

PAPER

Superiorization of incremental optimization algorithms for statistical tomographic image reconstruction

To cite this article: E S Helou *et al* 2017 *Inverse Problems* **33** 044010

View the [article online](#) for updates and enhancements.

Related content

- [String-averaging incremental subgradients for constrained convex optimization with applications to reconstruction of tomographic images](#)
Rafael Massambone de Oliveira, Elias Salomão Helou and Eduardo Fontoura Costa
- [subgradient algorithms for bilevel convex optimization](#)
Elias S Helou and Lucas E A Simões
- [A new convergence analysis and perturbation resilience of some accelerated proximal forward-backward algorithms with errors](#)
Daniel Reem and Alvaro De Pierro

Recent citations

- [Strong convergence and bounded perturbation resilience of a modified proximal gradient algorithm](#)
Yanni Guo and Wei Cui
- [Exponential convergence rate of distributed optimisation for multi-agent systems with constraints set over a directed graph](#)
Zhu Wang *et al*
- [Total variation superiorized conjugate gradient method for image reconstruction](#)
Marcelo V W Zibetti *et al*



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Superiorization of incremental optimization algorithms for statistical tomographic image reconstruction

E S Helou^{1,4}, M V W Zibetti² and E X Miqueles³

¹ Department of Applied Mathematics and Statistics, State University of São Paulo, São Carlos, Brazil

² Federal University of Technology, Curitiba, Brazil

³ Brazilian Synchrotron Light Source, CNPEM, Campinas, Brazil

E-mail: elias@icmc.usp.br

Received 29 April 2016, revised 14 July 2016

Accepted for publication 12 August 2016

Published 1 March 2017



CrossMark

Abstract

We propose the superiorization of incremental algorithms for tomographic image reconstruction. The resulting methods follow a better path in its way to finding the optimal solution for the maximum likelihood problem in the sense that they are closer to the Pareto optimal curve than the non-superiorized techniques. A new scaled gradient iteration is proposed and three superiorization schemes are evaluated. Theoretical analysis of the methods as well as computational experiments with both synthetic and real data are provided.

Keywords: superiorization, convex optimization, tomographic image reconstruction

(Some figures may appear in colour only in the online journal)

1. Introduction

Tomographic images reconstructed from projection data are important tools in various applications, for example, ranging from medicine to materials science and from geosciences to astronomy. It is therefore useful to develop the techniques that enable good reconstruction from a variety of data acquisition modalities. That means being able to cope with statistical error, poor angular sampling, truncated data, among other difficulties. Sometimes a combination of factors must be dealt with.

⁴ Author to whom any correspondence should be addressed.

Statistical methods were developed in order to handle poor photon counts, mainly for use in emission tomography modalities. Among those we can mention, in order of appearance, EM [13, 15], OS-EM [10], RAMLA [2], BSREM [4], DRAMA [8, 14] and string-averaging expectation maximization (SAEM) [7]. The original numerical approach to solve the statistical optimization model was the EM algorithm, which was deemed too slow, taking many iterations to provide reasonably accurate images; OS-EM subdivides the data into subsets and processes each of these incrementally in order to achieve an order of magnitude speedup compared to EM, but leads to oscillatory behavior when applied to inconsistent data; RAMLA takes the subset approach to the extreme using a single datum at a time, but prevents oscillation through the use of relaxation in order to ensure convergence; BSREM generalizes RAMLA by considering more flexible subset divisions and by allowing the use of regularization in the objective function; DRAMA introduced variable relaxation within an iteration cycle aiming at a more even noise contribution from each individual datum in the resulting images; finally, SAEM further enhances the possibilities by considering a parallelization approach whereby the same algorithm is applied in parallel to subsets of the data and the results are averaged to form the next iterate.

In the present paper, we consider yet another direction for statistical methods. All of the aforementioned algorithms use a diagonal scaling of the descent direction, which provides desirable convergence characteristics and helps in maintaining non-negativity of the iterations. This diagonal scaling has the drawback of making convergence analysis more difficult in the case of a non-differentiable objective function, which precludes several useful regularization functions. We, therefore, make use of the superiorization framework in order to analyze algorithms for this case. Our analysis will cover all of the above cited methods, allowing for the use of more interesting regularization functions alongside scaled incremental algorithms for the statistical tomographic reconstruction problem. Moreover, we present a new algorithmic framework which has better theoretical characteristics than the techniques mentioned in the previous paragraph, with similar practical performance.

We use both the superiorized SAEM algorithm and the newly presented superiorized approach in order to solve real world tomographic problems arising from synchrotron illuminated radiographic data. Implementation of the techniques was very inhomogeneous with the former being implemented using Matlab and the second running on GPUs, so no direct comparison was intended to be made.

1.1. Tomographic imaging from projections

The most common transmission tomography technique makes use of X-rays, based on the Beer–Lambert law:

$$\frac{I_d}{I_e} = e^{-\int_L \mu(x) ds}, \quad (1)$$

where I_d and I_e are, respectively, the detected and the emitted X-ray beam intensities, L is the line segment connecting emitter and detector and $\mu : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ is the non-negative linear attenuation factor. Different attenuation factor values usually correspond to features of interest, for example, the body anatomy in human patients, and therefore its knowledge brings important information for applications. The problem is now to recover the attenuation factor μ at each point in the plane from the non-invasive measurements of line integrals of μ . While, for simplicity of exposition, we focus on the two-dimensional case in the present paper, volumetric images can be obtained either by stacking planar tomographic images or by directly extending the approach to 3D datasets.

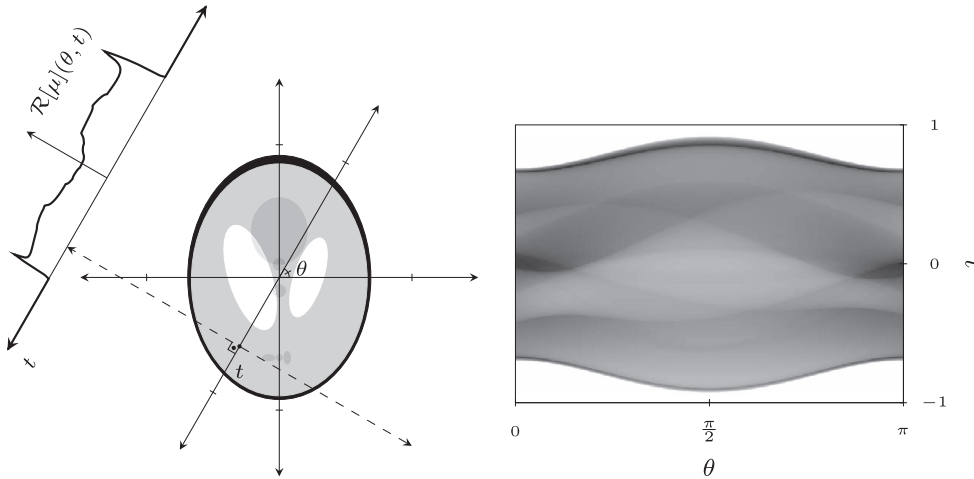


Figure 1. Left: meaning of the arguments of the Radon transform. Here, θ is the angle between the normal to the integration path and the horizontal axis, and t is the distance from the line of integration to the origin. Right: sinogram, i.e., an image of the Radon transform of the image shown on the left in the $\theta \times t$ coordinate system (adapted with permission from [7], copyright 2014 IOP Publishing Ltd).

We simplify the notation by parameterizing the data by the angle θ between the normal to the integration path and the horizontal axis and by the distance t of this integration path to the origin of the coordinate system. In doing so, we define the Radon transform \mathcal{R} , which takes functions on the plane to functions on the cylinder as follows:

$$\mathcal{R}[\mu](\theta, t) := \int_{\mathbb{R}} \mu \left(t \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} + s \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix} \right) ds. \quad (2)$$

Figure 1 depicts this definition. The Radon transform is mathematically rich, possessing several useful properties. Among those we can mention, for example, the Fourier slice property, which relates the one-dimensional Fourier transforms of $\mathcal{R}[\mu]$ along the t coordinate to ‘slices’ of the two-dimensional Fourier transform of the image μ . More importantly for us here, the Radon transform turns out to be a compact linear operator which has, in consequence, an ill-conditioned inverse or pseudo-inverse [11].

1.2. Iterative image reconstruction

While exploring some deep mathematical properties of the Radon transform can lead to useful inversion formulas, sometimes it is better to consider all of the involved physical effects more carefully. For example, the observation that the emission of photons follows a stochastic rule can lead to improved imaging by the introduction of maximum likelihood models whereby the reconstructed image is selected as the one that would give the maximum possible likelihood for the observed data. Whichever the imaging modality is, this approach will give rise to an optimization problem to be solved, therefore bringing the need to use iterative algorithms in order to approximate the solution of the model.

In every optimization model, whether based on statistical ideas or not, a discretization of the imaged object has to be taken into consideration and, given the finite nature of the data, the Radon transform, or more physically precise stripe-based integral transforms, can be

represented by a matrix $R \in \mathbb{R}^{m \times n}$. In this setting, the image is written as

$$\sum_{j=1}^n x_j \mu_j,$$

where $\{\mu_1, \mu_2, \dots, \mu_n\}$ is a basis for the space where the imaged object is assumed to be and x_j are the components of vector $\mathbf{x} \in \mathbb{R}^n$. In all of our models, the basis will be the indicator functions of square pixels which also means that the x_j are the actual values of the pixelized image. If taken literally, the problem would then become to find the solution of the linear system of equations

$$R\mathbf{x} = \mathbf{b},$$

where the coefficients of R are $r_{ij} = \mathcal{R}[\mu_j](\theta_i, t_i)$ and the components of \mathbf{b} are approximations to the Radon transform: $b_i \approx \mathcal{R}[\mu](\theta_i, t_i)$, where μ is the desired true image. The system matrix arising in tomographic problems ranges from mildly to severely ill-posed, which brings difficulties because of the experimental error in the collected tomographic data \mathbf{b} .

Therefore, instead of directly solving the above linear system of equations, it is usually more profitable to consider a (regularized) maximum likelihood model. For example, in emission tomography the desired image is the intensity of photon emission across the imaged plane, i.e., the number of emitted photons on each square pixel [15]. In his case, maximizing the likelihood of obtaining the data is equivalent to solve the following optimization problem:

$$\begin{aligned} \min \quad & L_E(\mathbf{x}) := \sum_{i=1}^m \{ (R\mathbf{x})_i - b_i \ln (R\mathbf{x})_i \} \\ \text{s.t. :} \quad & \mathbf{x} \in \mathbb{R}_+^n, \end{aligned}$$

where, in this case, b_i represents the attenuation corrected number of photons detected on bin i .

