Special Topics in Management Science 26:711:685:01

Convex Analysis and Optimization

Fall 2023 Rutgers University Prof. Eckstein

Solutions to Homework 7 / Take-Home Final

1. Fix any $r \in \mathbb{R}^n$. Since h is closed proper convex, Proposition 2.14 in the class notes assures that

$$(\exists!\, x, y \in \mathbb{R}^n) : \qquad y \in \partial h(x) \qquad x + cy = r,$$

where "$\exists!$" means "there exist unique". One then has $\operatorname{prox}_{ch}(r) = x$. Using the same login with the closed proper function h^* instead of h, the positive constant $1/c$ instead of c, and $(1/c)r$ instead of r yields

$$(\exists!\, u, v \in \mathbb{R}^n) : \qquad v \in \partial h^*(u) \qquad u + \tfrac{1}{c}v = \tfrac{1}{c}r,$$

from which one has $\operatorname{prox}_{(1/c)h^*}\big(\tfrac{1}{c}r\big) = u$. Using that $v \in \partial h^*(u)$ if and only $u \in \partial h(v)$ and multiplying through the equation $u + \frac{1}{c}v = \frac{1}{c}r$ by $c \neq 0$, conditions equivalent to those immediately above are

$$u \in \partial h(v) \qquad cu + v = r.$$

These conditions on u and v are exactly the same as those respectively given for y and x above, so by the uniqueness guaranteed by the representation lemma one has $u = y$ and $v = x$. Consequently,

$$\operatorname{prox}_{ch}(r) + c \operatorname{prox}_{(1/c)h^*}\big(\tfrac{1}{c}r\big) = x + cu = x + cy = r.$$

Since $r \in \mathbb{R}^n$ was arbitrary, the claim is now established.

Aside: with some algrebraic manipulations and changes of variables, the identity proved here can be turned into a formula allowing $\operatorname{prox}_{ah^*}$ to be computed from $\operatorname{prox}_{(1/a)h}$ for any positive scalar a, namely

$$(\forall\, s \in \mathbb{R}^n) \qquad \operatorname{prox}_{ah^*}(s) = s - a \operatorname{prox}_{(1/a)h}\big(\tfrac{1}{a}s\big).$$

2. (a) The Lagrangian is defined as $L(x,p) = \inf_{u \in \mathbb{R}^m}\{F(x,u) - \langle p, u\rangle\}$. Note that the set within the "inf" contains only $+\infty$ if $x \notin \operatorname{dom} f$. When $x \in \operatorname{dom} f$, we have the more interesting expression

$$\begin{aligned} L(x,p) &= \inf\left\{ f(x) - \langle p, u\rangle \;\middle|\; u \in \mathbb{R}^m : Ax - b + u \in K \right\} \\ &= f(x) - \sup\left\{ \langle p, u\rangle \;\middle|\; u \in \mathbb{R}^m : Ax - b + u \in K \right\}. \end{aligned}$$

Considering the last expression, we have

$$\begin{aligned}
&\sup\{\langle p,u\rangle \mid u\in\mathbb{R}^m : Ax-b+u\in K\}\\
&=\sup\{\langle p,u\rangle \mid u\in K-Ax+b+u\}\\
&=\sup\{\langle p,u\rangle \mid u=v-Ax+b, v\in K\}\\
&=\sup\{\langle p,v-Ax+b\rangle \mid v\in K\}\\
&=\langle p,b-Ax\rangle+\sup_{v\in K}\{\langle p,v\rangle\}\\
&=\langle p,b-Ax\rangle+\delta_{K^*}(p),
\end{aligned}$$

where $\sup_{v\in K}\{\langle p,v\rangle\}=\delta_{K^*}(p)$ was proven as part of problem 2 on Homework 6, or may be easily derived directly. Combining these observations, we come to the conclusion that

$$L(x,p)=\begin{cases} +\infty, & x\notin \operatorname{dom} f,\\ f(x)+\langle p,Ax-b\rangle, & x\in\operatorname{dom} f, p\in K^*,\\ -\infty, & x\in\operatorname{dom} f, p\notin K^*.\end{cases}$$

Note that basically we have $L(x,p)=f(x)+\langle p,Ax-b\rangle$ as usual, with the restriction that $p\in K^*$, and this directly generalizes results we obtained for $K=\{0\}$ and K being the nonpositive orthant (although in that case we allowed nonlinear functions in place of $Ax-b$ — that can also be done in general, but we have to enforce some restrictions to obtain a convex problem).

(b) $w=\operatorname{proj}_C(v)$ if and only if w is the unique solution of $\min_{x\in C}\{\frac{1}{2}\|x-v\|^2\}$, or equivalently $\min_{x\in\mathbb{R}^n}\{\frac{1}{2}\|x-v\|^2+\delta_C(x)\}$. A simple application of the Rockafellar-Moreau theorem gives that an equivalent condition is

$$0\in w-v+\partial\delta_C(w)=w-v+N_C(w) \quad\Leftrightarrow\quad v-w\in N_C(w).$$

In problem 3(b) of Homework 5, we established that for a convex cone, $N_C(w)=\{y\in C^* \mid \langle y,w\rangle=0\}$. Thus, we obtain the equivalent condition that $v-w\in C^*$ and $\langle v-w,w\rangle=0$.

