algorithm 35 iterations to converge to a solution $\boldsymbol{\theta}^*$. The SNR of the deconvolved signal was found to be 27.2745 dB which is practically the same as that obtained by Algorithm 4.7. As expected, Algorithm 4.9 was considerably faster: the average CPU time it required was approximately 20% of that required by Algorithm 4.7. ■

4.5 Alternating Direction Methods

Alternating direction methods have become increasingly important because of their ability to deal with large scale convex problems. This section presents two representative classes of alternating direction methods known as *alternating direction methods of multipliers* and *alternating minimization algorithms*.

4.5.1 Alternating direction method of multipliers

The alternating direction methods of multipliers (ADMM) [8] are aimed at solving the class of convex problems

$$\begin{aligned} &\text{minimize} \;\; f(\boldsymbol{x}) + h(\boldsymbol{y}) \\ &\text{subject to:} \; \boldsymbol{Ax} + \boldsymbol{By} = \boldsymbol{c} \end{aligned} \qquad (4.76\text{a-b})$$

where $\boldsymbol{x} \in R^n$ and $\boldsymbol{y} \in R^m$ are variables, $\boldsymbol{A} \in R^{p\times n}$, $\boldsymbol{B} \in R^{p\times m}$, $\boldsymbol{c} \in R^{p\times 1}$, and $f(\boldsymbol{x})$ and $h(\boldsymbol{y})$ are convex functions. Note that in Eq. (4.76) the variables in both objective function and constraint are split into two parts, each involves only one set of variables. By definition, the Lagrangian for the problem in Eq. (4.76) is given by

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{y}) + \boldsymbol{\lambda}^T(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c})$$

If both $f(\boldsymbol{x})$ and $h(\boldsymbol{y})$ are differentiable, the KKT conditions for the problem in Eq. (4.76) are given by

$$\begin{aligned} &\boldsymbol{Ax} + \boldsymbol{By} = \boldsymbol{c} \\ &\nabla f(\boldsymbol{x}) + \boldsymbol{A}^T\boldsymbol{\lambda} = \boldsymbol{0} \\ &\nabla h(\boldsymbol{y}) + \boldsymbol{B}^T\boldsymbol{\lambda} = \boldsymbol{0} \end{aligned} \qquad (4.77\text{a-c})$$

The Lagrange dual of Eq. (4.76) assumes the form

$$\text{maximize} \;\; q(\boldsymbol{\lambda}) \qquad (4.78)$$

where

$$q(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x},\boldsymbol{y}} \left[f(\boldsymbol{x}) + h(\boldsymbol{y}) + \boldsymbol{\lambda}^T(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c}) \right]$$

which in conjunction with Eq. (4.21) leads to

$$\begin{aligned} q(\boldsymbol{\lambda}) &= \inf_{\boldsymbol{x}} \left[f(\boldsymbol{x}) + \boldsymbol{\lambda}^T\boldsymbol{Ax} \right] + \inf_{\boldsymbol{y}} \left[h(\boldsymbol{y}) + \boldsymbol{\lambda}^T\boldsymbol{By} \right] - \boldsymbol{\lambda}^T\boldsymbol{c} \\ &= -\sup_{\boldsymbol{x}} \left[(-\boldsymbol{A}^T\boldsymbol{\lambda})^T\boldsymbol{x} - f(\boldsymbol{x}) \right] - \sup_{\boldsymbol{y}} \left[(-\boldsymbol{B}^T\boldsymbol{\lambda})^T\boldsymbol{y} - h(\boldsymbol{y}) \right] - \boldsymbol{\lambda}^T\boldsymbol{c} \end{aligned} \qquad (4.79)$$

$$= -f^*(-\boldsymbol{A}^T\boldsymbol{\lambda}) - h^*(-\boldsymbol{B}^T\boldsymbol{\lambda}) - \boldsymbol{c}^T\boldsymbol{\lambda}$$

By the properties that $\boldsymbol{u} = \nabla f(\boldsymbol{v})$ if and only if $\boldsymbol{v} = \nabla f^*(\boldsymbol{u})$ and that $\nabla f^*(-\boldsymbol{A}^T\boldsymbol{\lambda}) = -\boldsymbol{A}\nabla f^*(\boldsymbol{\lambda})$ (see Sec. 4.2.4), Eq. (4.79) implies that

$$\nabla q(\boldsymbol{\lambda}) = \boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c} \tag{4.80}$$

where $\{\boldsymbol{x}, \boldsymbol{y}\}$ minimizes $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$ for a given $\boldsymbol{\lambda}$.