For the case of transmission tomography, if we denote as ρ_i the number of events detected by detector i during a dark scan (i.e., no X-ray source turned on), β_i to be the number of photons detected by detector i during a blank scan and α_i the number of photons detected by detector i during the actual tomographic scan of the object, the maximum likelihood problem is equivalent to [12]:

$$\begin{aligned} \min \quad & L_{TR}(\mathbf{x}) := \sum_{i=1}^m \{ \beta_i e^{-(R\mathbf{x})_i} - \alpha_i \ln (e^{-(R\mathbf{x})_i} + \rho_i) \} \\ \text{s.t. :} \quad & \mathbf{x} \in \mathbb{R}_+^n. \end{aligned}$$

Both of the above functions are convex [9] (convexity of L_{TR} requires assumptions on the data which are always satisfied in practice), although not always strictly convex. Non-convex models do exist for the tomographic imaging problem, but due to the usually large dimensions of the problem algorithms for the convex optimization problem are more practical and much more common in practice. We will focus only on convex models in the present paper. In the next section we present the proposed algorithms and the general optimization model they intend to solve.

2. Algorithm description

For optimization problems like

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t. :} \quad & \mathbf{x} \in \mathbb{R}_+^n, \end{aligned} \quad (3)$$

with $f : \mathbb{R}^n \rightarrow \mathbb{R}$ convex and differentiable, we will consider algorithms of the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)}) + \epsilon^{(k)}, \quad (4)$$

where the diagonal scaling matrix will be given differently according to one of two algorithmic schemes to be described later.

Furthermore, it is assumed that

$$\lambda_k > 0, \quad \sum_{k=0}^{\infty} \lambda_k = \infty.$$

For us, the usefulness of allowing the error sequence $\{\epsilon^{(k)}\}$ in the analysis will be twofold. First, as already observed before in the literature [2, 4, 7, 8], it allows for flexible, from sequential to parallel with many intermediate instances, processing of the data giving rise to fast and stable algorithms. Second, it will allow us to apply the superiorization concept in order to drive the iterates to the optimum through a path with smoother intermediate iterates, as follows. We write an algorithm following (4) in a compact form as

$$\mathbf{x}^{(k+1)} = \mathcal{O}(\lambda_k, \mathbf{x}^{(k)}), \quad (5)$$

and we create a new algorithm following

$$\mathbf{x}^{(k+1)} = \tilde{\mathcal{O}}(\lambda_k, \mathbf{x}^{(k)}) := \mathcal{O}(\lambda_k, \mathbf{x}^{(k)}) + \mathbf{s}^{(k)}, \quad (6)$$

where we name $\{\mathbf{s}^{(k)}\}$ the *superiorization sequence*. Algorithm (6) is correspondingly named the *superiorized* version of algorithm (4). Notice that if we define $\tilde{\epsilon}^{(k)} := \epsilon^{(k)} + \mathbf{s}^{(k)}$, we then have that iterations (6) are equivalent to

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)}) + \tilde{\epsilon}^{(k)}.$$

That is, the essential approximation characteristics of the method are not ruined by adding a deliberate, appropriately small, perturbation. We shall worry about convergence issues in the next section.

In the remainder of the present section we describe two options for the operator \mathcal{O} , one of which is an existing algorithm while the second is our new proposal. Furthermore, we also describe more precisely the alternatives for the superiorization sequence we will try in the numerical section.

2.1. String-averaging expectation maximization

The SAEM algorithm was introduced in [7] and applied to a maximum likelihood problem in tomographic reconstruction. Convergence analysis contained in [7] embraces somewhat general smooth convex cost functions, and therefore we will keep the acronym SAEM even when not necessarily referring to applications to likelihood maximization problems.

In order to describe SAEM, we assume that the objective function can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^p f_i(\mathbf{x}),$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is supposed to be a convex function and $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are sufficiently smooth functions.

We then assume that the sequence $\{1, 2, \dots, p\}$ is split in s (ordered) subsets, $\{S_1, S_2, \dots, S_s\}$, each of which is called a string, satisfying

$$\bigcup_{l=1}^s S_l = \{1, 2, \dots, p\} \quad \text{and} \quad S_l \cap S_\ell = \emptyset \quad \forall l \neq \ell.$$

We make each of the subsets explicit through the following notation for the elements of each string:

$$S_l := \{l_1^l, l_2^l, \dots, l_{\nu(l)}^l\}.$$

That is, l_i^l is the i th element of string S_l and $\nu(l)$ is the cardinality of string S_l .

Now, for each string we iterate a scaled incremental gradient for the function $F_l := \sum_{i \in S_l} f_i$. Let us define the string operators $\mathcal{S}_l : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ for this purpose:

$$\mathcal{S}_l(\lambda, \mathbf{x}) := \mathbf{y}^{(\nu(l))},$$

where λ is a positive scalar, $\mathbf{x} \in \mathbb{R}^n$ and the vectors $\mathbf{y}^{(i)} \in \mathbb{R}^n$, $i \in \{1, 2, \dots, \nu(l)\}$ are recursively computed from the identity below:

$$\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} - \lambda D(\mathbf{y}^{(i-1)}) \nabla f_{l_i^l}(\mathbf{y}^{(i-1)}), \quad i \in \{1, 2, \dots, \nu(l)\}.$$

Recursion starts from $\mathbf{y}^{(0)} = \mathbf{x}$, and the scaling matrix is given by

$$D(\mathbf{x}) := \begin{pmatrix} \frac{x_1}{p_1} & & & & \\ & \frac{x_2}{p_2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \frac{x_n}{p_n} \end{pmatrix}, \quad (7)$$

where $p_j > 0$ and x_1, x_2, \dots, x_n are the components of \mathbf{x} .

Each of these operators is no more than a block-RAMLA (BRAMLA) iteration over a specific data subset [2, 4]. A BRAMLA iteration is a generalization of a RAMLA iteration whereby blocks of data can be used at each subiteration. Under mild smoothness assumptions on the gradients ∇f_i , it is possible to show that

$$\mathcal{S}_l(\lambda, \mathbf{x}) = \mathbf{x} - \lambda D(\mathbf{x}) \nabla F_l(\mathbf{x}) + O(\lambda^2).$$

See, e.g., [7, proposition 5.3]. Therefore, by setting weights $\{\omega_1, \omega_2, \dots, \omega_s\}$ such that $\sum_{l=1}^s \omega_l = 1$ and $\omega_l \geq 0$, we can finally define the iterative procedure:

$$\mathbf{x}^{(k+1)} = \sum_{l=1}^s \omega_l \mathcal{S}_l(\mathbf{x}^{(k)}, \lambda_k).$$

For which, then, there holds

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \nabla \tilde{f}(\mathbf{x}_k) + O(\lambda_k^2),$$

where $\tilde{f} := \sum_{l=1}^s \omega_l F_l$ is a weighted version of the original objective function $f = \sum_{l=1}^s F_l$.

A key, albeit very simple, observation here is that the convergence analysis naturally absorbs an error term, which makes it attractive for the superiorization approach because we can add suitable perturbations to this error term and still maintain a convergent algorithm. In order to do that, the magnitude of the perturbation must be appropriately controlled, which we will analyze theoretically in depth in the next section. The remaining of this section presents an option to SAEM and then discusses the three forms of the perturbation $\{s^{(k)}\}$ we consider, which we recall that we have named *superiorization sequences*.

2.2. New superiorized scaled gradient algorithm

In the present paper, we not only introduce superiorization of well known maximum likelihood algorithms, but we also present a different method with improved theoretical characteristics when compared to RAMLA and its relatives, while maintaining the good practical performance of these techniques. In order to describe the new algorithmic framework, there is a key idea: that the diagonal scaling matrix cannot have components vanishing when the corresponding entry in the gradient is positive, in order to avoid the possibility of convergence to a non-optimal point when the objective function is convex but not strictly convex. Of course the precise way of interpreting and implementing this statement is very important to the final computational results, and we describe it in details below.

In the following description we use the same notation for the strings given in section 2.1. The concrete algorithm that we propose uses strings similarly to SAEM, but is stabilized in a sense that it actually approximates a better-scaled iteration. We name it SSAEM from stabilized SAEM, and the iterations are as follows:

$$\tilde{S}_l(\lambda_k, \mathbf{x}^{(k)}) := \mathbf{y}^{(k,l,\nu(l))},$$

where

$$\mathbf{y}^{(k,l,i)} = \mathbf{y}^{(k,l,i-1)} - \lambda_k \bar{D}(\mathbf{y}^{(k,l,i-1)}) \nabla f_{i_l}(\mathbf{y}^{(k,l,i-1)}), \quad i \in \{1, 2, \dots, \nu(l)\},$$

and

$$\mathbf{y}^{(k,0)} = \mathbf{x}^{(k)}.$$

The diagonal scaling matrix \bar{D} is given componentwise, for $\mathbf{x} \in \mathbb{R}_+^n$, by

$$\bar{D}(\mathbf{x})_{jj} := \begin{cases} \frac{x_j}{p_j} & \text{if } x_j > \tau, \\ \frac{\tau}{p_j} & \text{otherwise,} \end{cases}$$

where each p_j , $j \in \{1, 2, \dots, n\}$ is positive.

The next iterate is computed in two steps. First the averaging:

$$\tilde{\mathbf{x}}^{(k+1)} = \sum_{l=1}^s \omega_l \tilde{S}_l(\lambda_k, \mathbf{x}^{(k)}),$$

and then a componentwise correction:

$$x_j^{(k+1)} = \begin{cases} x_j^{(k)} + \frac{x_j^{(k)}}{\tau} (\tilde{x}_j^{(k+1)} - x_j^{(k)}) & \text{if } x_j^{(k)} \leq \tau \quad \text{and} \quad \tilde{x}_j^{(k+1)} < x_j^{(k)}, \\ \tilde{x}_j^{(k+1)} & \text{otherwise.} \end{cases}$$

We discuss some useful facts about these iterations in preparation for the complete convergence analysis to be provided in the next section.