(c) Let $w=\operatorname{proj}_C(v)$, from which part (b) implies $w\in C$, $v-w\in C^*$ and $\langle v-w,w\rangle=0$. Now consider $\alpha w=\alpha\operatorname{proj}_C(v)$:

- Since C is a cone, $\alpha w\in C$.
- Since C^* is a cone and $v-w\in C^*$, we have that $\alpha v-\alpha w=\alpha(v-w)\in C^*$.
- Finally, since $\langle w-v,w\rangle=0$, we have that $\langle \alpha v-\alpha w,\alpha v\rangle=\alpha^2\langle w-v,w\rangle=\alpha^2\cdot 0=0$.

From part (b), these conditions are sufficient for αw to be the projection of αv onto C.

(d) Again, let $w=\operatorname{proj}_C(v)$, so we have from part (b) that $w\in C$, $v-w\in C^*$ and $\langle v-w,w\rangle=0$. Let $t=v-w$; then

- $t=v-w\in C^*$

- $v - t = v - (v - w) = w \in C = C^{**}$, where we know $C = C^{**}$ from the polar cone theorem because C is closed and convex.
- $\langle v - t, t\rangle = \langle v - (v - w), v - w\rangle = \langle w, v - w\rangle = \langle v - w, w\rangle = 0$.

Summarizing, we have now shown that

$$t \in C^* \qquad\qquad v - t \in C^{**} \qquad\qquad \langle v - t, t\rangle = 0.$$

From part (b), these conditions are sufficient for t to equal $\mathrm{proj}_{C^*}(v)$, so we conclude that $\mathrm{proj}_{C^*}(v) = t = v - w = v - \mathrm{proj}_C(v)$. This formula generalizes the familar projection properties of orthogonal linear subspaces.

(e) We obtain exactly the same solution set if we minimize $\|u - \frac{1}{c}\ell\|^2$ subject to $u + t \in K$. Setting $v = u + t$, and hence $u = v - t$, we may equivalently minimize $\|v - t - \frac{1}{c}\ell\|^2 = \|v - (t + \frac{1}{c}\ell)\|^2$ over $v \in K$ and then set $u = v - t$. The solution of this last problem over v is clearly $\mathrm{proj}_K(t + \frac{1}{c}\ell)$, so we obtain that the solution of the original problem is $u = \mathrm{proj}_K(t + \frac{1}{c}\ell) - t$.

(f) The generic augmented Lagrangian algorithm in Proposition 3.8 and (3.23)-(3.24) in the notes may be described as follows:

- Find $x^{k+1} \in \mathbb{R}^n$, $u^{k+1} \in \mathbb{R}^m$ achieving the respective infima in

$$\inf_{x\in\mathbb{R}^m} \left\{ \inf_{u\in\mathbb{R}_m} \left\{ F(x,u) + \tfrac{1}{2c_k}\left\|p^k - c_k u\right\|^2 \right\} \right\}$$

- Set $p^{k+1} = p^k - c_k u^{k+1}$.

From the form of F, we can express the "inf" expression as

$$\inf_{x\in\mathbb{R}^n} \left\{ f(x) + \inf \left\{ \tfrac{1}{2c_k}\left\|p^k - c_k u\right\|^2 \;\middle|\; u + (Ax - b) \in K \right\} \right\}.$$

Part (e) tells us that inner "inf" is attained by setting

$$u = \mathrm{proj}_K(Ax - b + \tfrac{1}{c_k}p^k) - (Ax - b).$$

Inserting the optimal inner value of $u = \mathrm{proj}_K(Ax - b + \frac{1}{c_k}p^k) - (Ax - b)$, the inner "inf" may be expressed as

$$\begin{aligned}
&\tfrac{1}{2c_k}\left\|p^k - c_k\left(\mathrm{proj}_K(Ax - b + \tfrac{1}{c_k}p^k) - (Ax - b)\right)\right\|^2 \\
&= \tfrac{1}{2c_k}\left\|p^k + c_k(Ax - b) - c_k\,\mathrm{proj}_K(Ax - b + \tfrac{1}{c_k}p^k)\right\|^2 \\
&= \tfrac{1}{2c_k}\left\|p^k + c_k(Ax - b) - \mathrm{proj}_K\left(p^k + c_k(Ax - b)\right)\right\|^2 \qquad \text{[by part (c)]} \\
&= \tfrac{1}{2c_k}\left[\mathrm{dist}_K(p^k + c_k(Ax - b))\right]^2
\end{aligned}$$

So, we obtain x^{k+1} by solving the problem

$$\min_{x\in\mathbb{R}^n} \left\{ f(x) + \tfrac{1}{2c_k}\left[\mathrm{dist}_K\big(p^k + c_k(Ax - b)\big)\right]^2 \right\},$$

which is identical to (3). From the analysis above concerning u, the inner "inf" is attained at

$$u^{k+1} = \mathrm{proj}_K(Ax^{k+1} - b + \tfrac{1}{c_k}p^k) - (Ax^{k+1} - b),$$

