Pr. 1.
For $\boldsymbol{A} \in \mathbb{F}^{M \times N}$, $\boldsymbol{b} \in \mathbb{F}^M$ and $\boldsymbol{x} \in \mathbb{F}^N$, show that $(\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{x}$ and $\boldsymbol{A}^\dagger \boldsymbol{b}$ are **orthogonal** vectors, where $\boldsymbol{A}^\dagger$ denotes the **Moore-Penrose pseudo-inverse** of $\boldsymbol{A}$. Hint: use a **compact SVD**.

Pr. 2.
Consider the plane $ax + by + cz = 0$ that intersects the origin.

(a) Describe how you would use an **SVD** to find **basis vectors** for the plane.
How many bases vectors are required to express a point on the plane?

(b) Find the point on the plane that is closest to an arbitrary point $(\alpha, \beta, \gamma) \in \mathbb{R}^3$. Your expression should be general and reasonably simple.

Anytime a problem says "find" something, it is also implied that you must also show how *you found it.*

(c) Using your preceding answer (and probably `Julia` or a calculator), find the point on the plane $x + 2y + 3z = 0$ that is closest to the point $(4, 5, 6)$.

Pr. 3.
Let $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ denote two orthonormal vectors and $\boldsymbol{b}$ some fixed vector, all in $\mathbb{R}^n$.

(a) Find the optimal linear combination $\alpha \boldsymbol{q}_1 + \beta \boldsymbol{q}_2$ that is **closest** to $\boldsymbol{b}$ (in the **2-norm sense**).

(b) Let $\boldsymbol{r} = \boldsymbol{b} - \alpha \boldsymbol{q}_1 - \beta \boldsymbol{q}_2$ denote the "residual error vector." Show that $\boldsymbol{r}$ is **orthogonal** to both $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

Pr. 4.
Find all solutions of the **linear least squares problem** $\arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ when $\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Pr. 5.
A function $f : \mathcal{V} \mapsto \mathbb{R}$ is called a **convex function** on vector space $\mathcal{V}$ iff for every $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{V}$, and every $\alpha \in [0, 1]$,

$$f(\alpha \boldsymbol{x}_1 + (1 - \alpha)\boldsymbol{x}_2) \leq \alpha f(\boldsymbol{x}_1) + (1 - \alpha) f(\boldsymbol{x}_2).$$

Show that:

(a) $f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ is a convex function on $\mathbb{R}^n$ when matrix $\boldsymbol{A}$ has n columns. Hint: Use the **triangle inequality**: $\|\boldsymbol{x} + \boldsymbol{y}\|_2 \leq \|\boldsymbol{x}\|_2 + \|\boldsymbol{y}\|_2$.

(b) The largest singular value of a matrix, *i.e.*, the function $\sigma_1(\boldsymbol{X}) : \mathbb{F}^{M \times N} \mapsto \mathbb{R}$, is a convex function of the elements of the $M \times N$ matrix $\boldsymbol{X}$. Hint: Use the fact that $\sigma_1(\boldsymbol{X}) = \max_{\|\boldsymbol{u}\|_2 = 1} \|\boldsymbol{X}\boldsymbol{u}\|_2$.

Pr. 6.
(Photometric stereo continued)
(This problem statement looks long, but it is actually quite easy because it uses tools you developed previously. You need not write any new code for it.)
For $f : \mathbb{R}^2 \mapsto \mathbb{R}$, from vector calculus, we can express the **surface normal vector** of f at (x, y) as

$$\boldsymbol{n}(x,y) = \frac{1}{\sqrt{1 + \left(\frac{\partial}{\partial x} f(x,y)\right)^2 + \left(\frac{\partial}{\partial y} f(x,y)\right)^2}} \begin{bmatrix} -\frac{\partial}{\partial x} f(x,y) \\ -\frac{\partial}{\partial y} f(x,y) \\ 1 \end{bmatrix} \triangleq \begin{bmatrix} n_1(x,y) \\ n_2(x,y) \\ n_3(x,y) \end{bmatrix}, \tag{1}$$

where $\frac{\partial}{\partial x} f$ and $\frac{\partial}{\partial y} f$ denote the partial derivatives of depth $f(x, y)$ with respect to x and y, respectively.
From (1), we can compute the partial derivatives as

$$\frac{\partial f(x,y)}{\partial x} = -\frac{n_1(x,y)}{n_3(x,y)}, \quad \frac{\partial f(x,y)}{\partial y} = -\frac{n_2(x,y)}{n_3(x,y)}. \tag{2}$$

In a previous HW you constructed a matrix $\boldsymbol{A}$ satisfying $\begin{bmatrix} \texttt{dfdx} \\ \texttt{dfdy} \end{bmatrix} = \boldsymbol{A}\,\texttt{fxy}$, where `dfdx` and `dfdy` denote the vectorized approximations of $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ and `fxy` denotes the vectorized approximation of $f(x, y)$. Using (2), we can compute `dfdx` and `dfdy` from our normal vectors, and, using our $\boldsymbol{A}$ matrix, we can obtain the surface corresponding to our normal vectors by solving the following **least squares** problem:

$$\texttt{fxy} \;=\; \underset{\boldsymbol{f} \in \mathbb{R}^{mn}}{\arg\min} \left\| \begin{bmatrix} \texttt{dfdx} \\ \texttt{dfdy} \end{bmatrix} - \boldsymbol{A}\boldsymbol{f} \right\|_2^2 .$$