Proposition 1. Fix $l \in \{1, 2, \dots, s\}$. Suppose that $x_j^{(k)} \geq 0$ for all $j \in \{1, 2, \dots, n\}$, and that, for every $i \in \{1, 2, \dots, \nu(l)\}$, $\{\nabla f_{l_i}^i(\mathbf{y}^{(k,l,i-1)})\}_{k \in \mathbb{N}}$ is bounded, $\nabla f_{l_i}^i$ is Lipschitz in

$$C_{l,i} := \text{the closure of the convex hull of } \{\mathbf{x}^{(k)}, \mathbf{y}^{(k,l,i-1)}\}_{k \in \mathbb{N}},$$

and that $C_{l,i}$ is in turn bounded. Then:

$$\tilde{S}_l(\lambda_k, \mathbf{x}^{(k)}) = \mathbf{x}^{(k)} - \lambda_k \bar{D}(\mathbf{x}^{(k)}) (\nabla F_l(\mathbf{x}^{(k)}) + O(\lambda_k)).$$

Proof. Notice that the hypothesis ensure that each function $\bar{D}(\cdot) \nabla f_{l_i}^i(\cdot)$ is Lipschitz, say with Lipschitz constant M , in C_i and thus, if we define

$$\begin{aligned} \epsilon^{(k,l)} &:= \bar{D}(\mathbf{x}^{(k)}) \nabla F_l(\mathbf{x}^{(k)}) - \sum_{i=1}^{\nu(l)} \bar{D}(\mathbf{y}^{(k,l,i-1)}) \nabla f_{l_i}^i(\mathbf{y}^{(k,l,i-1)}) \\ &= \sum_{i=1}^{\nu(l)} \{\bar{D}(\mathbf{x}^{(k)}) \nabla f_{l_i}^i(\mathbf{x}^{(k)}) - \bar{D}(\mathbf{y}^{(k,l,i-1)}) \nabla f_{l_i}^i(\mathbf{y}^{(k,l,i-1)})\}, \end{aligned}$$

we then have

$$\begin{aligned} \|\epsilon^{(k,l)}\| &\leq \sum_{i=1}^{\nu(l)} \|\bar{D}(\mathbf{x}^{(k)}) \nabla f_{l_i}^i(\mathbf{x}^{(k)}) - \bar{D}(\mathbf{y}^{(k,l,i-1)}) \nabla f_{l_i}^i(\mathbf{y}^{(k,l,i-1)})\| \\ &\leq M \sum_{i=1}^{\nu(l)} \|\mathbf{x}^{(k)} - \mathbf{y}^{(k,l,i-1)}\|. \end{aligned} \quad (8)$$

Now, we observe that the boundedness assumptions imply that there is a real number N large enough such that

$$\|\mathbf{x}^{(k)} - \mathbf{y}^{(k,l,i-1)}\| \leq (i-1) \lambda_k N.$$

Therefore, using this inequality in (8) we conclude that

$$\|\epsilon^{(k,l)}\| \leq \lambda_k N M \frac{\nu(l)(\nu(l)-1)}{2}. \quad (9)$$

Thus, from the definition of the method, the definition of $\epsilon^{(k,l)}$ and (9):

$$\begin{aligned} \tilde{S}_l(\lambda_k, \mathbf{x}^{(k)}) &= \mathbf{x}^{(k)} - \lambda_k \sum_{i=1}^{\nu(l)} \bar{D}(\mathbf{y}^{(k,l,i-1)}) \nabla f_{l_i}^i(\mathbf{y}^{(k,l,i-1)}) \\ &= \mathbf{x}^{(k)} - \lambda_k \bar{D}(\mathbf{x}^{(k)}) \nabla F_l(\mathbf{x}^{(k)}) + \lambda_k \epsilon^{(k,l)} \\ &= \mathbf{x}^{(k)} - \lambda_k \bar{D}(\mathbf{x}^{(k)}) \nabla F_l(\mathbf{x}^{(k)}) + O(\lambda_k^2). \end{aligned}$$

Finally, because the diagonal elements of diagonal matrix $\bar{D}(\mathbf{x}^{(k)})$ are positive and bounded from below by construction, the claim follows. \square

Let us introduce the simplifying notation

$$\mathcal{I} := \{(l, i) : l \in \{1, 2, \dots, s\} \text{ and } i \in \{1, 2, \dots, \nu(l)\}\}.$$

Corollary 1. Suppose $x_j^{(k)} \geq 0$ for all $j \in \{1, 2, \dots, n\}$, and that, for every $(l, i) \in \mathcal{I}$, $\{\nabla f_{l_i}^i(\mathbf{y}^{(k,l,i-1)})\}_{k \in \mathbb{N}}$ is bounded, $\nabla f_{l_i}^i$ is Lipschitz in

$C_{l,i} :=$ the closure of the convex hull of $\{\mathbf{x}^{(k)}, \mathbf{y}^{(k,l,i-1)}\}_{k \in \mathbb{N}}$,

and that $C_{l,i}$ is in turn bounded. Then:

$$\tilde{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \bar{D}(\mathbf{x}^{(k)}) (\nabla \tilde{f}(\mathbf{x}^{(k)}) + O(\lambda_k)).$$

Proof. Use the definition of $\tilde{\mathbf{x}}^{(k+1)}$ and proposition 1 in order to get

$$\begin{aligned} \tilde{\mathbf{x}}^{(k+1)} &= \sum_{l=1}^s \omega_l \tilde{S}_l(\lambda_k, \mathbf{x}^{(k)}) = \mathbf{x}^{(k)} - \lambda_k \sum_{l=1}^s \omega_l \bar{D}(\mathbf{x}^{(k)}) (\nabla F_l(\mathbf{x}^{(k)}) + O(\lambda_k)) \\ &= \mathbf{x}^{(k)} - \lambda_k \bar{D}(\mathbf{x}^{(k)}) \left(\sum_{l=1}^s \omega_l \nabla F_l(\mathbf{x}^{(k)}) + \sum_{l=1}^s \omega_l O(\lambda_k) \right), \end{aligned}$$

from where the claim follows. \square

Corollary 1 actually says that the iterations before renormalization satisfy

$$\tilde{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \bar{D}(\mathbf{x}^{(k)}) \mathbf{g}^{(k)},$$

where

$$\mathbf{g}^{(k)} = \nabla \tilde{f}(\mathbf{x}^{(k)}) + O(\lambda_k).$$

Thus, given the fact that $\bar{D}(\mathbf{x}^{(k)})_{jj} \geq 0$, and by the way the normalization is done, we have

$$\tilde{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \tilde{D}(\mathbf{x}^{(k)}, \mathbf{g}^{(k)}) \mathbf{g}^{(k)},$$

where the diagonal scaling matrix is given by

$$\tilde{D}(\mathbf{x}, \mathbf{g})_{jj} = \begin{cases} \frac{\tau}{p_j} & \text{if } x_j \leq \tau \text{ and } g_j \leq 0, \\ \frac{x_j}{p_j} & \text{otherwise.} \end{cases}$$

2.3. Superiorization sequences

The main advantage of a superiorization technique is that there is a lot of freedom for the sequence of perturbations to be added to the traditional optimization method. It is appealing to obtain a convergent method for a maximum likelihood solution, but, in practice, when iterated to full convergence, the maximum likelihood solution in emission or low count transmission tomography is degraded by noise. Therefore, and given the high computational cost of iterative techniques, one common way of smoothing the results is to stop the algorithm prematurely, having started it from a very smooth initial image.

While reasonably successful, the early stopping strategy cannot provide images as good as explicit regularization of the objective function in a model such as

$$\begin{aligned} \min \quad & f(\mathbf{x}) + \zeta r(\mathbf{x}) \\ \text{s.t. :} \quad & \mathbf{x} \in \mathbb{R}_+^n, \end{aligned} \tag{10}$$

where $\zeta > 0$ is the regularization parameter, and $r : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function that penalizes roughness, or otherwise undesirable features, in the image. More precisely, by denoting \mathbf{x}_ζ the optimizer of problem (10), it is possible to see that

$$r(\mathbf{x}_\zeta) = \min \{r(\mathbf{x}) : \mathbf{x} \in \mathbb{R}_+^n, f(\mathbf{x}) = f(\mathbf{x}_\zeta)\}.$$

That is, explicit regularization in the optimization model will provide the smoothest possible image among those with the same fitness to data. Although this is a desirable feature, there remains the problem of selecting a proper regularization parameter ζ under a reasonable computational effort. This task would demand the solution of problem (10) for several tentative values of the regularization parameter, which would therefore lead to a heavy computational overhead to the reconstruction process.

We claim that the superiorization framework can be useful in this setting, because by a proper selection of the regularization sequence used to perturb the unregularized method we can obtain iterates that are closer to the optimal r versus f curve traced by the explicitly regularized optimization method. This could potentially lead to improved results, for example, when consistency-based early stopping strategies are used.

For illustrative purposes, we will focus on $r = \text{TV}$, the total variation functional:

$$\text{TV}(\mathbf{x}) := \sum_{i=1}^{\sqrt{n}} \sum_{j=1}^{\sqrt{n}} \sqrt{(x_{i,j} - x_{i-1,j})^2 + (x_{i,j} - x_{i,j-1})^2},$$

where we have used the simplifying assumption that the reconstructed images are square (which can be readily dropped), the lexicographic ordering of elements, and a periodic boundary condition:

$$x_{0,j} := x_{n,j} \quad \text{and} \quad x_{i,0} := x_{i,n}.$$

2.3.1. Standard superiorization procedure

Algorithm 1. General superiorized version of algorithm $\mathcal{O}(\mathbf{x}^{(k)})$

```

1: set  $k \leftarrow 0$ 
2: set  $\mathbf{x}^{(0)}$  as initial solution
3: while  $f(\mathbf{x}^{(k)}) > \varepsilon$ 
4:   set  $\mathbf{x}^{(k+1/2)} \leftarrow \mathcal{O}(\lambda_k, \mathbf{x}^{(k)})$ 
5:   set  $\mathbf{x}^{(k+1)} \leftarrow \mathbf{R}_r(\mathbf{x}^{(k+1/2)})$ 
6:   set  $k \leftarrow k + 1$ 
7: return  $\mathbf{x}^{(k)}$ 

```

The general superiorized algorithm (6) is exposed in algorithm 1, where it is implicit in the superiorization step $\mathbf{R}_r(\mathbf{x}^{(k+1/2)})$ the procedure for obtaining the superiorized sequence $\mathbf{s}^{(k)} = \mathbf{R}_r(\mathbf{x}^{(k+1/2)}) - \mathbf{x}^{(k+1/2)}$. It is expected that the superiorization step improves the solution moving the current iterate towards the minimum of $r(\mathbf{x})$. Also, note that one could choose $\mathbf{x}^{(k)}$ or $\mathbf{x}^{(k+1/2)}$ as the reconstructed image, it is just a matter of visual preference⁵. Algorithm 2 shows the method proposed in [5] to produce a sequence which it is called here the *standard* superiorization sequence. In algorithm 2, $\beta_0 > 0$ and $\alpha \in (0, 1)$, and $\beta_0 \alpha^k$ is the starting trial stepsize at iteration k , which is repeatedly reduced by a factor of α until the decrease criterion is satisfied.

⁵ In [5], the reconstructed image is $\mathbf{x}^{(k+1/2)}$, and it used in the stopping criteria $f(\mathbf{x}^{(k+1/2)}) > \varepsilon$.