Next, the multiplier update $p^{k+1} = p^k - c_k u^{k+1}$ (for $\rho_k = 1$) becomes

$$\begin{aligned}
p^{k+1} &= p^k - c_k u^{k+1} \\
&= p^k - c_k\Big[\mathrm{proj}_K(Ax^{k+1} - b + \tfrac{1}{c_k}p^k) - (Ax^{k+1} - b)\Big] \\
&= p^k + c_k(Ax^{k+1} - b) - c_k\,\mathrm{proj}_K(Ax^{k+1} - b + \tfrac{1}{c_k}p^k) \\
&= p^k + c_k(Ax^{k+1} - b) - \mathrm{proj}_K\big(p^k + c_k(Ax^{k+1} - b)\big) && \text{[by part (c)]} \\
&= \mathrm{proj}_{K^*}\big(p^k + c_k(Ax^{k+1} - b)\big) && \text{[by part (d)]},
\end{aligned}$$

thus verifying the multiplier update formula (4).

Note that the recursion (3)-(4) generalizes the methods of multipliers we derived in class for both equality constraints (the case $K = \{0\}$) and inequality constraints (the case $K = \mathbb{R}^m_- = \{v \in \mathbb{R}^m \mid v \le 0\}$).

(g) Proposition 3.8 guarantees that $\{p^k\}$ will converge to an optimal dual solution. It also guarantees that $u^k \to 0$ and

$$\limsup_{k\to\infty} F(x^k, u^k) \le \inf\{F(x,0) \mid x \in \mathbb{R}^n\}$$

Since the minimum in the (3.23) will only occur at a point where $F(x^{k+1}, u^{k+1})$ is finite and hence equal to $f(x^{k+1})$, the values $F(x^k, u^k)$ on the left if this inequality will always equal $f(x^k)$. For the particular F, one will have $F(x,0) = +\infty$ unless $Ax - b \in K$, so the "inf" on the right may be rewritten as

$$\inf\{f(x) \mid x \in \mathbb{R}^n, Ax + b \in K\}.$$

Together, this means that above inequality may be rewritten as $\limsup_{k\to\infty} f(x^k) \le \inf\{f(x) \mid x \in \mathbb{R}^n, Ax + b \in K\}$ as claimed.

Finally, Proposition 3.8 guarantees that $u^k \to 0$. In the expression for u^{k+1} above, the point $\mathrm{proj}_K(Ax^{k+1} - b + \frac{1}{c_k}p^k)$ is in K. Therefore,

$$\begin{aligned}
(\forall\, k) \quad \mathrm{dist}_K(Ax^{k+1} - b) &= \inf_{w\in K}\left\{\left\|w - (Ax^{k+1} - b)\right\|\right\} \\
&\le \left\|\mathrm{proj}_K(Ax^{k+1} - b + \tfrac{1}{c_k}p^k) - (Ax^{k+1} - b)\right\| \\
&= \left\|u^{k+1}\right\|.
\end{aligned}$$

Since $u^k \to 0$ and $\mathrm{dist}_K(Ax^{k+1} - b)$ is nonnegative, it immediately follows that $\mathrm{dist}_K(Ax^{k+1} - b) \to 0$.

3. (a) To establish that F is closed and convex:

- The function $F_1 : (x, u) \mapsto f(x)$ is convex since it is the composition of the convex function f with the linear map $L : (x, u) \mapsto x$.
- The function $F_2 : (x, u) \mapsto g(Mx + u)$ is convex, since g is convex and the map $H : (x, u) \mapsto Mx + u$ is linear.
- F is convex because it is the sum of the two convex functions F_1 and F_2.
- Since f is lower semicontinuous (being closed), g is lower semicontinuous (also being closed), and linear map H defined immediately above is continuous, it is clear that $F : (x, u) \mapsto f(x) + g(Mx + u)$ is lower semicontinuous, hence closed.

We now show that F must be proper: since f and g are proper, there exist $\bar{x} \in \mathbb{R}^n$, $\bar{w} \in \mathbb{R}^m$ such that $f(\bar{x}), g(\bar{w}) \leq +\infty$. Setting $\bar{u} = \bar{w} - M\bar{x}$, we have $M\bar{x} + \bar{u} = \bar{w}$ and therefore $F(\bar{x}, \bar{u}) = f(\bar{x}) + g(\bar{w}) < +\infty$. Note that this result does not require there to be any point $x \in \mathbb{R}^n$ for which $f(x) + g(Mx)$ is finite. So F is proper even when $f + g \circ M$ is not.

