NUMERICAL MATHEMATICS: Theory, Methods and Applications
Numer. Math. Theor. Meth. Appl., Vol. **xx**, No. x, pp. 1-15 (200x)

Iterative methods of Richardson-Lucy-type for image deblurring

M. K. Khan[1] and S. Morigi[2] and L. Reichel[1] and F. Sgallari[4*]

[1] *Department of Mathematical Sciences*
Kent State University,
Kent, OH 44242, USA.
kazim@math.kent.edu
reichel@math.kent.edu
[2] *Department of Mathematics*
University of Bologna
P.zza Porta San Donato, 5, Bologna, Italy
serena.morigi@unibo.it
[4] *CIRAM*
University of Bologna
Via Saragozza, 8, Bologna,Italy
fiorella.sgallari@unibo.it

Abstract. Image deconvolution problems with a symmetric point-spread function arise in many areas of science and engineering. These problems often are solved by the Richardson-Lucy method, a nonlinear iterative method. We first show a convergence result for the Richardson-Lucy method. The proof sheds light on why the method may converge slowly. Subsequently, we describe an iterative active set method that imposes the same constraints on the computed solution as the Richardson-Lucy method. Computed examples show the latter method to yield better restorations than the Richardson-Lucy method and typically require less computational effort.

1. Introduction

This paper is concerned with the restoration of images that have been contaminated by blur and noise. We consider two-dimensional gray-scale images, whose brightness is represented by a real-valued function defined on a square or rectangular region $\Omega \in \mathbb{R}^2$. Let the function b^δ represent the available observed blur- and noise-contaminated image, and let the function $\hat{x}$ represent the unknown associated blur- and noise-free image that we would like to recover. We assume the functions b^δ and $\hat{x}$ to be related by the

*Corresponding author.

degradation model

$$b^\delta(s) = \int_\Omega h(s,t)\hat{x}(t)dt + \eta^\delta(s), \qquad s \in \Omega, \tag{1.1}$$

where η^δ represents additive noise (error) in the data b^δ. The kernel h models the blurring and is often referred to as the point-spread function (PSF). In many applications, the integral is a symmetric convolution, i.e., the kernel h is of the form

$$h(s,t) = h(t,s) = k(s-t), \qquad s,t \in \Omega, \tag{1.2}$$

for some function k. This situation is of primary interest to us; however, the method described in Section 3 can be applied to restoration problems with a more general kernel. We also will assume that

$$\int_\Omega h(s,t)ds = 1, \qquad t \in \Omega. \tag{1.3}$$

Then the blurring does not change the total brightness of the image.

The PSF in typical image restoration problems is smooth or piecewise smooth with jump discontinuities. The integral operator (1.1) then is compact and therefore its singular values cluster at the origin. Consequently, the integral operator does not have a bounded inverse.

We would like to determine an accurate approximation of the unknown blur- and noise-free image $\hat{x}$ when the observed image b^δ and the kernel h are available. Straightforward solution of

$$\int_\Omega h(s,t)x(t)dt = b^\delta(s), \qquad s \in \Omega, \tag{1.4}$$

for x generally does not yield a useful approximation of the desired blur- and noise-free image $\hat{x}$ because of the noise η^δ in b^δ and the fact that the integral operator does not have a bounded inverse. Due to the latter, the task of solving (1.4) is an ill-posed problem; see, e.g., Engl et al. [4] for discussions on ill-posed problems and on numerical methods for their solution.

We seek to determine an accurate approximation of $\hat{x}$ by computing a suitable approximate solution of (1.4). It is generally beneficial to impose constraints known to be satisfied by $\hat{x}$ on the computed approximation during the solution process. Since $\hat{x}$ represents the brightness of the image, it is nonnegative. We would like the computed approximation of $\hat{x}$ to satisfy the same constraint, i.e., would like our solution method to determine an approximate solution x of (1.4) that satisfies

$$x(t) \geq 0, \qquad t \in \Omega. \tag{1.5}$$

Integrating (1.4) with respect to s and using (1.3) yields

$$\int_\Omega x(t)dt = \int_\Omega b^\delta(s)ds. \tag{1.6}$$

We also would like the computed approximate solution to satisfy this constraint.

The present paper discusses two methods for determining approximate solutions of (1.4) that satisfy the constraints (1.5) and (1.6). The first method considered is the classical Richardson-Lucy method introduced by Richardson [13] and Lucy [8]. This method requires the kernel h to be nonnegative and of the form (1.2). It is a nonlinear iterative method, whose convergence may be quite slow. Nevertheless, the method is popular; see, e.g., [1, 3, 14–17] for a variety of recent applications. Richardson-Lucy iteration also is applied to solve blind deconvolution problems; see [2,5]. We review the Richardson-Lucy method in Section 2 and discuss some of its properties. In particular, we show convergence for a special class of functions. The proof sheds light on why Richardson-Lucy iteration may converge slowly. Section 3 describes an iterative active set method for the solution of (1.4). This method uses Lanczos bidiagonalization to reduce the large restoration problem to a smaller one, and applies an active set method to compute an approximation of $\hat{x}$ that satisfies the constraints (1.5) and (1.6). Section 4 presents a few computed examples that compare Richardson-Lucy iteration with the active set method of Section 3. Concluding remarks and comments on extensions can be found in Section 5.