Algorithm 2. Standard superiorization procedure to obtain $\mathbf{x}^{(k+1)}$ from point $\mathbf{x}^{(k+1/2)}$

```

1: set  $\ell \leftarrow k$ 
2: set  $n \leftarrow 0$ 
3: set  $\mathbf{b}^{(n)} \leftarrow \mathbf{x}^{(k+1/2)}$ 
4: while  $n < N$ 
5:   set  $\mathbf{v}^{(n)}$  to be a nonascending vector for  $r$  at  $\mathbf{b}^{(n)}$ 
6:   set loop  $\leftarrow$  true
7:   while loop
8:     set  $\ell \leftarrow \ell + 1$ 
9:     set  $\beta_n \leftarrow \beta_0 \times \alpha^\ell$ 
10:    set  $\mathbf{z} \leftarrow \mathbf{b}^{(n)} + \beta_n \mathbf{v}^{(n)}$ 
11:    if  $r(\mathbf{z}) \leq r(\mathbf{x}^{(k+1/2)})$  then
12:      set  $n \leftarrow n + 1$ 
13:      set  $\mathbf{b}^{(n)} \leftarrow \mathbf{z}$ 
14:      set loop  $\leftarrow$  false
15: return  $\mathbf{x}^{(k+1)} \leftarrow \mathbf{b}^{(n)}$ 

```

2.3.2. Projected subgradient iteration. A well known algorithm used for the minimization of convex, non-differentiable, functions under constraints is the projected subgradient iteration, where the iteration operator has the form

$$S(\mathbf{x}, \lambda) := \mathbf{x} - \lambda \tilde{\nabla} r(\mathbf{x}).$$

Here, $r : \mathbb{R}^n \rightarrow \mathbb{R}$ is the function to be minimized, $\tilde{\nabla} r(\mathbf{x}) \in \partial r(\mathbf{x})$ and

$$\partial r(\mathbf{x}) := \{\mathbf{v} \in \mathbb{R}^n : r(\mathbf{x}) + \nabla \mathbf{v}^T (\mathbf{y} - \mathbf{x}) \leq r(\mathbf{y})\}$$

is the subdifferential of r at \mathbf{x} . In our application, the subgradient-based superiorization operator is given as repeated application of a subgradient step under a diminishing stepsize scheme, followed by projection onto the non-negative orthant:

$$\begin{aligned} \mathbf{y}^{(0)} &= \mathbf{x} \\ \mathbf{y}^{(i)} &= \mathbf{y}^{(i-1)} - \frac{\gamma}{i} \tilde{\nabla} r(\mathbf{y}^{(i-1)}), \quad i = 1, 2, \dots, N \\ R_r(\mathbf{x}) &= \mathfrak{P}_{\mathbb{R}_+^n}(\mathbf{y}^{(N)}), \end{aligned} \tag{11}$$

where \mathfrak{P}_X is, for a non-empty convex and closed set X , the projection:

$$\mathfrak{P}_X(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|.$$

We will omit the starting stepsize γ from the notation, but it is implicitly assumed that each application of the technique has a well defined value for the parameter $\gamma > 0$. When considered as part of an iterative scheme, a sequence $\{\gamma_k\}$ is assumed.

The subgradient $\mathbf{t} \in \partial \operatorname{TV}(\mathbf{x})$ we have used in this superiorization scheme and as a nonascending direction for the standard superiorization procedure is given componentwise by:

$$t_{i,j} = \frac{2x_{i,j} - x_{i,j-1} - x_{i-1,j}}{\sqrt{(x_{i,j} - x_{i,j-1})^2 + (x_{i,j} - x_{i-1,j})^2}} + \frac{x_{i,j} - x_{i,j+1}}{\sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i,j+1} - x_{i-1,j+1})^2}} + \frac{x_{i,j} - x_{i+1,j}}{\sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i+1,j} - x_{i+1,j-1})^2}},$$

where the fractions that have denominator equal to 0 are ignored in the summation.

2.3.3. Fast gradient projection (FGP). In [1], Beck and Teboulle proposed an accelerated method to compute the proximal operator for positively constrained total variation given by:

$$\text{prox}_{\text{TV}, \mathbb{R}_+^n}(\mathbf{b}) := \arg \min_{\mathbf{x} \in \mathbb{R}_+^n} \|\mathbf{x} - \mathbf{b}\|^2 + \gamma \text{TV}(\mathbf{x}). \quad (12)$$

The approach in [1] is based on the dual formulation proposed by Chambolle [3], where TV is written as:

$$\text{TV}(\mathbf{x}) := \max_{\mathbf{p}, \mathbf{q} \in \mathcal{P}} T(\mathbf{x}, \mathbf{p}, \mathbf{q}), \quad (13)$$

where:

$$T(\mathbf{x}, \mathbf{p}, \mathbf{q}) := \sum_{i=1}^{\sqrt{n}} \sum_{j=1}^{\sqrt{n}} p_{i,j} (x_{i,j} - x_{i-1,j}) + q_{i,j} (x_{i,j} - x_{i,j-1})$$

and \mathcal{P} is a set in which \mathbf{p} and \mathbf{q} satisfy:

$$p_{i,j}^2 + q_{i,j}^2 \leq 1 \quad \text{for } 1 \leq i, j \leq \sqrt{n}.$$

This way, (12) can be written as:

$$\text{prox}_{\text{TV}, \mathbb{R}_+^n}(\mathbf{b}) := \arg \min_{\mathbf{x} \in \mathbb{R}_+^n} \max_{\mathbf{p}, \mathbf{q} \in \mathcal{P}} \|\mathbf{x} - \mathbf{b}\|^2 + \gamma(\mathbf{p}, \mathbf{q})^T \mathcal{D}\mathbf{x}, \quad (14)$$

where \mathcal{D} is the linear operator that takes \mathbf{x} and returns the pair \mathbf{u} and \mathbf{v} such that

$$u_{i,j} = x_{i,j} - x_{i-1,j} \quad \text{and} \quad v_{i,j} = x_{i,j} - x_{i,j-1}.$$

Equation (14) is solved with gradient projection algorithm in [3], and with a FGP in [1], the approach we are utilizing in this paper to produce the superiorized sequence. Notice that applying (14) just after the main algorithm:

$$\mathbf{x}^{(k+1)} = \text{prox}_{\text{TV}, \mathbb{R}_+^n}(\mathcal{O}(\lambda_k, \mathbf{x}^{(k)})) = \mathcal{O}(\lambda_k, \mathbf{x}^{(k)}) + \mathbf{s}^{(k)},$$

produces the superiorized sequence. Again, we avoid cluttering the notation by omitting the positive parameter γ .

3. Theoretical convergence analysis

3.1. SAEM-SUP

While the concrete form of our algorithm is based on the perturbed SAEM as described in the previous section, we will focus on a more general diagonally scaled gradient descent algorithm of the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)}) + o(\lambda_k), \quad (15)$$

where the diagonal scaling matrix $D(\mathbf{x})$ is given in (7) and

$$\lim_{\lambda \rightarrow 0} \frac{\|o(\lambda)\|}{\lambda} = 0.$$

It will be useful to separate the algorithm in two steps, as with algorithm 1:

$$\begin{aligned} \mathbf{x}^{(k+1/2)} &= \mathcal{O}(\lambda_k, \mathbf{x}^{(k)}) \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k+1/2)} + \mathbf{s}^{(k)}. \end{aligned} \quad (16)$$

We will assume in this subsection that the algorithm is able to maintain positivity for both subsequences $\{\mathbf{x}^{(k+1/2)}\}$ and $\{\mathbf{x}^{(k)}\}$. This will be the case for our techniques and is easy to achieve when using the diagonal scaling matrix $D(\mathbf{x})$ with the gradient descent direction in the first step and a projected subgradient or constrained proximal step in the second part of the algorithm.

Proposition 2. Consider the iterative procedure described in (16). Assume the sequences $\{\mathbf{x}^{(k)}\}$ and $\{\mathbf{x}^{(k+1/2)}\}$ are bounded and positive, and that $\{\nabla f(\mathbf{x}^{(k)})\}$ and $\{\nabla f(\mathbf{x}^{(k+1/2)})\}$ are bounded. Suppose also that the first step operator satisfies

$$\mathcal{O}(\lambda, \mathbf{x}) = \mathbf{x} - \lambda D(\mathbf{x}) \nabla f(\mathbf{x}) + o(\lambda), \quad (17)$$

and that

$$\lambda_k \rightarrow 0^+, \quad \sum_{k=0}^{\infty} \lambda_k = \infty, \quad \text{and} \quad \frac{\|\mathbf{s}^{(k)}\|}{\lambda_k} \rightarrow 0.$$

Then either there is a subsequence $\{\mathbf{x}^{(l_k)}\}$ satisfying $D(\mathbf{x}^{(l_k)}) \nabla f(\mathbf{x}^{(l_k)}) \rightarrow \mathbf{0}$ or we have $f(\mathbf{x}^{(k)}) \rightarrow -\infty$.

Proof. Let us assume that there is no subsequence $\{\mathbf{x}^{(l_k)}\}$ satisfying the limit $D(\mathbf{x}^{(l_k)}) \nabla f(\mathbf{x}^{(l_k)}) \rightarrow \mathbf{0}$. Notice that this statement is equivalent to saying that there is no subsequence such that $D(\mathbf{x}^{(l_k)})^{1/2} \nabla f(\mathbf{x}^{(l_k)}) \rightarrow \mathbf{0}$. In this case, there is $\epsilon > 0$ such that $\|D(\mathbf{x}^{(k)})^{1/2} \nabla f(\mathbf{x}^{(k)})\|^2 \geq \epsilon$ for every k .

Therefore, we have

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &= f(\mathbf{x}^{(k)} - \lambda_k D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)}) + o(\lambda_k) + \mathbf{s}^{(k)}) \\ &= f(\mathbf{x}^{(k)} - \lambda_k D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)}) + o(\lambda_k)) \\ &= f(\mathbf{x}^{(k)}) - \lambda_k \nabla f(\mathbf{x}^{(k)})^T D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)}) + o(\lambda_k) \\ &= f(\mathbf{x}^{(k)}) - \lambda_k \|D(\mathbf{x}^{(k)})^{1/2} \nabla f(\mathbf{x}^{(k)})\|^2 + o(\lambda_k) \\ &\leq f(\mathbf{x}^{(k)}) - \lambda_k \epsilon + o(\lambda_k), \end{aligned}$$

where the first equality is obtained replacing $\mathbf{x}^{(k+1)}$ according to the definition of algorithm in (16) and assumption (17), the second equality comes from $\|\mathbf{s}^{(k)}\|/\lambda_k \rightarrow 0$ and the third from the smoothness and boundedness assumptions which allow us to use the Taylor expansion of f around $f(\mathbf{x}^{(k)})$ in order to estimate $f(\mathbf{x}^{(k+1)})$. The fourth equation is merely formal and the last equation finally comes from $\|D(\mathbf{x}^{(k)})^{1/2} \nabla f(\mathbf{x}^{(k)})\|^2 \geq \epsilon > 0$.

