_1 *norm, whose weights depend on the entries of* $\boldsymbol{v}$.

The following property is at the core of iterative reweighted least-squares algorithms [21, 22, 23, 24, 25], and of half quadratic methods [26, 27, 28, 29, 30, 31, 32] for the minimization of penalized quadratic functions.

PROPERTY 1.7 *Let* $f : \mathbb{R} \to \mathbb{R}$ *be a continuous function such that, for every* $u \in \mathbb{R}$, $f(u) = \psi(|u|)$, *where:*

1) See also `http://proximity-operator.net`

(i) *ψ is differentiable on* $]0, +\infty[$,
(ii) *$\psi(\sqrt{\cdot})$ is concave on* $]0, +\infty[$,
(iii) *ψ is increasing on* $]0, +\infty[$.

Then, for all $v \in \mathbb{R}^*$,

$$(\forall u \in \mathbb{R}) \quad f(u) \leq f(v) + \dot{f}(v)(u - v) + \frac{1}{2}\omega(|v|)(u - v)^2$$

where, for every $\xi \in]0, +\infty[$, $\omega(\xi) = \dot{\psi}(\xi)/\xi$ and $\dot{f}$ denotes the derivative of the scalar function f.

EXAMPLE 1.8 *Let us focus on the search for a minimizer on $\mathbb{R}^N$ of the non smooth function $f : \mathbb{R}^N \to \mathbb{R}$ defined as:*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \sum_{s=1}^{S} \|\boldsymbol{L}_s \boldsymbol{u} - \boldsymbol{c}_s\| \tag{1.20}$$

with $(\forall s \in \{1, \ldots, S\})$ $\boldsymbol{L}_s \in \mathbb{R}^{P_s \times N}$, $\boldsymbol{c}_s \in \mathbb{R}^{P_s}$. Applying Property 1.7 with f the absolute value function, yields, for every $v \in \mathbb{R}^*$,

$$(\forall u \in \mathbb{R}) \quad |u| \leq |v| + \operatorname{sign}(v)(u - v) + \frac{1}{2|v|}(u - v)^2 = \frac{u^2}{2|v|} + \frac{|v|}{2}.$$

A direct consequence is that, for all $\boldsymbol{v}$ belonging to the set

$$\mathcal{C} = \left\{ \boldsymbol{v} \in \mathbb{R}^N \mid (\forall s \in \{1, \ldots, S\})\, \boldsymbol{L}_s \boldsymbol{v} - \boldsymbol{c}_s \neq 0 \right\},$$

the quadratic function

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = \frac{1}{2} \sum_{s=1}^{S} \frac{\|\boldsymbol{L}_s \boldsymbol{u} - \boldsymbol{c}_s\|^2}{\|\boldsymbol{L}_s \boldsymbol{v} - \boldsymbol{c}_s\|} + \|\boldsymbol{L}_s \boldsymbol{v} - \boldsymbol{c}_s\|,$$

is a majorant function for f at $\boldsymbol{v}$. This leads to an MM algorithm for minimizing f usually known as the iterative reweigthed least-squares *(IRLS) algorithm. A difficulty is that IRLS needs to be stopped as soon as an iterate does not belong to $\mathcal{C}$ (since the majorant function would not be defined anymore). This constitutes a difficulty to prove the convergence of the method. In particular, one must ensure that the algorithm does not stopped before reaching the convergence. An analysis of IRLS convergence properties is available in specific cases, namely if for every $s \in \{1, \ldots, S\}$, $\boldsymbol{L}_s = \boldsymbol{I}$, IRLS identifying with the Weiszfeld algorithm [23, 33], or when for every $s \in \{1, \ldots, S\}$, $P_s = 1$, $\boldsymbol{L}_s = \boldsymbol{e}_s^\top$, with $\boldsymbol{e}_s$ the s-th canonical basis vector of $\mathbb{R}^N$, and f contains an additional quadratic term [21]. Note that, even if the iterates $(\boldsymbol{u}_k)_{k \in \mathbb{N}}$ stay in $\mathcal{C}$, numerical issues may arise if, for some k, some component of $(\|\boldsymbol{L}_s \boldsymbol{u}_k - \boldsymbol{c}_s\|)_{1 \leq s \leq S}$ becomes close to zero. A common strategy adopted in the literature to avoid this problem is to replace the expression of the*

majorant function, by a smooth approximation of it, defined on the whole space $\mathbb{R}^N$. *For instance, [34, 35, 12] define, for every* $\boldsymbol{v} \in \mathbb{R}^N$, *the smooth majorant function*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad \widetilde{h}(\boldsymbol{u}|\boldsymbol{v}) = \frac{1}{2}\sum_{s=1}^{S} \frac{\|\boldsymbol{L}_s\boldsymbol{u} - \boldsymbol{c}_s\|^2}{\sqrt{\|\boldsymbol{L}_s\boldsymbol{v} - \boldsymbol{c}_s\|^2 + \eta^2}} + C(\boldsymbol{v}),$$

where $C(\boldsymbol{v})$ *is constant with respect to* $\boldsymbol{u}$ *such that* $f(\boldsymbol{v}) = \widetilde{h}(\boldsymbol{v}|\boldsymbol{v})$, *and* $\eta > 0$ *is a smoothing parameter. It can be easily shown that this modified IRLS algorithm identifies with an MM algorithm for minimizing a smooth approximation to* (1.20), *given by*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad \widetilde{f}(\boldsymbol{u}) = \sum_{s=1}^{S} \sqrt{\|\boldsymbol{L}_s\boldsymbol{u} - \boldsymbol{c}_s\|^2 + \eta^2} - \eta.$$

In practice, parameter η *is chosen close to zero, so that the minimizers of* f *and* $\widetilde{f}$ *almost coincide, while numerical issues are avoided.*

We finish this section by stating a property resulting from Jensen's inequality which is recalled below:

LEMMA 1.9 *Let* $\psi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ *be a convex function and let* $\boldsymbol{\omega} = (\omega_n)_{1\leq n\leq N} \in [0, +\infty[^N$ *be such that* $\sum_{n=1}^{N} \omega_n = 1$. *Then,*

$$(\forall (u_1, \ldots, u_N) \in \mathbb{R}^N) \quad \psi\left(\sum_{n=1}^{N} \omega_n u_n\right) \leq \sum_{n=1}^{N} \omega_n \psi(u_n).$$

PROPERTY 1.10 *Let* $\psi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ *be a convex function.*

(i) $(\forall (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{c}) \in (]0, +\infty[^N)^3) \quad \psi(\boldsymbol{c}^\top \boldsymbol{u}) \leq \sum_{n=1}^{N} \frac{c_n v_n}{\boldsymbol{c}^\top \boldsymbol{v}} \psi\left(\frac{\boldsymbol{c}^\top \boldsymbol{v}}{v_n} u_n\right)$.

(ii) *Let* $\boldsymbol{\omega} \in]0, +\infty[^N$ *such that* $\sum_{n=1}^{N} \omega_n = 1$. *We have*

$$(\forall (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{c}) \in (]-\infty, +\infty[^N)^3)$$
$$\psi(\boldsymbol{c}^\top \boldsymbol{u}) \leq \sum_{n=1}^{N} \omega_n \psi\left(\frac{c_n}{\omega_n}(u_n - v_n) + \boldsymbol{c}^\top \boldsymbol{v}\right).$$

This property is useful in the derivation of several optimization algorithms in the context of signal/image restoration.

EXAMPLE 1.11 *Let us consider the resolution of Problem* (1.12) *where*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \sum_{m=1}^{M} \psi((\boldsymbol{H}\boldsymbol{u})_m; z_m), \tag{1.21}$$

$\boldsymbol{H} \in \mathbb{R}^{M\times N}$, $\boldsymbol{z} \in [0,+\infty[^M$, *and* ψ *is the Kullback-Leibler divergence defined as*

$$(\forall u \in \mathbb{R})(\forall z \in [0,+\infty[) \quad \psi(u;z) = \begin{cases} u - z\log(u), & \text{if} \quad u>0 \text{ and } z>0 \\ 0 & \text{if} \quad u=0 \text{ and } z=0 \\ +\infty, & \text{elsewhere.} \end{cases} \tag{1.22}$$

Function f is convex but its minimizers have no closed form expression. A strategy is then to build a majorant function of f, which is simple to minimize. The Richardson-Lucy algorithm [36] consists in applying the MM method (1.13) where the majorant function is given by

$$(\forall \boldsymbol{u} \in]0,+\infty[^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = \sum_{n=1}^{N} a_n u_n - b_n(\boldsymbol{v})\log(u_n) + c_n(\boldsymbol{v}),$$

with, for every $n \in \{1,\dots,N\}$,

$$\begin{aligned} a_n &= [\boldsymbol{H}^\top \mathbf{1}]_n \\ b_n(\boldsymbol{v}) &= [\boldsymbol{H}^\top(\boldsymbol{z} \oslash \boldsymbol{H}\boldsymbol{v})]_n v_n \\ c_n(\boldsymbol{v}) &= [\boldsymbol{H}^\top(\boldsymbol{z} \odot \log(\boldsymbol{H}\boldsymbol{v}) \oslash \boldsymbol{H}\boldsymbol{v})]_n v_n - [\boldsymbol{H}^\top(\boldsymbol{z} \oslash \boldsymbol{H}\boldsymbol{v})]_n v_n \log(v_n) \end{aligned} \tag{1.23}$$

(consequence of Property 1.10(i)). This majorant function is easy to optimize since it is separable with respect to $\boldsymbol{u}$. Its minimization leads to the following multiplicative update rule:

$$\begin{aligned} &\boldsymbol{u}_0 \in]0,+\infty[^N \\ &\textit{For } k = 0,1,\dots \\ &\quad \textit{For } n = 1,2,\dots,N \\ &\quad\quad u_{n,k+1} = u_{n,k}\frac{[\boldsymbol{H}^\top(\boldsymbol{z}\oslash\boldsymbol{H}\boldsymbol{u}_k)]_n}{a_n}. \end{aligned} \tag{1.24}$$

The convergence of the iterations (1.24) has been studied for instance in [37]. Note that extensions of the Richardson-Lucy algorithm to the case of a penalized KL criterion are available (see, for instance, [38]).

1.2.3 MM quadratic methods

The numerical efficiency of the MM algorithm relies on the use of simple majorant functions providing good approximations to f and whose minimizer is simple to compute. An intuitive choice is thus to choose quadratic majorant functions, which leads to the MM quadratic algorithm [10]. Half-quadratic methods [29, 30, 32, 31] are a particular case of this method.

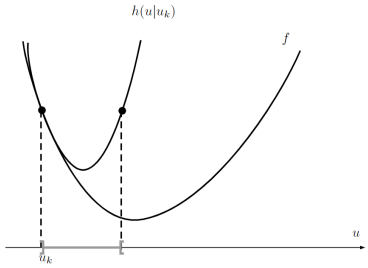

Figure 1.5 MM quadratic algorithm. Function f and a quadratic majorizing function $h(\cdot|u_k)$. The gray interval depicts associated values for u_{k+1} for $\theta_k \in]0,2[$. The choice $\theta_k = 1$ corresponds to choose u_{k+1} as the minimizer of $h(\cdot|u_k)$

1.2.3.1 MM quadratic algorithm

Assume that, for every $\boldsymbol{v} \in \mathbb{R}^N$, there exists a positive definite symmetric matrix $\boldsymbol{A}(\boldsymbol{v}) \in \mathbb{R}^{N\times N}$ such that the quadratic function

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f(\boldsymbol{v}) + (\boldsymbol{u}-\boldsymbol{v})^\mathsf{T}\nabla f(\boldsymbol{v}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|^2_{\boldsymbol{A}(\boldsymbol{v})} \tag{1.25}$$

is a majorant function of f at $\boldsymbol{v}$. The MM quadratic algorithm then reads:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0,1,\ldots \\ \left\lfloor \begin{array}{l} \theta_k \in]0,2[\\ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \theta_k \boldsymbol{A}(\boldsymbol{u}_k)^{-1}\nabla f(\boldsymbol{u}_k). \end{array} \right. \end{array} \tag{1.26}$$

The parameters $(\theta_k)_{k\in\mathbb{N}}$ act as stepsizes. For $\theta_k \equiv 1$, we recover the basic MM algorithm. An illustration of the role of the stepsize is illustrated in the scalar case in Figure 1.5. The simple Algorithm (1.26), which can be viewed as a preconditioned gradient algorithm, for the specific class of preconditioners $(\boldsymbol{A}(\boldsymbol{u}_k)^{-1})_{k\in\mathbb{N}}$ [39, Sec.1.3], enjoys the following convergence properties:

THEOREM 1.12 *Let $f : \mathbb{R}^N \to \mathbb{R}$ be a differentiable function such that $\lim_{\|\boldsymbol{u}\|\to+\infty} f(\boldsymbol{u}) = +\infty$. Assume that there exists $(\underline{\nu},\overline{\nu}) \in]0,+\infty[^2$ such that $(\forall k \in \mathbb{N})$ $\underline{\nu}\boldsymbol{I} \preceq \boldsymbol{A}(\boldsymbol{u}_k) \preceq \overline{\nu}\boldsymbol{I}$,[2] and $(\underline{\theta},\overline{\theta}) \in]0,+\infty[^2$ such that, $(\forall k \in \mathbb{N})$ $\underline{\theta} \leq \theta_k \leq 2-\overline{\theta}$. Then, the following statements hold:*

1) *As $k \to +\infty$, $\nabla f(\boldsymbol{u}_k) \to 0$ and $f(\boldsymbol{u}_k) \searrow f(\widetilde{\boldsymbol{u}})$[3], where $\widetilde{\boldsymbol{u}}$ is a critical point of f.*
2) *If f is convex, any sequential cluster point of $(\boldsymbol{u}_k)_{k\in\mathbb{N}}$ is a minimizer of f.*
3) *If f is strictly convex, then $\boldsymbol{u}_k \to \widehat{\boldsymbol{u}}$ where $\widehat{\boldsymbol{u}}$ is the unique minimizer of f.*

2) We use $\preceq$ to define the Loewner partial ordering on $\mathbb{R}^{N\times N}$: For every $(\boldsymbol{A},\boldsymbol{B}) \in \mathbb{R}^{N\times N}\times\mathbb{R}^{N\times N}$, $\boldsymbol{A} \preceq \boldsymbol{B}$ if and only if, for every $\boldsymbol{u} \in \mathbb{R}^N$, $\boldsymbol{u}^\top\boldsymbol{A}\boldsymbol{u} \leq \boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u}$.
3) We use the shorter notation $\searrow$ to indicate monotonically decreasing convergence.

EXAMPLE 1.13 *Algorithm (1.26) has been applied to the problem of chromatogram baseline correction and noise reduction in [40]. Chromatogram measurements* $\boldsymbol{y} \in \mathbb{R}^N$ *are modeled as* $\boldsymbol{y} = \boldsymbol{u} + \boldsymbol{v} + \boldsymbol{w}$ *with* $\boldsymbol{u} \in \mathbb{R}^N$ *the unknown signal of interest,* $\boldsymbol{v} \in \mathbb{R}^N$ *an unknown smooth background also called baseline and* $\boldsymbol{w} \in \mathbb{R}^N$ *a noise term. The goal is to retrieve* $\boldsymbol{u}$ *under some prior assumptions on* $\boldsymbol{u}$ *and* $\boldsymbol{v}$*. In [40], the authors proposed the following penalized cost function*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{H}(\boldsymbol{y} - \boldsymbol{u})\|^2 + \sum_{i=0}^{M} \lambda_i \phi(\boldsymbol{D}_i \boldsymbol{u}), \tag{1.27}$$

with $\boldsymbol{H}$ *a suitable high-pass filter, and, for every* $i = 0, \ldots, M$*,* $\boldsymbol{D}_i$ *the* i*-th order difference operator,* $\lambda_i > 0$ *and* ϕ *a smooth even potential function. The minimization of* f *is performed by Algorithm (1.26), leading to so-called Baseline Estimation And Denoising with Sparsity (BEADS) method. An illustration of result obtained by BEADS on real chromatogram data is displayed in Figure 1.6.*

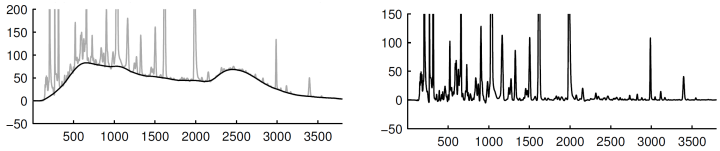

Figure 1.6 Baseline correction using BEADS. Baselines estimated in left and estimated peaks (obtained by subtraction of estimated baseline from data) in right.