In the computed examples of this paper, we will assume that a bound for the norm of the error η^δ is available. This allows us to terminate the iterations with the aid of the discrepancy principle. However, the solution methods discussed also can be applied with other stopping criteria, such as generalized cross validation; see, e.g., [12] for an illustration. A variety of so-called heuristic parameter choice rules and their properties have recently been investigated in [7].

2. The Richardson-Lucy method

We define the Richardson-Lucy method for $\Omega = \mathbb{R}^p$ with $p \geq 1$ an integer. This choice of Ω avoids boundary effects. Richardson [13] and Lucy [8] proposed the following nonlinear iterative method to determine a sequence of approximations x_r of the blur- and noise-free image $\hat{x}$:

$$x_{r+1}(u) = x_r(u) \int_{\mathbb{R}^p} \frac{k(s-u)b^\delta(s)}{\int_{\mathbb{R}^p} k(s-t)x_r(t)dt} ds, \quad u \in \mathbb{R}^p, \quad r = 0, 1, \ldots \ , \tag{2.1}$$

with $x_0 = b^\delta$. We refer to (2.1) as the Richardson-Lucy method, or briefly as the RL method. It is easy to verify that the iterates x_r are nonnegative,

$$x_r(t) \geq 0, \qquad t \in \mathbb{R}^p, \qquad r = 1, 2, \ldots \ ,$$

provided that

$$k(t) \geq 0, \qquad x_0(t) \geq 0, \qquad t \in \mathbb{R}^p.$$

We assume the latter inequalities to hold. Moreover, the x_r satisfy

$$\int_{\mathbb{R}^p} x_r(s)ds = \int_{\mathbb{R}^p} b^\delta(t)dt, \qquad r = 0, 1, \ldots \ .$$

Thus, the RL iterates x_r satisfy the conditions (1.5) and (1.6).

The main drawback of the RL method is that for many restoration problems a fairly large number of iterations are required to determine an accurate approximation of $\hat{x}$; see, e.g., Section 4. After discretization, each iteration requires the evaluation of two matrix-vector products with the blurring matrix.

The remainder of this section discusses some properties of the RL method, which shed light on the convergence behavior. For notational simplicity, we let $\Omega = \mathbb{R}$. However, our discussion, suitably modified, can be extended to higher space-dimensions.

We first note that there are problems for which the RL method convergences rapidly. Let $\Omega = \mathbb{R}$ and let $k(t)$ be a symmetric probability kernel (density with respect to the measure ν) with mean one. Then for any linear signal $x(t) = c_0 + c_1 t$, the convolution transform reproduces the signal, i.e.,

$$\int_{\mathbb{R}} k(s-t)x(t)d\nu(t) = x(s), \qquad s \in \mathbb{R}.$$

Therefore, if the RL method is started with a linear signal $x_0 = b^\delta$, then the method converges in one step. Thus, the linear functions are eigenfunctions of the integral operator associated with the eigenvalue one. This observation suggests that we may gain insight into the convergence properties of the RL method by identifying invariant sets of the convolution transform.

Consider the Gaussian kernel in one space-dimension

$$k(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-t^2/2\right), \qquad t \in \mathbb{R},$$

and the associated convolution transform

$$\int_{\mathbb{R}} \frac{1}{\sigma} k((t-u)/\sigma)x(u)du, \tag{2.2}$$

where the parameter $\sigma > 0$ determines the spread of the kernel. It follows from the above discussion that the linear functions are eigenfunctions of the transform (2.2) with eigenvalue one. The theorem below describes a nontrivial invariant set for this transform and discusses convergence.

Theorem 2.1. *The space of bell-shaped functions*

$$x(t) = \widehat{a} \exp\left(c_0 t - c_1 t^2\right), \tag{2.3}$$

where $\widehat{a}, c_0, c_1$ are real constants with $c_1 > 0$, forms an invariant set for the Gaussian convolution transform (2.2)*. The RL method, when started with an observed signal of the form* (2.3)*, converges point-wise to the input signal.*

Proof. The functions (2.3) can be expressed as

$$x(t) = a \exp\left(-c(t-\mu)^2\right) \tag{2.4}$$

for certain constants $c > 0$ and $a, \mu \in \mathbb{R}$. Let f be a function of the form (2.4) and introduce the "blurred" observed signal

$$H(t) = \int_{\mathbb{R}} \frac{1}{\sigma} k((t-u)/\sigma) f(u) du$$

associated with the original "unknown" signal f. The observed signal can be expressed as

$$\begin{aligned} x_0(t) &= H(t) = a \int_{\mathbb{R}} \frac{1}{\sigma} k((t-\mu-u)/\sigma) \exp\left(-cu^2\right) du \\ &= \frac{a}{\sqrt{1+2c\sigma^2}} \exp\left(-\frac{c(t-\mu)^2}{1+2c\sigma^2}\right) \\ &= a_0 \exp\left(-c_0(t-\mu)^2\right), \end{aligned}$$

where

$$a_0 = \frac{a}{\sqrt{1+2c\sigma^2}}, \qquad c_0 = \frac{c}{1+2c\sigma^2}. \tag{2.5}$$

