Pr. 1.

(a) $\tilde{x} = \beta_1/\beta_3$, $\tilde{y} = \beta_2/\beta_3$.

(b) From the first and third elements of vector $\boldsymbol{\beta}$, we get

$$\beta_3 \tilde{x} = h_1^T \boldsymbol{\alpha} = \boldsymbol{\alpha}^T h_1 \tag{1}$$

$$\beta_3 = h_3^T \boldsymbol{\alpha} = \boldsymbol{\alpha}^T h_3 \tag{2}$$

from which we see that

$$\boldsymbol{\alpha}^T h_1 - \tilde{x}\boldsymbol{\alpha}^T h_3 = 0.$$

In matrix-vector form:

$$\begin{bmatrix} \boldsymbol{\alpha}^T & \mathbf{0}^T & -\tilde{x}\boldsymbol{\alpha}^T \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}^T & \mathbf{0}^T & -\tilde{x}\boldsymbol{\alpha}^T \end{bmatrix} \operatorname{vec}(\boldsymbol{H}) = 0.$$

Following the same argument with the second and third elements of β, we get

$$\beta_3 \tilde{y} = h_2^T \boldsymbol{\alpha} = \boldsymbol{\alpha}^T h_2 \tag{3}$$

$$\beta_3 = h_3^T \boldsymbol{\alpha} = \boldsymbol{\alpha}^T h_3$$

from which

$$\begin{bmatrix} \mathbf{0}^T & \boldsymbol{\alpha}^T & -\tilde{y}\boldsymbol{\alpha}^T \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0$$

so that the required matrix A is

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{\alpha}^T & \mathbf{0}^T & -\tilde{x}\boldsymbol{\alpha}^T \\ \mathbf{0}^T & \boldsymbol{\alpha}^T & -\tilde{y}\boldsymbol{\alpha}^T \end{bmatrix}.$$

Alternatively, one could combine (1), (2) and (3) to see that

$$\boldsymbol{\alpha}^T h_1 + \boldsymbol{\alpha}^T h_2 - (\tilde{x} + \tilde{y}) h_3 = \beta_3 \tilde{x} + \beta_3 \tilde{y} - (\tilde{x} + \tilde{y})\beta_3 = 0,$$

so that

$$A = \begin{bmatrix} \boldsymbol{\alpha}^T & \boldsymbol{\alpha}^T & -(\tilde{x} + \tilde{y})\boldsymbol{\alpha}^T \\ \boldsymbol{\alpha}^T & \boldsymbol{\alpha}^T & -(\tilde{x} + \tilde{y})\boldsymbol{\alpha}^T \end{bmatrix}.$$

(c) Vector $\boldsymbol{h}$ is in the *null space* of $\boldsymbol{A}$, *i.e.*, $\boldsymbol{h} \in \mathcal{N}(\boldsymbol{A})$.

Pr. 2.

$$\begin{aligned}(\boldsymbol{A}^\dagger \boldsymbol{b})^T (\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}) \boldsymbol{y} &= \boldsymbol{b}^T (\boldsymbol{A}^\dagger)^T (\boldsymbol{I} - \boldsymbol{V}\boldsymbol{\Sigma}^\dagger \boldsymbol{U}^T \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T)\boldsymbol{y} = \boldsymbol{b}^T \boldsymbol{U} (\boldsymbol{\Sigma}^\dagger)^T \boldsymbol{V}^T (\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{V}\boldsymbol{\Sigma}^\dagger \boldsymbol{\Sigma} \boldsymbol{V}^T)\boldsymbol{y} \\ &= \boldsymbol{b}^T \boldsymbol{U}((\boldsymbol{\Sigma}^\dagger)^T - (\boldsymbol{\Sigma}^\dagger)^T \boldsymbol{\Sigma}^\dagger \boldsymbol{\Sigma}) \boldsymbol{V}^T \boldsymbol{y}.\end{aligned}$$

If $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$, then we can verify that $(\boldsymbol{\Sigma}^\dagger)^T = (\boldsymbol{\Sigma}^\dagger)^T \boldsymbol{\Sigma}^\dagger \boldsymbol{\Sigma}$ through direct multiplication, so that $(\boldsymbol{A}^\dagger \boldsymbol{b})^T (\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}) \boldsymbol{y} = 0$.