EXAMPLE 1.14 *Let us consider the resolution of Problem (1.12) where*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{z}\|^2, \tag{1.28}$$

$\boldsymbol{H} \in \mathbb{R}^{M \times N}$*,* $\boldsymbol{z} \in \mathbb{R}^M$*. According to Property 1.10(ii), for every* $\boldsymbol{v} \in \mathbb{R}^N$*, a separable majorant function of* f *at* $\boldsymbol{v}$ *is:*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f(\boldsymbol{v}) + \nabla f(\boldsymbol{v})^\mathsf{T}(\boldsymbol{u} - \boldsymbol{v}) + \frac{1}{2}\sum_{n=1}^{N} a_n (u_n - v_n)^2,$$

with, for every $n \in \{1, \ldots, N\}$*,*

$$a_n = [|\boldsymbol{H}|^\mathsf{T}|\boldsymbol{H}|\,\boldsymbol{1}]_n,$$

where $|\boldsymbol{H}|$ *denotes a matrix with same size than* $\boldsymbol{H}$*, whose entries are equal to the absolute value of those of* $\boldsymbol{H}$*. In that case, Algorithm (1.30) becomes equivalent to the* simultaneous algebraic reconstruction technique *(SART) [41] for tomography reconstruction:*

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \textit{For } k = 0, 1, \ldots \\ \quad \textit{For } n = 1, 2, \ldots, N \\ \quad\quad u_{n,k+1} = u_{n,k} - \frac{[\boldsymbol{H}^\mathsf{T}(\boldsymbol{H}\boldsymbol{u}_k - \boldsymbol{z})]_n}{a_n}. \end{array} \tag{1.29}$$

Note that the separable form of the quadratic majorant function in that case leads to a fully parallelizable algorithm, in the sense that each component of $\boldsymbol{u}$ is updated in a parallel manner. Generalizations of the SART method to the minimization of a smoothed regularized least squares criterion can be found for instance in [42, 43].

Remark 1.1 In the context of large scale optimization, the exact inversion of the majorant matrix involved at each iteration of Algorithm (1.26) can be computationally intensive. In practice, one can instead solve approximately the associated linear system using a linear solver, such as the conjugate gradient algorithm. A nice feature of Algorithm (1.26) is that its convergence properties are not modified if we replace the update by

$$\begin{array}{l} \text{For } k = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \theta_k \in]0, 2[\\ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \theta_k \boldsymbol{d}_k. \end{array} \right. \end{array} \tag{1.30}$$

where $\boldsymbol{d}_k$ results from $J_k \geq 1$ iterations of a conjugate gradient algorithm (possibly preconditioned) applied to the system $\boldsymbol{A}(\boldsymbol{u}_k)\boldsymbol{d} = \nabla f(\boldsymbol{u}_k)$ [44]. Surprisingly, the best performance in practice for the modified Algorithm (1.30) are achieved for small values of J_k (typically, $J_k = 5$).

Remark 1.2 Let us consider the minimization of a twice differentiable, strictly convex function $f : \mathbb{R}^N \to \mathbb{R}$, by using the *Newton algorithm*:

$$\begin{array}{l} \text{For } k = 0, 1, \ldots \\ \left\lfloor \; \boldsymbol{u}_{k+1} = \boldsymbol{u}_k - (\nabla^2 f(\boldsymbol{u}_k))^{-1} \nabla f(\boldsymbol{u}_k). \right. \end{array} \tag{1.31}$$

The strict convexity of f is however not sufficient to guarantee the convergence of Algorithm (1.31), nor to ensure the monotonicity of the sequence $(f(\boldsymbol{u}_k))_{k\in\mathbb{N}}$, which makes the Newton method with unit stepsize inadequate for general use [10]. The convergence of the algorithm can actually be ensured, according to Theorem 1.12, if, for every $\boldsymbol{u} \in \mathbb{R}^N$, the following majorization holds:

$$f(\boldsymbol{u}) \leq f(\boldsymbol{v}) + (\boldsymbol{u} - \boldsymbol{v})^{\mathsf{T}} \nabla f(\boldsymbol{v}) + \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{v}\|^2_{\nabla^2 f(\boldsymbol{v})}. \tag{1.32}$$

However, this assumption is rarely satisfied in practice. Three practical strategies to secure the convergence of Algorithm (1.31) are discussed in [14, Chap.6],[39, Sec.1.4]: (i) the introduction of a backtracking linesearch ensuring that the Newton step satisfies some sufficient decrease condition, (ii) the addition of a definite positive matrix, possibly varying along the iterations, to the Hessian matrix in order to satisfy a majorizing condition similar to (1.32), or (iii) the combination of the previous technique with a trust region approach that check the majorization inequality only within a neighborhood of the current iterate.

1.2.3.2 Half-quadratic algorithms

Half-quadratic methods [26, 27, 28, 29, 30, 31, 32] can be interpreted as MM quadratic algorithms for minimizing on $\mathbb{R}^N$ the following class of functions:

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{z}\|^2 + \sum_{s=1}^{S} \psi(\|\boldsymbol{L}_s\boldsymbol{u} - \boldsymbol{c}_s\|) \tag{1.33}$$

with $\boldsymbol{H} \in \mathbb{R}^{M\times N}$, $\boldsymbol{z} \in \mathbb{R}^M$, $\psi : \mathbb{R} \mapsto \mathbb{R}$, and $(\forall s \in \{1, \dots, S\})$ $\boldsymbol{L}_s \in \mathbb{R}^{N\times P_s}$, $\boldsymbol{c}_s \in \mathbb{R}^{P_s}$. Two particular choices for matrix $\boldsymbol{A}(\cdot)$ are employed in this context, depending on the assumptions on ψ.

The first strategy [30], relies on the assumption that ψ has a β-Lipschitzian gradient on $\mathbb{R}$. Let us define

$$(\forall \boldsymbol{v} \in \mathbb{R}^N) \quad \boldsymbol{A}(\boldsymbol{v}) = \boldsymbol{H}^\top\boldsymbol{H} + \mu\boldsymbol{L}^\top\boldsymbol{L}, \quad \mu \geq \beta, \tag{1.34}$$

with $\boldsymbol{L} = \left[\boldsymbol{L}_1^\top \mid \dots \mid \boldsymbol{L}_S^\top\right]^\top \in \mathbb{R}^{P\times N}$, $P = \sum_{s=1}^{S} P_s$. Then, according to Property 1.3(i), (1.25) is a majorant function of (1.33).

The second strategy [29] assumes that ψ is differentiable on $]0, +\infty[$, $\psi(\sqrt{\cdot})$ is concave, $(\forall x \in [0, +\infty[)$, $\dot{\psi}(x) \geq 0$, and $\lim_{\substack{x\to 0\\ x>0}} \left(\omega(x) = \frac{\dot{\psi}(x)}{x}\right) \in \mathbb{R}$. Then, Property 1.7 implies that (1.25) is a majorant function of (1.33), if

$$(\forall \boldsymbol{v} \in \mathbb{R}^N) \quad \boldsymbol{A}(\boldsymbol{v}) = \boldsymbol{H}^\top\boldsymbol{H} + \boldsymbol{L}^\top \operatorname{Diag}\left((\omega(\|\boldsymbol{L}_s\boldsymbol{v} - \boldsymbol{c}_s\|)\mathbf{1})_{1\leq s\leq S}\right)\boldsymbol{L}. \tag{1.35}$$

The later approach leads to a better convergence rate in theory [26, 32], but its computational cost may be higher. Indeed, in this case, the majorant matrix (1.35), as well as its inverse, must be recomputed at each iteration, which can be prohibitive in the context of large scale problems. Figure **??** illustrates on a simple scalar example the two different approaches for half-quadratic majorant construction.

The term *half-quadratic* stems from the fact that the application of the MM quadratic algorithm to the minimization of (1.33), with majorant matrix (1.34) or (1.35) can be written as an alternating minimization algorithm:

$$\begin{array}{l} \text{For } k = 0, 1, \dots \\ \left|\begin{array}{l} \boldsymbol{b}_k \in \underset{\boldsymbol{b}\in\mathcal{D}}{\operatorname{Argmin}}\ \Phi(\boldsymbol{u}, \boldsymbol{b}) \\ \bar{\boldsymbol{u}}_k \in \underset{\boldsymbol{u}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \Phi(\boldsymbol{u}, \boldsymbol{b}) \\ \boldsymbol{u}_{k+1} = \theta_k\bar{\boldsymbol{u}}_k + (1-\theta_k)\boldsymbol{u}_k, \qquad \theta_k \in]0, 2[, \end{array}\right. \end{array} \tag{1.36}$$

where $\Phi : \mathbb{R}^N \times \mathcal{D} \to (\mathbb{R} \cup \{+\infty\}) \times (\mathbb{R} \cup \{+\infty\})$ is such that for every $\boldsymbol{b} \in \mathcal{D}$, $\boldsymbol{u} \mapsto \Phi(\boldsymbol{u}, \boldsymbol{b})$ is a *quadratic* function.

1.2.3.3 Subspace acceleration strategy

In the context of large scale optimization, the minimization of a quadratic majorant function at each iteration may become untractable. The main idea of subspace acceleration is to restrict the minimization space to a subspace spanned by a small

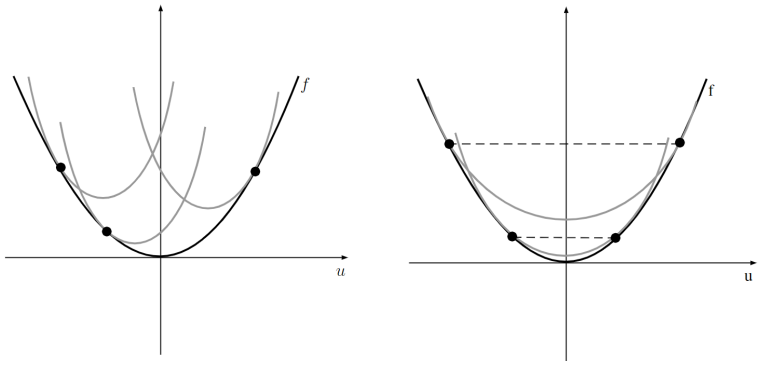

Figure 1.7 Half-quadratic majorant function. Function $f(u) = \sqrt{1+u^2}$ and quadratic majorizing functions $h(\cdot|v)$ for different values of the tangency point v, built according to the strategy (1.34) (left) and (1.35) (right). All the majorizing functions obtained from (1.34) share the same curvature, corresponding to the maximum curvature of f (here, obtained at $u = 0$). In contrast, the majorizing functions resulting from (1.35) are even functions, with curvature depending on the tangency point v.

number of vectors, instead of minimizing the majorant over the whole space $\mathbb{R}^N$. The general form of an MM subspace algorithm is thus

$$(\forall k \in \mathbb{N}) \quad \boldsymbol{u}_{k+1} \in \underset{\boldsymbol{u} \in \operatorname{span}\left\{\boldsymbol{d}_k^1, \boldsymbol{d}_k^2, \ldots, \boldsymbol{d}_k^{M_k}\right\}}{\operatorname{Argmin}} h(\boldsymbol{u}|\boldsymbol{u}_k),$$

where, for every $k \in \mathbb{N}$, $M_k \geq 1$. Equivalently, we obtain:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \text{Choose } \boldsymbol{D}_k \in \mathbb{R}^{N \times M_k}, \\ \boldsymbol{s}_k \in \underset{\boldsymbol{s} \in \mathbb{R}^{M_k}}{\operatorname{Argmin}}\ h\left(\boldsymbol{u}_k + \boldsymbol{D}_k \boldsymbol{s} | \, \boldsymbol{u}_k\right), \\ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \boldsymbol{D}_k \boldsymbol{s}_k, \end{array} \right. \end{array} \tag{1.37}$$

where, for every $k \in \mathbb{N}$, $\boldsymbol{D}_k = \left[\boldsymbol{d}_k^1 \,|\, \boldsymbol{d}_k^2 \,|\, \ldots \,|\, \boldsymbol{d}_k^{M_k}\right] \in \mathbb{R}^{N \times M_k}$ gathers some search directions (see [45, Tab.1] for examples) and $\boldsymbol{s}_k \in \mathbb{R}^{M_k}$ is a multi-dimensional stepsize allowing to optimally combined these directions. By adopting a subspace spanned by a small number of directions, one can expect to reduce the computational cost of the algorithm.

Assume that, for every $\boldsymbol{v} \in \mathbb{R}^N$, there exists a strongly positive self-adjoint operator $\boldsymbol{A}(\boldsymbol{v})$ such that the quadratic function (1.25) is a majorant function of f at $\boldsymbol{v}$. The MM subspace quadratic algorithm reads as follows, where $\dagger$ denotes the

pseudo-inverse operation:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \text{Choose } \boldsymbol{D}_k \in \mathbb{R}^{M_k}, \\ \boldsymbol{s}_k = -\left(\boldsymbol{D}_k^\top \boldsymbol{A}(\boldsymbol{u}_k)\boldsymbol{D}_k\right)^\dagger \left(\boldsymbol{D}_k^\top \nabla f(\boldsymbol{u}_k)\right), \\ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \boldsymbol{D}_k \boldsymbol{s}_k. \end{array} \right. \end{array} \tag{1.38}$$

A convergence result of Algorithm (1.38) similar to Theorem 1.12 has been established in [45], under the assumption that, for every $k \in \mathbb{N}$, the gradient direction $\nabla f(\boldsymbol{u}_k)$ belongs to the vector subspace spanned by the columns of $\boldsymbol{D}_k$. This result has recently been extended to the case of non-convex functions [46], where the convergence of the iterates of Algorithm (1.38) to a critical point of the objective function is proved. Note that Algorithm (1.38) has recently been extended to the online case when only a stochastic approximation of the criterion is employed at each iteration [47].

The MM subspace algorithm shows very good performance in practice, when compared with nonlinear conjugate gradient algorithms [48], low memory BFGS strategy [49], and also with graph-cut based discrete optimization methods, as well as proximal algorithms [45, 50, 51]. All the related works illustrate the fact that the choice of the subspace has a major impact on the practical convergence speed of Algorithm (1.38) (see, for instance [45, Section 5], [46, Section 5.1]). In particular, it seems that the worse performance is obtained in the case of a gradient-like algorithm, i.e., when, for every $k \in \mathbb{N}$, $M_k = 1$ and $\boldsymbol{D}_k = -\nabla f(\boldsymbol{u}_k)$. On the opposite, when the search subspace is the full space, Algorithm (1.38) becomes equivalent to the MM quadratic method mentioned in Section 1.2.3.1. This methods can converge in a very small number of iterations, but each of them requires the inversion of an $N \times N$ matrix, which may have a high computational cost. A good compromise is obtained for the memory gradient subspace [52], spanned by the current gradient and the previous direction, leading to the so-called MM Memory Gradient (3MG) algorithm. These practical observations are confirmed by the theoretical analysis of the rate of convergence of Algorithm (1.38) [53].

1.2.4 Variable metric forward-backward algorithm

Let us now focus on the minimization of $f = f_1 + f_2$ with f_1 differentiable and f_2 convex non necessarily differentiable (for instance, a regularization term enforcing sparsity). Assume that, for every $\boldsymbol{v} \in \mathbb{R}^N$, there exists a positive definite matrix $\boldsymbol{A}(\boldsymbol{v}) \in \mathbb{R}^{N \times N}$ such that the quadratic function

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h(\boldsymbol{u}|\boldsymbol{v}) = f_1(\boldsymbol{v}) + \nabla f_1(\boldsymbol{v})^\top (\boldsymbol{u} - \boldsymbol{v}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|^2_{\boldsymbol{A}(\boldsymbol{v})}$$

is a majorant function of f_1 at $\boldsymbol{v}$. The variable metric forward-backward (VMFB) algorithm [54, 55] reads

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \theta_k \in]0, 2[, \\ \boldsymbol{u}_{k+1} = \operatorname{prox}_{f_2}^{\theta_k^{-1}\boldsymbol{A}(\boldsymbol{u}_k)} \left(\boldsymbol{u}_k - \theta_k \boldsymbol{A}(\boldsymbol{u}_k)^{-1} \nabla f_1(\boldsymbol{u}_k)\right), \end{array} \right. \end{array} \tag{1.39}$$

where the proximity operator has been defined in (1.15). Algorithm (1.39) can actually be viewed as a relaxed form of an MM algorithm. Let $\theta_k \equiv 1$. According to the definition of the proximity operator,

$$\begin{aligned} \boldsymbol{u}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^N} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{u}_k + \boldsymbol{A}(\boldsymbol{u}_k)^{-1}\nabla f_1(\boldsymbol{u}_k)\|^2_{\boldsymbol{A}(\boldsymbol{u}_k)} + f_2(\boldsymbol{u}) \\ &= \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^N} (\boldsymbol{u} - \boldsymbol{u}_k)^\top \nabla f_1(\boldsymbol{u}_k) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{u}_k\|^2_{\boldsymbol{A}(\boldsymbol{u}_k)} + f_2(\boldsymbol{u}) \\ &= \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^N} h(\boldsymbol{u}|\boldsymbol{u}_k) + f_2(\boldsymbol{u}), \end{aligned}$$

which identifies with a MM algorithm to minimize f.

A convergence result similar to Theorem 1.12 can be stated for the VMFB algorithm, under the assumption that f_2 is convex, with non empty domain, and continuous on its domain. An extension of the convergence properties of VMFB in the context of non convex optimization can be found in [55].