Using x_0 as the initial function for the RL method, the next iterate can be written as

$$\begin{aligned} x_1(t) &= x_0(t) \int_{\mathbb{R}} \frac{\frac{1}{\sigma} k((t-u)/\sigma) H(u) du}{\int_{\mathbb{R}} \frac{1}{\sigma} k((u-v)/\sigma) x_0(v) dv} \\ &= e^{-c_0(t-\mu)^2} \int_{\mathbb{R}} \frac{\frac{1}{\sigma} k((t-u)/\sigma)\, a_0 e^{-c_0(u-\mu)^2} du}{\int_{\mathbb{R}} \frac{1}{\sigma} k((u-v)/\sigma) e^{-c_0(v-\mu)^2} dv} \\ &= e^{-c_0(t-\mu)^2} \int_{\mathbb{R}} \frac{\frac{1}{\sigma} k((t-u)/\sigma) a_0 e^{-c_0(u-\mu)^2} du}{\frac{1}{\sqrt{1+2c_0\sigma^2}} e^{-c_0(u-\mu)^2/(1+2c_0\sigma^2)}}. \end{aligned}$$

Further calculations show that

$$x_1(t) = a_1 \exp\left(-c_1(t-\mu)^2\right),$$

where

$$a_1 = \frac{a_0\sqrt{1+2c_0\sigma^2}}{\sqrt{1+2\sigma^2\left(c_0 - \frac{c_0}{1+2c_0\sigma^2}\right)}}, \qquad c_1 = c_0 + \frac{c_0 - \frac{c_0}{1+2\sigma^2 c_0}}{1+2\sigma^2\left(c_0 - \frac{c_0}{1+2\sigma^2 c_0}\right)}.$$

Repeating the iterations, we obtain

$$x_r(t) = a_r \exp\left(-c_r(t-\mu)^2\right), \qquad r = 2, 3, \dots ,$$

with

$$a_{r+1} = \frac{a_0\sqrt{1+2c_r\sigma^2}}{\sqrt{1+2\sigma^2\left(c_0 - \frac{c_r}{1+2c_r\sigma^2}\right)}}, \qquad c_{r+1} = c_r + \frac{c_0 - \frac{c_r}{1+2\sigma^2 c_r}}{1+2\sigma^2\left(c_0 - \frac{c_r}{1+2\sigma^2 c_r}\right)}. \tag{2.6}$$

This establishes that the functions (2.3) form an invariant set.

We turn to the convergence of the coefficients c_r. Introduce the function

$$h(u) = \frac{u}{1 + 2\sigma^2 u},$$

which satisfies

$$\begin{aligned} h(u) &< \min\left\{u, \frac{1}{2\sigma^2}\right\}, && (2.7)\\ 0 &< h'(u) < 1 && (2.8) \end{aligned}$$

for $0 < u < \infty$.

The recursion formula (2.6) for the coefficients c_r can be written as

$$c_{r+1} = c_r + \frac{c_0 - h(c_r)}{1 + 2\sigma^2(c_0 - h(c_r))}, \qquad r = 0, 1, \dots\ . \tag{2.9}$$

Assume for a moment that the sequence $c_0, c_1, \dots$ has the limit α. Then

$$\alpha = \alpha + \frac{c_0 - h(\alpha)}{1 + 2\sigma^2(c_0 - h(\alpha))}.$$

Solving for α yields

$$\alpha = \frac{c_0}{1 - 2\sigma^2 c_0} = c,$$

where the right-hand side equality follows from (2.5). Moreover,

$$0 < h(\alpha) = c_0 < c. \tag{2.10}$$

We will show that the coefficients c_r, $r = 1, 2, \dots$, converge monotonically to c.

Assume that $0 < c_r < \alpha$. These inequalities hold for c_0. It follows from (2.9) that $c_{r+1} > c_r$. Writing (2.9) as

$$c_{r+1} = c_r + h(c_0 - h(c_r))$$

and using (2.7), (2.10), and (2.8), in order, yields

$$c_{r+1} < c_r + c_0 - h(c_r) = c_r + (h(\alpha) - h(c_r)) = c_r + h'(\xi)(\alpha - c_r) < \alpha,$$

for some $c_r < \xi < \alpha$. Thus, $c_r < c_{r+1} < \alpha$. These inequalities hold for $r = 0, 1, \dots$, and show the monotonic convergence of the coefficients c_r. Convergence of the c_r, $r = 1, 2, \dots$, secures that the coefficients a_r, $r = 1, 2, \dots$, in (2.6) converge as well. This, in turn, implies point-wise convergence of the iterates $x_1, x_2, x_3, \dots$ to the function f.

