$$x_{k+1} = \arg\min_{x}\left(f(x_k) + \nabla f(x_k)^T(x - x_k) + \tfrac{L}{2}\| x - x_k \|_2^2 + \psi(x)\right)$$

which is the same as

$$x_{k+1} = \arg\min_{x}\left(\tfrac{L}{2}\left\| x - \left(x_k - \tfrac{1}{L}\nabla f(x_k)\right)\right\|_2^2 + \psi(x)\right)$$

or

$$x_{k+1} = \arg\min_{x}\left(\tfrac{L}{2}\left\| x - b_k \right\|_2^2 + \psi(x)\right) \tag{4.64a}$$

where

$$b_k = x_k - \tfrac{1}{L}\nabla f(x_k) \tag{4.64b}$$

**Algorithm 4.6 Proximal-point algorithm for the problem in Eq. (4.56)**

**Step 1** Input initial point $x_0$, Lipschitz constant $L$ for $\nabla f(x)$, and tolerance $\varepsilon$.

Set $k = 0$.

**Step 2** Compute $x_{k+1}$ solving the convex problem in Eq. (4.64).

**Step 3** If $\| x_{k+1} - x_k \|_2 / \sqrt{n} < \varepsilon$, output solution $x^* = x_{k+1}$ and stop; otherwise

set $k = k + 1$ and repeat from Step 2.

A problem of particular interest arising from several statistics and signal processing applications that fits well into the formulation in Eq. (4.56) is the $l_1$-$l_2$ minimization problem

$$\text{minimize} \quad F(x) = \tfrac{1}{2}\|Ax - b\|_2^2 + \mu \| x \|_1 \tag{4.65}$$

where data $A \in R^{m\times n}, b \in R^{m\times 1}$ and *regularization* parameter $\mu > 0$ are given. The two component functions in this case are $f(x) = \tfrac{1}{2}\|Ax - b\|_2^2$ which is convex with Lipschitz continuous gradient and $L = \lambda_{\max}(A^T A)$ (see Example 4.2), and $\Psi(x) = \mu \| x \|_1$ which is convex but non-differentiable. Therefore, the $l_1$-$l_2$ minimization problem fits nicely into the formulation in Eq. (4.56). Applying Algorithm 4.6, the iteration step in Eq. (14.64) becomes

$$x_{k+1} = \arg\min_{x}\left(\tfrac{1}{2}\left\| x - b_k \right\|_2^2 + \tfrac{\mu}{L}\| x \|_1\right) \tag{4.66a}$$

where

$$b_k = \frac{1}{L}A^T\left(b - Ax_k\right) + x_k \tag{4.66b}$$

It is well known [7] that the minimizer of the problem in (4.66) can be obtained by applying the *soft-shrinkage* by $\mu / L$ to $b_k$, i.e.,

$$\boldsymbol{x}_{k+1} = S_{\mu/L}\left\{\frac{1}{L}\boldsymbol{A}^T\left(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k\right)+\boldsymbol{x}_k\right\} \tag{4.67}$$

where operator $S_\delta$ is defined by

$$S_\delta(z) \triangleq \text{sign}(z)\cdot\max\{|z|-\delta, \boldsymbol{0}\} \tag{4.68}$$

where the operations "sign", "max" and "$|z|-\delta$" are performed componentwisely.

**Algorithm 4.7 Proximal-point algorithm for the problem in Eq. (4.65)**

**Step 1** Input data $\{\boldsymbol{A}, \boldsymbol{b}\}$, regularization parameter $\mu$, initial point $\boldsymbol{x}_0$, Lipschitz constant $L$, and number of iterations $K$. Set $k = 0$.

**Step 2** Compute $\boldsymbol{x}_{k+1}$ using Eq. (4.67).

**Step 3** If $k = K$, output solution $\boldsymbol{x}^* = \boldsymbol{x}_{k+1}$ and stop; otherwise set $k = k + 1$ and repeat from Step 2.

### 4.4.2 A Fast Algorithm for Solving (4.56) [6] [7]

Nesterov [6] presents an algorithm for the problem in Eq. (4.56) with $O(k^{-2})$ rate of convergence in the sense that

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_{\text{optimal}}) \le \frac{2L\left\|\boldsymbol{x}_0 - \boldsymbol{x}_{\text{optimal}}\right\|_2^2}{(k+1)^2} \tag{4.69}$$

An algorithm, known as *fast iterative shrinkage-thresholding algorithm* (FISTA), is developed in [7], which is essentially a modified version of Nesterov's algorithm, in that soft-thresholding is not directly applied to the preceding iterate but to a subtly modified previous iterate. As such, FISTA is a fast algorithm with $O(k^{-2})$ rate of convergence in the sense of Eq. (4.69). The algorithmic steps of FISTA are outlined as follows.

**Algorithm 4.8 FISTA for the problem in Eq. (4.56)**

**Step 1** Input initial point $\boldsymbol{x}_0$, Lipschitz constant $L$ for $\nabla f(\boldsymbol{x})$, and number of iterations $K$. Set $\boldsymbol{z}_1 = \boldsymbol{x}_0$, $t_1 = 1$, and $k = 1$.

**Step 2** Compute $\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}}\left(\frac{L}{2}\left\|\boldsymbol{x} - \left(\boldsymbol{z}_k - \frac{1}{L}\nabla f(\boldsymbol{z}_k)\right)\right\|_2^2 + \psi(\boldsymbol{x})\right)$

**Step 3** Compute $t_{k+1} = \dfrac{1+\sqrt{1+4t_k^2}}{2}$