Remark 1.3 Assume that f_1 is β-Lipschitz differentiable. According to Property1.3(i), a valid choice is $\boldsymbol{A}(\boldsymbol{u}_k) \equiv \beta^{-1}\boldsymbol{I}$. In that case, VMFB algorithm becomes equivalent to the forward-backward algorithm [56] given below

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \theta_k \in]0, 2\beta^{-1}[, \\ \boldsymbol{u}_{k+1} = \operatorname{prox}_{\theta_k f_2} \left(\boldsymbol{u}_k - \theta_k \nabla f_1(\boldsymbol{u}_k)\right). \end{array} \right. \end{array} \tag{1.40}$$

The *iterative soft thresholding algorithm* (ISTA) [57] is a special case of this algorithm in the context of sparse signal restoration when f_2 is a (possibly weighted) ℓ_1 norm.

Remark 1.4 There exist plenty of properties for the proximity operator, so that a number of functions have a closed form expression for their prox [15][4]. Moreover, if f is a separable function, i.e.

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \sum_{n=1}^{N} f_n(u_n), \tag{1.41}$$

with $(f_n)_{1\leq n\leq N}$ functions defined on $\mathbb{R}$, then

$$\operatorname{prox}_f^{\operatorname{Diag}\{\boldsymbol{a}\}}(\boldsymbol{u}) = \left(\operatorname{prox}_{a_n^{-1} f_n}(u_n)\right)_{1\leq n\leq N}. \tag{1.42}$$

4) See also the web repository http://proximity-operator.net/.

Note that the later property is the cornerstone of the so-called *separable surrogate functionals* algorithm proposed in [42] in the context of sparse signal reconstruction, which identifies with a VMFB algorithm with a diagonal metric.

When the form of the function and/or the metric is more involved, one must resort to an iterative strategy in order to compute the proximity step. Attention must be paid to this problem since the overall computation cost of the optimization method becomes then strongly dependent on the efficiency of the subiterations implemented for computing the proximity operator. A very efficient algorithm for performing the inner loop is the dual forward-backward algorithm [58] (sometimes also known as the dual ascent algorithm) which simply consists in applying the above forward-backward algorithm to the dual of the proximal problem. Note that several recent works [59, 60, 61, 62] have proposed block-alternating, parallel, or distributed versions of this method, in order to reduce its computational and memory cost in the context of large scale optimization problems.

EXAMPLE 1.15 *Assume that*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f_2(\boldsymbol{u}) = \iota_C(\boldsymbol{u}), \tag{1.43}$$

where C is a closed non empty convex subset of $\mathbb{R}^N$. Then, Algorithm (1.39) becomes equivalent to the scaled projected gradient algorithm

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \theta_k \in]0, 2[, \\ \widetilde{\boldsymbol{u}}_k = \boldsymbol{u}_k - \theta_k \boldsymbol{A}(\boldsymbol{u}_k)^{-1} \nabla f_1(\boldsymbol{u}_k), \\ \boldsymbol{u}_{k+1} = \underset{\boldsymbol{v} \in C}{\operatorname{argmin}} \|\boldsymbol{v} - \widetilde{\boldsymbol{u}}_k\|_{\boldsymbol{A}(\boldsymbol{u}_k)}. \end{array} \right. \end{array} \tag{1.44}$$

The convergence of Algorithm (1.44) has been studied, for instance, in [63, 64] under various assumptions on the metric matrices, and the stepsize values. Note that the standard projected gradient [65] is recovered if the preconditioning matrices are scaled versions of the identity matrix.

EXAMPLE 1.16 *The recent work [66, 67] presents an application of the variable metric forward-backward approach in the context of mass spectrometry (MS). MS is a fundamental technology of analytical chemistry. In the context of protein analysis, the MS spectra $\boldsymbol{y} \in \mathbb{R}^M$ can be written as $\boldsymbol{y} = \boldsymbol{K}\boldsymbol{u} + \boldsymbol{w}$ with $\boldsymbol{K} \in \mathbb{R}^{M \times N}$ the MS averagine dictionnary, $\boldsymbol{u} \in \mathbb{R}^N$ a sparse positive-valued signal to be estimated and $\boldsymbol{w}$ a noise term. The dictionary size N is simply equal to $N = MZ$, with M the grid size for the mass and Z the number of charges to be explored. The estimation of $\boldsymbol{u}$ can be performed by minimizing $f = f_1 + f_2$ with f_1 a smooth term favorizing the sparsity of $\boldsymbol{u}$ and $f_2(\boldsymbol{u}) = \iota_{\mathcal{C}}(\boldsymbol{u})$ with*

$$\mathcal{C} = \left\{ \boldsymbol{u} \in \mathbb{R}^N \,|\, \|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{y}\| \leq \xi \text{ and } \boldsymbol{u} \geq 0 \right\} \tag{1.45}$$

where $\xi > 0$ depends of the noise level. The authors in [66, 67] introduced the so-called SPOQ ((Smoothed p-Over-q) penalty:

$$f_1(\boldsymbol{u}) = \log\left(\frac{\left(\ell_{p,\alpha}^p(\boldsymbol{u}) + \beta^p\right)^{1/p}}{\ell_{q,\eta}(\boldsymbol{u})}\right), \tag{1.46}$$

where $\beta > 0$, $\ell_{p,\alpha}$ and $\ell_{q,\eta}$ are smoothed versions of ℓ_p and ℓ_q norms parametrized by $(\alpha, \eta) \in]0, +\infty[^2$. SPOQ penalty consists of a non-convex Lipschitz-differentiable surrogate for ℓ_p-over-ℓ_q quasi-norm/norm ratios, with the ability to enforce sparsity property while preserving the signal scale. An accelerated version of Algorithm (1.39), is then proposed to minimize efficiently the resulting non-convex non-smooth objective function f, where the majorization condition is relaxed, using a trust-region based strategy. Figure 1.8 illustrates the results obtained when using SPOQ procedure to restore synthetic MS spectra.

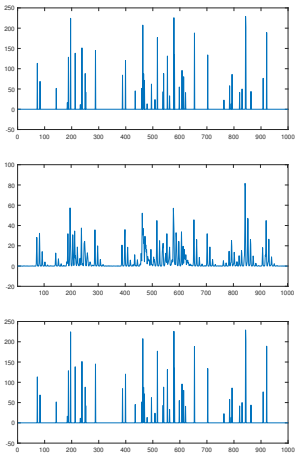

Figure 1.8 Original sparse signal (top), observed measurements (middle) and restored signal (bottom) using SPOQ with $p = 0.75$ and $q = 2$, for a synthetic dataset with $M = 1000$ and mono-charged profile, i.e. $Z = 1$.

1.2.5 Block-coordinate MM algorithms

1.2.5.1 General principle

As emphasized in Section 1.1, most of the objective functions encountered in the context of source separation involve blocks of variables (very often, two blocks) with different significance from the application standpoint, which makes the resulting optimization problems intrinsically adapted to block alternating minimization methods. Such methods can also be very appealing in the context of large scale problems when the whole variable cannot be stored entirely in the local memory of

the machine, so that only update subsets of it can be loaded and updated at each iteration of the algorithm. The aim of this section is to show how MM algorithms can be combined with block alternating schemes.

For every $\boldsymbol{v} \in \mathbb{R}^N$, for every $j \in \{1, \ldots, J\}$, assume that there exists a majorant function $h_j(\cdot|\boldsymbol{v}) : \mathbb{R}^{N_j} \to \mathbb{R}$, which majorizes the restriction of f to its j-th block at $\boldsymbol{v}^{(j)}$, i.e. $h_j(\boldsymbol{v}^{(j)}|\boldsymbol{v}) = f(\boldsymbol{v})$ and,

$$(\forall \boldsymbol{s} \in \mathbb{R}^{N_j}) \qquad h_j(\boldsymbol{s}|\boldsymbol{v}) \leq f(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(j-1)}, \boldsymbol{s}, \boldsymbol{v}^{(j+1)}, \ldots, \boldsymbol{v}^{(J)}).$$

Then, a block-coordinate MM algorithm for minimizing f reads:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \text{Select } j_k \in \{1, \ldots, J\}, \\ \boldsymbol{u}_{k+1}^{(j_k)} = \underset{\boldsymbol{s} \in \mathbb{R}^{N_{j_k}}}{\operatorname{argmin}}\, h_j(\boldsymbol{s}|\boldsymbol{u}_k), \\ \boldsymbol{u}_{k+1}^{(\bar{j}_k)} = \boldsymbol{u}_k^{(j_k)}. \end{array} \right. \end{array} \tag{1.47}$$

where $\bar{j}_k = \{1, \ldots, J\} \setminus \{j_k\}$. When $J = 1$, one recovers the standard MM algorithm (1.13). For $J > 1$, at each iteration $k \in \mathbb{N}$, $j_k \in \{1, \ldots, J\}$ can be chosen according to:

- the *cyclic* rule:
$$(\forall k \in \mathbb{N}) \quad j_k - 1 = k \bmod (J),$$
- a *quasi-cyclic* rule:
There exists a constant $K \geq J$ such that, for every $k \in \mathbb{N}$,
$$\{1, \ldots, J\} \subset \{j_k, \ldots, j_{k+K-1}\},$$
- a *random* rule:
For every $k \in \mathbb{N}$, j_k is a realization of a random variable, for example uniformly distributed on $\{1, \ldots, J\}$.

The convergence properties of the general scheme (1.47), also known as *block successive upper-bound minimization* [12], have been studied in [68, 69] under various assumptions on the function f, on the majorant function sequence, and on the block selection rule. Discussions regarding the practical implementation of Algorithm (1.47) can be found for instance in [70] in the context of image reconstruction and in [11] in the context of machine learning.

1.2.5.2 Block-coordinate quadratic MM algorithm

Let us now describe a block-coordinate version of the quadratic MM algorithm (1.26). For every $\boldsymbol{v} \in \mathbb{R}^N$, for every $j \in \{1, \ldots, J\}$, assume that there exists a symmetric positive definite matrix $\boldsymbol{A}_j(\boldsymbol{v}) \in \mathbb{R}^{N_j \times N_j}$ such that the quadratic function

$$(\forall \boldsymbol{s} \in \mathbb{R}^{N_j})\, h_j(\boldsymbol{s}|\boldsymbol{v}) = f(\boldsymbol{v}) + (\boldsymbol{s} - \boldsymbol{v}^{(j)})^\mathsf{T} \nabla_j f(\boldsymbol{v}) + \frac{1}{2} \|\boldsymbol{s} - \boldsymbol{v}^{(j)}\|^2_{\boldsymbol{A}_j(\boldsymbol{v})}$$

is a majorant function at $\boldsymbol{v}^{(j)}$ of the restriction of f to its j-th block. Then, the block-coordinate MM quadratic algorithm reads:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \text{Select } j_k \in \{1, \ldots, J\}, \\ \theta_k \in]0, 2[, \\ \boldsymbol{u}_{k+1}^{(j_k)} = \boldsymbol{u}_k^{(j_k)} - \theta_k \boldsymbol{A}_{j_k}(\boldsymbol{u}_k)^{-1} \nabla_{j_k} f(\boldsymbol{u}_k), \\ \boldsymbol{u}_{k+1}^{(\bar{j}_k)} = \boldsymbol{u}_k^{(j_k)}. \end{array} \right. \end{array} \tag{1.48}$$

For $J = 1$, we recover Algorithm (1.26). For $J > 1$, the convergence properties of Algorithm (1.48), also known as *coordinate-ascent paraboloidal surrogate* method [70], depend on the block selection rule. For example, in the case of a deterministic quasi-cyclic rule, a convergence result similar to Theorem 1.12 holds, under the assumption that there exists $(\underline{\nu}, \overline{\nu}) \in]0, +\infty[^2$ such that $(\forall k \in \mathbb{N})$ $\underline{\nu}\boldsymbol{I} \preceq \boldsymbol{A}_{j_k}(\boldsymbol{u}_k) \preceq \overline{\nu}\boldsymbol{I}$.

EXAMPLE 1.17 *Let us consider the problem of minimizing the following function:*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{z}\|^2 + \sum_{n=1}^{N} \sqrt{1 + u_n^2}, \tag{1.49}$$

with $\boldsymbol{H} \in \mathbb{R}^{M \times N}$ *and* $\boldsymbol{z} \in \mathbb{R}^M$. *According to Property* (1.7), *for every* $\boldsymbol{v} \in \mathbb{R}^N$,

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad f(\boldsymbol{u}) \leq f(\boldsymbol{v}) + \nabla f(\boldsymbol{v})^\top (\boldsymbol{u} - \boldsymbol{v}) + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{v})^\top \boldsymbol{A}(\boldsymbol{v})(\boldsymbol{u} - \boldsymbol{v}), \tag{1.50}$$

with

$$(\forall \boldsymbol{v} \in \mathbb{R}^N) \quad \boldsymbol{A}(\boldsymbol{v}) = \boldsymbol{H}^\top \boldsymbol{H} + \mathrm{Diag}\{(1 + v_n^2)_{1 \leq n \leq N}\}^{-1/2}. \tag{1.51}$$

Consequently, for every $\boldsymbol{v} \in \mathbb{R}^N$, *a majorant approximation of the restriction of* f *to the* n*-th component of* $\boldsymbol{u}$, *i.e.* $u \mapsto f([v_1, \ldots, v_{n-1}, u, v_n, \ldots, v_N]^\top)$, *is:*

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad h_n(u_n|\boldsymbol{v}) = f(\boldsymbol{v}) + \nabla_n f(\boldsymbol{v})(u_n - v_n) + \frac{a_n(\boldsymbol{v})}{2}(u_n - v_n)^2, \tag{1.52}$$

with

$$(\forall \boldsymbol{v} \in \mathbb{R}^N)(\forall n \in \{1, \ldots, N\}) \quad a_n(\boldsymbol{v}) = [\boldsymbol{H}^\top \boldsymbol{H}]_{n,n} + (1 + v_n^2)^{-1/2}, \tag{1.53}$$

and $\nabla_n f(\boldsymbol{v})$ *the* n*-th component of* $\nabla f(\boldsymbol{u})$. *We can deduce the following alternating algorithm for minimizing* f, *referred to as the* coordinate ascent with parabola

surrogate *in [70]:*

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \textit{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \textit{Select } n \in \{1, \ldots, N\}, \\ \theta_k \in]0, 2[, \\ u_{n,k+1} = u_{n,k} - \theta_k a_n(\boldsymbol{u}_k)^{-1} \nabla_n f(\boldsymbol{u}_k), \\ (\forall n' \in \{1, \ldots, N\} \setminus \{n\}) \quad u_{n',k+1} = u_{n',k}. \end{array} \right. \end{array} \tag{1.54}$$

1.2.5.3 Block-coordinate VMFB algorithm

The VMFB algorithm (1.39) can also be extended to a block alternating version of it. Consider the minimization of $f = f_1 + f_2$ with f_1 differentiable and f_2 convex non necessarily differentiable. Assume additionally that f_2 is block-separable, i.e.

$$(\forall \boldsymbol{u} = (\boldsymbol{u}^{(j)})_{1 \leq j \leq J}) \qquad f_2(\boldsymbol{u}) = \sum_{j=1}^{J} f_{2,j}(\boldsymbol{u}^{(j)}).$$

For every $\boldsymbol{v} \in \mathbb{R}^N$, for every $j \in \{1, \ldots, J\}$, assume that there exists a symmetric positive definite matrix $\boldsymbol{A}_j(\boldsymbol{v}) \in \mathbb{R}^{N_j \times N_j}$ such that the quadratic function

$$(\forall \boldsymbol{u}^{(j)} \in \mathbb{R}^{N_j})\, h_j(\boldsymbol{u}^{(j)} | \boldsymbol{v}) = f_1(\boldsymbol{v}) + (\boldsymbol{u}^{(j)} - \boldsymbol{v}^{(j)})^{\mathsf{T}} \nabla_j f_1(\boldsymbol{v}) + \frac{1}{2} \|\boldsymbol{u}^{(j)} - \boldsymbol{v}^{(j)}\|^2_{\boldsymbol{A}_j(\boldsymbol{v})}$$

is a majorant function at $\boldsymbol{v}^{(j)}$ of the restriction of f_1 to its j-th block. Then, the block-coordinate VMFB (BC-VMFB) algorithm reads:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \text{Select } j_k \in \{1, \ldots, J\}, \\ \theta_k \in]0, 2[, \\ \boldsymbol{u}_{k+1}^{(j_k)} = \operatorname{prox}_{f_{2,j_k}}^{\theta_k^{-1} \boldsymbol{A}_{j_k}(\boldsymbol{u}_k)} \left(\boldsymbol{u}_k^{(j_k)} - \theta_k \boldsymbol{A}_{j_k}(\boldsymbol{u}_k)^{-1} \nabla_{j_k} f_1(\boldsymbol{u}_k) \right), \\ \boldsymbol{u}_{k+1}^{(\bar{j}_k)} = \boldsymbol{u}_k^{(j_k)}. \end{array} \right. \end{array} \tag{1.55}$$

Like its non alternating version, this algorithm can be understood as a relaxed MM algorithm. Its convergence properties have been analyzed in the non convex case in [71]. In the convex case, the convergence properties are the same as for the VMFB algorithm, assuming that the update rule is quasi-cyclic.