Now, let $\kappa \in \mathbb{N}$ be such that $k > \kappa \Rightarrow o(\lambda_k)/\lambda_k < \epsilon/2$. Then, iterating the above inequality from $\mathbf{x}^{(\kappa)}$ we get:

$$\begin{aligned}
f(\mathbf{x}^{(\kappa+n)}) &\leq f(\mathbf{x}^{(\kappa)}) - \sum_{k=0}^{n-1} \lambda_{\kappa+k} \left\{ \epsilon - \frac{o(\lambda_{\kappa+k})}{\lambda_{\kappa+k}} \right\} \\
&\leq f(\mathbf{x}^{(\kappa)}) - \frac{\epsilon}{2} \sum_{k=0}^{n-1} \lambda_{\kappa+k}.
\end{aligned}$$

From this inequality we see that $\sum_{k=0}^{\infty} \lambda_k = \infty$ implies that $f(\mathbf{x}^{(k)}) \rightarrow -\infty$. \square

Now we use the convexity of the objective function and an extra *a priori* assumption in order to obtain global convergence of the algorithm. We shall not give the proofs here, instead referring, respectively, to [8, propositions 7 and 5] for demonstrations.

Proposition 3. *Suppose that beyond the assumptions of proposition 2, we further have that $f(\mathbf{x}^{(k)})$ converges and that f is strictly convex. Then*

$$f(\mathbf{x}^{(k)}) \rightarrow f^*,$$

where f^* is the optimal value of f over \mathbb{R}_+^n .

Proposition 4. *Suppose that beyond the assumptions of proposition 2, we further have to assume that $\sum_{k=0}^{\infty} \|o(\lambda_k)\| < \infty$ and that f is strictly convex. Then*

$$f(\mathbf{x}^{(k)}) \rightarrow f^*,$$

where f^* is the optimal value of f over \mathbb{R}_+^n .

While these results may be useful in many cases, our interest is in exploring the circumstances where there may be several optima and we wish to select one among these, according to some criterion. Therefore, the strict convexity hypothesis is too restrictive (albeit superiorized techniques can be useful in the unique solution case too). Furthermore, summability of the perturbations could make the influence of the superiorization small in the overall optimization process.

A difficulty in proving convergence results for algorithms of the form (16) is that the step direction can vanish at a non-optimal point. This fact leads to the possibility that a sequence could converge to non-optimality. This possibility cannot be ruled out because there is no way to guarantee that the error term is small compared to the components of the iterates in case $x_j^{(k)} \rightarrow 0$. The technical way found in the literature is to use summability of the perturbation term to obtain convergence of the objective value sequence. Then, assuming strict convexity one can prove convergence by finding out that the existence of a non-optimal limit point would imply the existence of another different limit point and that both of these are optimizers of the original problem with added constraints of the form $x_j = 0$. The next subsection analyses the alternative SSAEM, which avoids these difficulties.

3.2. SSAEM-SUP

We now analyze iterative techniques of the form:

$$\begin{aligned}
\mathbf{x}^{(k+1/2)} &= \mathbf{x}^{(k)} - \lambda_k \tilde{D}(\mathbf{x}^{(k)}, \mathbf{g}^{(k)}) \mathbf{g}^{(k)} \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k+1/2)} + \mathbf{s}^{(k)},
\end{aligned} \tag{18}$$

where

$$\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)}) + \boldsymbol{\epsilon}_k \quad (19)$$

with

$$\boldsymbol{\epsilon}_k \rightarrow \mathbf{0} \quad (20)$$

and where $\tilde{D}(\mathbf{x}, \mathbf{g})$ is the diagonal matrix given componentwise as

$$\begin{aligned} \tilde{D}(\mathbf{x}, \mathbf{g})_{jj} &:= \begin{cases} \frac{\tau}{p_j} & \text{if } x_j \leq \tau \text{ and } g_j \leq 0, \\ \frac{x_j}{p_j} & \text{otherwise,} \end{cases} \\ &= \begin{cases} \frac{x_j}{p_j} & \text{if } g_j > 0, \\ \frac{\max\{x_j, \tau\}}{p_j} & \text{otherwise.} \end{cases} \end{aligned}$$

We have already discussed how to implement this kind of algorithm in the previous section, and it consists of a string-averaged scaled incremental gradient iteration where the scaling matrix does not contain too small entries, followed by a correction in those components which are small and have positive corresponding component $g_j^{(k)}$. The scaling matrix $\tilde{D}(\mathbf{x}^{(k)}, \mathbf{g}^{(k)})$ can be seen as an approximation to $\hat{D}(\mathbf{x}^{(k)})$ where

$$\hat{D}(\mathbf{x})_{jj} := \begin{cases} \frac{x_j}{p_j} & \text{if } \frac{\partial f}{\partial x_j}(\mathbf{x}) > 0, \\ \frac{\max\{x_j, \tau\}}{p_j} & \text{otherwise.} \end{cases}$$

Notice that, if $\mathbf{x}^* \in \mathbb{R}_+^n$, $\hat{D}(\mathbf{x}^*)^{1/2} \nabla f(\mathbf{x}^*) = \mathbf{0}$ is a necessary and sufficient optimality condition for \mathbf{x}^* to be an optimizer of problem (3) given that f is convex and $\tau > 0$. As a basic tool for further analysis, we give the practical approximation \tilde{D} to \hat{D} a more precise and useful mathematical characterization. We will simplify the notation by using $D^k := \tilde{D}(\mathbf{x}^{(k)}, \mathbf{g}^{(k)})$.

Proposition 5. *Suppose f is convex and continuously differentiable, that $\{\mathbf{x}^{(k)}\} \subset \mathbb{R}_+^n$ generated by algorithm (18) is bounded and that (19) and (20) hold. Then, for every $\epsilon > 0$, there are $\kappa \in \mathbb{N}$ and $\delta_\epsilon > 0$ such that for every $k \geq \kappa$ we have*

$$f(\mathbf{x}^{(k)}) \geq f^* + \epsilon \Rightarrow \|(D^k)^{1/2} \mathbf{g}^{(k)}\|^2 \geq \delta_\epsilon.$$

Proof. If the claim were false, there would exist $\epsilon > 0$ and a subsequence $\{\mathbf{x}^{(l_k)}\}$ such that $f(\mathbf{x}^{(l_k)}) \geq f^* + \epsilon$ and $\|(D^{l_k})^{1/2} \mathbf{g}^{(l_k)}\| \rightarrow 0$. Because of the boundedness assumptions, there is no loss of generality in assuming that we have the limits $\mathbf{x}^{(l_k)} \rightarrow \bar{\mathbf{x}}$ and $D^{l_k} \rightarrow \bar{D}$. Notice also that we have $\mathbf{g}^{(l_k)} \rightarrow \nabla f(\bar{\mathbf{x}})$.

Furthermore, we claim that $\hat{D}(\bar{\mathbf{x}}) \nabla f(\bar{\mathbf{x}}) = \mathbf{0}$. If this is not true, it is because at least one of the two cases below hold for a given index j :

$$(1) \bar{x}_j > 0 \text{ and } \frac{\partial f}{\partial x_j}(\bar{\mathbf{x}}) > 0;$$

$$(2) \frac{\partial f}{\partial x_j}(\bar{\mathbf{x}}) < 0.$$

If case 1 were true, for large enough k we would have $x_j^{(k)} \geq 1/2\bar{x}_j$ and $g_j^{(k)} \geq 1/2 \frac{\partial f}{\partial x_j}(\bar{\mathbf{x}})$. Therefore, there would hold $D_{jj}^k \frac{\partial f}{\partial x_j}(\mathbf{x}^{(k)}) \geq 1/4 D(\bar{\mathbf{x}})_{jj} \frac{\partial f}{\partial x_j}(\bar{\mathbf{x}}) > 0$, which contradicts $\|(D^k)^{1/2} \mathbf{g}^{(k)}\| \rightarrow 0$. If, on the other hand, case 2 holds, then, for large enough k , we have $g_j^{(k)} \leq 1/2 \frac{\partial f}{\partial x_j}(\bar{\mathbf{x}})$ and therefore $D_{jj}^k \frac{\partial f}{\partial x_j}(\mathbf{x}^{(k)}) \leq 1/2\tau/p_j \frac{\partial f}{\partial x_j}(\bar{\mathbf{x}}) < 0$, again a contradiction.

Now, since $\bar{\mathbf{x}}$ satisfies $f(\bar{\mathbf{x}}) \geq f^* + \epsilon$ it cannot be an optimal point. However, $\hat{D}(\bar{\mathbf{x}})\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$ is a necessary and sufficient condition for $\bar{\mathbf{x}}$ to be optimal if f is convex. This contradiction is derived from the assumption that the specified $\{\mathbf{x}^{(k)}\}$ exists, and therefore the claim is proven. \square

With this result in hand we can prove convergence of the method. The key idea is to notice that if $f(\mathbf{x}^{(k)})$ differs from f^* by more than some threshold, then we can prove that the algorithm reduces the value of the objective function at small enough stepsize regimes. This implies in convergence because it results that once we are inside a sublevel set we cannot ‘get away from it’ by much. The details are in the theorem below.

Theorem 1. Assume f is convex and continuously differentiable, the sequence of stepsizes satisfy

$$\sum_{k=0}^{\infty} \lambda_k = \infty, \quad \lambda_k \rightarrow 0^+ \quad \text{and} \quad \frac{\|\mathbf{s}^{(k)}\|}{\lambda_k} \rightarrow 0,$$

and that $\{\mathbf{x}^{(k)}\}$, as generated by algorithm (18) satisfying (19) and (20) is bounded and nonnegative. Then

$$f(\mathbf{x}^{(k)}) \rightarrow f^*.$$

Proof. Let us first provide an estimate of the decrease in function value based on the optimality measure $(D^k)^{1/2} \mathbf{g}^{(k)}$:

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &= f\left(\mathbf{x}^{(k)} - \lambda_k \left[D^k \mathbf{g}^{(k)} + \frac{\mathbf{s}^{(k)}}{\lambda_k} \right]\right) \\ &= f(\mathbf{x}^{(k)}) - \lambda_k \nabla f(\mathbf{x}^{(k)})^T \left[D^k \mathbf{g}^{(k)} + \frac{\mathbf{s}^{(k)}}{\lambda_k} \right] + o(\lambda_k) \\ &= f(\mathbf{x}^{(k)}) - \lambda_k \mathbf{g}^{(k)T} D^k \mathbf{g}^{(k)} + o(\lambda_k) \\ &= f(\mathbf{x}^{(k)}) - \lambda_k \|(D^k)^{1/2} \mathbf{g}^{(k)}\|^2 + o(\lambda_k). \end{aligned} \tag{21}$$

In the above sequence, the first equality is from (18), the second from differentiability of f and boundedness of $\{D^k \mathbf{g}^{(k)} + \mathbf{s}^{(k)}/\lambda_k\}$, which is a consequence of the boundedness of $\{\mathbf{x}^{(k)}\}$ and of $\mathbf{s}^{(k)}/\lambda_k \rightarrow 0$. The third equality is based on (19) and (20), while the fourth equality is merely formal.