Pr. 3.

(a) Every point $[x\, y\, z]^T$ on the plane satisfies the equation $\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0$. Therefore every point on the plane must satisfy $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathcal{N}([a \quad b \quad c])$. Clearly: $\begin{bmatrix} a & b & c \end{bmatrix} = \underbrace{1}_{=\boldsymbol{u}_1} \underbrace{\sqrt{a^2 + b^2 + c^2}}_{=\sigma_1} \underbrace{\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{\sqrt{a^2 + b^2 + c^2}}}_{=\boldsymbol{v}_1^T}$.

Note that $\mathcal{N}(\begin{bmatrix} a & b & c \end{bmatrix}) = \operatorname{span}(\{\boldsymbol{v}_2, \boldsymbol{v}_3\})$, since we are considering a rank–1 matrix. Therefore $\{\boldsymbol{v}_2, \boldsymbol{v}_3\}$ are an orthonormal basis for the plane. Two basis vectors are required to express every point on the plane.

(b) The nearest point on the plane is given by

$$P_{\mathcal{R}\{\boldsymbol{v}_2, \boldsymbol{v}_3\}} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = (\boldsymbol{v}_2 \boldsymbol{v}_2^T + \boldsymbol{v}_3 \boldsymbol{v}_3^T) \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = (\boldsymbol{I} - \boldsymbol{v}_1 \boldsymbol{v}_1^T) \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}.$$

Pr. 4.

Define: $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{q}_1 & \boldsymbol{q}_2\end{bmatrix}$, and $\boldsymbol{x} = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$.

(a) The linear least squares estimate that minimizes $||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2$ is given by $\widehat{\boldsymbol{x}} = \boldsymbol{A}^{\dagger}\boldsymbol{b}$. The SVD of $\boldsymbol{A}$ is simply:

$$\boldsymbol{A} = \sum_{i=1}^{2} 1 \boldsymbol{q}_i \boldsymbol{e}_i^T,$$

where $\boldsymbol{e}_i$ denotes the ith unit vector Thus:

$$\boldsymbol{A}^{\dagger} = \sum_{i=1}^{2} 1 \boldsymbol{e}_i \boldsymbol{q}_i^T,$$

and hence

$$\widehat{\boldsymbol{x}} = \boldsymbol{A}^{\dagger}\boldsymbol{b} = \sum_{i=1}^{2} 1 \boldsymbol{e}_i \boldsymbol{q}_i^T \boldsymbol{b} = \boldsymbol{A}^T\boldsymbol{b}.$$

This solution is unsurprising, because what we get is precisely the first two “coordinates” of the vector $\boldsymbol{b}$ relative to the basis whose first two basis vectors correspond to the columns of $\boldsymbol{A}$.

(b) Here we have that the residual (or error) vector:

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{b} = \sum_{i=3}^{n} \boldsymbol{q}_i \boldsymbol{q}_i^T \boldsymbol{b},$$

where $\boldsymbol{q}_i$ for $i = 3, \ldots, n$ are the $n-2$ (unit norm) basis vectors, orthogonal to $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ so that $\operatorname{span}(\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}) = \mathbb{R}^n$. Hence $\boldsymbol{q}_1^T\boldsymbol{r} = \sum_{i=3}^{n} \boldsymbol{q}_1^T \boldsymbol{q}_i \boldsymbol{b} = 0$ and $\boldsymbol{q}_2^T\boldsymbol{r} = \sum_{i=3}^{n} \boldsymbol{q}_2^T \boldsymbol{q}_i \boldsymbol{b} = 0$.

Pr. 5.