EXAMPLE 1.18 *Let us consider the minimization of*

$$\underset{\boldsymbol{A} \in \mathbb{R}^{P \times R}, \boldsymbol{S} \in \mathbb{R}^{R \times T}}{\text{minimize}} F(\boldsymbol{A}, \boldsymbol{S}) = \frac{1}{2} \|\boldsymbol{A}\boldsymbol{S} - \boldsymbol{X}\|_{\mathrm{F}}^2 + \iota_{\mathcal{A}}(\boldsymbol{A}) + \iota_{\mathcal{S}}(\boldsymbol{S}), \tag{1.56}$$

where $\mathcal{A} = [a_{\min}, a_{\max}]^{P \times R}$, *with* $0 < a_{\min} \leq a_{\max}$, *and* $\mathcal{S} = [s_{\min}, s_{\max}]^{R \times T}$, *with* $0 < s_{\min} \leq s_{\max}$. *In order to apply Algorithm* (1.55), *we need to define, for a*

given $(\boldsymbol{A}', \boldsymbol{S}') \in \mathcal{A} \times \mathcal{S}$, quadratic majorants of $F(\cdot, \boldsymbol{S}')$ (resp. $F(\boldsymbol{A}', \cdot)$). Since matrices $\boldsymbol{A}' \in \mathcal{A}$ and $\boldsymbol{S}' \in \mathcal{S}$ have positive elements, we can derive the following quadratic majorant functions (consequence of Lemma 1.10)

$$\begin{aligned}(\forall \boldsymbol{A} \in \mathcal{A}) \quad & F(\boldsymbol{A}, \boldsymbol{S}') \leq F(\boldsymbol{A}', \boldsymbol{S}') + \operatorname{tr}\left((\boldsymbol{A} - \boldsymbol{A}')\nabla_1 F(\boldsymbol{A}', \boldsymbol{S}')^{\mathsf{T}}\right) \\ & + \frac{1}{2}\operatorname{tr}\left(((\boldsymbol{A} - \boldsymbol{A}') \boxdot (\boldsymbol{A}'(\boldsymbol{S}'\boldsymbol{S}'^{\mathsf{T}})) \oslash \boldsymbol{A}')(\boldsymbol{A} - \boldsymbol{A}')^{\mathsf{T}}\right), \end{aligned} \tag{1.57}$$

$$\begin{aligned}(\forall \boldsymbol{S} \in \mathcal{S}) \quad & F(\boldsymbol{A}', \boldsymbol{S}) \leq F(\boldsymbol{A}', \boldsymbol{S}') + \operatorname{tr}\left((\boldsymbol{S} - \boldsymbol{S}')\nabla_2 F(\boldsymbol{A}', \boldsymbol{S}')^{\mathsf{T}}\right) \\ & + \frac{1}{2}\operatorname{tr}\left(((\boldsymbol{S} - \boldsymbol{S}') \boxdot (\boldsymbol{A}'^{\mathsf{T}}\boldsymbol{A}'\boldsymbol{S}') \oslash \boldsymbol{S}')(\boldsymbol{S} - \boldsymbol{S}')^{\mathsf{T}}\right), \end{aligned} \tag{1.58}$$

where $\operatorname{tr}(\cdot)$ denotes the trace operator, and $\boxdot$ (resp. $\oslash$) the component-wise product (resp. division) between matrices of the same size. Then, the BC-VMFB algorithm, using a cyclic rule, reads:

$$\begin{array}{l} \boldsymbol{A}_0 \in \mathcal{A},\ \boldsymbol{S}_0 \in \mathcal{S} \\ \textit{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \widetilde{\boldsymbol{A}}_k = \boldsymbol{A}_k - (\boldsymbol{A}_k \oslash (\boldsymbol{A}_k \boldsymbol{S}_k \boldsymbol{S}_k^{\mathsf{T}})) \boxdot \nabla_1 F(\boldsymbol{A}_k, \boldsymbol{S}_k), \\ \boldsymbol{A}_{k+1} = \mathrm{P}_{\mathcal{A}}\left(\widetilde{\boldsymbol{A}}_{\mathrm{k}}\right) = \min\left(\max\left(\widetilde{\boldsymbol{A}}_{\mathrm{k}}, \mathrm{a}_{\min}\right), \mathrm{a}_{\max}\right), \\ \widetilde{\boldsymbol{S}}_k = \boldsymbol{S}_k - (\boldsymbol{S}_k \oslash (\boldsymbol{A}_{k+1}^{\mathsf{T}} \boldsymbol{A}_{k+1} \boldsymbol{S}_k)) \boxdot \nabla_2 F(\boldsymbol{A}_{k+1}, \boldsymbol{S}_k), \\ \boldsymbol{S}_{k+1} = \mathrm{P}_{\mathcal{S}}\left(\widetilde{\boldsymbol{S}}_{\mathrm{k}}\right) = \min\left(\max\left(\widetilde{\boldsymbol{S}}_{\mathrm{k}}, \mathrm{s}_{\min}\right), \mathrm{s}_{\max}\right). \end{array} \right. \end{array} \tag{1.59}$$

Let us formulate a few remarks about the above algorithm:

- *The projections onto sets $\mathcal{A}$ and $\mathcal{S}$ are here computed in the standard Euclidean metric. This is a direct consequence of the separability of the majorant functions, and of the simple form of the constraint sets. More complex constraints on the sought solution (e.g., sum-to-one, sparsity) can be included at the price of modifying the underlying metric in the projection steps (see for example [72]).*
- *Algorithm (1.59) is reminiscent from the alternating minimization algorithm proposed in [2] in the context of NMF with Euclidean distance, the main difference being the strict positivity constraint on the components of the involved matrices, introduced in the projection step. This constraint allows us to guarantee the convergence of the iterates of the algorithm to a local minimizer of F, according to the convergence theorem from [71].*
- *A cyclic rule has been used in Algorithm (1.59) for updating matrices $\boldsymbol{A}$ and $\boldsymbol{S}$. In fact, the convergence of the algorithm still holds when using a quasi-cyclic rule i.e., when each block is updated at least once every $K \geq 0$ iteration. A particularly interesting strategy is to adopt an inner looping approach, where $\boldsymbol{A}$ (resp. $\boldsymbol{S}$) is updated several times before updating $\boldsymbol{S}$ (resp. $\boldsymbol{A}$). Such a technique has been applied for instance in [73, 74] in the context of blind deconvolution.*

1.3 Primal-dual methods

In this section, we present algorithms making use of primal-dual formulations for dealing with the minimization of a composite cost function of the form

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \qquad f(\boldsymbol{u}) = f_1(\boldsymbol{u}) + f_2(\boldsymbol{C}\boldsymbol{u}) \tag{1.60}$$

where $f_1 \colon \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$, $f_2 \colon \mathbb{R}^M \to \mathbb{R} \cup \{+\infty\}$, and $\boldsymbol{C} \in \mathbb{R}^{M \times N}$. Subsequently, functions f_1 and f_2 will be assumed to be convex so as to facilitate our presentation. The presented algorithms can however be applied to the non convex case, even if convergence guaranties become hazardous.

1.3.1 Lagrange duality

By introducing an auxiliary variable $\boldsymbol{v}$, the minimization of f can be reformulated as the minimization of a separable sum of functions subject to the linear constraint $\boldsymbol{v} = \boldsymbol{C}\boldsymbol{u}$, that is

$$\underset{\boldsymbol{u} \in \mathbb{R}^N,\, \boldsymbol{v} \in \mathbb{R}^M}{\text{minimize}} \; f_1(\boldsymbol{u}) + f_2(\boldsymbol{v}) + \iota_{\{0\}}(\boldsymbol{C}\boldsymbol{u} - \boldsymbol{v}), \tag{1.61}$$

with $\iota_{\{0\}}$ is the indicator function of the singleton $\{0\}$. A well-known strategy in optimization to address constrained problems is to resort to the Lagrange multiplier method. This classical technique makes use of the Lagrange function defined as

$$(\forall \boldsymbol{u} \in \mathbb{R}^N)\big(\forall (\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^M)^2\big) \quad \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = f_1(\boldsymbol{u}) + f_2(\boldsymbol{v}) + \boldsymbol{w}^{\mathsf{T}}(\boldsymbol{C}\boldsymbol{u} - \boldsymbol{v}), \tag{1.62}$$

where the parameter $\boldsymbol{w}$ is the so-called Lagrange multiplier vector. More precisely, it can be shown that under some mild qualification conditions (for example, the intersection of the interior of the domain of f_2 and the image of the domain of f_1 by $\boldsymbol{C}$ is nonempty), if $(\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{w}})$ is a saddle point of $\mathcal{L}$, then $\widehat{\boldsymbol{u}}$ is a minimizer of f. Recall that such a saddle point is defined as follows:

$$(\forall \boldsymbol{u} \in \mathbb{R}^N)\big(\forall (\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^M)^2\big) \quad \mathcal{L}(\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{v}}, \boldsymbol{w}) \leq \mathcal{L}(\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{w}}) \leq \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \widehat{\boldsymbol{w}}). \tag{1.63}$$

It is characterized by the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\left\{ \begin{array}{rcl} -\boldsymbol{C}^{\mathsf{T}}\widehat{\boldsymbol{w}} & \in & \partial f_1(\widehat{\boldsymbol{u}}) \\ \widehat{\boldsymbol{w}} & \in & \partial f_2(\widehat{\boldsymbol{v}}) \\ \boldsymbol{C}\widehat{\boldsymbol{u}} & = & \widehat{\boldsymbol{v}}. \end{array} \right. \tag{1.64}$$

1.3.2 Alternating direction method of multipliers

1.3.2.1 Basic form

The saddle point property of the Lagrangian suggests to find a minimizer of the cost function f by alternating between a minimization step with respect to the primal variables $\boldsymbol{u}$ and $\boldsymbol{v}$, and a maximization step with respect to the dual variable $\boldsymbol{w}$. However, in order to obtain good convergence properties of the resulting algorithm, the Lagrange function must be modified as follows:

$$(\forall \boldsymbol{u} \in \mathbb{R}^N)(\forall (\boldsymbol{v}, \boldsymbol{z}) \in (\mathbb{R}^M)^2) \quad \widetilde{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}) = \mathcal{L}(\boldsymbol{u}, \boldsymbol{v}, \gamma \boldsymbol{z}) + \frac{\gamma}{2} \|\boldsymbol{C}\boldsymbol{u} - \boldsymbol{v}\|^2, \tag{1.65}$$

where, for convenience, we have performed a variable change for the Lagrange multiplier by setting $\boldsymbol{w} = \gamma \boldsymbol{z}$ with $\gamma \in]0, +\infty[$. Due to the additional quadratic term, $\widetilde{\mathcal{L}}$ is called an augmented Lagrangian. It can be noticed that $\widetilde{\mathcal{L}}$ is a majorant function of $\mathcal{L}$, which coincides with it when the constraint $\boldsymbol{v} = \boldsymbol{C}\boldsymbol{u}$ is satisfied. The proposed optimization algorithm then generates a sequence $(\boldsymbol{u}_k, \boldsymbol{v}_k, \boldsymbol{z}_k)_{k \geq 1}$ as follows:

$$\begin{array}{l}
\boldsymbol{v}_0 \in \mathbb{R}^M, \boldsymbol{z}_0 \in \mathbb{R}^M \\
\text{For } k = 0, 1, \ldots \\
\left| \begin{array}{l}
\boldsymbol{u}_{k+1} = \underset{\boldsymbol{u} \in \mathbb{R}^N}{\operatorname{argmin}} \widetilde{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{v}_k, \boldsymbol{z}_k) \\
\boldsymbol{v}_{k+1} = \underset{\boldsymbol{v} \in \mathbb{R}^N}{\operatorname{argmin}} \widetilde{\mathcal{L}}(\boldsymbol{u}_{k+1}, \boldsymbol{v}, \boldsymbol{z}_k) \\
\boldsymbol{z}_{k+1} \text{ such that } \widetilde{\mathcal{L}}(\boldsymbol{u}_{k+1}, \boldsymbol{v}_{k+1}, \boldsymbol{z}_{k+1}) \geq \widetilde{\mathcal{L}}(\boldsymbol{u}_{k+1}, \boldsymbol{v}_{k+1}, \boldsymbol{z}_k).
\end{array} \right.
\end{array} \tag{1.66}$$

If the last step is performed through a gradient ascent step with stepsize $1/\gamma$, we obtain the Alternating Direction Method of Multipliers (ADMM) [75, 76, 77, 78], which reads

$$\begin{array}{l}
\boldsymbol{v}_0 \in \mathbb{R}^M, \boldsymbol{z}_0 \in \mathbb{R}^M \\
\text{For } k = 0, 1, \ldots \\
\left| \begin{array}{l}
\boldsymbol{u}_{k+1} = \underset{\boldsymbol{u} \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2} \left\|\boldsymbol{C}\boldsymbol{u} - \boldsymbol{v}_k + \boldsymbol{z}_k\right\|^2 + \frac{1}{\gamma} f_1(\boldsymbol{u}) \\
\boldsymbol{s}_k = \boldsymbol{C}\boldsymbol{u}_{k+1} \\
\boldsymbol{v}_{k+1} = \operatorname{prox}_{\frac{f_2}{\gamma}} (\boldsymbol{z}_k + \boldsymbol{s}_k) \\
\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \boldsymbol{s}_k - \boldsymbol{v}_{k+1}.
\end{array} \right.
\end{array} \tag{1.67}$$

The convergence of the sequence $(\boldsymbol{u}_k)_{k \in \mathbb{N}}$ to a minimizer of the cost function f is then secured provided that $\boldsymbol{C}$ has full column rank, that is $\boldsymbol{C}^\top \boldsymbol{C}$ is invertible. Note that, by duality arguments, ADMM is strongly related to another famous algorithm in convex optimization, the Douglas-Rachford algorithm [79, 80].

1.3.2.2 Minimizing a sum of more than two functions

In practice, one may be interested in more involved cost functions of the form:

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \qquad f(\boldsymbol{u}) = \sum_{j=1}^{J} f_j(\boldsymbol{C}_j \boldsymbol{u}), \tag{1.68}$$

where, for every $j = 1, \ldots, J$, f_j is a convex function from $\mathbb{R}^{M_j}$ to $\mathbb{R} \cup \{+\infty\}$, and $\boldsymbol{C}_j \in \mathbb{R}^{M_j \times N}$. One might think of a direct extension of ADMM to this context by introducing auxiliary variables $\boldsymbol{v}_j = \boldsymbol{C}_j \boldsymbol{u}$, but one has to be very cautious in doing so, since convergence guaranties may be lost [81]. A better approach may consist in resorting to consensus-like techniques leading to parallel forms of ADMM [82]. Such an efficient algorithm is the Parallel ProXimal Algorithm (PPXA) which was designed in [83]. An extended version of this algorithm (PPXA+) [84] is described next:

$$\begin{array}{l}
(\boldsymbol{y}_{0,j})_{1 \leq j \leq J} \in \mathbb{R}^N, \boldsymbol{u}_0 = (\sum_{j=1}^{J} \boldsymbol{C}_j^\top \boldsymbol{C}_j)^{-1} \sum_{j=1}^{J} \boldsymbol{C}_j^\top \boldsymbol{y}_{0,j} \\
\text{For } k = 0, 1, \ldots \\
\left| \begin{array}{l}
\boldsymbol{v}_{k,j} = \operatorname{prox}_{\gamma f_j}(\boldsymbol{y}_{k,j}), \quad j = 1, \ldots, J \\
\boldsymbol{c}_k = (\sum_{j=1}^{J} \boldsymbol{C}_j^\top \boldsymbol{C}_j)^{-1} \sum_{j=1}^{J} \boldsymbol{C}_j^\top \boldsymbol{v}_{k,j} \\
\boldsymbol{y}_{k+1,j} = \boldsymbol{y}_{k,j} + \lambda(\boldsymbol{C}_j(2\boldsymbol{c}_k - \boldsymbol{u}_k) - \boldsymbol{v}_{k,j}), \quad j = 1, \ldots, J \\
\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \lambda(\boldsymbol{c}_k - \boldsymbol{u}_k),
\end{array} \right.
\end{array} \tag{1.69}$$

where $\gamma \in]0, +\infty[$ and $\lambda \in]0, 2[$ are two parameters of the algorithm. The use of this algorithm requires that $\sum_{j=1}^{J} \boldsymbol{C}_j^\top \boldsymbol{C}_j$ be invertible. One of the key additional advantages of PPXA+ is that many operations can be performed in parallel on J processors.