Now let us fix any $\epsilon > 0$ and assume that κ is large for proposition 5 to hold for this ϵ and also that $k \geq \kappa$ implies that $o(\lambda_k)/\lambda_k \leq \delta_\epsilon/2$, where δ_ϵ is from proposition 5.

We then split in two cases:

- (1) $f(\mathbf{x}^{(k)}) > f^* + \epsilon$;
 (2) $f(\mathbf{x}^{(k)}) \leq f^* + \epsilon$.

In case 1, proposition 5, equation (21) and $o(\lambda_k)/\lambda_k \leq \delta_\epsilon/2$ lead to

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &= f(\mathbf{x}^{(k)}) - \lambda_k \|(D^k)^{1/2} \mathbf{g}^{(k)}\|^2 + o(\lambda_k) \\ &\leq f(\mathbf{x}^{(k)}) - \lambda_k \delta_\epsilon + \lambda_k \frac{\delta_\epsilon}{2} \\ &\leq f(\mathbf{x}^{(k)}) - \lambda_k \frac{\delta_\epsilon}{2}. \end{aligned} \quad (22)$$

Therefore, because $\sum_{k=0}^{\infty} \lambda_k = \infty$ we know that either the second case occurs infinitely many times or $f(\mathbf{x}^{(k)}) \rightarrow -\infty$. If this second possibility is true, the claim is proven, if not, we proceed with the argument.

Now if case 2 holds, then we have

$$f(\mathbf{x}^{(k+1)}) = f(\mathbf{x}^{(k)}) + O(\lambda_k)(\mathbf{x}^{(k)}) + \lambda_k M$$

for some large enough M . Therefore, a consequence of the fact that we reach this case infinitely many times coupled with (22) for case 1 is that after reaching the sublevel set $\text{lev}_\epsilon(f) := \{\mathbf{x} : f(\mathbf{x}) \leq f^* + \epsilon\}$, say in iterate k_0 , the algorithm never escapes from the larger, but ever reducing, sublevel set $\text{lev}_{\epsilon+\lambda_{k_1}}(f)$ for some $k_1 \geq k_0$. Thus

$$\limsup_{k \rightarrow \infty} f(\mathbf{x}^{(k)}) \leq f^* + \epsilon.$$

But since $\epsilon > 0$ was arbitrary and $f(\mathbf{x}^{(k)}) \geq f^*$, we then have the claim proven. \square

4. Numerical experimentation

The numerical experiments are divided in two independent sets. In the first of these, the SAEM [7] is compared with the EM [15] considering two kind of superiorization sequences for non-negatively constrained total variation as a secondary criteria. The first superiorization sequence is produced according to algorithm 2, following the ideas presented in [5], the second superiorization sequence is produced by the FGP algorithm from [1].

Moreover, this first set of experiments considers 15 repetitions of the error simulating procedure where the MSE and SSIM [16] figures of merit of the reconstructed image of each algorithmic variation are computed alongside with other numerical and performance indicators. These experiments were done on a quad core Intel i5-4570S CPU @2.9 GHz, with 32 GB DDR3 memory.

The second test is used in order to assess the viability of the SSAEM with or without superiorization for large tomographic reconstruction from actual data. It uses a specialized GPU implementation which is efficient only for $s = 1$. The objective of this set of experiments is to both give a proof of concept showing that the algorithm is efficient in practical applications and to assess the effects of superiorization in its performance for use with the maximum likelihood model for transmission tomography. In this case a GeForce GTX 745 was used.

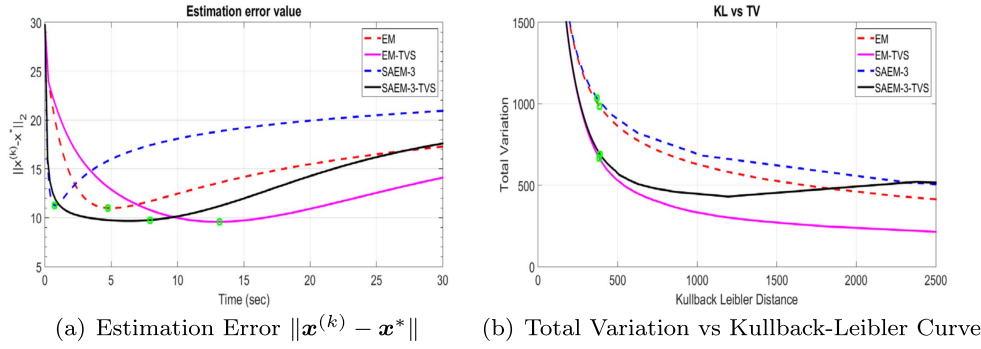


Figure 2. Comparison between EM and SAEM with and without superiorization.

4.1. SAEM

This technique was used, as already mentioned, in order to solve problems of the form

$$\begin{aligned} \min \quad & \sum_{i=1}^m \{ (R\mathbf{x})_i - b_i \ln (R\mathbf{x})_i \} \\ \text{s.t. : } \quad & \mathbf{x} \in \mathbb{R}_+^n, \end{aligned}$$

which can be interpreted both as the maximum likelihood model for emission tomography and as the minimal Kullback–Leibler distance model for a general non-consistent non-negative system of equations. In this case, each f_i was only one term of the above sum, i.e.

$$f_i(\mathbf{x}) = (R\mathbf{x})_i - b_i \ln (R\mathbf{x})_i$$

and, therefore

$$\nabla f_i(\mathbf{x}) = R_i^T \left(1 - \frac{b_i}{(R\mathbf{x})_i} \right),$$

where R_i is the i th line of R .

Tomographic setup. For this first set of tests we utilized 32 angles with 182 line integrals each. The reconstructed images and the original numeric phantoms have dimensions of 128×128 pixels. Poisson noise was added to this sparse-angle simulated acquisition resulting in a signal to noise ratio of around 18 dB in the data for each of the 15 repetitions.

4.1.1. Algorithmic parameters. *Starting image.* All of the algorithms, including SAEM with or without superiorization, were started from a uniform image $x_j^{(0)} = \phi$ such that $\sum_{i=1}^m (R\mathbf{x}^{(0)})_i = \sum_{i=1}^m b_i$. It can be noticed that the appropriate value for this to hold is

$$\phi = \frac{\sum_{i=1}^m b_i}{\sum_{i=1}^m (R\mathbf{1})_i},$$

where $\mathbf{1}$ is a vector in \mathbb{R}^n with all entries equal to 1.

Strings formation and weights. We denote as SAEM- s the SAEM algorithm with s strings, and as SAEM- s -TVS and SAEM- s -TVS-FGP its standard superiorized version and its FGP superiorized version, respectively. All of SAEM- s versions, with or without superiorization, used weights $w_i = 1/s$, thereby maintaining the unweighted likelihood model. Strings were selected by randomly shuffling the data and partitioning into s strings of equal length

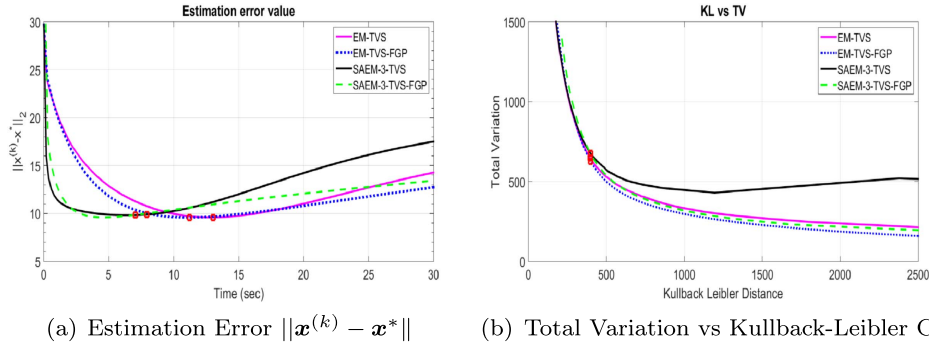


Figure 3. Comparison between EM and SAEM with standard superiorization and FGP superiorization.

(depending on divisibility of the data size by the number of strings, some strings actually contained one element more than others). All algorithms were ran for several iterations, however the final result was assumed when $f(\mathbf{x}^{(k)}) \leq 400$. This value was experimentally chosen corresponding to a point where all algorithms are close to their minimum MSE (see figures 2 and 3).

Scaling matrix. The coefficients p_i of $D(\mathbf{x})$ must be selected, we have followed [7]:

$$p_j = \sum_{i=1}^n a_{ij} \quad j \in \{1, 2, \dots, n\}.$$

This scaling was selected because then SAEM- m with fixed stepsize $\lambda = m$ corresponds to the EM algorithm.

Stepsize sequence. The stepsize sequence λ_k was chosen following [7]: for SAEM- s (i.e., s strings) and its superiorized versions we have used

$$\lambda_k = \frac{\lambda_{0,s}}{k^{0.51}/s + 1},$$

where $\lambda_{0,s}$ was the largest value such that SAEM- s would lead to a positive $\mathbf{x}^{(1)}$.

Superiorization sequences. For the standard superiorization sequence, the step $\beta_{n,k}$ is unknown a priori and depends on the choice of the nonascending direction $\mathbf{v}^{(n,k)}$. So the size of the vector $\mathbf{s}^{(k)}$ depends on β_0 and N . The larger these two variables the larger is the visual effect in the superiorization sequence. We have used $\beta_0 = 1$ and $N = 10$ or $N = 20$ for EM and SAEM, respectively, and we have fixed $\alpha = 0.95$.

For the superiorization sequence produced by the proximal operator via FGP, the size of $\mathbf{s}^{(k)}$ depends on the γ_k used in (12). The γ_k chosen for this experiments is:

$$\gamma_k = \frac{\gamma_0}{(k+1)^{1+\text{eps}}}, \quad (23)$$

where eps is the machine epsilon⁶. The values used for γ_0 used were manually adjusted to 0.15 for EM-TVS-FGP and 0.3 for SAEM-TVS-FGP, which were values that produced a competitive performance. The sequence with step from (23) is also summable, which is a requisite from the convergence analysis. We expect the sequence produced by the

⁶ The machine epsilon is an upper bound on the relative error due to rounding in floating point arithmetic. For double precision it is around 2.22×10^{-16} .