(b) Using the definition $L(x,p) = \inf_{u \in \mathbb{R}^m} \big\{F(x,u) - \langle p, u\rangle\big\}$ and the given form for F, we have for any x for which $f(x) < \infty$ that

$$
\begin{aligned}
L(x,p) &= \inf_{u \in \mathbb{R}^m} \{f(x) + g(Mx + u) - \langle u, p\rangle\} \\
&= f(x) + \inf_{u \in \mathbb{R}^m} \{g(Mx + u) - \langle u, p\rangle\} \qquad\qquad \text{(when } f(x) < +\infty) \\
&= f(x) + \inf_{v \in \mathbb{R}^m} \{g(v) - \langle v - Mx, p\rangle\} \\
&= f(x) + \langle Mx, p\rangle + \inf_{v \in \mathbb{R}^m} \{g(v) - \langle v, p\rangle\} \\
&= f(x) + p^\top Mx - g^*(p),
\end{aligned}
$$

where the third equality follows by making the substitution $v = Mx + u$ and thus $u = v - Mx$ (since u can range throughout $\mathbb{R}^m$, $Mx + u$ can also take any value in $\mathbb{R}^m$). Note that if $f(x) = +\infty$, then the first infimand in the chain above is $+\infty$ for all u, and therefore $L(x, u) = +\infty$ regardless of the value of $g^*(p)$. Thus, we may write

$$
L(x,p) = \begin{cases} +\infty, & \text{if } f(x) = +\infty, \\ f(x) + p^\top Mx - g^*(p), & \text{if } f(x) < +\infty. \end{cases}
$$

The second case below includes the possibility that $L(x, p) = -\infty$, which will occur when $g^*(p) = +\infty$.

(c) The dual objective obtained from the above setup is

$$
\begin{aligned}
D^*(0,p) &= \inf_{x \in \mathbb{R}^n} \{L(x,p)\} \\
&= \inf_{x \in \mathbb{R}^n} \{f(x) + p^\top Mx - g^*(p)\} \\
&= \inf_{x \in \mathbb{R}^n} \{f(x) + \langle M^\top p, x\rangle\} - g^*(p) \\
&= \inf_{x \in \mathbb{R}^n} \{f(x) - \langle x, -M^\top p\rangle\} - g^*(p) \\
&= -f^*(-M^\top p) - g^*(p),
\end{aligned}
$$

precisely as obtained with the Fenchel approach and also on page 67 of the class notes.

(d) In our case, (3.17) reduces to

$$(x^{k+1}, u^{k+1}) \in \operatorname*{arg\,min}_{\substack{x\in\mathbb{R}^n \\ u\in\mathbb{R}^m}} \left\{ f(x) + g(Mx+u) - \langle p^k, u\rangle + \frac{c_k}{2}\|u\|^2 \right\}.$$

With a change of variables $z = Mx + u$, hence $u = z - Mx$, the minimand may be written

$$\begin{aligned} f(x) + g(z) - \langle p^k, z - Mx\rangle &+ \frac{c_k}{2}\|z - Mx\|^2 \\ &= f(x) + g(z) + \langle p^k, Mx - z\rangle + \frac{c_k}{2}\|Mx - z\|^2. \end{aligned}$$

Allowing (x, z) to range of over all of $\mathbb{R}^n \times \mathbb{R}^m$ is equivalent to allowing $(x, u) = (x, z - Mx)$ to range over all of $\mathbb{R}^n \times \mathbb{R}^m$, so one may rewrite the minimization calculation as

$$\begin{aligned} (x^{k+1}, z^{k+1}) &\in \operatorname*{arg\,min}_{\substack{x\in\mathbb{R}^n \\ z\in\mathbb{R}^m}} \left\{ f(x) + g(z) - \langle p^k, z - Mx\rangle + \frac{c_k}{2}\|Mx - z\|^2 \right\} \\ u^{k+1} &= z^{k+1} - Mx^{k+1}. \end{aligned}$$

The update $p^{k+1} = p^k - c_k u^{k+1}$ then takes the form $p^{k+1} = p^k - c_k(z^{k+1} - Mx^{k+1}) = p^k + c_k(Mx^{k+1} - z^{k+1})$, so the entire augmented Lagrangian procedure is equivalent to

$$\begin{aligned} (x^{k+1}, z^{k+1}) &\in \operatorname*{arg\,min}_{\substack{x\in\mathbb{R}^n \\ z\in\mathbb{R}^m}} \left\{ f(x) + g(z) - \langle p^k, z - Mx\rangle + \frac{c_k}{2}\|Mx - z\|^2 \right\} \\ p^{k+1} &= p^k + c_k(Mx^{k+1} - z^{k+1}), \end{aligned}$$

which is the same procedure found on page 67.