EXAMPLE 1.19 *An illustration of the great performance of Algorithm (1.69) can be found in [85], in the context of signal restoration from DOSY NMR measurements. The applicative context for this modality is explained in detail in Section 1.4, but we introduce it briefly here for the sake of clarity. We are giving measurements $\boldsymbol{y} \in \mathbb{R}^M$, related to the sought DOSY spectra $\overline{\boldsymbol{u}} \in \mathbb{R}^N$ through the linear relation $\boldsymbol{y} = \boldsymbol{K}\overline{\boldsymbol{u}} + \boldsymbol{w}$, with $\boldsymbol{K} \in \mathbb{R}^{M \times N}$ the observation matrix associated to the DOSY physical model, and $\boldsymbol{w} \in \mathbb{R}^M$ a noise vector. The authors propose to solve the inverse problem of retrieving an estimate of $\overline{\boldsymbol{u}}$ from $\boldsymbol{y}$ and $\boldsymbol{K}$, by minimizing (1.68) where $J = 2$, $\boldsymbol{C}_1 = \boldsymbol{K}$, $f_1 = \iota_{\mathcal{C}}$ with*

$$\mathcal{C} = \left\{ \boldsymbol{z} \in \mathbb{R}^M | \|\boldsymbol{z} - \boldsymbol{y}\| \leq \xi \right\}, \tag{1.70}$$

$\xi > 0$, $\boldsymbol{C}_2 = \boldsymbol{I}_N$ and f_2 is an hybrid penalty combining an entropy term and an ℓ_1 regularization [86]. The application of Algorithm (1.69) leads to the so-called PALMA algorithm, standing for "Proximal Algorithm for L1 combined with MAxent prior". An example of result obtained by PALMA on the DOSY NMR analysis of crude ethanolic plant extract obtained from brown algae is provided in Figure 1.9.

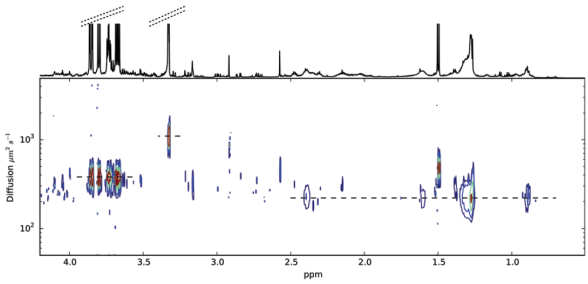

Figure 1.9 Result of PALMA method on an NMR DOSY experiment performed on a brown algae methanol/water extract.

1.3.3 Primal-dual proximal algorithms

It can be observed that the computation of $\boldsymbol{u}_{k+1}$ at the k-th iteration of ADMM is generally non explicit. Even when $J = 2$, PPXA+ does not present this shortcoming, but it requires a matrix inversion for computing each variable $\boldsymbol{c}_k$. When the involved matrices do not have a simple structure and they are of large-size, such inversion has a high computational cost. A number of primal-dual proximal algorithms have been proposed to circumvent this difficulty [87, 88, 89], which are applicable to possibly non smooth optimization problems. A simple way for deriving one of the most popular primal-dual proximal algorithms consists in starting from (1.67) and to replace the update of $\boldsymbol{u}_{k+1}$ by the following semi-implicit subgradient step:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \tau\gamma\Big(\boldsymbol{C}^\mathsf{T}(\boldsymbol{C}\boldsymbol{u}_k - \boldsymbol{v}_k + \boldsymbol{z}_k) + \frac{1}{\gamma}\boldsymbol{t}_k\Big) \tag{1.71}$$

where $\boldsymbol{t}_k$ is a subgradient of f_1 at $\boldsymbol{u}_{k+1}$ and τ is a positive stepsize. This is also equivalent to

$$\boldsymbol{y}_k = \gamma(\boldsymbol{C}\boldsymbol{u}_k - \boldsymbol{v}_k + \boldsymbol{z}_k) \tag{1.72}$$

$$\boldsymbol{u}_{k+1} = \operatorname{prox}_{\tau f_1}\left(\boldsymbol{u}_k - \tau\boldsymbol{C}^\mathsf{T}\boldsymbol{y}_k\right). \tag{1.73}$$

Now reverting the order of the updates of $\boldsymbol{v}_k$ and $\boldsymbol{z}_k$ with respect to the original form of ADMM and performing some algebra lead to

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N, \boldsymbol{y}_0 \in \mathbb{R}^M \\ \text{For } k = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \boldsymbol{u}_{k+1} = \operatorname{prox}_{\tau f_1}\left(\boldsymbol{u}_k - \tau\boldsymbol{C}^\mathsf{T}\boldsymbol{y}_k\right) \\ \boldsymbol{d}_k = \boldsymbol{y}_k + \gamma\boldsymbol{C}(2\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) \\ \boldsymbol{y}_{k+1} = \boldsymbol{d}_k - \gamma \operatorname{prox}_{\frac{f_2}{\gamma}}\left(\frac{\boldsymbol{d}_k}{\gamma}\right). \end{array} \right. \end{array} \tag{1.74}$$

The convergence of $(\boldsymbol{u}_k)_{k\in\mathbb{N}}$ to a minimizer of f can then be shown if $\tau\gamma|||\boldsymbol{C}|||^2 \leq 1$, where we recall that $|||\cdot|||$ denotes the spectral norm. Let us emphasize that no

matrix inversion is required in the previous algorithm. Various extensions of this framework are possible. In particular, other forms of primal-dual proximal algorithms can be obtained [89]. It is also possible to add in the original criterion a Lipschitz-continuous term, which is addressed in the algorithm through its gradient [90, 91, 92, 93]. Variants of this algorithm have been proposed so as to minimize composite function of the form (1.68) [89]. Block-alternating implementations of primal-dual proximal algorithms, along with their convergence properties, can be found for instance in [94, 95].

EXAMPLE 1.20 *The primal-dual proximal algorithm is applied in [96] to mass spectrometry. Following a framework similar to the one of Example 1.16, the authors consider now the ℓ_1 norm penalty function $f_1(\boldsymbol{u}) = \sum_{n=1}^{N} |u_n|$. The resulting optimization problem is then convex non-differentiable, and thus Algorithm 1.74 is particularly well suited. Example of results obtained for the analysis of a real MS dataset measured on a Brucker Solarix 15 T, FT-ICR instrument with an ESI source, is provided in Figure 1.10. The considered sample was constituted of 3 μM of the peptide EVEALEKKVAALESKVQALEKKVEALEHG-NH2 (C140H240N38O45) in its trimer form within 50 mM of NH4OAc, acquired in native conditions. Despite the very large size of the problem, the processing time was of about 108 min on a standard laptop.*

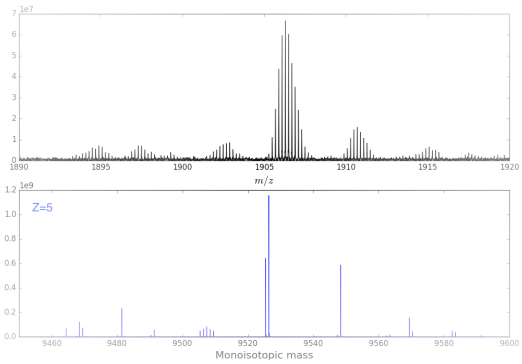

Figure 1.10 Analysis of the real FT-ICR-MS spectrum of a peptide in trimer form: (top) zoom on the acquired data; bottom) recovered spectrum for the charge $z = 5$, using Algorithm 1.74 for a mass grid size of $M = 8130981$ and total charge number of $Z = 5$.

1.3.4
Primal-dual interior point algorithm

Let us consider the optimization problem

$$\underset{\boldsymbol{u}\in\mathcal{C}}{\text{minimize}}\ f_1(\boldsymbol{u}), \tag{1.75}$$

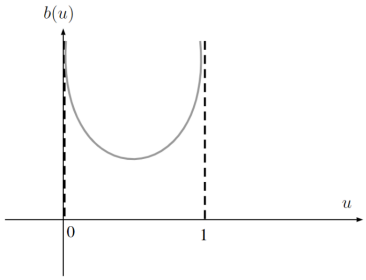

Figure 1.11 Logarithmic barrier function $b(u) = -\log(u) - \log(1-u)$ associated to the constraint $u \in [0,1]$.

subject to the linear equality constraints

$$\mathcal{C} = \{\boldsymbol{u} \in \mathbb{R}^N \mid \boldsymbol{C}\boldsymbol{u} + \boldsymbol{\rho} \in]0,+\infty[^N\}, \tag{1.76}$$

f_1 is a convex twice differentiable function, $\boldsymbol{C} \in \mathbb{R}^{M\times N}$, and $\boldsymbol{\rho} = (\rho_m)_{1\leq m\leq M} \in \mathbb{R}^M$. The existence of a minimizer $\widehat{\boldsymbol{u}}$ will be subsequently assumed. Interior point methods, introduced in [97], solve the constrained optimization problem (1.75) by introducing a sequence of unconstrained optimization subproblems:

$$(\forall k \in \mathbb{N}) \quad \underset{\boldsymbol{u}\in\mathbb{R}^N}{\text{minimize}}\ f_{\mu_k}(\boldsymbol{u}) = f_1(\boldsymbol{u}) + \mu_k\, b(\boldsymbol{u}) \tag{1.77}$$

for positive barrier parameter values $(\mu_k)_{k\in\mathbb{N}}$ decaying to 0. The auxiliary function b, called *barrier function*, penalizes the closeness to the constraint boundaries and maintains the iterates inside the strict interior of the constrained domain. The most widely used auxiliary function is the logarithmic barrier

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad b(\boldsymbol{u}) = f_2(\boldsymbol{C}\boldsymbol{u}), \tag{1.78}$$

where

$$(\forall \boldsymbol{v} = (v_m)_{1\leq m\leq M} \in \mathbb{R}^M)$$
$$f_2(\boldsymbol{v}) = \begin{cases} -\displaystyle\sum_{m=1}^{M} \log(v_m + \rho_m) & \text{if } \boldsymbol{v} \in]0,+\infty[^M \\ +\infty & \text{otherwise.} \end{cases} \tag{1.79}$$

For every $\mu \in]0,+\infty[$, this choice makes the penalized criterion f_μ unbounded at the boundary of the feasible region so that its minimizers $\widehat{\boldsymbol{u}}_\mu$ fulfill strictly the constraints. An example is presented in Figure **??**.

Although various classical interior-point methods can be envisaged to solve Problem (1.77), a primal-dual approach can be followed by reformulating the problem under the form (1.61). The resulting primal-dual interior-point methods simultaneously estimate the primal variables $\boldsymbol{u}$ and the dual Lagrange multiplier vector

$\boldsymbol{\lambda} = -\boldsymbol{w}$ associated with the constraints [98]. The joint estimation of primal and dual variables is performed through KKT optimality conditions (1.64):

$$\left\{\begin{array}{rcl} \nabla f_1(\widehat{\boldsymbol{u}}_\mu) - \boldsymbol{C}^\mathsf{T}\widehat{\boldsymbol{\lambda}}_\mu & = & \boldsymbol{0} \\ \widehat{\boldsymbol{\Lambda}}_\mu(\boldsymbol{C}\widehat{\boldsymbol{u}}_\mu + \boldsymbol{\rho}) & = & \boldsymbol{\mu} \\ \boldsymbol{C}\widehat{\boldsymbol{u}}_\mu + \boldsymbol{\rho} & > & \boldsymbol{0}. \end{array}\right. \tag{1.80}$$

where $\widehat{\boldsymbol{\Lambda}}_\mu = \mathrm{Diag}\{\widehat{\boldsymbol{\lambda}}_\mu\}$ and $\boldsymbol{\mu} = \mu[1,\ldots,1]^\mathsf{T} \in]0,+\infty[^M$.

Numerically, a sequence of optimization problems is solved for a sequence of penalization (also called perturbation) parameters $(\mu_k)_{k\in\mathbb{N}}$ converging to 0. At each iteration $k \in \mathbb{N}$ of the algorithm, a pair of primal-dual variables $(\boldsymbol{u}_{k+1}, \boldsymbol{\lambda}_{k+1})$ is firstly computed from an approximate solution to (1.80) through a Newton algorithm step on the equality conditions, in association with a linesearch strategy on a merit function incorporating some barrier terms, allowing to ensure the inequality condition [9, Chap.11]. The update strategy is then given by

$$(\boldsymbol{u}_{k+1}, \boldsymbol{\lambda}_{k+1}) = (\boldsymbol{u}_k - \alpha_k \boldsymbol{c}_k, \boldsymbol{\lambda}_k - \alpha_k \boldsymbol{d}_k), \tag{1.81}$$

where α_k is the stepsize and $(\boldsymbol{c}_k, \boldsymbol{d}_k)$ are the primal and dual Newton directions. Finally, the perturbation parameter μ_{k+1} is updated in order to ensure the algorithm convergence. Based on the iterative scheme (1.81), several primal-dual interior point methods have been proposed in the literature, each of them calling for its own strategy for the computation of the primal-dual directions, the derivation of a suitable stepsize, and the update of the perturbation parameter (see [98, 99] for a review). We focus here on the iterative scheme introduced in [100, 101]. Note that the algorithm described above is reminiscent from the one from [102], with the introduction of novel tools based on the MM principle aiming at accelerating the practical convergence, as well as reducing the computational cost per iteration.

1.3.4.1 Primal-dual directions

The Newton directions $(\boldsymbol{c}_k, \boldsymbol{d}_k)$ are obtained by solving the linear system

$$\begin{bmatrix} \nabla^2 f_1(\boldsymbol{u}_k) & -\boldsymbol{C}^\mathsf{T} \\ \boldsymbol{\Lambda}_k \boldsymbol{C} & \mathrm{Diag}\{\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}\} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_k \\ \boldsymbol{d}_k \end{bmatrix} = \boldsymbol{r}(\boldsymbol{u}_k, \boldsymbol{\lambda}_k, \mu_k), \tag{1.82}$$

where $\boldsymbol{\Lambda}_k = \mathrm{Diag}\{\boldsymbol{\lambda}_k\}$ and $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\lambda}, \mu)$ is the primal-dual residual defined as

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\lambda}, \mu) = \begin{pmatrix} \boldsymbol{r}^{\mathrm{prim}}(\boldsymbol{u}, \boldsymbol{\lambda}) \\ \boldsymbol{r}^{\mathrm{dual}}(\boldsymbol{u}, \boldsymbol{\lambda}, \mu) \end{pmatrix} = \begin{pmatrix} \nabla f_1(\boldsymbol{u}) - \boldsymbol{C}^\mathsf{T}\boldsymbol{\lambda} \\ \boldsymbol{\Lambda}(\boldsymbol{C}\boldsymbol{u} + \boldsymbol{\rho}) - \boldsymbol{\mu} \end{pmatrix}. \tag{1.83}$$

As pointed out in [103, 104], the primal-dual matrix in the left side of (1.82) suffers from ill-conditioning as soon as some $[\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}]_m$ or some λ_m gets closer to zero. Moreover, this matrix is not guaranteed to be symmetric nor definite positive [99, 105], so that the linear system (1.82) is difficult to solve. Therefore, rather than solving directly (1.82), variable substitution is used [102, 106]. From the second

equation of (1.82), one calculates the dual direction according to

$$\begin{aligned}\boldsymbol{d}_k &= \operatorname{Diag}\{\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}\}^{-1}\big(\boldsymbol{\Lambda}_k(\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}) - \boldsymbol{\mu}_k - \boldsymbol{\Lambda}_k\boldsymbol{C}\boldsymbol{c}_k\big)\\ &= \boldsymbol{\lambda}_k - \operatorname{Diag}\{\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}\}^{-1}\big(\boldsymbol{\mu}_k + \boldsymbol{\Lambda}_k\boldsymbol{C}\boldsymbol{c}_k\big).\end{aligned} \tag{1.84}$$

Then, the primal direction is obtained by solving the linear system

$$\boldsymbol{H}_k\boldsymbol{c}_k = \boldsymbol{g}_k \tag{1.85}$$

with

$$\boldsymbol{H}_k = \nabla^2 f_1(\boldsymbol{u}_k) + \boldsymbol{C}^\mathsf{T}\operatorname{Diag}\{\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}\}^{-1}\boldsymbol{\Lambda}_k\boldsymbol{C} \tag{1.86}$$

$$\boldsymbol{g}_k = \nabla f_1(\boldsymbol{u}_k) - \boldsymbol{C}^\mathsf{T}\operatorname{Diag}\{\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho}\}^{-1}\boldsymbol{\mu}_k. \tag{1.87}$$

The computation cost of the primal-dual interior point algorithm is almost exclusively due to the resolution of the primal system (1.85). It is shown in [107] that an approximate solution of this system is sufficient to ensure the convergence of the method. Several solutions exist to calculate such approximation (see [108] and references therein). For instance, [100] proposes to perform an approximate solution using a preconditioned bi-conjugate gradient algorithm based on an incomplete LU factorization of matrix $\boldsymbol{H}_k$. Moreover, [101] emphasizes that, when $\boldsymbol{H}_k$ has a block-diagonal matrix, the resolution of (1.85) reduces to the resolution of a family of linear systems of small-size and such structure is well suited for parallel computing as the blocks can be processed independently. When $\boldsymbol{H}_k$ is not easily invertible, a separable quadratic majorization of it can be employed, and an MM algorithm can be applied to solve (1.85). More precisely, let us remark that solving (1.85) is equivalent to the resolution of a quadratic minimization problem of the form

$$\underset{\boldsymbol{c}\in\mathbb{R}^M}{\text{minimize}}\ \frac{1}{2}\boldsymbol{c}^\mathsf{T}\boldsymbol{H}_k\boldsymbol{c} - \boldsymbol{g}_k^\mathsf{T}\boldsymbol{c}. \tag{1.88}$$

Let $\boldsymbol{B}_k$ be a symmetric positive definite matrix such that $\boldsymbol{H}_k \preceq \boldsymbol{B}_k$, and whose inverse is simple to compute (for instance, when $\boldsymbol{B}_k$ is a block diagonal matrix). Then, the solution of (1.88) is computed thanks to the following iterations:

$$\begin{aligned}&\boldsymbol{c}_{k,0} = 0,\\ &\text{For } j = 0, 1, \ldots\\ &\quad \boldsymbol{c}_{k,j+1} = \boldsymbol{c}_{k,j} - \boldsymbol{B}_k^{-1}\left(\boldsymbol{H}_k\boldsymbol{c}_{k,j} - \boldsymbol{g}_k\right),\end{aligned} \tag{1.89}$$

until the fulfillment of the following stopping criterion on the primal residual: $\|\boldsymbol{H}_k\boldsymbol{d}_{k,j} - \boldsymbol{g}_k\| \leqslant \mu_k\|\boldsymbol{g}_k\|$ where μ_k is the barrier parameter. The benefit of this method has been illustrated in [109] through an example of spectral unmixing. Note that this MM-like approach is not only faster, it is also better suited for parallel implementation [110].