If in addition we assume that $f(\boldsymbol{x})$ and $h(\boldsymbol{y})$ are strictly convex, a solution of the problem in Eq. (4.76) can be found by minimizing the Lagranging $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}^*)$ with respect to primal variables $\boldsymbol{x}$ and $\boldsymbol{y}$, where $\boldsymbol{\lambda}^*$ maximizes the dual function $q(\boldsymbol{\lambda})$ in Eq. (4.79). This in conjunction with Eq. (4.80) suggests *dual ascent* iterations for the problem in Eq. (4.76) as follows:

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{y}_k, \boldsymbol{\lambda}_k) = \arg\min_{\boldsymbol{x}} \left[f(\boldsymbol{x}) + \boldsymbol{\lambda}_k^T \boldsymbol{Ax} \right] \\
\boldsymbol{y}_{k+1} &= \arg\min_{\boldsymbol{y}} L(\boldsymbol{x}_k, \boldsymbol{y}, \boldsymbol{\lambda}_k) = \arg\min_{\boldsymbol{y}} \left[h(\boldsymbol{y}) + \boldsymbol{\lambda}_k^T \boldsymbol{By} \right] \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \alpha_k (\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_{k+1} - \boldsymbol{c})
\end{aligned} \tag{4.81a-c}$$

where scalar $\alpha_k > 0$ is chosen to maximize $q(\boldsymbol{\lambda})$ (see Eq. (4.78)) along the direction $\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_{k+1} - \boldsymbol{c}$.

It is well known that convex problems of form in Eq. (4.76) with less restrictive $f(\boldsymbol{x})$ and $h(\boldsymbol{y})$ and data matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can be handled by augmented dual based on the *augmented Lagrangian* [8]

$$L_\alpha(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{y}) + \boldsymbol{\lambda}^T(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c}) + \tfrac{\alpha}{2} \| \boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c} \|_2^2 \tag{4.82}$$

which includes the conventional Lagrangian $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$ as a special case when parameter α is set to zero. The introduction of augmented Lagrangian may be understood by considering the following [8]: if we modify the objective function in Eq. (4.76) by adding a penalty term $\tfrac{\alpha}{2} \| \boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c} \|_2^2$ for violation of the equality constraint, namely,

$$\begin{aligned}
&\text{minimize} \quad f(\boldsymbol{x}) + h(\boldsymbol{y}) + \tfrac{\alpha}{2} \| \boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c} \|_2^2 \\
&\text{subject to:} \ \boldsymbol{Ax} + \boldsymbol{By} = \boldsymbol{c}
\end{aligned} \tag{4.83}$$

then the conventional Lagrangian of problem (4.83) is exactly equal to $L_\alpha(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$ in Eq. (4.82).

Associated with $L_\alpha(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$, the augmented dual problem is given by

$$\text{maximize} \quad q_\alpha(\boldsymbol{\lambda})$$

where

$$q_\alpha(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x},\boldsymbol{y}} \left[f(\boldsymbol{x}) + h(\boldsymbol{y}) + \boldsymbol{\lambda}^T(\boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c}) + \tfrac{\alpha}{2} \| \boldsymbol{Ax} + \boldsymbol{By} - \boldsymbol{c} \|_2^2 \right]$$

Unlike the dual ascent iterations in Eq. (4.81) where the minimization of the Lagrangian with respect to variables $\{\boldsymbol{x}, \boldsymbol{y}\}$ is split into two separate steps with reduced problem size, the augmented Lagrangian are no longer separable in variables $\boldsymbol{x}$ and $\boldsymbol{y}$ because of the presence of the penalty term. In ADMM iterations, this issue is addressed by *alternating* updates of the primal variables $\boldsymbol{x}$ and $\boldsymbol{y}$, namely,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \left[f(\boldsymbol{x}) + \boldsymbol{\lambda}_k^T \boldsymbol{Ax} + \tfrac{\alpha}{2} \| \boldsymbol{Ax} + \boldsymbol{By}_k - \boldsymbol{c} \|_2^2 \right] \\ \boldsymbol{y}_{k+1} &= \arg\min_{\boldsymbol{y}} \left[h(\boldsymbol{y}) + \boldsymbol{\lambda}_k^T \boldsymbol{By} + \tfrac{\alpha}{2} \| \boldsymbol{Ax}_{k+1} + \boldsymbol{By} - \boldsymbol{c} \|_2^2 \right] \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \alpha(\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_{k+1} - \boldsymbol{c}) \end{aligned} \tag{4.84a-c}$$

Note that parameter α from the quadratic penalty term is used in Eq. (4.84c) to update Lagrange multiplier $\boldsymbol{\lambda}_k$, thereby eliminating a line search step to compute α_k as required in Eq. (4.81c). To justify Eq. (4.84), note that $\boldsymbol{y}_{k+1}$ minimizes $h(\boldsymbol{y}) + \boldsymbol{\lambda}_k^T \boldsymbol{By} + \tfrac{\alpha}{2} \| \boldsymbol{Ax}_{k+1} + \boldsymbol{By} - \boldsymbol{c} \|_2^2$, hence

$$\begin{aligned} \boldsymbol{0} &= \nabla h(\boldsymbol{y}_{k+1}) + \boldsymbol{B}^T \boldsymbol{\lambda}_k + \alpha \boldsymbol{B}^T(\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_{k+1} - \boldsymbol{c}) \\ &= \nabla h(\boldsymbol{y}_{k+1}) + \boldsymbol{B}^T \left[\boldsymbol{\lambda}_k + \alpha(\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_{k+1} - \boldsymbol{c}) \right] \end{aligned}$$

which in conjunction with Eq. (4.84c) leads to

$$\nabla h(\boldsymbol{y}_{k+1}) + \boldsymbol{B}^T \boldsymbol{\lambda}_{k+1} = 0$$