The above proof shows that convergence of the RL method may be slow when σ is large, because then the corrections $c_{r+1} - c_r$ are "tiny."

3. An iterative active set method

The method of this section is based on the iterative active set scheme described in [10] for finite dimensional problems. We therefore consider a discretization

$$Ax = b^\delta \tag{3.1}$$

of (1.4).

Let the available image that we would like to restore be represented by an $n \times n$ array of pixels. Ordering these pixels column-wise yields the right-hand side $b^\delta \in \mathbb{R}^m$ of (3.1) with $m = n^2$. The matrix $A \in \mathbb{R}^{m\times m}$ in (3.1) represents a discretization of the integral operator in (1.4), and the entries of $x \in \mathbb{R}^m$ are pixel values, ordered column-wise, of an approximation of the desired blur- and noise-free image.

The entries of b^δ are contaminated by noise. Let $b \in \mathbb{R}^m$ be the associated vector with the unknown noise-free entries, i.e.,

$$b^\delta = b + \eta^\delta, \tag{3.2}$$

where the vector η^δ represents the noise. In the present section and in the computed examples, we will assume that a fairly accurate bound

$$\|\eta^\delta\| \leq \delta \tag{3.3}$$

is known, where $\|\cdot\|$ denotes the Euclidean vector norm, and that the linear system of equations with the noise-free right-hand side,

$$Ax = b, \tag{3.4}$$

is consistent.

Let $\hat{x} \in \mathbb{R}^m$ denote the solution of minimal Euclidean norm of (3.4). We are interested in computing an approximation of $\hat{x}$ that satisfies discrete analogues of the constraints (1.5) and (1.6).

The iterative active set method in [10] is designed to determine an approximate solution of the constraint minimization problem

$$\min_{x\in\mathbb{S}} \|Ax - b^\delta\|,$$

where $\mathbb{S} \subset \mathbb{R}^m$ is a convex set of feasible solutions defined by box constraints.

A vector $x \in \mathbb{R}^m$ is said to satisfy the discrepancy principle if

$$\|Ax - b^\delta\| \leq \gamma\delta, \tag{3.5}$$

where $\gamma > 1$ is a user-chosen constant. The size of γ depends on the accuracy in the estimate δ. If δ is know to be a tight bound for the norm of the noise, then γ is generally chosen to be close to unity. We note that the vector $\hat{x}$ satisfies (3.5).

The active set method [10] first determines an approximate solution of (3.1) with the LSQR iterative method. This is a minimal residual Krylov subspace method; see [11] for details. We use the initial iterate $x_0 = 0$ and terminate the iterations as soon as an iterate x_k that satisfies the discrepancy principle (3.5) has been found.

The vector x_k is not guaranteed live in $\mathbb{S}$. The scheme [10] therefore projects x_k orthogonally into $\mathbb{S}$ and if necessary applies an active set method to determine an approximate solution of (3.1) that lies in $\mathbb{S}$ and satisfies the discrepancy principle. The active set method uses LSQR. We have found it useful to iterate with LSQR until the discrepancy principle (3.5) is satisfied after each update of the active set. Each update generally affects several of the components of the computed approximate solutions; see [10]. A related method and theoretical results are shown in [9]; we will comment on the latter method below.

We now describe how the outlined active set method can be applied to enforce a constraint analogous to (1.6). Define the norm

$$\|x\|_1 = \sum_{j=1}^{m} |x_j|, \qquad x = [x_1, x_2, \ldots, x_m]^T.$$

Let $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^m$ and assume that $x \geq 0$. Here and below vector inequalities are understood element-wise. Then the constraint

$$\|x\|_1 = \|b^\delta\|_1$$

can be expressed as the linear constraint

$$e^T x = \|b^\delta\|_1, \tag{3.6}$$

which is a discrete analogue of (1.6).

Introduce the orthogonal projectors

$$P_e = \frac{1}{m} e e^T, \quad P_e^\perp = I - P_e, \quad P_{Ae} = \frac{Ae(Ae)^T}{\|Ae\|^2}, \quad P_{Ae}^\perp = I - P_{Ae}.$$

We use these projectors to split the computed approximate solution into

$$x = P_e^\perp x + P_e x. \tag{3.7}$$

The condition (3.6) can be expressed as

$$P_e x = \frac{1}{m} \|b^\delta\|_1 e. \tag{3.8}$$

We would like to determine an approximate solution x of (3.1) that satisfies (3.8). The linear system of equations (3.1) can be written as

$$\begin{aligned} P_{Ae} A P_e x + P_{Ae} A P_e^\perp x &= P_{Ae} b^\delta, \\ P_{Ae}^\perp A P_e x + P_{Ae}^\perp A P_e^\perp x &= P_{Ae}^\perp b^\delta. \end{aligned}$$