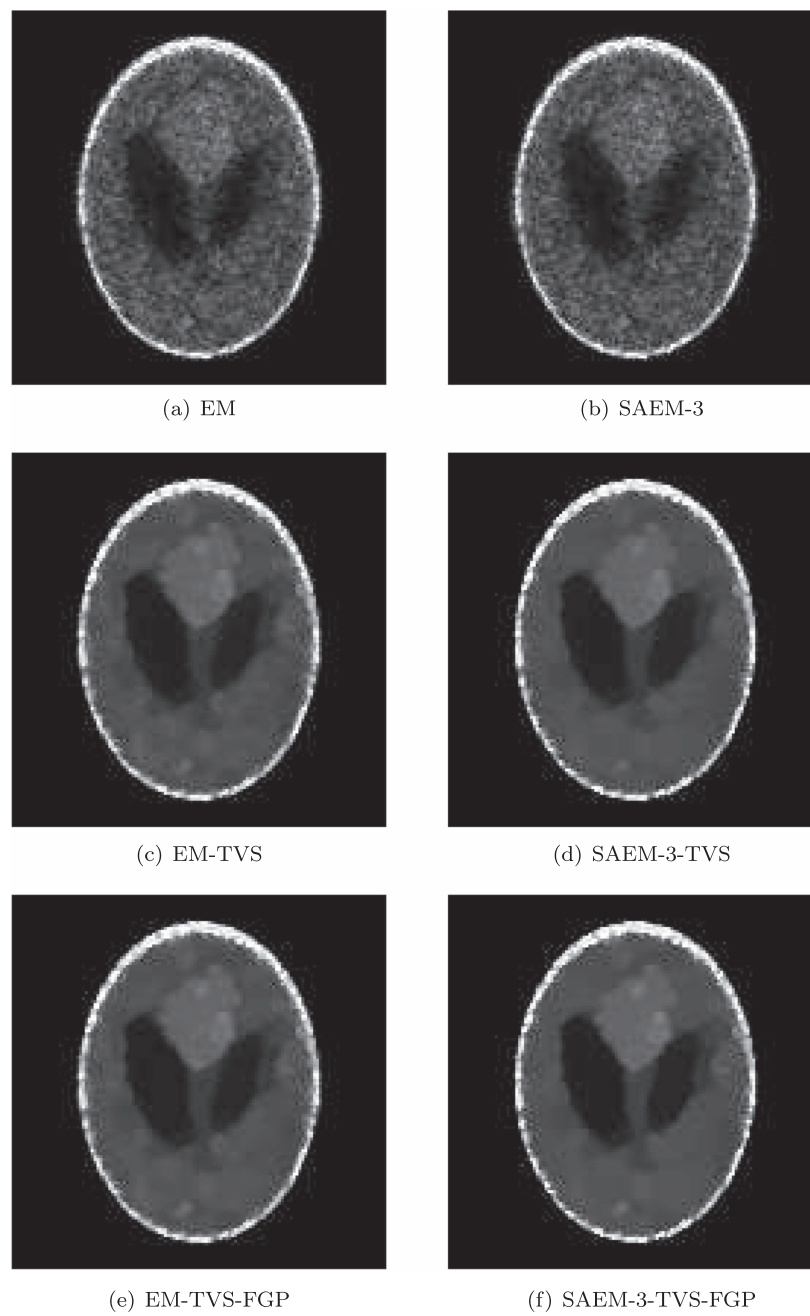


Figure 4. Visual comparison between EM and SAEM images.

proximal operator to be more efficient than subgradient, in the sense that the superiorized sequence is closer to the Pareto optimal curve, but without much more computational cost per iteration.

4.1.2. Numerical results. The tests compared EM and SAEM, considering the use or not of a superiorization sequence. In figure 2 we show the comparison with the standard superiorization sequence. The plot in figure 2(a) shows one realization of the experiment where the estimation error from the current solution to the original phantom image \mathbf{x}^* . The green boxes in the error curves correspond to the first image in the iteration process, of each method, to satisfy the stopping criteria. The reconstructed images from one of the 15 repetitions are shown in figure 4 and table 1 brings some statistical information about experiment.

Figure 2(b) shows a curve of Kullback–Leibler distance versus total variation measurement at each iteration. This figure, which resembles an L -curve [6], helps to see how much the total variation measurement increases while the iterations runs reducing the Kullback–Leibler distance, basically moving from the lower right to the upper left of figure 2(b). The points corresponding to the images satisfying the stopping criteria are also marked in green boxes on the plots.

In figure 3 we compare the effect of the two superiorization sequences, the standard one against the proximal, produced by FGP. The plot in figure 3(a) shows the estimation error from the current solution to the original phantom image \mathbf{x}^* . Now the points corresponding to images satisfying the stopping criteria are marked as red boxes. The plot in figure 3(b) shows a curve of Kullback–Leibler distance and total variation measurement. This plot also shows, as iteration runs, an evolution of the results moving the lower right to the upper left of figure 3(b). Note that the final result according to the stopping criteria never reaches the upper left part, it ends in the red boxes which is the first iteration where $f(\mathbf{x}^{(k)}) \leq 400$.

4.2. SSAEM

Now, the problem is to reconstruct images from transmission data. Therefore, the maximum likelihood model is

$$\begin{aligned} \min \quad & \sum_{i=1}^m \{ \beta_i e^{-(R\mathbf{x})_i} - \alpha_i \log(e^{-(R\mathbf{x})_i} + \rho_i) \} \\ \text{s.t. :} \quad & \mathbf{x} \in \mathbb{R}_+^n. \end{aligned}$$

In this case, we split the data in s subsets (following the same construction rules than the strings) S_l , $l \in \{1, 2, \dots, s\}$, such that

$$f(\mathbf{x}) = \sum_{l=1}^s F_l(\mathbf{x}),$$

where

$$F_l(\mathbf{x}) = \sum_{i \in S_l} \{ \beta_i e^{-(R\mathbf{x})_i} - \alpha_i \log(e^{-(R\mathbf{x})_i} + \rho_i) \}.$$

Therefore

$$\frac{\partial F_l}{\partial x_j}(\mathbf{x}) = \sum_{i \in S_l} r_{ij} \beta_i e^{-(R\mathbf{x})_i} \left\{ \frac{\alpha_i}{\beta_i e^{-(R\mathbf{x})_i} + \rho_i} - 1 \right\}.$$

Tomographic setup. Data for this set of experiments was obtained at the Imaging Beamline (IMX) of the Brazilian National Synchrotron Light Source (LNLS) through illumination of an apple seed by X-rays. Each radiographic image was 2048×2048 pixels in size and the sample was rotated by $\pi/512$ between each of the 512 images, thereby totaling, for

Table 1. Statistical indicators at the point where the stopping criterion was reached (only for SSIM higher is better). The algorithms were repeated 15 times with different noise. The values shown are the sample means with 99% confidence interval.

	EM	SAEM-3	EM-TVS	SAEM-3-TVS	EM-TVS-FGP	SAEM-3-TVS-FGP
KL	389.5 ± 21.9	329.5 ± 103.4	395.9 ± 5.7	398.6 ± 3.0	397.8 ± 4.4	398.1 ± 3.7
TV	935.2 ± 62.0	1076.9 ± 165.4	612.7 ± 50.6	670.9 ± 54.1	592.1 ± 49.7	653.0 ± 54.1
MSE	10.6 ± 0.5	11.0 ± 0.6	9.2 ± 0.6	9.4 ± 0.7	9.2 ± 0.6	9.4 ± 0.8
SSIM	0.72 ± 0.01	0.71 ± 0.01	0.85 ± 0.01	0.85 ± 0.01	0.85 ± 0.01	0.86 ± 0.01
iter.	21.2 ± 1.3	4.8 ± 1.2	47.1 ± 4.5	24.5 ± 9.0	46.0 ± 5.8	30.2 ± 6.3
time	2.9 ± 0.2 s	0.4 ± 0.2 s	8.8 ± 0.9 s	4.6 ± 1.7 s	7.1 ± 0.9 s	3.8 ± 0.9 s

each slice to be reconstructed, 512 views times 2048 rays of tomographic count data α_i . Reconstructed images have dimensions of 2048×2048 pixels.

Blank scans were measured before and after the actual tomographic measurements of the sample and then data β_i was obtained by linear interpolation between these two measurements. This is necessary because because LNLS is a 2nd generation synchrotron that does not operate in a top-up mode, i.e., the storage ring is not injected periodically in order to maintain constant current. Therefore, flux is not constant and decreases exponentially with time. As this measurement was obtained with a short exposure time, we expect that ring current (hence flux) behaves approximately as a linear model. A 4th generation synchrotron source, such as the forthcoming brazilian source, Sirius, will operate in a top-up mode with more flux, i.e., samples will be measured with a shorter exposure time.

It is important to note that iterative methods such as those presented in this paper are extremely attractive for ultra-fast tomographic experiments, with an ultra-short exposure time and a considerable amount of noise. This is the case of soft-tissue samples where low dose should be considered and a analytical reconstruction method like filtered backprojection [11] produce low quality reconstructed images with strong streak artifacts.

4.2.1. Parameter selection. Scaling matrix. For this algorithm we have used

$$p_j = \sum_{i=1}^m r_{ij} \{\alpha_i - \rho_i\},$$

which provides a reasonable scaling in the sense that if the approximation $\alpha_i \approx \beta_i e^{-(R\mathbf{x}^{(k)})_i} + \rho_i$ holds, then iterations of the form $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - D(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)})$ become close to a multiplicative method similar to the EM for emission tomography, and thus it can be written approximately as a product of coordinates of $\mathbf{x}^{(k)}$ with corresponding ratios of expected by observed quantities. The lower threshold was set to be $\tau = 10^{-14}$.

Subsets formation. Notation for SSAEM has a different meaning, where SSAEM- s represents SSAEM using s subsets of data and always only one string. The construction of the subsets was done by including all rays in a sequence of views. This sequence of views had approximately the same size on each subset. Subset processing within the string was done selecting a different random permutation for each iteration. Because only one string was used, the experimented algorithm could be described as a less general stabilized BRAMLA. This is because it was suitable for massive paralelization in a GPU which was necessary given the large size of the dataset. However, if more than one GPU were to be used, one string for each of them could be a reasonable way of taking advantage of the multi-GPU setup.

Stepsize sequence. As stepsize sequence for SSAEM- s and its superiorized version we have used

$$\lambda_k = \frac{\lambda_{0,s}}{(ks + 1)^{0.25}},$$

where $\lambda_{0,s}$ was the largest value such that the first iteration of SSAEM- s would not lead to a negative value.