4. (a) The normal cone map of a set $C \subseteq \mathbb{R}^m$ is defined by

$$N_C(x) \doteq \{ y \in \mathbb{R}^n \mid (\forall\, x' \in C)\ \langle y, x' - x\rangle \le 0 \}.$$

When $C = S + d$, the "for all" condition inside the above definition may be written

$$(\forall\, s' \in S) \quad \langle y, s' + d - x\rangle \le 0.$$

If one fixes any $x \in S + d$ and sets $s \doteq x - d \in S$, the "for all" condition is in turn equivalent to

$$\begin{aligned} y \in N_{S+d}(x) \quad &\Leftrightarrow \quad (\forall\, s' \in S) \quad \langle y, s' - s\rangle \le 0 \\ &\Leftrightarrow \quad (\forall\, t \in S) \quad \langle y, t\rangle \le 0 \\ &\Leftrightarrow \quad t \in S^{\perp}, \end{aligned}$$

where the second equivalence comes from a change of variables $t = s' - s$, which can still range exactly over all of S if s' does. Since $N_C(x) = \emptyset$ whenever $x \notin C$, the full formula for N_{S+d} is

$$N_{S+d}(x) = \begin{cases} S^\perp, & \text{if } x \in S + d \\ \emptyset, & \text{if } x \notin S + d. \end{cases}$$

The conditions in this formula may of course be equivalently written as $x - d \in S$ and $x - d \notin S$, respectively. The resulting formula is essentially the same formula we already encountered for the normal cone map of a subspace, which is the case $d = 0$.

(b) Referring to the characterization of projection in Proposition 2.2.1(b) of the Bertsekas *et al.* textbook and then using part (a) of this problem

$$\begin{aligned} z = \mathrm{proj}_{S+d}(y) \quad &\Leftrightarrow \quad y - z \in N_{S+d}(y) \\ &\Leftrightarrow \quad z \in S + d \quad \wedge \quad y - z \in S^\perp. \end{aligned}$$

If one lets $u \doteq \mathrm{proj}_S(y - d) \in S$, then $(y - d) - u = \mathrm{proj}_{S^\perp}(y - d) \in S^\perp$. If one then lets $z = u + d$, it then follows that

$$z = u + d \in S + d \qquad y - z = y - (u + d) = (y - d) - u \in S^\perp,$$

meaning that $z = \mathrm{proj}_S(y)$. Therefore, the procedure for computing $\mathrm{proj}_{S+d}(y)$ is simply to return $\mathrm{proj}_S(y - d) + d$.

(c) Fixing any $y \in \mathbb{R}^m$ and $w \in S^\perp$,

$$\begin{aligned} \mathrm{prox}_{S+d}(y + w) &= \mathrm{proj}_S(y + w - d) + d && \text{[by part (b)]} \\ &= \mathrm{proj}_S(y - d) + \mathrm{proj}_S(w) + d && \text{[since } \mathrm{proj}_S \text{ is linear]} \\ &= \mathrm{proj}_S(y - d) + 0 + d && \text{[since } w \in S^\perp\text{]} \\ &= \mathrm{proj}_S(y - d) + d && \\ &= \mathrm{proj}_{S+d}(y) && \text{[by part (b) again].} \end{aligned}$$

(d) Consider the ADMM (3.84)-(3.86) in the class notes, with $g = \delta_{S+d}$. Fix any $k \geq 0$, suppose $p^k \in S^\perp$, and consider executing first (3.84) and then (3.85). Then, (3.85) performs the calculation

$$\begin{aligned} z^{k+1} &= \arg\min_{z \in \mathbb{R}^n} \left\{ g(z) - \langle p^k, z \rangle + \frac{c}{2} \left\| M x^{k+1} - z \right\|^2 \right\} \\ &= \arg\min_{z \in \mathbb{R}^n} \left\{ \delta_{S+d}(z) - \langle p^k, z \rangle + \frac{c}{2} \left\| M x^{k+1} - z \right\|^2 \right\} \\ &= \arg\min_{z \in S+d} \left\{ -\langle p^k, z \rangle + \frac{c}{2} \left\| z - M x^{k+1} \right\|^2 \right\} \\ &= \arg\min_{z \in S+d} \left\{ \frac{c}{2} \left\| z - (M x^{k+1} + \tfrac{1}{c} p^k) \right\|^2 \right\} && (*) \text{ (see below)} \\ &= \mathrm{proj}_{S+d}(M x^{k+1} + \tfrac{1}{c} p^k) \\ &= \mathrm{proj}_{S+d}(M x^{k+1}), && (\dagger) \text{ (see below)} \end{aligned}$$

where

- Step $(*)$ follows by completing the square and dropping constants in the manner demonstrated in the last class of the semester.
- The last equality $(\dagger)$ follows from part (b) of this problem because $p^k \in S^\perp$ and thus $(1/c)p^k \in S^\perp$.

In summary, the second step of the ADMM reduces to $z^{k+1} = \operatorname{proj}_{S+d}(Mx^{k+1})$ when $g = \delta_{S+d}$ and $p^k \in S^\perp$.

Turn now to the multiplier update (3.86), which involves the vector

$$\begin{aligned} Mx_{k+1} - z^{k+1} &= Mx_{k+1} - \operatorname{proj}_{S+d}(Mx^{k+1}) \\ &\in N_{S+d}\big(\operatorname{proj}_{S+d}(Mx^{k+1})\big) && \text{[by textbook Proposition 2.2.1(b)]} \\ &= S^\perp && \text{[by part (a)].} \end{aligned}$$

Thus, $c(Mx_{k+1} - z^{k+1}) \in S^\perp$ since linear subspaces are closed under scalar multiplication, and so $p^{k+1} = p^k + c(Mx_{k+1} - z^{k+1}) \in S^\perp$ since it is the sum of two vectors in $S^\perp$.