It is easy to see that $P_{Ae}^{\perp} A P_e = 0$. Therefore, the second equation simplifies to

$$P_{Ae}^{\perp} A P_e^{\perp} x = P_{Ae}^{\perp} b^{\delta}. \tag{3.9}$$

Using (3.8), the nonnegativity constraint (1.5) can be expressed as

$$0 \leq P_e x + P_e^{\perp} x = \frac{1}{m} \|b^{\delta}\|_1 e + P_e^{\perp} x,$$

which yields

$$P_e^{\perp} x \geq -\frac{1}{m} \|b^{\delta}\|_1 e. \tag{3.10}$$

The images in the computed examples of Section 4 are represented by 8-bit pixels. Therefore, each entry of the solution of (3.1) should not exceed $2^8 - 1 = 255$. Consequently, the entries of $P_e^{\perp} x$ should satisfy the upper bound

$$P_e^{\perp} x \leq (255 - \frac{1}{m} \|b^{\delta}\|_1) e. \tag{3.11}$$

We apply the active set method [10] to the solution of (3.9) with the set $\mathbb{S}$ determined by the constraints (3.10) and (3.11). This yields the component of the solution orthogonal to the vector e, that is $P_e^{\perp} x$. The desired approximate solution of (3.1) is obtained according to (3.7), where $P_e x$ is given by (3.8). The following section illustrates the performance of this solution method.

4. Computed examples

This section demonstrates the performance of the active set method of Section 3 when applied to both synthetic and real images that have been contaminated by blur and noise. We compare the performance with that of the RL method described in Section 2.

Let $\tilde{x}$ be an available approximation of the desired blur- and noise-free image $\hat{x}$. The Peak Signal-to-Noise Ratio (PSNR),

$$\mathrm{PSNR}(\tilde{x}, \hat{x}) = 20 \log_{10} \frac{255}{\|\tilde{x} - \hat{x}\|} \text{ dB},$$

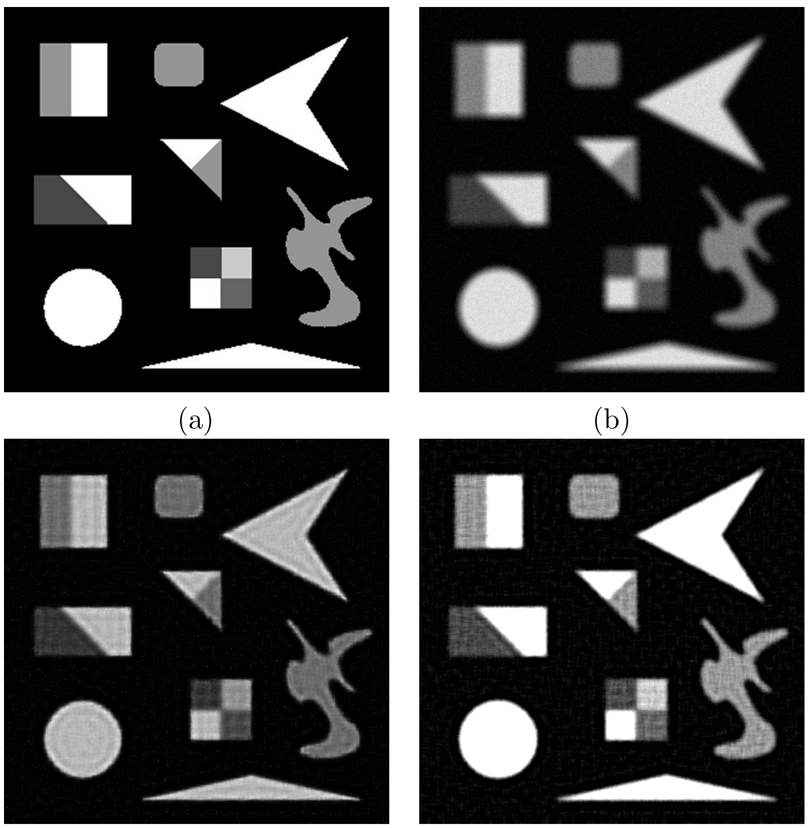
provides a quantitative measure of the quality of $\tilde{x}$. The norm $\|\tilde{x} - \hat{x}\|$ is the Root Mean Squared Error (RMSE) of $\tilde{x} - \hat{x}$. The numerator, 255, is the largest pixel-value that can be represented with 8 bits. A large PSNR-value indicates that $\tilde{x}$ is an accurate approximation of $\hat{x}$; however, the PSNR-values are not always in agreement with visual perception. All computations are carried out in MATLAB with about 16 significant decimal digits.

We assume an accurate estimate of the noise-level

$$\nu = \frac{\|\eta^{\delta}\|}{\|b\|}$$

to be available in all examples of this section, cf. (3.2)-(3.3), and therefore choose the parameter γ in (3.5) close to unity; we set $\gamma = 1.001$. The noise is white and Gaussian in all examples.