Here

$$\boldsymbol{A} = \begin{bmatrix}1 & 1 \\ 1 & 1\end{bmatrix} = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 \\ 1 & -1\end{bmatrix}}_{=\boldsymbol{U}} \underbrace{\begin{bmatrix}2 & 0 \\ 0 & 0\end{bmatrix}}_{=\boldsymbol{\Sigma}} \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1 \\ 1 & -1\end{bmatrix}^T}_{=\boldsymbol{V}^T},$$

since $\boldsymbol{A}$ is rank-1 and can be written as an outerproduct $\boldsymbol{A} = \boldsymbol{z}\boldsymbol{z}^T$ where $\boldsymbol{z} = \begin{bmatrix}1 \\ 1\end{bmatrix}$ and we know the eigen-decomposition of such rank-1 matrices from previous homeworks. Recall that since $\boldsymbol{A}$ is symmetric, positive-semidefinite, its eigen-decomposition is the same as a singular value decomposition.

Consider the minimum norm solution given by

$$\begin{aligned}\widehat{\boldsymbol{x}} = \boldsymbol{A}^{\dagger}\boldsymbol{b} &= \begin{bmatrix}1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2}\end{bmatrix}\begin{bmatrix}1/2 & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2}\end{bmatrix}^T \boldsymbol{b} \\ &= \begin{bmatrix}0.25 & 0.25 \\ 0.25 & 0.25\end{bmatrix}\begin{bmatrix}1 \\ 2\end{bmatrix} = \begin{bmatrix}0.75 \\ 0.75\end{bmatrix}.\end{aligned}$$

Here $\operatorname{rank}(\boldsymbol{A}) = 1 < 2$. Moreover

$$\ker(A) = \operatorname{span}\left(\begin{bmatrix}1 \\ -1\end{bmatrix}\right),$$

so that for $\alpha \in \mathbb{R}$, all

$$\widetilde{\boldsymbol{x}} = \widehat{\boldsymbol{x}} + \alpha\begin{bmatrix}1 \\ -1\end{bmatrix}$$

will yield the same the (minimum) squared error.

Pr. 6.

(a)

$$\begin{aligned}
&(\boldsymbol{U}_x \otimes \boldsymbol{U}_y)(\boldsymbol{\Sigma}_x \otimes \boldsymbol{\Sigma}_y)(\boldsymbol{V}_x \otimes \boldsymbol{V}_y)^H \\
=&(\boldsymbol{U}_x \otimes \boldsymbol{U}_y)(\boldsymbol{\Sigma}_x \otimes \boldsymbol{\Sigma}_y)(\boldsymbol{V}_x^H \otimes \boldsymbol{V}_y^H) \\
=&(\boldsymbol{U}_x \otimes \boldsymbol{U}_y)((\boldsymbol{\Sigma}_x \boldsymbol{V}_x^H) \otimes (\boldsymbol{\Sigma}_y \boldsymbol{V}_y^H)) \\
=&(\boldsymbol{U}_x \boldsymbol{\Sigma}_x \boldsymbol{V}_x^H) \otimes (\boldsymbol{U}_y \boldsymbol{\Sigma}_y \boldsymbol{V}_y^H) = \boldsymbol{X} \otimes \boldsymbol{Y}.
\end{aligned}$$

To show that $\boldsymbol{U}_x \otimes \boldsymbol{U}_y$ is unitary, observe that

$$\begin{aligned}
&(\boldsymbol{U}_x \otimes \boldsymbol{U}_y)^H(\boldsymbol{U}_x \otimes \boldsymbol{U}_y) \\
=&(\boldsymbol{U}_x^H \otimes \boldsymbol{U}_y^H)(\boldsymbol{U}_x \otimes \boldsymbol{U}_y) \\
=&(\boldsymbol{U}_x^H \boldsymbol{U}_x) \otimes (\boldsymbol{U}_y^H \boldsymbol{U}_y) \\
=&\boldsymbol{I}_m \otimes \boldsymbol{I}_p = \boldsymbol{I}_{mp}.
\end{aligned}$$

The above argument can be repeated to show that

- $(\boldsymbol{U}_x \otimes \boldsymbol{U}_y)(\boldsymbol{U}_x \otimes \boldsymbol{U}_y)^H = \boldsymbol{I}_{mp}$
- $(\boldsymbol{V}_x \otimes \boldsymbol{V}_y)^H(\boldsymbol{V}_x \otimes \boldsymbol{V}_y) = \boldsymbol{I}_{nq}$
- $(\boldsymbol{V}_x \otimes \boldsymbol{V}_y)(\boldsymbol{V}_x \otimes \boldsymbol{V}_y)^H = \boldsymbol{I}_{nq}$.