Superiorization sequence. The method termed SSAEM-TV- s corresponds to the superiorized version of SSAEM- s where the superiorization was obtained by application of (11) to the result of each SSAEM- s iteration. In this case, $N = 50$ and the sequence $\{\gamma_k\}$ was chosen to be

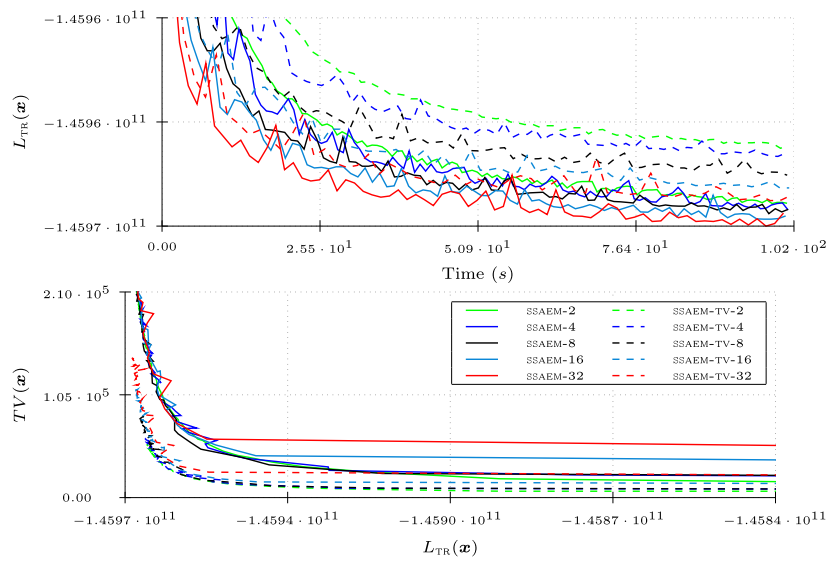


Figure 5. Comparison of several SSAEM variations and its respective superiorized versions. Top: objective function value evolution over time. Bottom: objective function versus total variation. Legend is the same for both graphics.

$$\gamma_k = \frac{\gamma_{0,s}}{(ks + 1)^{0.35}}.$$

Now, $\gamma_{0,s}$ was selected so that

$$\frac{\|\mathbf{x}^{(1/2)} - \mathbf{x}^{(1)}\|}{\|\mathbf{x}^{(0)} - \mathbf{x}^{(1/2)}\|} \approx 10^{-2}$$

in the following manner. First, $\mathbf{x}^{(1/2)}$ is computed normally using the stepsize λ_0 as described above. Then, a tentative value $\tilde{\mu}_{0,s} = 1$ is used in order to compute $\tilde{\mathbf{x}}^{(1)}$. Then we set

$$\mu_{0,s} = 10^{-2} \frac{\|\mathbf{x}^{(0)} - \mathbf{x}^{(1/2)}\|}{\|\mathbf{x}^{(1/2)} - \mathbf{x}^{(1)}\|}.$$

4.2.2. Numerical results. The left side of figure 5 shows the objective function value as a function of computation time for SSAEM- s and SSAEM-TV- s with $s \in \{2, 4, 8, 16, 32\}$. Notice that increasing the number of subsets speeds up convergence in terms of reduction of objective function value over time. Superiorization reduces this convergence speed, but the fastest methods without superiorization give rise to the fastest superiorized methods as a rule of thumb.

The graphic on the right of figure 5 displays plots of the objective function value versus total variation for the iterates of each method. The curves make clear that the superiorized version of the algorithms always present the better compromise between data adhesion and smoothness when compared to its respective non-superiorized version. Figure 6 shows how this property translates to better image quality in practice. The lower row of figure 6 shows isocontours overlaid on top of details of the images, in order to show that the superiorized image is noticeably smoother, therefore potentially more useful for visualization tasks. If, on

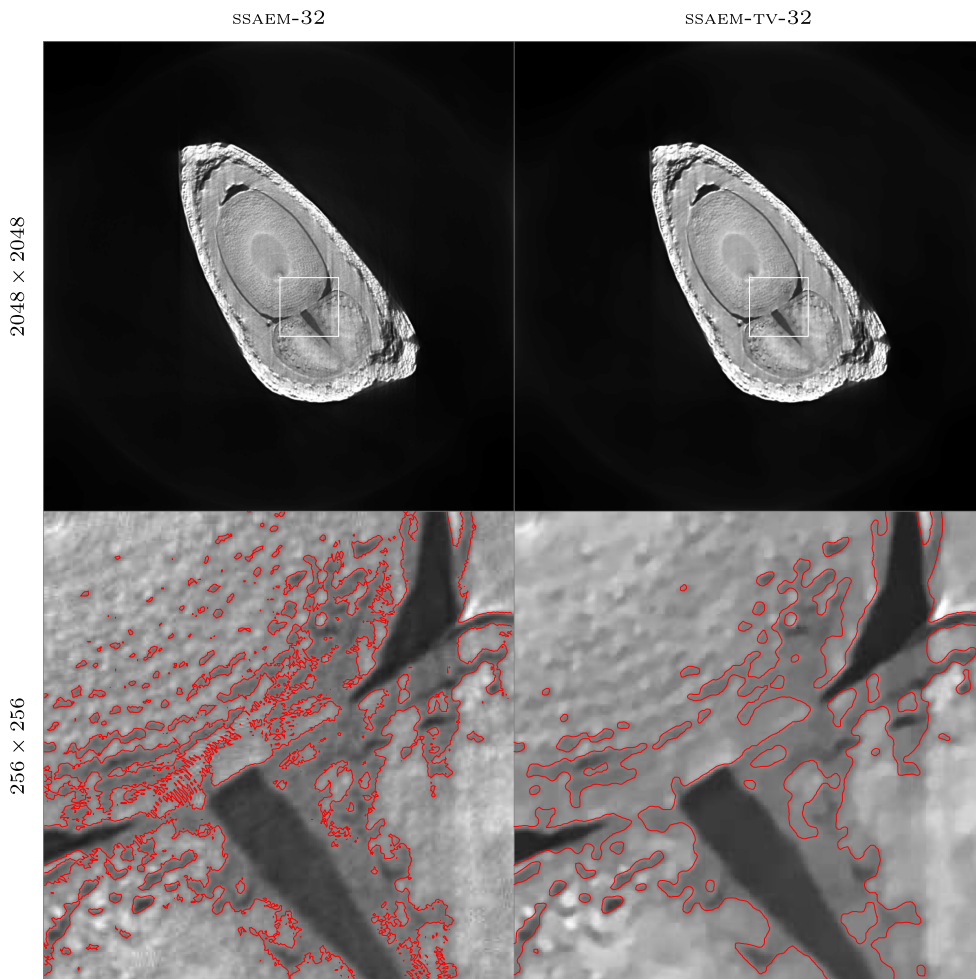


Figure 6. Reconstructions of apple seed slice from synchrotron radiation transmission data. Top-row: full slice images. Bottom row: details, with overlaid isocontour and location in respective images above shown as a white square. These images are the first iterates to satisfy $L_{\text{TR}}(\mathbf{x}^{(k)}) \leq -1.4596 \times 10^{11}$.

the other hand, some apparent detail seems to have been removed from the image, part of these fine details may be reconstruction artifacts but we make no claims in this direction. Notice that these results are in accordance with those obtained using SAEM and its superiorized versions presented before.

5. Conclusions

We have studied the possibility of superiorizing incremental algorithms for maximum likelihood tomographic image reconstruction both from the theoretical and practical point of view. Several options for the superiorization were evaluated, all of which showed promising results. Furthermore, two maximum likelihood algorithms were studied, namely SAEM [7] and the newly introduced SSAEM.

Provided numerical evidence indicates that the extra computational burden posed by the superiorization approach is justifiable given the observed improvement in image quality. Furthermore, the new algorithm SSAEM was shown to have better convergent properties than SAEM and related methods while retaining its good practical characteristics.

Acknowledgments

The authors have been partially funded by the FAPESP grant 2013/07375-0. ESH was partially supported by the CNPq grant 311476/2014-7, MVWZ was partially supported by the CNPq grant 475553/2013-6, EXM was partially supported by the CNPq grant 442000/2014-6. We are grateful to LNLS for providing the beamtime for the acquisition of tomographic data for the apple seed reconstruction experiments.

References

- [1] Beck A and Teboulle M 2009 Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems *IEEE Trans. Image Process.* **18** 2419–34
- [2] Browne J and De Pierro Á R 1996 A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography *IEEE Trans. Med. Imaging* **15** 687–99
- [3] Chambolle A 2004 An algorithm for total variation minimization and applications *J. Math. Imaging Vis.* **20** 89–97
- [4] De Pierro Á R and Yamagishi M E B 2001 Fast EM-like methods for maximum ‘*a posteriori*’ estimates in emission tomography *IEEE Trans. Med. Imaging* **20** 280–8
- [5] Garduño E and Herman G T 2014 Superiorization of the ML-EM algorithm *IEEE Trans. Nucl. Sci.* **61** 162–72
- [6] Hansen P C 1992 Analysis of discrete ill-posed problems by means of the *L*-curve *SIAM Rev.* **34** 561–80
- [7] Helou E S, Censor Y, Chen T-B, Chern I-L, De Pierro Á R, Jiang M and Lu H H-S 2014 String-averaging expectation-maximization for maximum likelihood estimation in emission tomography *Inverse Problems* **30** 055003
- [8] Helou N E S and De Pierro Á R 2005 Convergence results for scaled gradient algorithms in positron emission tomography *Inverse Problems* **21** 1905–14
- [9] Hiriart-Urruty J-B and Lemaréchal C 1993 *Convex Analysis and Minimization Algorithms (A Series of Comprehensive Studies in Mathematics)* (Berlin: Springer)
- [10] Hudson H M and Larkin R S 1994 Accelerated image reconstruction using ordered subsets of projection data *IEEE Trans. Med. Imaging* **13** 601–9
- [11] Natterer F 1986 *The Mathematics of Computerized Tomography* (New York: Wiley)
- [12] Rashed E A and Kudo H 2013 Towards high-resolution synchrotron radiation imaging with statistical iterative reconstruction *J. Synchrotron Radiat.* **20** 116–24
- [13] Shepp L A and Vardi Y 1982 Maximum likelihood reconstruction for emission tomography *IEEE Trans. Med. Imaging* **1** 113–22
- [14] Tanaka E and Kudo H 2003 Subset-dependent relaxation in block-iterative algorithms for image reconstruction in emission tomography *Phys. Med. Biol.* **48** 1405–22
- [15] Vardi Y, Shepp L A and Kaufman L 1985 A statistical model for positron emission tomography *J. Am. Stat. Assoc.* **80** 8–20
- [16] Wang Z, Bovik A C, Sheikh H R and Simoncelli E P 2004 Image quality assessment: from error visibility to structural similarity *IEEE Trans. Image Process.* **13** 600–12