Therefore, the KKT condition in Eq. (4.77c) is satisfied by ADMM iterations. In addition, since $\boldsymbol{x}_{k+1}$ minimizes $f(\boldsymbol{x}) + \boldsymbol{\lambda}_k^T \boldsymbol{Ax} + \tfrac{\alpha}{2} \| \boldsymbol{Ax} + \boldsymbol{By}_k - \boldsymbol{c} \|_2^2$, we have

$$\begin{aligned} \boldsymbol{0} &= \nabla f(\boldsymbol{x}_{k+1}) + \boldsymbol{A}^T \boldsymbol{\lambda}_k + \alpha \boldsymbol{A}^T(\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_k - \boldsymbol{c}) \\ &= \nabla f(\boldsymbol{x}_{k+1}) + \boldsymbol{A}^T \left[\boldsymbol{\lambda}_k + \alpha(\boldsymbol{Ax}_{k+1} + \boldsymbol{By}_k - \boldsymbol{c}) \right] \\ &= \nabla f(\boldsymbol{x}_{k+1}) + \boldsymbol{A}^T \boldsymbol{\lambda}_{k+1} - \alpha \boldsymbol{A}^T \boldsymbol{B}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \end{aligned}$$

i.e.,

$$\nabla f(\boldsymbol{x}_{k+1}) + \boldsymbol{A}^T \boldsymbol{\lambda}_{k+1} = \alpha \boldsymbol{A}^T \boldsymbol{B}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \tag{4.85}$$

On comparing Eq. (4.85) with Eq. (4.77b), a *dual residual* in the kth iteration can be defined as

$$\boldsymbol{d}_k = \alpha \boldsymbol{A}^T \boldsymbol{B}(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k) \tag{4.86}$$

From (4.77a), a *primal residual* in the kth iteration is defined as

$$\boldsymbol{r}_k = \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{y}_{k+1} - \boldsymbol{c} \tag{4.87}$$

Together, $\{\boldsymbol{r}_k, \boldsymbol{d}_k\}$ measures closeness of the kth ADMM iteration $\{\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{\lambda}_k\}$ to the solution of the problem in Eq. (4.76), thus a reasonable criteria for terminating ADMM iterations is when

$$\| \boldsymbol{r}_k \| < \varepsilon_p \ \text{ and } \ \|\boldsymbol{d}_k \| < \varepsilon_d \tag{4.88}$$

where ε_p and ε_d are prescribed tolerances for primal and dual residuals, respectively.

Convergence of the ADMM iterations in Eq. (4.84) have been investigated under various assumptions, see [8] and [9] and the references cited therein. If both $f(\boldsymbol{x})$ and $h(\boldsymbol{y})$ are strongly convex with parameters m_f and m_h, respectively, and parameter α is chosen to satisfy

$$\alpha^3 \le \frac{m_f m_h^2}{\rho(\boldsymbol{A}^T\boldsymbol{A})\rho(\boldsymbol{B}^T\boldsymbol{B})^2}$$

where $\rho(\boldsymbol{M})$ denotes the largest eigenvalue of symmetric matrix $\boldsymbol{M}$, then both primal and dual residuals vanish at rate $O(1/k)$ [9], namely,

$$\| \boldsymbol{r}_k \|_2 \le O(1/k) \ \text{ and } \ \| \boldsymbol{d}_k \|_2 \le O(1/k)$$

We now summarize the method for solving the problem in Eq. (4.76) as an algorithm.

Algorithm 4.10 ADMM for the problem in Eq. (4.76)

Step 1 Input parameter $\alpha > 0$, $\boldsymbol{y}_0$, $\boldsymbol{\lambda}_0$, and tolerance $\varepsilon_p > 0$, $\varepsilon_d > 0$.

Set $k = 0$.

Step 2 Compute $\{\{\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}, \boldsymbol{\lambda}_{k+1}\}$ using Eq. (4.84).

Step 3 Compute $\boldsymbol{d}_k$ and $\boldsymbol{r}_k$ using Eqs. (4.86) and (4.87), respectively.

Step 4 If the conditions in Eq. (4. 88) are satisfied, output $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ as solution and stop; Otherwise, set $k = k + 1$ and repeat from Step 2.

Several variants of ADMM are available, one of them is that of the *scaled form* ADMM [8]. By letting

$$\boldsymbol{r} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c} \ \text{ and } \ \boldsymbol{v} = \boldsymbol{\lambda} / \alpha,$$

we write the augmented Lagrangian as

$$
\begin{aligned}
L_\alpha(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda}) &= f(\boldsymbol{x}) + h(\boldsymbol{y}) + \boldsymbol{\lambda}^T\boldsymbol{r} + \tfrac{\alpha}{2}\|\boldsymbol{r}\|_2^2 \\
&= f(\boldsymbol{x}) + h(\boldsymbol{y}) + \tfrac{\alpha}{2}\|\boldsymbol{r}+\boldsymbol{\nu}\|_2^2 - \tfrac{\alpha}{2}\|\boldsymbol{\nu}\|_2^2 \\
&= f(\boldsymbol{x}) + h(\boldsymbol{y}) + \tfrac{\alpha}{2}\|\boldsymbol{Ax}+\boldsymbol{By}-\boldsymbol{c}+\boldsymbol{\nu}\|_2^2 - \tfrac{\alpha}{2}\|\boldsymbol{\nu}\|_2^2
\end{aligned}
$$

Consequently, the scaled ADMM algorithm can be outlined as follows.