(b)

$$\begin{aligned}
&(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \otimes \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H \\
=&(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \otimes \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a^H \otimes \boldsymbol{Q}_b^H) \\
=&(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \boldsymbol{Q}_a^H \otimes \boldsymbol{\Lambda}_b \boldsymbol{Q}_b^H) \\
=&\boldsymbol{Q}_a \boldsymbol{\Lambda}_a \boldsymbol{Q}_a^H \otimes \boldsymbol{Q}_b \boldsymbol{\Lambda}_b \boldsymbol{Q}_b^H = \boldsymbol{A} \otimes \boldsymbol{B}.
\end{aligned}$$

To show that $\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b$ is unitary, observe that

$$\begin{aligned}
&(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b) \\
=&(\boldsymbol{Q}_a^H \otimes \boldsymbol{Q}_b^H)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b) \\
=&(\boldsymbol{Q}_a^H \boldsymbol{Q}_a \otimes \boldsymbol{Q}_b^H \boldsymbol{Q}_b) \\
=&\boldsymbol{I}_n \otimes \boldsymbol{I}_m = \boldsymbol{I}_{mn}.
\end{aligned}$$

The above argument can be repeated to show that $(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H = \boldsymbol{I}_{mn}$.

(c)

$$\begin{aligned}
\boldsymbol{A} \oplus \boldsymbol{B} &= \boldsymbol{A} \otimes \boldsymbol{I}_m + \boldsymbol{I}_n \otimes \boldsymbol{B} \\
&= \boldsymbol{A} \otimes (\boldsymbol{Q}_b \boldsymbol{I}_m \boldsymbol{Q}_b^H) + (\boldsymbol{Q}_a \boldsymbol{I}_n \boldsymbol{Q}_a^H) \otimes \boldsymbol{B} \\
\text{by (b)} \rightarrow &= (\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \otimes \boldsymbol{I}_m)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H + (\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{I}_n \otimes \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H \\
&= (\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \otimes \boldsymbol{I}_m + \boldsymbol{I}_n \otimes \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H \\
&= (\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \oplus \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)^H.
\end{aligned}$$

As in (b), $\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b$ is unitary.

Pr. 7.

We have to show that $||\boldsymbol{A}(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} - \boldsymbol{b}||_2 \leq \lambda\, ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2 + (1-\lambda)||\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}||_2$.
To that end note that

$$\begin{aligned}
f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) &= ||\boldsymbol{A}(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} - \boldsymbol{b}||_2 \\
&= ||\boldsymbol{A}(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} - \boldsymbol{b}(\lambda + 1 - \lambda)||_2 \\
&= ||(\lambda \boldsymbol{A}\boldsymbol{x} - \lambda \boldsymbol{b}) + ((1-\lambda)\boldsymbol{A}\boldsymbol{y} - (1-\lambda)\boldsymbol{b})||_2 \\
&\leq ||\lambda(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})||_2 + ||(1-\lambda)(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b})||_2, \qquad \text{(via the triangle inequality)} \\
&\leq \lambda ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2 + (1-\lambda)||\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}||_2 \\
&\leq \lambda f(x) + (1-\lambda) f(y),
\end{aligned}$$

establishing convexity. Next we need to show that, given two matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\forall \lambda \in [0,1]$:

$$\sigma_1(\lambda \boldsymbol{A} + (1-\lambda)\boldsymbol{B}) \leq \lambda \sigma_1(\boldsymbol{A}) + (1-\lambda)\sigma_1(\boldsymbol{B}).$$

From the hint:

$$\sigma_1\left(\lambda \boldsymbol{A} + (1-\lambda)\boldsymbol{B}\right) = \max_{\|u\|_2=1} \left\| \left(\lambda \boldsymbol{A} + (1-\lambda)\boldsymbol{B}\right) \boldsymbol{u} \right\|_2$$