Summarizing, $z^k \in \operatorname{proj}_{S+d}(Mx^{k+1})$ and $p^{k+1} \in S^\perp$ if $p^k \in S^\perp$. If $p^0 \in S^\perp$, then by induction $p^k \in S^\perp$ for all k and the recursions take the claimed form.

Aside: it is always possible to assure $p^0 \in S^\perp$ by setting $p^0 = 0$. But if one starts with an arbitrary nonzero $p \in \mathbb{R}^m$, the second step of the algorithm should be $z^1 = \operatorname{proj}_{S+d}(Mx^1 + \frac{1}{c}p^0)$, not dropping the $\frac{1}{c}p^0$ term since p^0 might not be in $S^\perp$. The first multiplier update then sets

$$\begin{aligned} p^1 &= p^0 + c(Mx^1 - z^1) \\ &= p^0 + c\big(Mx^1 - \operatorname{proj}_{S+d}(Mx^1 + \tfrac{1}{c}p^0)\big) \\ &= c\big(Mx^1 + \tfrac{1}{c}p^0 - \operatorname{proj}_{S+d}(Mx^1 + \tfrac{1}{c}p^0)\big) \\ &\in S^\perp, \end{aligned}$$

where the last step uses similar logic to the analysis above. This operation places $p^1 \in S^\perp$, so all subsequent iterations may use the simpler formula $z^{k+1} = \operatorname{proj}_{S+d}(Mx^{k+1})$, with p^k remaining in $S^\perp$ for $k \geq 1$.

Another aside: that the multipliers p^k always reside in $S^\perp$ (except for perhaps at iteration 0) may also be viewed a consequence of the underlying recursions stated near the top of page 81 of the notes, which impose for all $k \geq 0$ the condition $z^{k+1} \in \partial g^*(p^{k+1})$, equivalent to $p^{k+1} \in \partial g(z^{k+1})$. In the case $g = \delta_{S+d}$, that implies that one always has $p^{k+1} \in S^\perp$ by part (a).

5. (a) Fix any $x = (x_1, \ldots, x_\ell) \in \mathbb{R}^n$. Then, using the given forms of f, g, and M, one has

$$\begin{aligned} f(x) + g(Mx) &= \left(\sum_{i=1}^{\ell} f_i(x_i)\right) + g(A_1x_1, A_2x_2, \ldots, A_\ell x_\ell) \\ &= \left(\sum_{i=1}^{\ell} f_i(x_i)\right) + \begin{cases} 0, & \text{if } \sum_{i=1}^{\ell} A_i x_i = b \\ +\infty, & \text{otherwise} \end{cases} \end{aligned}$$

$$= \begin{cases} \sum_{i=1}^{\ell} f_i(x_i), & \text{if } \sum_{i=1}^{\ell} A_i x_i = b \\ +\infty, & \text{otherwise.} \end{cases}$$

Therefore, minimizing $f(x) + g(Mx)$ is equivalent to minimizing $\sum_{i=1}^{\ell} f_i(x_i)$ subject to the constraint $A_i x_i = b$. It immediately follows that x is a minimizer of $f(x) + g(Mx)$ if and only if it minimizes $\sum_{i=1}^{\ell} f_i(x_i)$ subject to $A_i x_i = b$.

(b) The key observation here is that the set

$$\left\{ (z_1, \ldots, z_\ell) \in \mathbb{R}^{\ell m} \;\middle|\; \sum_{i=1}^{l} z_i = b \right\}$$

on which g returns 0 is of the form $S + d$, where

$$S \doteq \left\{ (s_1, \ldots, s_\ell) \in \mathbb{R}^{\ell m} \;\middle|\; \sum_{i=1}^{l} s_i = 0 \right\}$$

is clearly a linear subspace of $\mathbb{R}^{\ell m}$ and one sets for example $d \doteq (b, 0, 0, \ldots, 0)$. It suffices here to choose d to be any vector $(d_1, \ldots, d_\ell)$ such that $\sum_{i=1}^{\ell} d_i = b$; the choice $d = (b, 0, 0, \ldots, 0)$ is merely one of the simplest ones.

Thus, we are in the situation of the previous problem, meaning that if one starts with multipliers $p^0 \in S^\perp$, the ADMM will assume the form

$$\begin{aligned} x^{k+1} &\in \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\{ f(x) + \langle p^k, Mx \rangle + \frac{c}{2} \left\| Mx - z^k \right\|^2 \right\} \\ z^{k+1} &= \operatorname{proj}_{S+d}(Mx^{k+1}) \\ p^{k+1} &= p^k + c(Mx^{k+1} - z^{k+1}). \end{aligned}$$