Algorithm 4.11 Scaled ADMM for the problem in Eq. (4.76)

Step 1 Input parameter $\alpha > 0$, y_0, $\boldsymbol{\nu}_0$, and tolerance $\varepsilon_p > 0$, $\varepsilon_d > 0$.

Set $k = 0$.

Step 2 Compute

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}}\left[f(\boldsymbol{x}) + \tfrac{\alpha}{2}\|\boldsymbol{Ax}+\boldsymbol{By}_k-\boldsymbol{c}+\boldsymbol{\nu}_k\|_2^2\right] \\
\boldsymbol{y}_{k+1} &= \arg\min_{\boldsymbol{y}}\left[h(\boldsymbol{y}) + \tfrac{\alpha}{2}\|\boldsymbol{Ax}_{k+1}+\boldsymbol{By}-\boldsymbol{c}+\boldsymbol{\nu}_k\|_2^2\right] \\
\boldsymbol{\nu}_{k+1} &= \boldsymbol{\nu}_k + \boldsymbol{Ax}_{k+1} + \boldsymbol{By}_{k+1} - \boldsymbol{c}
\end{aligned}
\tag{4.89}
$$

Step 3 Compute $\boldsymbol{d}_k$ and $\boldsymbol{r}_k$ using Eqs. (4.86) and (4.87), respectively.

Step 4 If the conditions in Eq. (4. 88) are satisfied, output $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ as solution and stop; Otherwise, set $k = k + 1$ and repeat from Step 2.

A variant of scaled ADMM is the *over-relaxed* ADMM in that the term $\boldsymbol{Ax}_{k+1}$ in the $\boldsymbol{y}$- and $\boldsymbol{\nu}$-updates is replaced by $\tau\boldsymbol{Ax}_{k+1} - (1-\tau)(\boldsymbol{By}_k - \boldsymbol{c})$ with $\tau \in (0, 2]$. Thus the over-relaxed ADMM iterations assume the form

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}}\left[f(\boldsymbol{x}) + \tfrac{\alpha}{2}\|\boldsymbol{Ax}+\boldsymbol{By}_k-\boldsymbol{c}+\boldsymbol{\nu}_k\|_2^2\right] \\
\boldsymbol{y}_{k+1} &= \arg\min_{\boldsymbol{y}}\left[h(\boldsymbol{y}) + \tfrac{\alpha}{2}\|\tau\boldsymbol{Ax}_{k+1}-(1-\tau)\boldsymbol{By}_k+\boldsymbol{By}-\tau\boldsymbol{c}+\boldsymbol{\nu}_k\|_2^2\right] \\
\boldsymbol{\nu}_{k+1} &= \boldsymbol{\nu}_k + \tau\boldsymbol{Ax}_{k+1} - (1-\tau)\boldsymbol{By}_k + \boldsymbol{By}_{k+1} - \tau\boldsymbol{c}
\end{aligned}
\tag{4.90}
$$

In addition, the idea from Nesterov's accelerated gradient descent algorithm [2] has been extended to ADMM [9]. If both functions $f(\boldsymbol{x})$ and $h(\boldsymbol{y})$ are strongly convex, then a fast ADMM outlined below is shown to converge [9].

Algorithm 4.12 Accelerated ADMM for the problem in Eq. (4.76)

Step 1 Input parameter $\alpha > 0$, $\hat{\boldsymbol{y}}_0, \hat{\boldsymbol{\lambda}}_0$, and tolerance $\varepsilon_p > 0$, $\varepsilon_d > 0$.

Set $\boldsymbol{y}_{-1} = \hat{\boldsymbol{y}}_0, \boldsymbol{\lambda}_{-1} = \hat{\boldsymbol{\lambda}}_0$, $t_0 = 1$, and $k = 0$.