1.3.4.2 Linesearch

At the k-th iteration, the stepsize value α_k must be chosen so as to ensure the convergence of the algorithm and the fulfillment of the inequalities of the pertubed KKT system (1.80). The convergence study of the primal-dual algorithm presented in [102] requests that α_k ensures a sufficient decrease of the primal-dual merit function Ψ_{μ_k}, defined, for every $\boldsymbol{u} \in \mathbb{R}^N$ and $\boldsymbol{\lambda} \in \mathbb{R}^M$, as

$$\Psi_{\mu_k}(\boldsymbol{u}, \boldsymbol{\lambda}) = f_{\mu_k}(\boldsymbol{u}) + \boldsymbol{\lambda}^{\mathsf{T}}(\boldsymbol{C}\boldsymbol{u} + \boldsymbol{\rho}) - \mu_k \sum_{m=1}^{M} \log(\lambda_m [\boldsymbol{C}\boldsymbol{u} + \boldsymbol{\rho}]_m), \quad (1.90)$$

$$= \mathcal{L}(\boldsymbol{u}, \boldsymbol{C}\boldsymbol{u}, -\boldsymbol{\lambda}) - \mu_k \sum_{m=1}^{M} \log \lambda_m. \quad (1.91)$$

The sufficient decrease is assessed using the Armijo condition,

$$\psi_{\mu_k}(\alpha_k) - \psi_{\mu_k}(0) \leqslant \sigma\, \alpha_k \nabla \psi_{\mu_k}(0) \quad \text{with} \quad \sigma \in]0, 1/2[, \quad (1.92)$$

where, for every $\alpha \in [0, +\infty[$, $\psi_{\mu_k}(\alpha) \triangleq \Psi_{\mu_k}(\boldsymbol{u}_k - \alpha \boldsymbol{c}_k, \boldsymbol{\lambda}_k - \alpha \boldsymbol{d}_k)$. One can note that (1.91) contains two logarithmic barrier functions enforcing the fulfillment of the KKT inequalities, the positivity of the Lagrange multipliers $(\lambda_m)_{1 \leq m \leq M}$ being a straightforward consequence of (1.80). The presence of the barrier function may cause the inefficiency of general purpose backtracking line search methods for finding a stepsize satisfying (1.92) and, thus, the slowdown of the algorithm convergence. Several strategies have been proposed to override this issue [111, 112, 113]. In particular, the majorization-minimization strategy from [112], relying on the construction of a majorizing function made of a quadratic term and a logarithmic barrier term, was shown to lead to good practical performance through numerical applications to 2D nuclear magnetic resonance signal reconstruction under positivity constraints [114], and to sparse signal deconvolution [113].

1.3.4.3 Perturbation parameter update

The convergence analysis of interior point methods requires that the sequence $(\mu_k)_{k \in \mathbb{N}}$ tends to 0 as k tends to infinity. An efficient update strategy for the choice of the barrier parameter is the μ-criticity rule introduced in [115]:

$$(\forall k \in \mathbb{N}) \quad \mu_k = \theta \frac{\delta_k}{M}, \quad (1.93)$$

where $\delta_k = (\boldsymbol{C}\boldsymbol{u}_k + \boldsymbol{\rho})^{\mathsf{T}} \boldsymbol{\lambda}_k$ is the duality gap and $\theta \in]0, 1[$. The barrier parameter is updated as soon as the primal and dual directions fulfill the inner stopping rule ([106, 116]):

$$\|\boldsymbol{r}^{\text{prim}}(\boldsymbol{u}_k, \boldsymbol{\lambda}_k)\|_\infty \leqslant \eta^{\text{prim}} \mu_k \text{ and } \frac{\delta_k}{M} \leqslant \eta^{\text{dual}} \mu_k, \quad (1.94)$$

with $\eta^{\text{prim}} > 0$ and $\eta^{\text{dual}} \in]1, 1/\theta[$.

1.3.4.4 Resulting algorithm

The main steps of the proposed optimization method are summarized below:

$$
\begin{array}{l}
\boldsymbol{u}_0 \in \mathbb{R}^N \quad \text{s.t.} \quad \boldsymbol{C}\boldsymbol{u}_0 + \boldsymbol{\rho} \in]0, +\infty[^M \,,\ \boldsymbol{\lambda}_0 \in]0, +\infty[^M,\ \mu_0 \in]0, +\infty[. \\
\text{For } k = 0, 1, \ldots \\
\quad \left| \begin{array}{l}
\text{If (1.94) holds} \\
\quad \lfloor \ \text{Update} \quad \mu_k \quad \text{according to (1.93)} \\
\text{Calculate} \quad \boldsymbol{c}_k \quad \text{by solving (approximately) (1.85)} \\
\text{Deduce} \quad \boldsymbol{d}_k \quad \text{from (1.84)} \\
\text{Find} \quad \alpha_k > 0 \quad \text{satisfying (1.92)} \\
\text{Update} \quad (\boldsymbol{u}_{k+1}, \boldsymbol{\lambda}_{k+1}) \quad \text{according to (1.81).}
\end{array} \right.
\end{array}
\tag{1.95}
$$

The convergence properties of Algorithm (1.95) have been studied in [102, 107, 101], under the assumption that the linear constraint set is nonempty and bounded. In particular, it is shown that if f_1 is strictly convex, then the sequence $(\boldsymbol{u}_k)_{k\in\mathbb{N}}$ converges to the unique solution of (1.75).

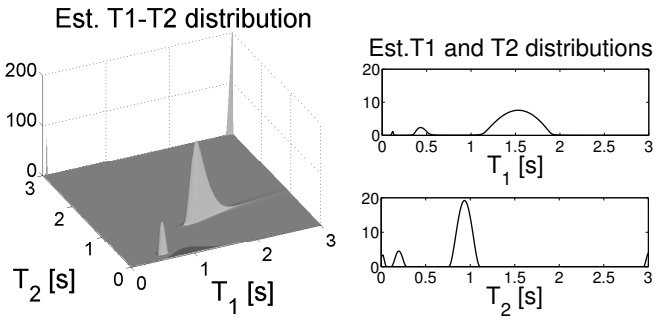

Figure 1.12 Estimated T_1-T_2 NMR distributions from real data (apple) using Algorithm (1.95).

EXAMPLE 1.21 *The primal-dual interior point algorithm* (1.95) *has been applied to two dimensional nuclear magnetic resonance in [114]. Classical NMR experiments analyze the spin relaxation process independently, either in terms of longitudinal or transverse relaxation, leading to the so-called T_1 and T_2 relaxation spectra. Joint measurements with respect to both relaxation parameters allow to build two-dimensional distribution which is of high interest for chemical structure determination [117]. Experimental data consist of a series of discrete noisy samples $\boldsymbol{Y} \in \mathbb{R}^{m_1 \times m_2}$ such that $\boldsymbol{Y} = \boldsymbol{K}_1 \boldsymbol{U} \boldsymbol{K}_2^\top + \boldsymbol{W}$ with $\boldsymbol{U} \in \mathbb{R}^{N_1 \times N_2}$ the sought distribution, $\boldsymbol{K}_1$, $\boldsymbol{K}_2$ matrices associated to the acquisition model, and $\boldsymbol{W}$ a noise term. T_1-T_2 NMR reconstruction aims at estimating $\boldsymbol{U}$ given $\boldsymbol{Y}$, $\boldsymbol{K}_1$, $\boldsymbol{K}_2$. This problem can be solved by minimizing a penalized least-squares term:*

$$
f_1(\boldsymbol{U}) = \frac{1}{2} \|\boldsymbol{K}_1 \boldsymbol{U} \boldsymbol{K}_2^\top - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{U}\|_F^2 \tag{1.96}
$$

with $\lambda > 0$, subject to the positivity constraint $\boldsymbol{U} \geq 0$. Figure 1.12 presents numerical results of 2D spectra obtained from the analysis of an apple matter sample ($m_1 = 50$, $m_2 = 10000$, $N_1 = 300$, $N_2 = 300$), using Algorithm (1.95) [114].

1.4 Illustration in the context of DOSY NMR signal restoration

This section illustrates the applicability of the presented optimization tools for signal restoration in nuclear magnetic resonance (NMR). The measurement of diffusion by NMR is used in various application fields (agroalimentary sector, pharmaceutical industry, ecology) to analyze the properties of complex chemical mixtures in order to determine their molecular structure and dynamics [85, 117, 118]. After the immersion of the matter in a strong magnetic field, all the nuclear spins align to an equilibrium state along the field orientation. The application of a short magnetic pulse, i.e. the pulsed field gradient, in resonance with the spin motion disturbs the spin orientation. NMR aims at analyzing the process which corresponds to the re-establishment of the spin into its equilibrium state. During the DOSY (Diffusion Order SpectroscopY) experiment [119], a series of measurements is acquired for different pulsed field gradient strengths. The data are then analyzed with the aim to separate different species according to their diffusion coefficient.

The DOSY NMR data $\boldsymbol{y} = (y^{(m)})_{1 \leq m \leq M} \in \mathbb{R}^M$ gathers the results of M experiments characterized by a vector $\boldsymbol{t} = (t^{(m)})_{1 \leq m \leq M} \in \mathbb{R}^M$ related to the pulsed field gradient strength and to the acquisition time. The relation between $\boldsymbol{y}$ and $\boldsymbol{t}$ can be expressed as the following Laplace transform:

$$(\forall m \in \{1, \ldots, M\}) \quad y^{(m)} = \int \chi(T) \exp(-t^{(m)} T) dT, \tag{1.97}$$

where $\chi(T)$ is the unknown diffusion distribution. The problem is then to reconstruct $\chi(T)$ on the sampled grid $\boldsymbol{T} = (T^{(n)})_{1 \leq n \leq N}$, from the measurements $\boldsymbol{y}$. After discretization and appropriate normalization, the observation model reads

$$\boldsymbol{y} = \boldsymbol{K}\overline{\boldsymbol{u}} + \boldsymbol{w}, \tag{1.98}$$

where $\boldsymbol{K} \in \mathbb{R}^{M \times N}$ is given by

$$(\forall m \in \{1, \ldots, M\})(\forall n \in \{1, \ldots, N\}) \qquad K^{(m,n)} = \exp(-T^{(n)} t^{(m)}), \tag{1.99}$$

$\overline{\boldsymbol{u}} \in \mathbb{R}^N$ is the sought signal related to $\left(\chi(T^{(n)})\right)_{1 \leq n \leq N}$ (up to a scaling factor depending on the discretization grid), and $\boldsymbol{w} \in \mathbb{R}^M$ is a perturbation noise.

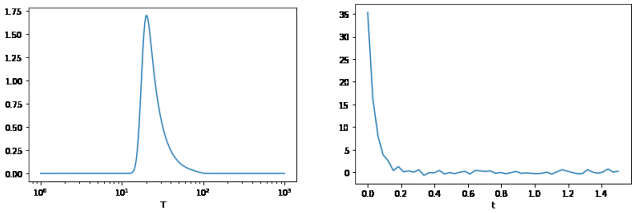

Figure 1.13 Original signal $\overline{\boldsymbol{u}}$ (left) and noisy acquired measure $\boldsymbol{y}$ (right).

We propose here to find an estimate $\widehat{\boldsymbol{u}} \in \mathbb{R}^N$ of $\overline{\boldsymbol{u}}$ by solving the following minimization problem, requiring the knowledge of $\boldsymbol{K}$ and $\boldsymbol{y}$:

$$\underset{\boldsymbol{u}\in\mathbb{R}^N}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{K}\boldsymbol{u}-\boldsymbol{y}\|^2+\beta g(\boldsymbol{u}) \tag{1.100}$$

where $g \in \Gamma_0(\mathbb{R}^N)$ denotes a regularization term and $\beta > 0$ is a regularization parameter. Note that, in practice, a very large number of DOSY NMR acquisitions (typically, 10^4) are conduced for various settings of the pulsed gradient field, so that Problem (1.100) must be solved many times, which motivates the search for a fast minimization algorithm [85]. In the following, we present optimization solutions, for various choices of function g, in the line of the work published in [86]. For illustration purpose, we will consider a test example with $\overline{\boldsymbol{u}}$ of size $N = 256$, and $\boldsymbol{y}$ of size $M = 50$, both represented in Figure 1.13. The noise $\boldsymbol{w}$ is a realization of a zero-mean white Gaussian noise, with standard deviation equals to $10^{-2}y_0 \approx 0.34$. The grid $\boldsymbol{T} = (T^{(n)})_{1\leq n\leq N}$ is sampled following a logarithmic rule, within the interval $[T_{\min}, T_{\max}] = [1, 10^3]\ \mu m^2\, s^{-1}$, while $\boldsymbol{t} = (t^{(m)})_{1\leq m\leq M}$ is regularly sampled on $[t_{\min}, t_{\max}] = [0, 1.5]$ seconds. The quality of the results obtained by the different tested restoration approaches will be assessed quantitatively by means of the normalized mean square error (NMSE):

$$\text{NMSE} = \frac{\|\overline{\boldsymbol{u}}-\widehat{\boldsymbol{u}}\|^2}{\|\overline{\boldsymbol{u}}\|^2}, \tag{1.101}$$

with $\widehat{\boldsymbol{u}}$ the result of each method.

1.4.1 Quadratic penalization

Let us start with the following regularization term serving to promote the reconstruction of smooth signals:

$$(\forall \boldsymbol{u}\in\mathbb{R}^N)\quad g(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{D}\boldsymbol{u}\|^2 \tag{1.102}$$

where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is the discrete gradient operator such that,

$$(\forall n\in\{1,\ldots,N\})\quad [\boldsymbol{D}\boldsymbol{u}]^{(n)} = u^{(n)} - u^{(n-1)} \tag{1.103}$$

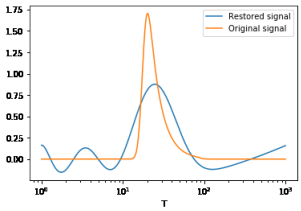

Figure 1.14 Restored signal (1.104) for $\beta = 1$, NMSE $= 0.29$.

with the circular convention $u^{(0)} = u^{(N)}$. In this case, the cost function involved in Problem (1.100) is convex and quadratic. Assuming that $\boldsymbol{K}^\top \boldsymbol{K} + \beta \boldsymbol{D}^\top \boldsymbol{D}$ is invertible, the solution to Problem (1.100) is unique and reads:

$$\widehat{\boldsymbol{u}} = \left(\boldsymbol{K}^\top \boldsymbol{K} + \beta \boldsymbol{D}^\top \boldsymbol{D}\right)^{-1} \boldsymbol{K}^\top \boldsymbol{y}. \tag{1.104}$$

An example of such restored signal is displayed in Figure 1.14.

Now, assume that we want to impose some value range constraints on the sought signal. Then a possible regularization function is:

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad g(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{D}\boldsymbol{u}\|^2 + \iota_{[u_{\min}, u_{\max}]^N}(\boldsymbol{u}) \tag{1.105}$$

with $0 \leq u_{\min} < u_{\max}$ the minimum and maximum values of the original signal $(\overline{u}^{(n)})_{1 \leq n \leq N}$. The quadratic function $\boldsymbol{u} \mapsto 1/2\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{y}\|^2 + \beta/2\|\boldsymbol{D}\boldsymbol{u}\|^2$ is Lipschitz differentiable, with constant $|||\boldsymbol{K}^\top \boldsymbol{K} + \beta \boldsymbol{D}^\top \boldsymbol{D}|||$. Problem (1.100) can thus be solved using the projected gradient algorithm introduced in Example 1.15, which reads in this case:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \alpha_k \in]0, 2/(|||\boldsymbol{K}^\top \boldsymbol{K} + \beta \boldsymbol{D}^\top \boldsymbol{D}|||)[, \\ \widetilde{\boldsymbol{u}}_k = \boldsymbol{u}_k - \alpha_k \left(\boldsymbol{K}^\top (\boldsymbol{K}\boldsymbol{u}_k - \boldsymbol{y}) + \beta \boldsymbol{D}^\top \boldsymbol{D}\boldsymbol{u}_k\right), \\ \boldsymbol{u}_{k+1} = \mathrm{P}_{[u_{\min}, u_{\max}]^N}(\widetilde{\boldsymbol{u}}_\mathrm{k}) = \min\left(\max\left(\widetilde{\boldsymbol{u}}_\mathrm{k}, u_{\min}\right), u_{\max}\right), \end{array} \right. \end{array} \tag{1.106}$$

where the min and max operations are performed componentwise. An example of solution obtained using the above algorithm is displayed in Figure 1.15.