By triangle inequality of norms,

$$\begin{aligned}
&\le \max_{\|\boldsymbol{u}\|_2=1} \|\lambda \boldsymbol{A}\boldsymbol{u}\|_2 + \|(1-\lambda)\boldsymbol{B}\boldsymbol{u}\|_2 \\
&\le \max_{\|\boldsymbol{u}\|_2=1} \lambda\|\boldsymbol{A}\boldsymbol{u}\|_2 + (1-\lambda)\|Bu\|_2 \\
&\le \max_{\|\boldsymbol{u}\|_2=1} \lambda\|\boldsymbol{A}\boldsymbol{u}\|_2 + \max_{\|\boldsymbol{u}\|_2=1} (1-\lambda)\|\boldsymbol{B}\boldsymbol{u}\|_2 \\
&= \lambda\, \sigma_1(\boldsymbol{A}) + (1-\lambda)\, \sigma_1(\boldsymbol{B}).
\end{aligned}$$

Thus, $\sigma_1(\cdot)$ is a convex function.

Pr. 8.

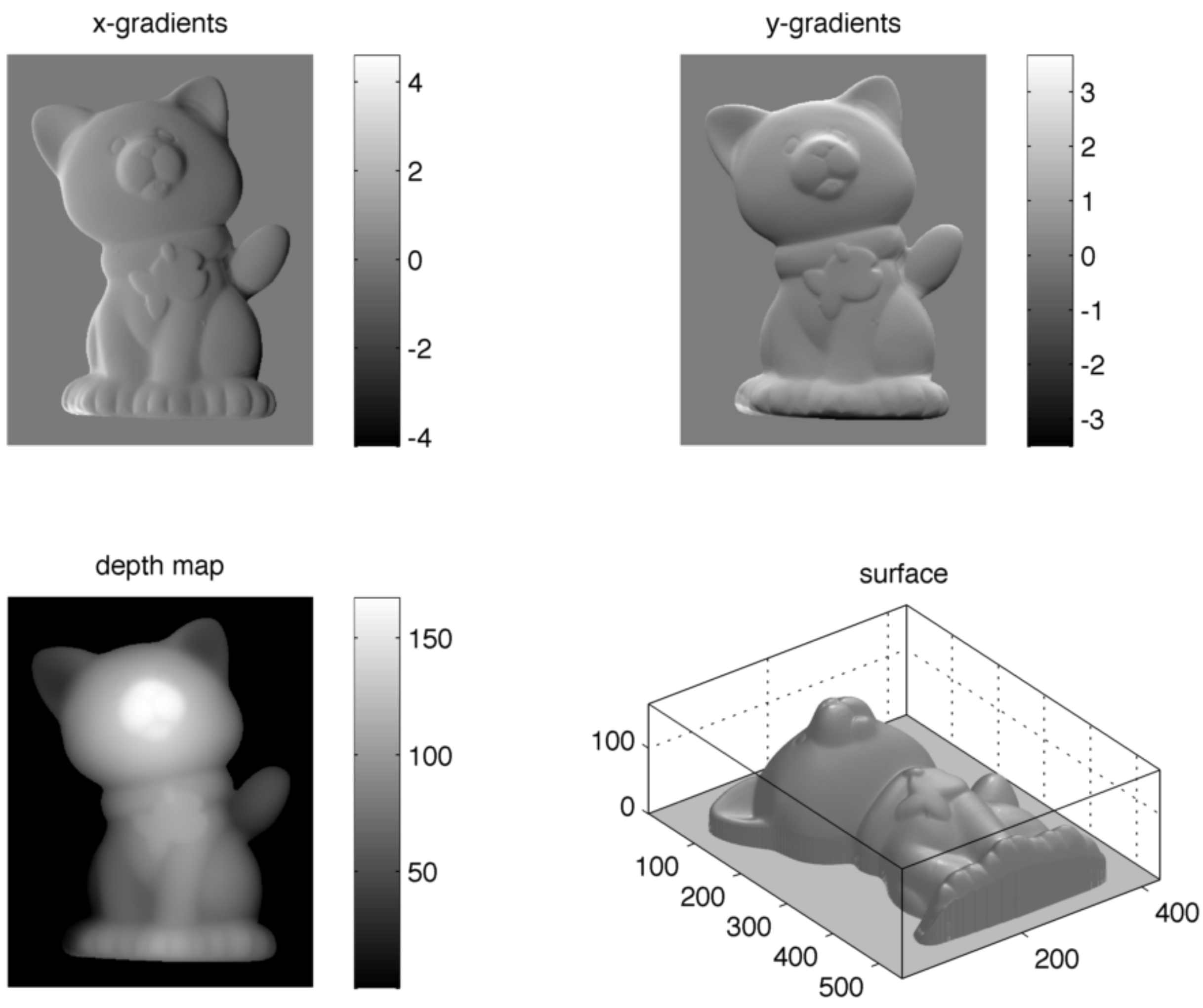

FIGURE 1. Photometric stereo reconstruction

Pr. 9.

(a) A possible `Julia` implementation is

```
function lsgd(A, b, mu, x0, nIters)
#
# Syntax:       x = lsgd(A, b, mu, x0, nIters)
```

```
#
# Inputs:       A is a m x n matrix
#
#               b is a vector of length m
#
#               mu is the step size to use, and must satisy
#               0 < mu < 2 / norm(A)^2 to