Step 2 Compute

$$
\begin{aligned}
\boldsymbol{x}_k &= \arg\min_{\boldsymbol{x}} \left[f(\boldsymbol{x}) + \hat{\boldsymbol{\lambda}}_k^T \boldsymbol{A}\boldsymbol{x} + \tfrac{\alpha}{2} \| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\hat{\boldsymbol{y}}_k - \boldsymbol{c} \|_2^2 \right] \\
\boldsymbol{y}_k &= \arg\min_{\boldsymbol{y}} \left[h(\boldsymbol{y}) + \hat{\boldsymbol{\lambda}}_k^T \boldsymbol{B}\boldsymbol{y} + \tfrac{\alpha}{2} \| \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c} \|_2^2 \right] \\
\boldsymbol{\lambda}_k &= \hat{\boldsymbol{\lambda}}_k + \alpha(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{y}_k - \boldsymbol{c}) \\
t_{k+1} &= \tfrac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right) \\
\hat{\boldsymbol{y}}_{k+1} &= \boldsymbol{y}_k + \tfrac{t_k - 1}{t_{k+1}}(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) \\
\hat{\boldsymbol{\lambda}}_{k+1} &= \boldsymbol{\lambda}_k + \tfrac{t_k - 1}{t_{k+1}}(\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1})
\end{aligned} \tag{4.91}
$$

Step 3 If $\| \alpha \boldsymbol{A}^T \boldsymbol{B}(\boldsymbol{y}_k - \boldsymbol{y}_{k-1}) \|_2 < \varepsilon_d$ and $\| \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{y}_k - \boldsymbol{c} \|_2 < \varepsilon_p$, output $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ as solution and stop; Otherwise, set $k = k + 1$ and repeat from Step 2.

Example 4.8

(a) Apply Algorithm 4.10 to solve the l_1-l_2 minimization problem

$$
\text{minimize} \quad \tfrac{1}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \|_2^2 + \mu \| \boldsymbol{x} \|_1 \tag{4.92}
$$

(b) Apply the results from part (a) to solve the deconvolution problem in Example 4.7.

Solution (a) The problem in Eq. (4.92) can be formulated as [8]

$$
\begin{aligned}
&\text{minimize} \quad \tfrac{1}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \|_2^2 + \mu \| \boldsymbol{y} \|_1 \\
&\text{subject to:} \quad \boldsymbol{x} - \boldsymbol{y} = \boldsymbol{0}
\end{aligned}
$$

which fits into the formulation in Eq. (4.76) with $f(\boldsymbol{x}) = \tfrac{1}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \|_2^2$ and $h(\boldsymbol{y}) = \mu \| \boldsymbol{y} \|_1$. The scaled ADMM iterations in this case assume the form

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \left[\tfrac{1}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \|_2^2 + \tfrac{\alpha}{2} \| \boldsymbol{x} - (\boldsymbol{y}_k - \boldsymbol{\lambda}_k / \alpha) \|_2^2 \right] \\
\boldsymbol{y}_{k+1} &= \arg\min_{\boldsymbol{y}} \left[\mu \| \boldsymbol{y} \|_1 + \tfrac{\alpha}{2} \| \boldsymbol{y} - (\boldsymbol{x}_{k+1} + \boldsymbol{\lambda}_k / \alpha) \|_2^2 \right] \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \alpha(\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1})
\end{aligned}
$$

Evidently, updating $\boldsymbol{x}_k$ amounts to minimizing a convex quadratic function and updating $\boldsymbol{y}_k$ can be done by soft-shrinkage of $\boldsymbol{x}_{k+1} + \boldsymbol{\lambda}_k / \alpha$ by μ / α, see Eq. (4.67). Thus we have

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= (\boldsymbol{A}^T \boldsymbol{A} + \alpha \boldsymbol{I})^{-1} (\boldsymbol{A}^T \boldsymbol{b} + \alpha \boldsymbol{y}_k - \boldsymbol{\lambda}_k) \\
\boldsymbol{y}_{k+1} &= \mathcal{S}_{\mu/\alpha}(\boldsymbol{x}_{k+1} + \boldsymbol{\lambda}_k / \alpha) \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \alpha(\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1})
\end{aligned} \tag{4.93}
$$

where operator $\mathcal{S}_{\mu/\alpha}$ is defined by Eq. (4.68). Note that matrix $(\boldsymbol{A}^T \boldsymbol{A} + \alpha \boldsymbol{I})^{-1}$ as well as vector $\boldsymbol{A}^T \boldsymbol{b}$ in Eq. (4.93) are independent of iterations, hence they need to be computed only once.

(b) By applying Eq. (4.93) to the data set described in Example 4.7 with y_0 = **0**, λ_0 = **0**, $\mu = 0.25$, and $\alpha = 0.11$, it took 32 ADMM iterations to yield a satisfactory estimation of the signal. The SNR of the estimated signal was found to be 27.2765 dB. The original, distorted, and recovered signals are depicted in Fig. 4.13. The profiles of the primal and dual residuals in terms of $\|\boldsymbol{r}_k\|_2$ and $\|\boldsymbol{d}_k\|_2$ are shown in Fig. 4.14. It is observed that both $\|\boldsymbol{r}_k\|_2$ and $\|\boldsymbol{d}_k\|_2$ fall below 5×10^{-3} after 32 iterations. On comparing with the proximal-point algorithms in Sec. 4.4, the average CPU time required by ADMM iterations was found be practically the same as that of Algorithm 14.9. ■