Now, S is the kernel $\ker B \doteq \left\{ s \in \mathbb{R}^{\ell m} \;\middle|\; Bs = 0 \right\}$ of the matrix B defined by

$$B \doteq \begin{bmatrix} \mathrm{Id}_m & \mathrm{Id}_m & \cdots & \mathrm{Id}_m \end{bmatrix},$$

where Id_m denotes the $m \times m$ identity matrix. From the basic linear algebra fact mentioned in the problem, that $S = \ker B$ implies that

$$\begin{aligned} S^\perp &= (\ker B)^\perp \\ &= \{ B^\top u \mid u \in \mathbb{R}^m \} \\ &= \left\{ \begin{bmatrix} \mathrm{Id}_m & \mathrm{Id}_m & \cdots & \mathrm{Id}_m \end{bmatrix}^\top u \;\middle|\; u \in \mathbb{R}^m \right\} \\ &= \left\{ (u, u, \ldots, u) \in \mathbb{R}^{\ell m} \;\middle|\; u \in \mathbb{R}^m \right\}. \end{aligned}$$

Thus, any initial Lagrange multiplier estimate of the form $(p^0, p^0, \ldots, p^0) \in \mathbb{R}^{\ell m}$, where p^0 is any vector in $\mathbb{R}^m$, will lie in $S^\perp$. By the previous problem, all subsequent Lagrange multiplier estimates generated by the ADMM will also lie in $S^\perp$, meaning that they will be of the form $(p^k, p^k, \ldots, p^k)$. So, for each $k \geq 0$, it suffices to use a single vector $p^k \in \mathbb{R}^m$ to represent the Lagrange multiplier estimates.

Next, consider the projecting and arbitrary $y = (y_1, y_2, \ldots, y_m) \in \mathbb{R}^{\ell m}$ onto $S+d$. As in problem 4, one has $z = \operatorname{proj}_{S+d}(y)$ if $z \in S+d$ and $y - z \in S^\perp$. From the form already demonstrated for $S^\perp$, having $y - z \in S^\perp$ means that

$$(y_1, y_2, \ldots, y_\ell) - (z_1, z_2, \ldots, z_\ell) = (u, u, \ldots, u)$$

for some $u \in \mathbb{R}^m$. That is, $y_i - z_i = u$ and thus $z_i = y_i - u$ for $i = 1, \ldots, m$. In view of these equations, having $z \in S+d$ is equivalent to

$$\begin{aligned}
\sum_{i=1}^{\ell} z_i = b \qquad &\Leftrightarrow \qquad \sum_{i=1}^{\ell} (y_i - u) = b \\
&\Leftrightarrow \qquad \left(\sum_{i=1}^{\ell} y_i\right) - \ell u = b \\
&\Leftrightarrow \qquad \left(\sum_{i=1}^{\ell} y_i\right) - b = \ell u \\
&\Leftrightarrow \qquad u = \frac{1}{\ell}\left(\sum_{i=1}^{\ell} y_i - b\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\operatorname{proj}_{S+d}(y_1, y_2, \ldots, y_\ell) &= (y_1 - u, y_2 - u, \ldots, y_\ell - u) \\
&= (z_1, z_2, \ldots, z_\ell),
\end{aligned}$$

where

$$(\forall\, i = 1, \ldots, \ell) \qquad z_i = y_i - \frac{1}{\ell}\left(\sum_{i=1}^{\ell} y_i - b\right) = y_i + \frac{1}{\ell}\left(b - \sum_{i=1}^{\ell} y_i\right).$$

We now consider how to compute each of the ADMM operations. We will use the notation that the working vectors of the recursion are

$$\begin{aligned}
x^k &= (x_1^k, x_2^k, \ldots, x_\ell^k) \in \mathbb{R}^n, && \text{with each } x_i^k \in \mathbb{R}^{n_i} \\
z^k &= (z_1^k, z_2^k, \ldots, z_\ell^k) \in \mathbb{R}^{\ell m}, && \text{with each } z_i^k \in \mathbb{R}^m \\
&(p^k, p^k, \ldots, p^k) \in \mathbb{R}^{\ell m}, && p^k \in \mathbb{R}^m
\end{aligned}$$

the form of the multiplier vector $(p^k, p^k, \ldots, p^k)$ being justified by the form of $S^\perp$ derived above. Consider first the "x" recursion $x^{k+1} \in \arg\min_{x \in \mathbb{R}^n} \big\{ f(x) + \langle p^k, Mx \rangle + \frac{c}{2}\|Mx - z^k\|^2 \big\}$. With the given choices of f and M and the iteration notation above, this calculation is