1.4.2 Sparsity prior in the signal domain

Another strategy for regularization is to enforce the sparsity of the sought signal, in addition to the positivity constraint. In that respect, one can use

$$(\forall \boldsymbol{u} = (u_n)_{1 \leq n \leq N} \in \mathbb{R}^N) \quad g(\boldsymbol{u}) = \|\boldsymbol{u}\|_1 + \iota_{[0, +\infty[^N}(\boldsymbol{u}). \tag{1.107}$$

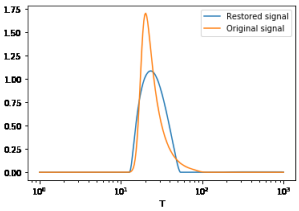

Figure 1.15 Restored signal using Algorithm (1.106) for $\beta = 0.0625$, NMSE $= 0.14$.

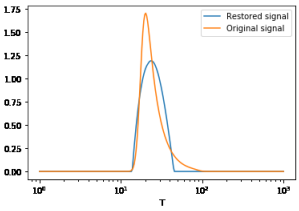

Figure 1.16 Restored signal using Algorithm (1.108) for $\beta = 1$, NMSE $= 0.11$.

Function g is convex, but it is not differentiable. Function $\boldsymbol{u} \mapsto 1/2\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{y}\|^2$ is Lipschitz differentiable with constant $|||\boldsymbol{K}|||^2$. Problem (1.100) can be solved using the forward-backward algorithm presented in Remark 1.3:

$$\begin{array}{l} \boldsymbol{u}_0 \in \mathbb{R}^N \\ \text{For } k = 0, 1, \ldots \\ \left| \begin{array}{l} \alpha_k \in]0, 2/|||\boldsymbol{K}|||^2[, \\ \widetilde{\boldsymbol{u}}_k = \boldsymbol{u}_k - \theta_k \boldsymbol{K}^\top (\boldsymbol{K}\boldsymbol{u}_k - \boldsymbol{y}), \\ \boldsymbol{u}_{k+1} = \operatorname{prox}_{\alpha_k \beta(\|\cdot\|_1 + \iota_{[0,+\infty[^N}}(\widetilde{\boldsymbol{u}}_k) = \max\left(|\widetilde{\boldsymbol{u}}_k| - \alpha_k\beta, 0\right). \end{array} \right. \end{array} \tag{1.108}$$

The convergence rate of the above method can be improved, using a variable metric approach ([55], Section 1.2.4) or a Nesterov-based scheme (see for instance, [120]). An example of solution obtained using Algorithm (1.108) is displayed in Figure 1.16.

1.4.3
Sparsity prior in a transformed domain

It can finally be useful to impose the sparsity of the signal in a transformed domain, in order to impose some regularity properties. For instance, it can be assumed that $\boldsymbol{W}\overline{\boldsymbol{u}}$ is sparse, where $\boldsymbol{W}$ is a (possibly overcomplete) wavelet analysis operator [121]. In this case, we may use

$$(\forall \boldsymbol{u} \in \mathbb{R}^N) \quad g(\boldsymbol{u}) = \|\boldsymbol{W}\boldsymbol{u}\|_1. \tag{1.109}$$

Because of the presence of matrix $\boldsymbol{W}$, the proximal step in the forward-backward algorithm is not explicit anymore. In such context, an efficient strategy is to resort

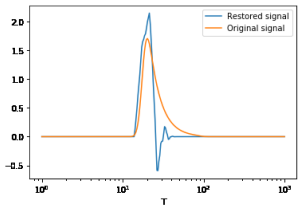

Figure 1.17 Restored signal using Algorithm (1.110) for $\beta = 27$, NMSE $= 0.39$.

to a primal-dual optimization approach as described in Section 1.3. For instance, the primal-dual proximal algorithm (1.74) [89] would read in this case:

$$
\begin{array}{l}
\boldsymbol{u}_0 \in \mathbb{R}^N, \boldsymbol{v}_0 \in \mathbb{R}^M \\
(\tau, \gamma) \text{ such that } \tau\gamma|||\boldsymbol{W}|||^2 \leq 1 \\
\text{For } k = 0, 1, \ldots \\
\left|\begin{array}{l}
\boldsymbol{u}_{k+1} = \operatorname{prox}_{\tau/2\|\boldsymbol{K}\cdot-\boldsymbol{y}\|^2}\left(\boldsymbol{u}_k - \tau\boldsymbol{W}^{\mathsf{T}}\boldsymbol{v}_k\right) \\
\qquad = (\boldsymbol{I} + \tau\boldsymbol{K}^\top\boldsymbol{K})^{-1}(\boldsymbol{u}_k - \tau\boldsymbol{W}^{\mathsf{T}}\boldsymbol{v}_k + \tau\boldsymbol{K}^\top\boldsymbol{y}) \\
\boldsymbol{d}_k = \boldsymbol{v}_k + \gamma\boldsymbol{W}(2\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) \\
\boldsymbol{v}_{k+1} = \boldsymbol{d}_k - \gamma \operatorname{prox}_{\beta\gamma^{-1}\|\cdot\|_1}\left(\frac{\boldsymbol{d}_k}{\gamma}\right) \\
\qquad = \boldsymbol{d}_k - \gamma\operatorname{sign}(\boldsymbol{d}_k) \boxdot \max\left(|\frac{\boldsymbol{d}_k}{\gamma}| - \beta\gamma^{-1}, 0\right).
\end{array}\right.
\end{array}
\tag{1.110}
$$

An example of solution obtained using Algorithm (1.110), where $\boldsymbol{W}$ represents the Symlet orthonormal wavelet analysis operator, with level 3, is displayed in Figure 1.17.

1.4.4 Sparsity prior and range constraints

Let us end this section by focusing on the case when the regularization term includes both sparsity constraints (possibly in a transformed domain), and value range constraints on the sought signal. The problem then becomes:

$$
\underset{\boldsymbol{u}\in\mathbb{R}^N}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{y}\|^2 + \beta\|\boldsymbol{W}\boldsymbol{u}\|_1 + \iota_{[u_{\min}, u_{\max}]^N}(\boldsymbol{u}). \tag{1.111}
$$

The cost function reads as the sum of three terms, namely a quadratic term, and two non-differentiable terms. An efficient solution to Problem (1.111) can be obtained

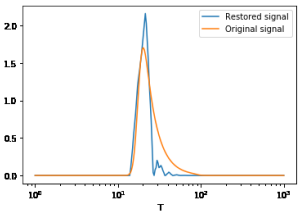

Figure 1.18 Restored signal using Algorithm (1.112) for $\beta = 20$, NMSE $= 0.15$.

by using the PPXA+ algorithm (1.69) [84], which reads in this case:

$$
\begin{array}{l}
(\boldsymbol{z}_{0,j})_{1\leq j\leq 3} \in \mathbb{R}^N \\
\gamma \in]0,+\infty[,\ \lambda \in]0,2[\\
\boldsymbol{u}_0 = (\boldsymbol{I} + \boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{W}^\top \boldsymbol{W})^{-1} \left(\boldsymbol{K}^\top \boldsymbol{z}_{0,1} + \boldsymbol{W}^\top \boldsymbol{z}_{0,2} + \boldsymbol{z}_{0,3}\right) \\
\text{For } k = 0, 1, \ldots \\
\left|\begin{array}{l}
\boldsymbol{v}_{k,1} = \operatorname{prox}_{\frac{\gamma}{2}\|\cdot - \boldsymbol{y}\|^2}(\boldsymbol{z}_{k,1}) = \boldsymbol{y} + (\boldsymbol{z}_{k,1} - \boldsymbol{y})/(\gamma+1) \\
\boldsymbol{v}_{k,2} = \operatorname{prox}_{\gamma\beta\|\cdot\|_1}(\boldsymbol{z}_{k,2}) = \operatorname{sign}(\boldsymbol{z}_{k,2}) \boxdot \max\left(|\boldsymbol{z}_{k,2}| - \beta\gamma, 0\right). \\
\boldsymbol{v}_{k,3} = \mathrm{P}_{[u_{\min},u_{\max}]^N}(\boldsymbol{z}_{\mathrm{k},3}) = \min\left(\max\left(\boldsymbol{z}_{\mathrm{k},3}, u_{\min}\right), u_{\max}\right) \\
\boldsymbol{c}_k = (\boldsymbol{I} + \boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{W}^\top \boldsymbol{W})^{-1} \left(\boldsymbol{K}^\top \boldsymbol{v}_{k,1} + \boldsymbol{W}^\top \boldsymbol{v}_{k,2} + \boldsymbol{v}_{k,3}\right) \\
\boldsymbol{z}_{k+1,1} = \boldsymbol{z}_{k,1} + \lambda\big(\boldsymbol{K}(2\boldsymbol{c}_k - \boldsymbol{u}_k) - \boldsymbol{v}_{k,1}\big) \\
\boldsymbol{z}_{k+1,2} = \boldsymbol{z}_{k,2} + \lambda\big(\boldsymbol{W}(2\boldsymbol{c}_k - \boldsymbol{u}_k) - \boldsymbol{v}_{k,2}\big) \\
\boldsymbol{z}_{k+1,3} = \boldsymbol{z}_{k,3} + \lambda\big(2\boldsymbol{c}_k - \boldsymbol{u}_k - \boldsymbol{v}_{k,3}\big) \\
\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \lambda(\boldsymbol{c}_k - \boldsymbol{u}_k).
\end{array}\right.
\end{array}
\tag{1.112}
$$

An example of solution obtained using Algorithm (1.112) is displayed in Figure 1.18.

1.4.5 Concluding remarks

We have illustrated the applicability of the introduced optimization methods, in the context of the restoration of a signal from NMR DOSY measurements. For various choices of the regularization function, we have provided a possible minimization scheme. It is worth noting that there is rarely a single technique available for the resolution of an optimization problem. For a given application, it is always recommended to test and compare different strategies, as we did in the presented example, in order to reach the best possible tradeoff in terms of computational complexity and convergence rate.

1.5 Conclusion

Table 1.2 presents a comprehensive list of the algorithms that have been presented in this chapter and the types of problem they can address. Available acceleration strategies to improve their computational efficiency are also mentioned. Since optimization constitutes a very rich and continuously evolving scientific area, it was not possible to provide an exhaustive description of all the existing methods and their various, sometimes empirical, extensions. We think however that the presented approaches provide a good basis for sound applicative developments grounded on strong theoretical foundations.

Algorithm name	Cost function	Particular cases	Acceleration techniques
Generic MM algorithm (1.13)	Any proper lower-semicontinuous function	Expectation-minimization, IRL1, IRLS, Richardson-Lucy, DC programming.	Linesearch strategy [122], block-coordinate approach [12], fixed-point schemes [123].
Quadratic MM algorithm (1.26)	Lipschitz differentiable function.	Gradient descent, SART, half-quadratic (in the smooth case), modified Newton.	Inexact version (1.30), subspace strategy (1.37), block coordinate approach (1.54).
Variable metric forward-backward algorithm (1.39)	Sum of a Lipschitz differentiable and of a convex non differentiable function.	Forward-backward, ISTA, (scaled) projected gradient, proximal point.	Inexact version [55], block-coordinate approach (1.55).
Primal-dual proximal algorithms [89]	Sum of convex functions composed with linear operators	ADMM (1.67), PPXA+ (1.69), primal-dual algorithm (1.74).	Preconditioning [124], block-coordinate approach [125], distributed strategy [95].
Primal-dual interior point algorithm (1.95)	Twice differentiable function subject to linear constraints		Inexact MM resolution (1.89), linesearch strategy [112].

Table 1.2 Summary of the algorithms presented in this chapter.

References

1. Rockafellar, R. and Wets, R. (1998) *Variational Analysis*, *Grundlehren der Mathematischen Wissenschaften*, vol. 317, Springer, Berlin, 1st edn..
2. Lee, D.D. and Seung, H.S. (2001) Algorithms for non-negative matrix factorization, in *Advances in Neural and Information Processing Systems*, vol. 13, vol. 13, pp. 556–562.
3. Févotte, C., Bertin, N., and Durrieu, J.L. (2009) Nonnegative matrix factorization with the itakura-saito divergence. with application to music analysis. *Neural Computation*, **21** (3), 793–830.
4. Févotte, C. and Idier, J. (2011) Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation*, **23** (9), 2421–2456.
5. Pham, D.T. and Garat, P. (1997) Blind separation of mixtures of independent sources through a quasi maximum likelihood approach. *IEEE Transactions on Signal Processing*, **45** (7), 1712–1725.
6. Cichocki, A., Mandic, D., Lathauwer, L.D., Zhou, G., Zhao, Q., Caiafa, C., and PHAN, H.A. (2015) Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, **32** (2), 145–163.
7. Hunter, D.R. and Lange, K. (2004) A tutorial on MM algorithms. *Amer. Stat.*, **58** (1), 30–37.
8. Lange, K., Hunter, D.R., and Yang, I. (2000) Optimization transfer using surrogate objective functions with discussion. *Journal of Computational and Graphical Statistics*, **9** (1), 1–59.
9. Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*, Cambridge University Press, New York, 1st edn..
10. Bohning, D. and Lindsay, B.G. (1988) Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, **40** (4), 641–663.
11. Zhang, Z., Kwok, J.T., and Yeung, D.Y. (2007) Surrogate maximization/minimization algorithms and extensions. *Mach. Learn.*, **69**, 1–33.
12. Hong, M., Razaviyayn, M., Luo, Z.Q., and Pang, J.S. (2016) A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Processing Magazine*, **33** (1), 57–77.
13. Horst, R. and Thoai, N.V. (1999) Dc programming: Overview. *Journal of Optimization Theory and Applications*, **103** (1), 1–43.
14. Nocedal, J. and Wright, S.J. (1999) *Numerical Optimization*, Springer-Verlag, New York, NY.
15. Combettes, P.L. and Pesquet, J.C. (2010) Proximal splitting methods in signal processing, in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (eds H.H. Bauschke, R. Burachik, P.L. Combettes, V. Elser, D.R. Luke, and H. Wolkowicz), Springer-Verlag, New York, pp. 185–212.
16. Shor, N.Z. (1985) *Minimization Methods for Non-Differentiable Functions*, *Springer Series in Computational Mathematics*, vol. 3, Springer Berlin Heidelberg.
17. Ahmad, R. and Schniter, P. (2015)

Iteratively reweighted ℓ_1 approaches to sparse composite regularization, *Tech. Rep.*. Http://arxiv.org/pdf/1504.05110.pdf.

18 Wipf, D. and Nagarajan, S. (2010) Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *IEEE Journal on Selected Topics in Signal Processing*, **4** (2), 317–329.

19 Carillo, R.E., McEwen, J.D., Van de Ville, D., Thiran, J.P., and Wiaux, Y. (2013) Sparsity averaging for compressive imaging. *IEEE Signal Processing Letters*, **20** (6), 591–594.

20 Candes, E.J., Wakin, M.B., and Boyd, S.P. (2008) Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, **14**, 877–905.

21 Fuchs, J.J. (2007) Convergence of a sparse representations algorithm applicable to real or complex data. *IEEE Transactions on Signal Processing*, **1** (4), 598–605.

22 Gorodnitsky, I.F. and Rao, B.D. (1997) Sparse signal reconstruction from limited data using focuss : a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, **45** (3), 600–616.

23 Weiszfeld, E. and Plastria, F. (2009) On the point for which the sum of the distances to n given points is minimum. *Ann Oper Res*, **167**, 7–41.

24 Byrd, R.H. and Payne, D.A. (1979) Convergence of the iteratively reweighted least squares algorithm for robust regression, *Tech. Rep. 131*, The Johns Hopkins University, Baltimore, MD.

25 Rao, B.D., Engan, K., Cotter, S.F., Palmer, J., and Kreutz-Delgao, K. (2003) Subset selection in noise based on diversity measure minimization. *IEEE Transactions on Signal Processing*, **51** (3), 760–770.

26 Allain, M., Idier, J., and Goussard, Y. (2006) On global and local convergence of half-quadratic algorithms. *IEEE Transactions on Image Processing*, **15** (5), 1130–1142.

27 Charbonnier, P., Blanc-Féraud, L., Aubert, G., and Barlaud, M. (1997) Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, **6**, 298–311.