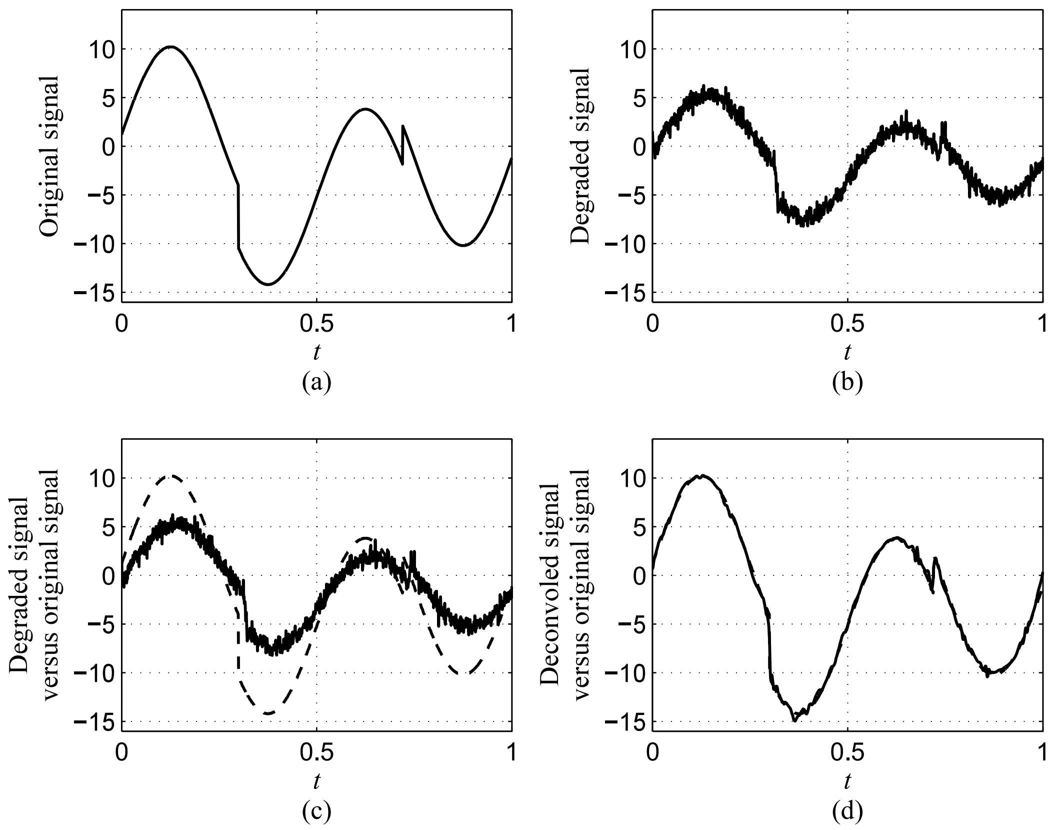

Figure 4.13. (a) Original signal heavisine (b) Distorted and noise-contaminated heavisine (c) Original (dashed) versus degraded (solid) heavisine, SNR = 5.7592 dB. (d) Original (dashed) versus reconstructed (solid) heavisine, SNR = 27.2765 dB.

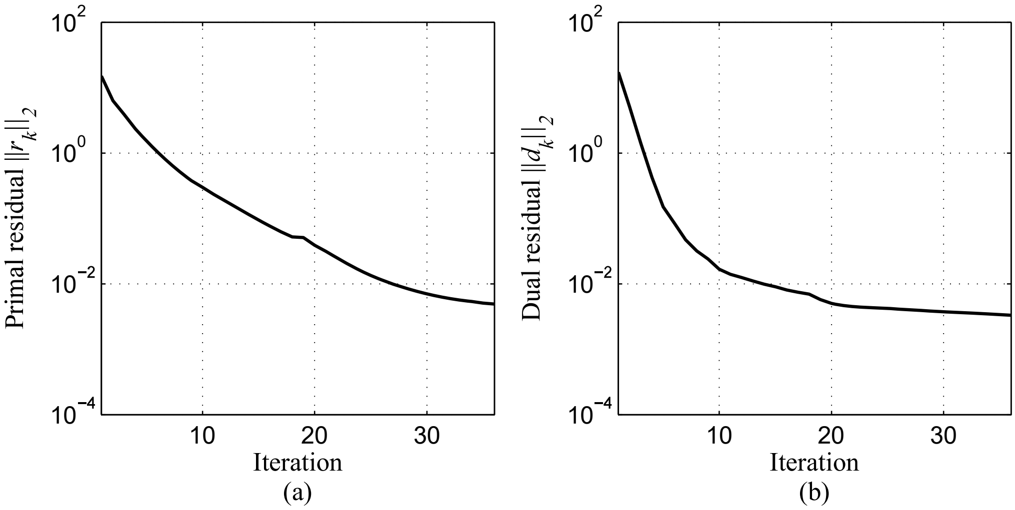

Figure 4.14. (a) Primal residual $\|\boldsymbol{r}_k\|_2$ versus iterations (b) Dual residual $\|\boldsymbol{d}_k\|_2$ versus iterations for Example 4.8.