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\ell \end{bmatrix} \in \mathop{\arg\min}_{\substack{x_i \in \mathbb{R}^{n_i} \\ i=1\ldots,\ell}} \left\{ \sum_{i=1}^{\ell} f_i(x_i) + \begin{bmatrix} p^k \\ p^k \\ \vdots \\ p^k \end{bmatrix}^\top \begin{bmatrix} A_1 x_1 \\ A_2 x_2 \\ \vdots \\ A_\ell x_\ell \end{bmatrix} + \frac{c}{2} \left\| \begin{bmatrix} A_1 x_1 \\ A_2 x_2 \\ \vdots \\ A_\ell x_\ell \end{bmatrix} - \begin{bmatrix} z_1^k \\ z_2^k \\ \vdots \\ z_\ell^k \end{bmatrix} \right\|^2 \right\}$$

$$= \operatorname*{arg\,min}_{\substack{x_i \in \mathbb{R}^{n_i} \\ i=1\ldots,\ell}} \left\{ \sum_{i=1}^{\ell} f_i(x_i) + \sum_{i=1}^{\ell} \langle p^k, A_i x_i \rangle + \frac{c}{2} \sum_{i=1}^{\ell} \left\| A_i x_i - z_i^k \right\|^2 \right\}$$

$$= \bigtimes_{i=1}^{\ell} \operatorname*{arg\,min}_{x_i \in \mathbb{R}^{n_i}} \left\{ f_i(x_i) + \langle p^k, A_i x_i \rangle + \frac{c}{2} \left\| A_i x_i - z_i^k \right\|^2 \right\},$$

where the final equality occurs because the preceding minimand clearly consists of triplets of terms each of which depend only on the single block of variables x_i, for $i = 1, \ldots, \ell$. Therefore, $x^{k+1} = (x_1^{k+1}, x_2^{k+1}, \ldots, x_\ell^{k+1})$ may be computed by executing

$$x_i^{k+1} \in \operatorname*{arg\,min}_{x_i \in \mathbb{R}^{n_i}} \left\{ f_i(x_i) + \langle p^k, A_i x_i \rangle + \frac{c}{2} \left\| A_i x - z_i^k \right\|^2 \right\} \qquad \forall\, i = 1, \ldots, \ell,$$

matching the first line in the recursion cycle the problem asked to be proved. Next we turn to the "z" recursion formula $z^{k+1} = \operatorname{proj}_{S+d}(Mx^{k+1})$. To find a more explicit form for this calculation, we may simply plug $y = Mx^{k+1} = (A_1 x_1^{k+1}, A_2 x_2^{k+1}, \ldots, A_\ell x_\ell^{k+1})$ into formula for $\operatorname{proj}_{S+d}(y)$ derived above to obtain obtain

$$z_i^{k+1} = A_i x_i^{k+1} + \frac{1}{\ell} \left(b - \sum_{j=1}^{\ell} A_j x_j^{k+1} \right) \qquad \forall\, i = 1, \ldots, \ell,$$

matching the second line in the desired cycle. Finally, the multiplier update takes the form

$$\begin{bmatrix} p^k \\ p^k \\ \vdots \\ p^k \end{bmatrix} + c(Mx^{k+1} - z^{k+1}) = \begin{bmatrix} p^k \\ p^k \\ \vdots \\ p^k \end{bmatrix} + c \left(\begin{bmatrix} A_1 x_1^{k+1} \\ A_2 x_2^{k+1} \\ \vdots \\ A_\ell x_\ell^{k+1} \end{bmatrix} - \begin{bmatrix} z_1^{k+1} \\ z_2^{k+1} \\ \vdots \\ z_\ell^{k+1} \end{bmatrix} \right)$$
$$= \begin{bmatrix} p^k \\ p^k \\ \vdots \\ p^k \end{bmatrix} + c \begin{bmatrix} A_1 x_1^{k+1} - z_1^{k+1} \\ A_2 x_2^{k+1} - z_2^{k+1} \\ \vdots \\ A_\ell x_\ell^{k+1} - z_\ell^{k+1} \end{bmatrix}.$$

For any $i = 1, \ldots, \ell$, consider any of the terms $A_i x_i^{k+1} - z_i^{k+1}$ in the last vector in this expression. Substituting the formula for z_i^{k+1} derived above, one has

$$A_i x_i^{k+1} - z_i^{k+1} = A_i x_i^{k+1} - \left(A_i x_i^{k+1} + \frac{1}{\ell} \left(b - \sum_{j=1}^{\ell} A_j x_j^{k+1} \right) \right)$$
$$= \frac{1}{\ell} \left(\sum_{j=1}^{\ell} A_j x_j^{k+1} - b \right),$$

which is independent of i. Thus, each block p^k of $(p^k, p^k, \ldots, p^k)$ is updated to p^{k+1} by adding the same value $c \cdot (1/\ell)\big(\sum_{j=1}^{\ell} A_j x_j^{k+1} - b\big)$, which may be written (changing summation the index from i to j) as

$$p^{k+1} = p^k + \frac{c}{\ell}\left(\sum_{i=1}^{\ell} A_i x_i^{k+1} - b\right),$$

which is the last recursion in the specified algorithm.

Aside: this kind of algorithm is straightforward to implement in a distributed computing environment, with one processing element for each of the ℓ blocks (or several blocks per processing element). The processor for block i would store A_i, x_i^k, z_i^k, and the information needed to compute f_i. The Lagrange multiplier estimates p^k would best be stored redundantly on each processor. The only communication between processors in each iteration would be the global sum needed to calculate $\sum_{i=1}^{\ell} A_i x_i^k$. Many distributed processing environments have efficient communication primitives for this kind of operation. All the other operations could be performed locally on each processing element.