28 Chan, T.F. and Mulet, P. (1999) On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, **36** (2), 354–367.

29 Geman, D. and Reynolds, G. (1992) Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14** (3), 367–383.

30 Geman, D. and Yang, C. (1995) Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, **4** (7), 932–946.

31 Idier, J. (2001) Convex half-quadratic criteria and interacting auxiliary variables for image restoration. *IEEE Transactions on Image Processing*, **10** (7), 1001–1009.

32 Nikolova, M. and Ng, M.K. (2005) Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific Computing*, **27**, 937–966.

33 Beck, A., Teboulle, M., and Chikishev, Z. (2008) Iterative minimization schemes for solving the single source localization problem. *SIAM Journal on Optimization*, **19** (3), 1397–1416.

34 Bissantz, N., Dumbgen, L., Munk, A., and Stratmann, B. (2009) Convergence analysis of generalized iteratively reweighted least squares algorithms on convex function spaces. *SIAM Journal on Optimization*, **19** (4), 1828–1845.

35 Lefkimmiatis, S., Bourquard, A., and Unser, M. (2012) Hessian-based norm regularization for image restoration with biomedical applications. *IEEE Transactions on Image Processing*, **21** (3), 983–995.

36 Fish, D.A., Walker, J.G., Brinicombe, A.M., and Pike, E.R. (1995) Blind deconvolution by means of the richardson-lucy algorithm. *J. Opt. Soc. Am. A*, **12**, 58–65.

37 Vardi, Y., Shepp, L.A., and Kaufman, L. (1985) A statistical model for positron emission tomography (with discussion). *Journal of American Statistical Association*, **80**, 8–38.

38 Lanteri, H., Roche, M., and Aime, C. (2002) Penalized maximum likelihood image restoration with positivity constraints : multiplicative algorithms. *Inverse Problems*, **18**, 1397–1419.

39 Bertsekas, D.P. (1999) *Nonlinear Programming*, Athena Scientific, Belmont, MA, 2nd edn..

40 Ning, X., Selesnick, I.W., and Duval, L. (2014) Chromatogram baseline estimation and denoising using sparsity (beads). *Chemometrics and Intelligent Laboratory Systems*, **139**, 156 – 167.

41 Jiang, M. and Wang, W. (2003) Convergence of the simultaneous algebraic reconstruction technique (SART). *IEEE Transactions on Image Processing*, **12** (8), 957–961.

42 Zibulevsky, M. and Elad, M. (2010) $\ell_2 - \ell_1$ optimization in signal and image processing. *IEEE Signal Processing Magazine*, **27**, 76–88.

43 De Pierro, A.R. (1995) A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Trans. Med. Imag.*, **14** (1), 132–137.

44 Labat, C. and Idier, J. (2007) Convergence of truncated half-quadratic and newton algorithms, with application to image restoration, *Tech. Rep.*, IRCCyN, Nantes, France.

45 Chouzenoux, E., Idier, J., and Moussaoui, S. (2011) A Majorize-Minimize strategy for subspace optimization applied to image restoration. *IEEE Transactions Image Processing*, **20** (18), 1517–1528.

46 Chouzenoux, E., Jezierska, A., Pesquet, J.C., and Talbot, H. (2013) A majorize-minimize subspace approach for ℓ_2-ℓ_0 image regularization. *SIAM Journal Imaging Science*, **6** (1), 563–591.

47 Chouzenoux, E. and Pesquet, J.C. (2017) A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation. *IEEE Transactions on Signal Processing*, **65** (18), 4770–4783.

48 Hager, W.W. and Zhang, H. (2006) A survey of nonlinear conjugate gradient methods. *Pacific Journal on Optimization*, **2** (1), 35–58.

49 Liu, D.C. and Nocedal, J. (1989) On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, **45** (3), 503–528.

50 Chouzenoux, E., Pesquet, J.C., Talbot, H., and Jezierska, A. (2011) A memory gradient algorithm for ℓ_2-ℓ_0 regularization with applications to image restoration, in *18th IEEE International Conference on Image Processing (ICIP 2011)*, Brussels, Belgium, pp. 2717–2720.

51 Florescu, A., Chouzenoux, E., Pesquet, J.C., Ciuciu, P., and Ciochina, S. (2014) A majorize-minimize memory gradient method for complex-valued inverse problem. *Signal Processing*, **103**, 285–295. Special issue on Image Restoration and Enhancement: Recent Advances and Applications.

52 Miele, A. and Cantrell, J.W. (1969) Study on a memory gradient method for the minimization of functions. *Journal on Optimization Theory and Applications*, **3** (6), 459–470.

53 Chouzenoux, E. and Pesquet, J.C. (2016) Convergence rate analysis of the majorize-minimize subspace algorithm. *IEEE Signal Processing Letters*, **23** (9), 1284–1288.

54 Combettes, P.L. and Vũ, B.C. (2014) Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization: A Journal of Mathematical Programming and Operations Research*, **63** (9), 1289–1318.

55 Chouzenoux, E., Pesquet, J.C., and Repetti, A. (2014) Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, **162** (1), 107–132.

56 Combettes, P.L. and Wajs, V.R. (2005) Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, **4** (4), 1168–1200.

57 Daubechies, I., Defrise, M., and De Mol, D. (2004) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, **57**, 1413–1457.

58 Combettes, P., Dung, D., and Vu, B. (2011) Proximity for sums of composite functions. *J. Math. Anal. Appl.*, **380** (2), 680–688.

59 Abboud, F., Chouzenoux, E., Pesquet, J.C., Chenot, J.H., and Laborelli, L.

(2015) A dual block coordinate proximal algorithm with application to deconvolution of interlaced video sequences, in *Proceedings of the 22nd IEEE International Conference on Image Processing (ICIP 2015)*, Quebec City, Canada, pp. 4917–4921.

60 Chambolle, A. and Pock, T. (2015) A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI-Journal of computational mathematics*, **1**, 29–54.

61 Jaggi, M., Smith, V., Takac, M., Terhorst, J., Krishnan, S., Hofmann, T., and Jordan, M.I. (2014) Communication-efficient distributed dual coordinate ascent, in *Advances in Neural Information Processing Systems 27* (eds Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger), Curran Associates, Inc., pp. 3068–3076.

62 Abboud, F., Chouzenoux, E., Pesquet, J.C., Chenot, J.H., and Laborelli, L. (2015) A distributed strategy for computing proximity operators, in *Proceedings of the 49th Asilomar Conference on Signals, Systems and Computers (ASILOMAR 2015)*, pp. 396–400.

63 Bertsekas, D.P. (1981) Projected Newton methods for optimization problems with simple constraints. *SIAM Journal Control and Optimization*, **20**, 762–767.

64 Bonettini, S., Zanella, R., and Zanni, L. (2009) A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, **25** (1), 015 002+.

65 Iusem, A.N. (2003) On the convergence properties of the projected gradient method for convex optimization. *Computational Applied Mathematics*, **22** (1), 37–52.

66 Cherni, A., Chouzenoux, E., Duval, L., and Pesquet, J.C. (2019) A novel smoothed norm ratio for sparse signal restoration application to mass spectrometry, in *Proceedings of Signal Processing with Adaptive Sparse Structured Representations (SPARS 2019)*, Toulouse, France.

67 Cherni, A., Chouzenoux, E., Duval, L., and Pesquet, J.C. (2020) SPOQ lp-Over-lq regularization for sparse signal recovery applied to mass spectrometry, *Tech. Rep.*. Https://arxiv.org/abs/2001.08496.

68 Jacobson, M.W. and Fessler, J.A. (2007) An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Transactions on Image Processing*, **16** (10), 2411–2422.

69 Razaviyayn, M., Hong, M., and Luo, Z. (2013) A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, **23** (2), 1126–1153.

70 Sotthivirat, S. and Fessler, J.A. (2002) Image recovery using partitioned-separable paraboloidal surrogate coordinate ascent algorithms. *IEEE Transactions on Signal Processing*, **11** (3), 306–317.

71 Chouzenoux, E., Pesquet, J.C., and Repetti, A. (2016) A block coordinate variable metric forward-backward algorithm. *Journal on Global Optimization*, **66** (3), 457–485.

72 Repetti, A., Chouzenoux, E., and Pesquet, J.C. (2014) A preconditioned forward-backward approach with application to large-scale nonconvex spectral unmixing problems, in *Proceedings of the 39th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, Firenze, Italy, pp. 1498–1502.

73 Repetti, A., Pham, M.Q., Duval, L., Chouzenoux, E., and Pesquet, J.C. (2015) Euclid in a taxicab: Sparse blind deconvolution with smoothed l1/l2 regularization. *IEEE Signal Processing Letters*, **22** (5), 539–543.

74 Abboud, F., Chouzenoux, E., Pesquet, J., Chenot, J., and Laborelli, L. (2019) An alternating proximal approach for blind video deconvolution. *Signal Processing: Image Communication*, **70**, 21–36.

75 Gabay, D. and Mercier, B. (1976) A dual algorithm for the solution of nonlinear variational problems via finite elements approximations. *Computers and Mathematics with Applications*, **2**, 17–40.

76 Fortin, M. and Glowinski, R. (eds) (1983) *Augmented Lagrangian Methods: Applications to the Numerical Solution of*

Boundary-Value Problems, Elsevier Science Ltd, Amsterdam: North-Holland.

77 Figueiredo, M.A.T. and Bioucas-Dias, J.M. (2010) Restoration of Poissonian images using alternating direction optimization. *IEEE Transactions on Image Processing*, **19** (12), 3133–3145.

78 Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, **3** (1), 1–222.

79 Lions, P.L. and Mercier, B. (1979) Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, **16**, 964–979.

80 Combettes, P.L. and Pesquet, J.C. (2007) A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, **1** (4), 564–574.

81 Chen, C., He, B., Ye, Y., and Yuan, X. (2014) The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, pp. 1–23.

82 Setzer, S., Steidl, G., and Teuber, T. (2010) Deblurring poissonian images by split bregman techniques. *Journal on Visual Communication and Image Representation*, **21** (3), 193–199.

83 Combettes, P.L. and Pesquet, J.C. (2008) A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, **24** (6).

84 Pesquet, J.C. and Pustelnik, N. (2012) A parallel inertial proximal optimization method. *Pacific Journal of Optimization*, **8** (2), 273–305.

85 Cherni, A., Chouzenoux, E., and Delsuc, M.A. (2016) PALMA, an improved algorithm for DOSY signal processing. *Analyst*, **142** (5), 772–779.

86 Cherni, A., Chouzenoux, E., and Delsuc, M.A. (2016) Proximity operators for a class of hybrid sparsity+entropy priors. application to DOSY NMR signal reconstruction, in *Proceedings of the International Symposium on Signal, Image, Video and Communications (ISIVC 2016)*, Tunis, Tunisia.

87 Chambolle, A. and Pock, T. (2011) A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, **40** (1), 120–145.

88 Esser, E., Zhang, X., and Chan, T. (2010) A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, **3** (4), 1015–1046.

89 Komodakis, N. and Pesquet, J.C. (2014) Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, **32** (6), 31–54.

90 Combettes, P.L. and Pesquet, J.C. (2012) Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators. *Set-Valued and Variational Analysis*, **20** (2), 307–330.

91 Condat, L. (2013) A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization, Theory and Applications*, **158** (2), 460–479.

92 Vũ, B.C. (2013) A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, **38** (3), 667–681.

93 Raguet, H., Fadili, J., , and Peyré, G. (2013) A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, **6** (3), 1199–1226.

94 Bricenos-Arias, L., Chierchia, G., Chouzenoux, E., and Pesquet, J.C. (2019) A random block-coordinate douglas-rachford splitting method with low computational complexity for binary logistic regression. *Computational Optimization and Applications*, **72** (3), 707–726.

95 Pesquet, J.C. and Repetti, A. (2015) A class of randomized primal-dual algorithms for distributed optimization. *Journal of nonlinear and convex analysis*, **16** (12), 2453–2490.

96 Cherni, A., Chouzenoux, E., and Delsuc, M.A. (2018) Fast dictionnary-based approach for mass spectrometry data

analysis, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018)*, Calgary, Canada.

97 Fiacco, A.V. and McCormick, G.P. (1967) The sequential unconstrained minimization technique (SUMT) without parameters. *Operations Research*, **15** (5), 820–827.

98 Wright, S.J. (1997) *Primal-dual interior-point methods*, SIAM, Philadelphia, PA, 1st edn..

99 Forsgren, A., Gill, P.E., and Wright, M.H. (2002) Interior methods for nonlinear optimization. *SIAM Review*, **44** (4), 525–597.

100 Moussaoui, S., Chouzenoux, E., and Idier, J. (2012) Primal-dual interior point optimization for penalized least squares estimation of abundance maps in hyperspectral imaging, in *Proceedings of the 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2012)*, Shangai, China.

101 Chouzenoux, E., Legendre, M., Moussaoui, S., and Idier, J. (2012) Fast constrained least squares spectral unmixing using primal-dual interior-point optimization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **7** (1), 59–69.

102 Armand, P., Gilbert, J.C., and Jan-Jégou, S. (2000) A feasible BFGS interior point algorithm for solving strongly convex minimization problems. *SIAM Journal on Optimization*, **11**, 199–222.

103 Wright, M.H. (1994) Some properties of the Hessian of the logarithmic barrier function. *Mathematical Programming*, **67** (2), 265–295.

104 Wright, M.H. (1998) Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM Journal on Optimization*, **9** (1), 84–111.

105 Friedlander, M.P. and Orban, D. (2012) A primal–dual regularized interior-point method for convex quadratic programs. *Mathematical Programming Computation*, **4** (1), 71–107.

106 Conn, A., Gould, N., and Toint, P.L. (1996) A primal-dual algorithm for minimizing a nonconvex function subject to bounds and nonlinear constraints, in *Nonlinear Optimization and Applications* (eds G. Di Pillo and F. Giannessi), Kluwer Academic Publishers, 2nd edn..

107 Armand, P., Benoist, J., and Dussault, J. (2012) Local path-following property of inexact interior methods in nonlinear programming. *Computational Optimization and Applications*, **52** (1), 209–238.

108 Bonettini, S., Galligani, E., and Ruggiero, V. (2007) Inner solvers for interior point methods for large scale nonlinear programming. *Computational Optimization and Applications*, **37** (1), 1–34.

109 Legendre, M., Moussaoui, S., Chouzenoux, E., and Idier, J. (2014) Primal-dual interior-point optimization based on majorization-minimization for edge preserving spectral unmixing, in *Proceedings of the 21st IEEE International Conference on Image Processing (ICIP 2014)*, Shangai, China, pp. 4161–4165.

110 Legendre, M., Moussaoui, S., Schmidt, F., and Idier, J. (2013) Parallel implementation of a primal-dual interior-point optimization method for fast abundance maps estimation, in *Proceedings of the 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2013)*, Gainesville, FL, USA.

111 Murray, W. and Wright, M.H. (1994) Line search procedures for the logarithmic barrier function. *SIAM Journal on Optimization*, **4** (2), 229–246.

112 Chouzenoux, E., Moussaoui, S., and Idier, J. (2012) Majorize-minimize linesearch for inversion methods involving barrier function optimization. *Inverse Problems*, **28** (6), 065 011.

113 Chouzenoux, E., Moussaoui, S., and Idier, J. (2011) Efficiency of line search strategies in interior point methods for linearly constrained optimization, in *Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP 2011)*, Nice, France, pp. 101–104.

114 Chouzenoux, E., Moussaoui, S., Idier, J., and Mariette, F. (2013) Primal-dual

interior point optimization for a regularized reconstruction of NMR relaxation time distributions, in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, pp. 8747–8750.

115 El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., and Zhang, Y. (1996) On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, **89**, 507–541.

116 Johnson, C.A., Seidel, J., and Sofer, A. (2000) Interior-point methodology for 3-D PET reconstruction. *IEEE Transactions on Medical Imaging*, **19** (4).

117 Chouzenoux, E., Moussaoui, S., Idier, I., and Mariette, F. (2010) Efficient maximum entropy reconstruction of nuclear magnetic resonance t1-t2 spectra. *IEEE Transactions on Signal Processing*, **58** (12), 6040–6051.

118 Kaipin, X. and Shanmin, Z. (2013) Trust-region algorithm for the inversion of molecular diffusion nmr data. *Analytical chemistry*, **86** (1), 592–599.

119 Johnson, C.S. (1999) Diffusion ordered nuclear magnetic resonance spectroscopy: principles and applications. **34**, 203–256.

120 Beck, A. and Teboulle, M. (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, **2** (1), 183–202.

121 Pustelnik, N., Benazza-Benhayia, A., Zheng, Y., and Pesquet, J.C. (2016) Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*.

122 Lange, K. (1995) A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 425–437.

123 Varadhan, R. and Roland, C. (2008) Simple and globally convergent methods for accelerating the convergence of any em algorithm. *Scandinavian Journal of Statistics*, **35** (2), 335–353.

124 Pock, T. and Chambolle, A. (2011) Diagonal preconditioning for first order primal-dual algorithms in convex optimization, in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, pp. 1762–1769.

125 Repetti, A., Chouzenoux, E., and Pesquet, J.C. (2015) A random block-coordinate primal-dual proximal algorithm with application to 3d mesh denoising, in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2015)*, Brisbane, Australia, pp. 3561–3565.