The matrix A in (3.1) in all examples is a symmetric block Toeplitz matrix with Toeplitz blocks and models a Gaussian blurring operator. It is determined with the MATLAB function `blur.m` from Regularization Tools [6]. This function has two parameters `band` and `sigma`. The former specifies the half-bandwidth of the Toeplitz blocks and the latter the variance of the Gaussian point spread function. The larger `sigma`, the more blurring. Enlarging `band` increases the storage requirement, the arithmetic work required for the evaluation of matrix-vector products with A, and to some extent the blurring.



(a) (b)

(c) (d)

Figure 1: Example 4.1. `corner` images: (a) blur- and noise-free image; (b) the corrupted image produced by Gaussian blur, defined by the parameters `band` $= 15$ and `sigma` $= 7$, and by 10% noise; (c) restoration with largest PSNR-value determined by the RL method; (d) restored image determined by the active set method.

Example 4.1. We consider the restoration of a blur- and noise-contaminated `corner` image represented by 512×512 pixels. Thus, the order of the blurring matrix

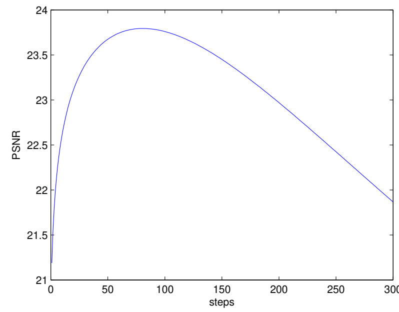


Figure 2: Example 4.1. Plot of the PSNR-value as a function of the number of RL iterations.

ν	PSNR_i	$\mathrm{PSNR}_{\mathrm{ASM}}$	$\mathrm{mvp}_{\mathrm{ASM}}$	$\mathrm{PSNR}_{\mathrm{RLd}}$	$\mathrm{mvp}_{\mathrm{RLd}}$	$\mathrm{PSNR}_{\mathrm{RL}}$	$\mathrm{mvp}_{\mathrm{RL}}$
0.05	20.28	26.44	68	22.22	8	25.60	80
0.10	19.80	24.40	49	20.85	2	23.34	48
0.30	16.59	19.29	21	18.03	2	18.45	16

Table 1: Example 4.1. Results for restorations of `corner` images that have been corrupted by significant Gaussian blur, defined by `band` $= 15$ and `sigma` $= 7$, and by noise corresponding to noise-level ν.

A is $m = 262144$. The desired blur- and noise-free image is depicted in Figure 1(a). A version of the image that has been contaminated by Gaussian blur, determined by the parameters `band` $= 15$ and `sigma` $= 7$, and by 10% noise is displayed in Figure 1(b).

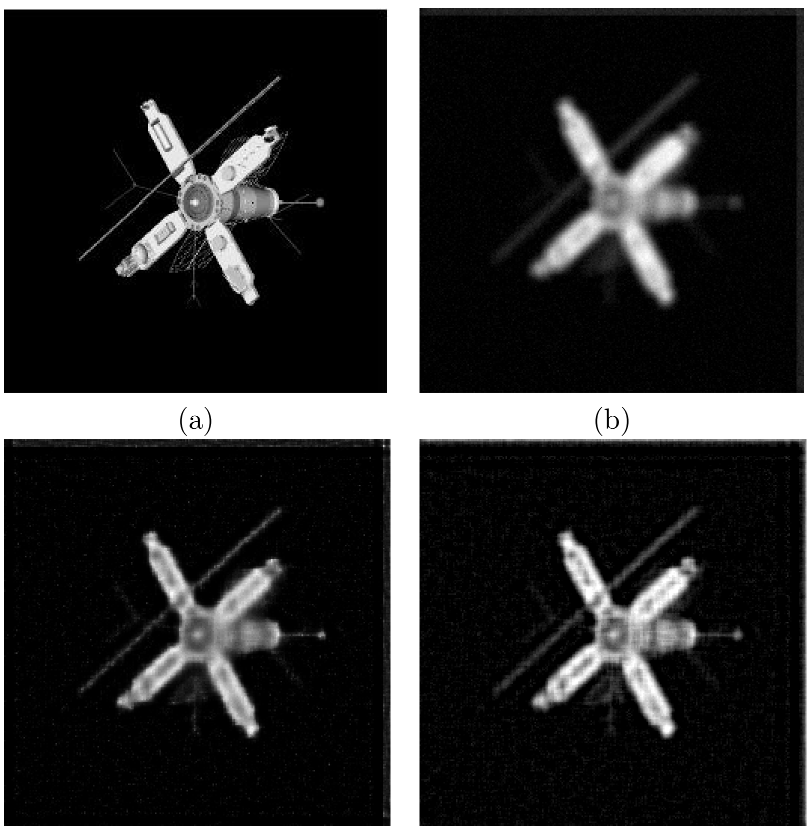
The second column of Table 1, with header PSNR_i, reports PSNR-values for images `corner` that have been corrupted by Gaussian blur, characterized by `band` $= 15$ and `sigma` $= 7$, and by noise of different noise-levels ν. The latter are shown in the first column. Column three, labeled $\mathrm{PSNR}_{\mathrm{ASM}}$, shows the PSNR-values for restorations determined by the active set method of Section 3. The column with header $\mathrm{mvp}_{\mathrm{ASM}}$ displays the number of matrix-vector product evaluations with the matrix A required by the active set method. The iterations with this method are terminated by using the discrepancy principle (3.5).