4.5.2 ADMM for general convex optimization

Consider the constrained convex problem

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to:} & \boldsymbol{x} \in C \end{array} \tag{4.94}$$

where $f(\boldsymbol{x})$ is a convex function and C is a convex set representing the feasible region of the problem. The problem in Eq. (4.94) can be formulated as

$$\text{minimize} \quad f(\boldsymbol{x}) + I_C(\boldsymbol{x}) \tag{4.95}$$

where $I_C(\boldsymbol{x})$ is the indicator function associated with set C:

$$I_C(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in C \\ +\infty & \text{otherwise} \end{cases}$$

The problem in Eq. (4.95) can be written as

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) + I_C(\boldsymbol{y}) \\ \text{subject to:} & \boldsymbol{x} - \boldsymbol{y} = \boldsymbol{0} \end{array} \tag{4.96}$$

That fits into the ADMM formulation in Eq. (4.76) [8]. The scaled ADMM iterations for Eq. (4.96) are given by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \left[f(\boldsymbol{x}) + \tfrac{\alpha}{2} \| \boldsymbol{x} - \boldsymbol{y}_k + \boldsymbol{v}_k \|_2^2 \right] \\ \boldsymbol{y}_{k+1} &= \arg\min_{\boldsymbol{y}} \left[I_C(\boldsymbol{y}) + \tfrac{\alpha}{2} \| \boldsymbol{y} - (\boldsymbol{x}_{k+1} + \boldsymbol{v}_k) \|_2^2 \right] \\ \boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + \boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1} \end{aligned}$$

where the y-minimization is obtained by minimizing $\| \boldsymbol{y} - (\boldsymbol{x}_{k+1} + \boldsymbol{v}_k) \|_2$ subject to $\boldsymbol{y} \in C$. This means that $\boldsymbol{y}_{k+1}$ can be obtained by projecting $\boldsymbol{x}_{k+1} + \boldsymbol{v}_k$ onto set C. Therefore, the ADMM iterations become

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \left[f(\boldsymbol{x}) + \tfrac{\alpha}{2} \| \boldsymbol{x} - \boldsymbol{y}_k + \boldsymbol{v}_k \|_2^2 \right] \\ \boldsymbol{y}_{k+1} &= P_C(\boldsymbol{x}_{k+1} + \boldsymbol{v}_k) \\ \boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + \boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1} \end{aligned} \tag{4.97a-c}$$

where $P_C(\boldsymbol{z})$ denotes the projection of point $\boldsymbol{z}$ onto convex set C. The projection cab be accomplished by solving the convex problem

$$\begin{aligned} &\text{minimize} \quad \| \boldsymbol{y} - \boldsymbol{z} \|_2 \\ &\text{subject to:} \ \ \boldsymbol{y} \in C \end{aligned}$$

Example 4.9

Find a sparse solution of an underdetermined system of linear equation $\boldsymbol{Ax} = \boldsymbol{b}$ by solving the constrained convex problem

$$\begin{aligned} &\text{minimize} \ \ \| \boldsymbol{x} \|_1 \\ &\text{subject to:} \ \ \boldsymbol{Ax} = \boldsymbol{b} \end{aligned} \tag{4.98}$$

Solution The problem fits into the formulation in Eq. (4.94) with $f(\boldsymbol{x}) = \| \boldsymbol{x} \|_1$ and $C = \{ \boldsymbol{x} : \boldsymbol{Ax} = \boldsymbol{b} \}$. The $\boldsymbol{x}$-minimization step in Eq. (4.97a) becomes

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \left[\| \boldsymbol{x} \|_1 + \tfrac{\alpha}{2} \| \boldsymbol{x} - \boldsymbol{y}_k + \boldsymbol{v}_k \|_2^2 \right]$$

hence

$$\boldsymbol{x}_{k+1} = \mathcal{S}_{1/\alpha}(\boldsymbol{y}_k - \boldsymbol{v}_k) \tag{4.99}$$

where operator $\mathcal{S}$ is defined by (4.68). The $\boldsymbol{y}$-minimization step in Eq. (4.94b) is carried out by solving the simple convex QP problem

$$\begin{aligned} &\text{minimize} \quad \| \boldsymbol{y} - (\boldsymbol{x}_{k+1} + \boldsymbol{v}_k) \|_2 \\ &\text{subject to:} \ \ \boldsymbol{Ay} = \boldsymbol{b} \end{aligned} \tag{4.100}$$

whose solution is given by

$$\boldsymbol{y}_{k+1} = \boldsymbol{A}^{+} \boldsymbol{b} + \boldsymbol{P}_A (\boldsymbol{x}_{k+1} + \boldsymbol{v}_k) \tag{4.101}$$

where $\boldsymbol{A}^{+} = \boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{A}^T)^{-1}$ and $\boldsymbol{P}_A = \boldsymbol{I} - \boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{A}^T)^{-1} \boldsymbol{A}$. To summarize, the ADMM iterations in Eq. (4.97) are realized using Eqs. (4.99), (4.101), and (4.97c).