guarantee convergence
#
#               x0 is the initial starting vector (of length n) to use
#
#               nIters is the number of iterations to perform
#
# Outputs:      x is a vector of length n containing the approximate solution
#
# Description:  Performs gradient descent to solve the least squares
#
#               \min_x \|b - A x\|_2
#

    # Parse inputs
    b = vec(b)
    x0 = vec(x0)

    # Gradient descent
    x = x0
    for _ in 1:nIters
        x -= mu * (A' * (A * x - b))
    end

    return x
end
```

(b) Figure 2 shows a plot of $||\boldsymbol{x}_k - \hat{\boldsymbol{x}}||$ versus iteration k for one realization of the system with step size $\mu = 1/\sigma_1^2(\boldsymbol{A})$, for four values of noise standard deviation σ. Clearly the $\boldsymbol{x}_k$ iterates are converging to $\hat{\boldsymbol{x}} = \boldsymbol{A}^{+}\boldsymbol{b}$.

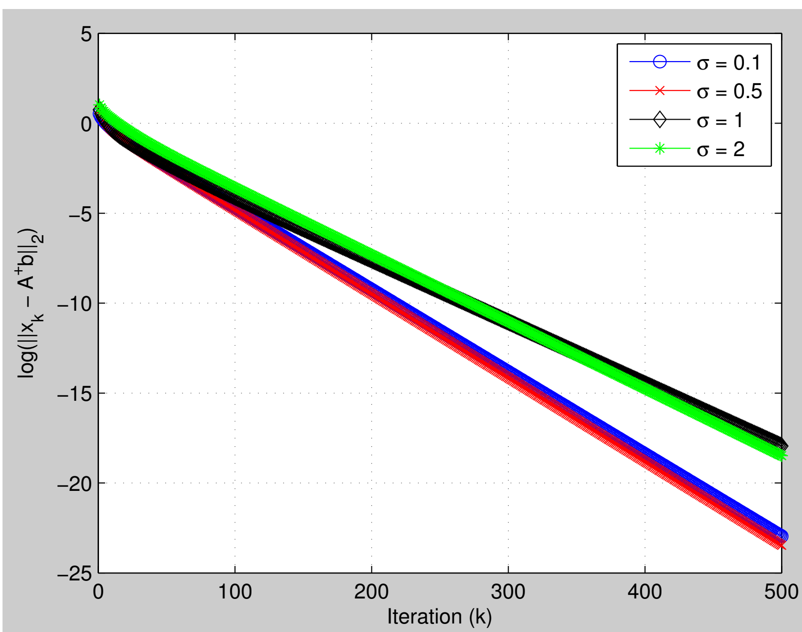

FIGURE 2. Convergence of gradient descent for least squares problems with different noise levels.

Pr. 10.