When the iterations with the RL method are terminated with the discrepancy principle, using the same value of γ, restorations with PSNR-values reported in column five of Table 1 are obtained. The required number of matrix-vector products are reported in column sixth under the header $\mathrm{mvp}_{\mathrm{RLd}}$. The RL method is seen to carry out only few iterations and produce restorations of lower quality than the active set method.

Running the RL method for 300 iterations, we noticed that the PSNR-values first increased and then decreased; see Figure 2. The seventh column of Table 1, labeled $\mathrm{PSNR}_{\mathrm{RL}}$, reports the largest PSNR-values achieved by the RL method during continued iterations, and column eight displays the number of matrix-vector product evaluations required to obtain a restoration with maximal PSNR-value. Comparing columns 3 and 7 of Table 1 shows that the active set method of Section 3 with the discrepancy principle determines restorations with larger PSNR-values evaluations than any restoration determined by the RL method. A comparison of columns 4 and 8 of Table 1 shows the active set method to require fewer matrix-vector product evaluations than the RL method when the noise-level ν is small. The saving in the number of matrix-vector product evaluations is the largest when ν is the smallest and the largest number of iterations are required. We recall that both the active set and RL methods require two matrix-vector product evaluations with A per iteration.

Figure 1(c) shows the best restoration of the image in Figure 1(b) achieved with the RL method. Thus, this restoration was obtained after 24 iterations, which required 48 matrix-vector product evaluations. Figure 1(d) displays the restoration computed with the active set method.

This example illustrates that it is important to terminate the iterations with the RL method before the quality of the computed restorations deteriorates. Moreover, the discrepancy principle is seen not to be a suitable stopping criterion for the method. □



(a) (b)

(c) (d)

Figure 3: Example 4.2. `satellite` images: (a) blur- and noise-free image; (b) the corrupted image produced by Gaussian blur, defined by the parameters `band` $= 9$ and `sigma` $= 7$, and by 10% noise; (c) restoration with largest PSNR-value determined by the RL method; (d) restored image determined by the active set method.

Example 4.2. The noise- and blur-free image `satellite` used in this example is

ν	PSNR_i	$\mathrm{PSNR}_{\mathrm{ASM}}$	$\mathrm{mvp}_{\mathrm{ASM}}$	$\mathrm{PSNR}_{\mathrm{RLd}}$	$\mathrm{mvp}_{\mathrm{RLd}}$	$\mathrm{PSNR}_{\mathrm{RL}}$	$\mathrm{mvp}_{\mathrm{RL}}$
0.05	19.55	24.47	63	17.57	>1000	20.46	66
0.10	19.45	22.32	26	20.06	8	20.35	42
0.30	18.48	19.65	11	19.12	2	19.40	16

Table 2: Example 4.2. Results for restorations of `satellite` images that have been corrupted by significant Gaussian blur, defined by `band` $= 9$ and `sigma` $= 7$, and by noise corresponding to noise-level ν.

`band`	`sigma`	PSNR_i	$\mathrm{PSNR}_{\mathrm{ASM}}$	$\mathrm{mvp}_{\mathrm{ASM}}$	$\mathrm{PSNR}_{\mathrm{RL}}$	$\mathrm{mvp}_{\mathrm{RL}}$
21	9	17.87	20.24	74	19.07	161
15	9	18.45	20.90	47	19.81	96
15	7	18.55	20.74	47	19.70	90
9	7	19.45	22.32	26	20.35	42
5	3	20.89	23.64	18	21.32	12
3	3	22.22	24.80	14	22.67	10

Table 3: Example 4.2. Results for restorations of `satellite` images that have been corrupted by symmetric Gaussian blur associated with several values of the parameters `band` and `sigma`, and noise of level $\nu = 0.1$.

shown in Figure 3(a). It is represented by 256×256 pixels. It follows that the order of the blurring matrix A is $m = 65536$. The corresponding image corrupted by Gaussian blur, defined by `band` $= 9$ and `sigma` $= 7$, and by 10% noise, is shown in Figure 3(b).

Table 2 reports PSNR-values for the contaminated images and for restorations determined by the RL method and the active set method. The table is analogous to Table 1. When the noise-level is $\nu = 0.05$, the RL method supplied with the discrepancy principle as stopping criterion does not terminate within 500 iterations. Table 2 shows the active set method with the discrepancy principle to determine restorations with larger PSNR-values and to require fewer matrix-vector product evaluations than Richardson-Lucy for every noise-level. Figure 3(c) shows the best restoration of the image of Figure 3(b) determined by the RL method. Its computation required 42 matrix-vector product evaluations. Figure 3(d) shows the corresponding restoration computed with the active set method.