For illustration purposes, above results were applied to the problem in Eq. (4.98) where $\boldsymbol{A}$ was a randomly generated matrix of size 20 by 50. A sparse column vector $\boldsymbol{x}_s$ of length 50 was produced by placing six randomly generated nonzero numbers in the zero vector of length 50 for six randomly selected coordinates, see Fig. 4.15a. Vector $\boldsymbol{b}$ was then generated as $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_s$. In this way, $\boldsymbol{x}_s$ is a known sparse solution of the underdetermined system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. With $\alpha = 0.5$, it took 49 ADMM iterations to converge to a solution $\boldsymbol{x}^*$ which is depicted in Fig. 4.15b where the ADMM-based solution is found to well recover the true sparse solution $\boldsymbol{x}_s$. The l_2 reconstruction error of the solution is found to be $\| \boldsymbol{x}^* - \boldsymbol{x}_s \|_2 = 4.1282 \times 10^{-4}$. ■

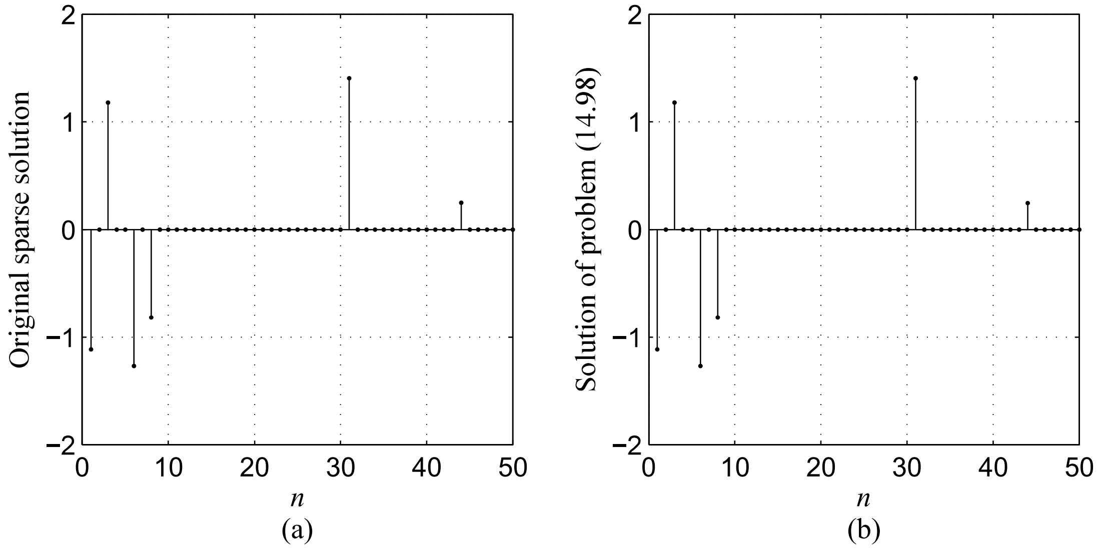

Figure 4.15. (a) Original sparse solution $\boldsymbol{x}_s$ and (b) Solution obtained by solving the problem in Eq. (4.98).

The primal and dual residuals are given by $\boldsymbol{r}_k = \boldsymbol{x}_k - \boldsymbol{y}_k$ and $\boldsymbol{d}_k = -\alpha(\boldsymbol{y}_k - \boldsymbol{y}_{k-1})$, respectively. The profiles of these residuals for the 49 ADMM iterations are shown in Fig. 4.16.

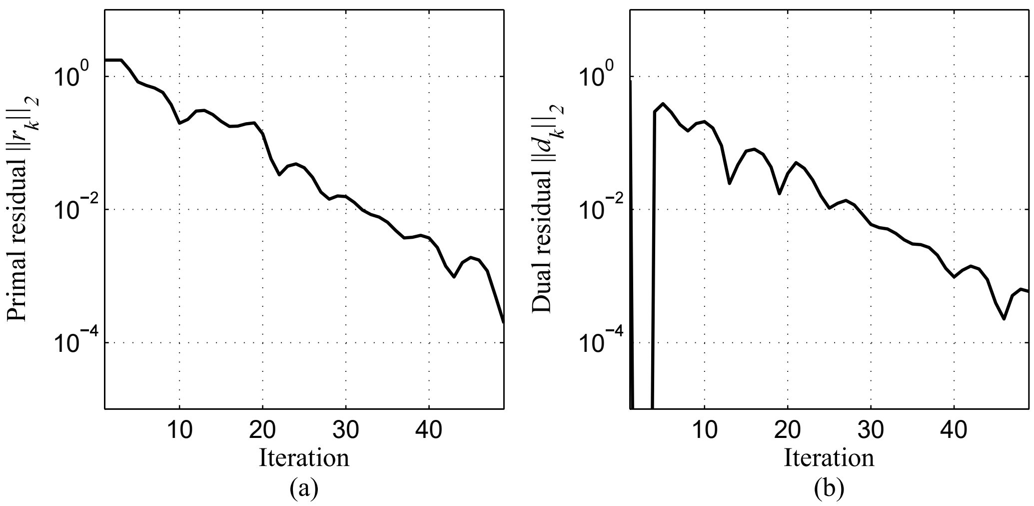

Figure 4.16. (a) Primal and (b) Dual residuals for the problem in Example 4.8.