(a) A possible `Julia` implementation is

```
function lsngd(A, b, mu, x0, nIters)
#
# Syntax:       x = lsngd(A, b, mu, x0, nIters)
#
# Inputs:       A is a m x n matrix
#
#               b is a vector of length m
#
#               mu is the step size to use, and must satisy
#               0 < mu < 1 / norm(A)^2 to guarantee convergence
#
#               x0 is the initial starting vector (of length n) to use
#
#               nIters is the number of iterations to perform
#
# Outputs:      x is a vector of length n containing the approximate solution
#
# Description:  Performs Nesterov-acclerated gradient descent to solve the least
#               squares problem
#
#               \min_x \|b - Ax\|_2
#

    # Parse inputs
    b = vec(b)
    x0 = vec(x0)

    # Nesterov-accelerated gradient descent
    t = 0
    xLast = x0
    x = x0
    for _ = 1:nIters
        # t update
        tLast = t
        t = 0.5 * (1 + sqrt(1 + 4 * t^2))

        # z update
        z = x + ((tLast - 1) / t) * (x - xLast)

        # x update
        xLast = x
        x = z - mu * (A' * (A * z - b))
    end

    return x
end
```

(b) Figure 3 compares sequences of $||\boldsymbol{x}_k - \hat{\boldsymbol{x}}||$ versus k for standard gradient descent and Nestorov-accelerated gradient descent.

(c) For step sizes $\mu = \{0.25, 0.5, 0.75, 1\}/\sigma_1^2(A)$, clearly Nesterov-accelerated gradient descent converges faster than standard gradient descent, although the convergence is not necessarily monotone.

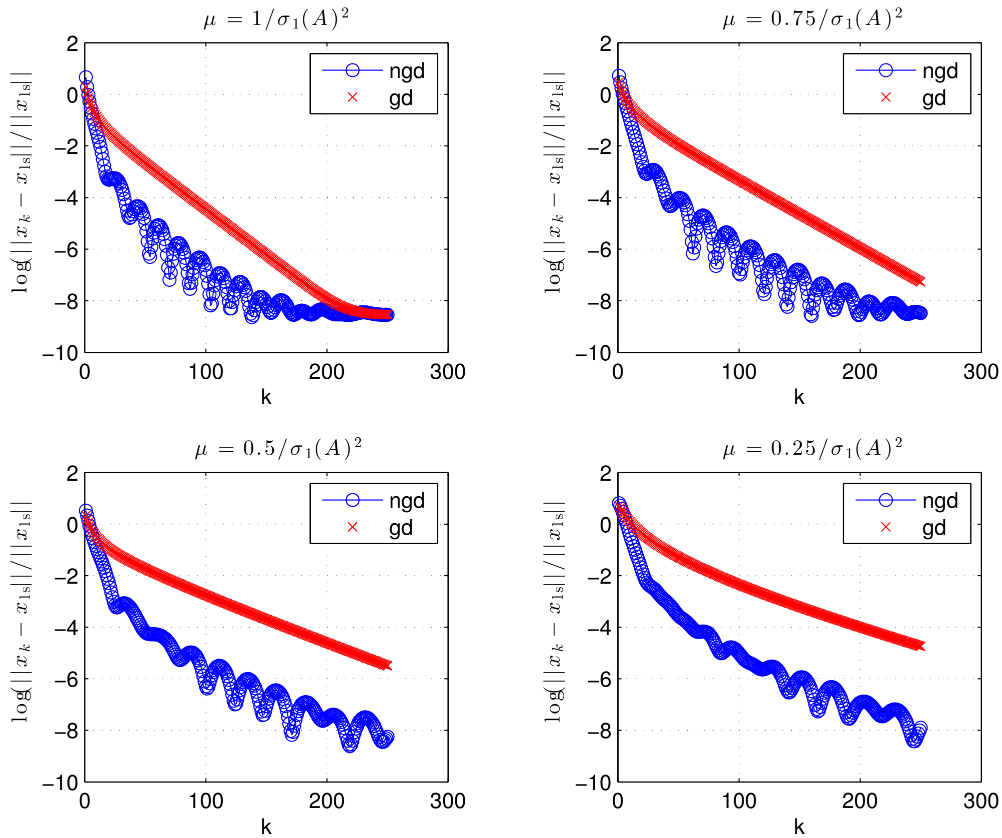

FIGURE 3. Nesterov-accelerated vs. standard gradient descent for a least squares problem, for four values of step size μ.