Finally, Table 3 compares the PSNR-values for restorations computed with the RL and active set methods for Gaussian blurs associated with different values of the parameters `band` and `sigma`, and 10% noise. We report the largest achievable PSNR-values for the RL method and show the associated number of matrix-vector product evaluations required. Our analysis of Section 2 suggests that the RL method converges slower the larger the value of the parameter `sigma`. This is in agreement with the results reported in Table 3. □

5. Conclusion and extensions

The computed examples show the proposed active set method with the discrepancy principle as stopping criterion to yield restorations of higher quality than any restoration determined by Richardson-Lucy. Moreover, for most examples the computation of the best restoration with the latter method required more matrix-vector product evaluations than the active set method with the discrepancy principle. The fact that the active set method performs well with the discrepancy principle makes the method easy to apply when an estimate of the norm of the error in the data is available or can be computed. However, we note that the active set method also can be used in conjunction with other stopping criteria. In view of the superior performance of the active set method, it may be interesting to investigate the performance of other implementations, such as an implementation based on a multilevel approach [9]. We also plan to investigate the performance of the active set method when applied to the solution of other ill-posed problems that often are solved by the Richardson-Lucy method.

Acknowledgements. We would like to thank the referees for comments. This work was supported by PRIN-MIUR-Cofin 2008 project, GNCS-INDAM, an OBR Research Challenge Grant, and NSF grant DMS-1115385.

References

[1] A. G. Al-Bakkar, S. S. Ipson, G. J. Porter, and D. G. Gleed, A parallel algorithm for a modified Richardson-Lucy algorithm for image de-blurring, Internat. J. Infrared Millimeter Waves, 18 (1997), pp. 555–575.

[2] G. R. Ayers and J. C. Dainty, Iterative blind deconvolution method and its applications, Opt. Lett., 1 (1988), pp. 547–549.

[3] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, Richardson-Lucy algorithm with total variation regularization for 3D confocal microsope deconvolution, Microscopy Research and Technique, 69 (2006), pp. 260–266.

[4] H. W. Engl, M. Hanke, and A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, 1996.

[5] D. A. Fish, A. M. Brinicombe, E. R. Pike, and J. G. Walker, Blind deconvolution by means of the Richardson-Lucy algorithm, J. Opt. Soc. Am. A, 12 (1995), pp. 58–65.

[6] P. C. Hansen, Regularization Tools version 4.0 for Matlab 7.3, Numer. Algorithms, 46 (2007), pp. 189–194.

[7] S. Kindermann, Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems, Electron. Trans. Numer. Anal., 38 (2011), pp. 233–257.

[8] L. B. Lucy, An iterative method for the rectification of observed distributions, Astronomical J., 79 (1974), pp. 745–754.

[9] S. Morigi, R. Plemmons, L. Reichel, and F. Sgallari, A hybrid multilevel-active set method for large box-constrained linear discrete ill-posed problems, Calcolo, 48 (2011), pp. 89–105.

[10] S. Morigi, L. Reichel, F. Sgallari, and F. Zama, An iterative method for linear discrete ill-posed problems with box constraints, J. Comput. Appl. Math., 198 (2007), pp. 505–520.

[11] C. C. Paige and M. A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[12] K. M. Perry and S. J. Reeves, Generalized cross-validation as a stopping rule for the Richardson-Lucy algorithm, in The Restoration of HST Images and Spectra II Space Telescope Science Institute, eds. R. J. Hanisch and R. L. White, 1994, pp. 97–103.

[13] W. H. Richardson, Bayesian-based iterative method of image restoration, J. Opt. Soc. Am., 62 (1972), pp. 55–59.

[14] G. Ricort, H. Lantéri, E. Aristidi, and C. Aime, Application of the Richardson-Lucy algorithm to the deconvolution of two-fold probability density functions, Pure Appl. Opt., 2 (1993), pp. 125–143.

[15] D. P. Yu, J. D. Zhang, V. Cheng, C. D. Beling, and S. Fung, Doppler broadening of annihilation radiation spectroscopy study using Richardson-Lucy, maximum entropy, and Huber methods, Phys. Stat. Sol. (c), 4 (2007), pp. 3997–4000.

[16] J. D. Zhang, T. J. Zhou, C. K. Cheung, C. D. Beling, S. Fung, and M. K. Ng, Deconvolution of 2D coincident Doppler broadening spectroscopy using the Richardson-Lucy algorithm, Nuclear Instruments and Methods in Physics Research A, 560 (2006) 552–557.

[17] Z. Zhao, Y. Ding, J. Dong, Y. Hao, S. Wu, L. Cao, and Y. Pu, Richardson-Lucy method for decoding x-ray ring code image, Plasma Phys. Control Fusion, 49 (2007), pp. 1145–1150.