Download the `photometric_stereo_xy_demo` notebook from the hw04 directory on Canvas, and copy your previous `compute_normals.jl` and `first_diffs_2d_matrix.jl` solution files into the same directory. Now use `Julia` to run that Jupyter notebook. If you have installed the necessary packages mentioned in the notebook, and if you have working versions of the `compute_normals` and `first_diffs_2d_matrix` functions, then the notebook will run properly and generate a surface view of the object.
After $f(x, y)$ is estimated, one can separately generate a stereolithography file (consisting of a collection of tessellated triangles) that can be rendered on a 3D display or printed by a 3D printer. Figure 1 depicts an actual 3D-print made from a solution to this problem generated with a Cube 3 printer.
Submit a screenshot of the final surface that your Jupyter notebook produces.

Figure 1: 3D printed reconstruction of the surface you'll reconstruct in Problem 6. The 3D printing quality was intentionally set coarse so you can see how the printer constructed the shape from its level curves.

Pr. 7.

Recall that the **least squares** problem

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2,$$

has the solution $\hat{\boldsymbol{x}} = \boldsymbol{A}^+\boldsymbol{b}$ where $\boldsymbol{A}^+$ denotes the **pseudo-inverse**. When the nullspace of $\boldsymbol{A}$ is non-trivial then $\hat{\boldsymbol{x}}$ is the **minimum norm** solution. When $\boldsymbol{A}$ is large, it can be computationally prohibitive to compute an SVD of $\boldsymbol{A}$ and then its pseudo-inverse before computing $\hat{\boldsymbol{x}}$. In such settings, the **gradient descent** iteration given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mu\boldsymbol{A}'\left(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\right),$$

will minimize $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ as iteration $k \to \infty$ when $0 < \mu < 2/\sigma_1^2(\boldsymbol{A})$. Note that the iteration has a fixed point, *i.e.*, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ if

$$\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}) = \boldsymbol{0},$$

which are exactly the normal equations. So any fixed point minimizes the least squares cost function.

(a) Write a function called `lsgd` that implements the above least squares gradient descent algorithm.

In `Julia`, your file should be named `lsgd.jl` and should contain the following function:

```
"""
`x = lsgd(A, b ; mu=0, x0=zeros(size(A,2)), nIters::Int=200)`

Performs gradient descent to solve the least squares problem:
``\\argmin_x 0.5 \\| b - A x \\|_2``

In:
- `A` `m x n` matrix
- `b` vector of length `m`

Option:
- `mu` step size to use, and must satisfy ``0 < mu < 2 / \\sigma_1(A)^2``
to guarantee convergence,
where ``\\sigma_1(A)`` is the first (largest) singular value.
Ch.5 will explain a default value for `mu`
- `x0` is the initial starting vector (of length `n`) to use.
Its default value is all zeros for simplicity.
- `nIters` is the number of iterations to perform (default 200)

Out:
`x` vector of length `n` containing the approximate LS solution
"""
function lsgd(A, b ; mu::Real=0, x0=zeros(size(A,2)), nIters::Int=200)
```

Submit your solution to the autograder by emailing it as an attachment to `eecs551@autograder.eecs.umich.edu`.

The function specification above uses a powerful feature of `Julia` where functions can have optional arguments with specified default values. If you simply call `lsgd(A,b)` then `mu`, `x0` and `nIters` will all have their default values. But if you call, say, `lsgd(A, b, nIters=5, mu=7)` then it will use the specified values for `nIters` and `mu` and the default for `x0`. Note that these named optional arguments can appear in any order. This approach is very convenient for functions with multiple arguments.

(b) After your code passes, use it to generate a plot of $\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|$ as a function of k using $\mu = 1/\sigma_1^2(\boldsymbol{A})$ for $\boldsymbol{A}$ and $\boldsymbol{b}$ generated as follows

```
using Random: seed!
m = 100; n = 50; sigma = 0.1
seed!(0) # seed random number generator
A = randn(m, n); xtrue = rand(n)
b = A * xtrue + sigma * randn(m)
```

Repeat for $\sigma = 0.5, 1, 2$ and turn in one plot with all four curves on it, using a logarithmic scale for the vertical axis. Does $\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|$ decrease monotonically with k in the plots? Note that $\sigma =$ `sigma` here is a noise standard deviation unrelated to singular values.

Optional problem(s) below
(not graded, but solutions will be provided for self check; do not submit to gradescope)

Pr. 8.
Let $\boldsymbol{A}$ be a **normal** matrix of the (unnamed) type where each eigenvalue either has a different magnitude than all either eigenvalues, or has the same value as all other eigenvalues with its magnitude. In other words, having both $|\lambda_j| = |\lambda_i|$ and $\lambda_j \neq \lambda_i$ is not allowed. For example, $\boldsymbol{A}$ might have eigenvalues $(3, 4\imath, 4\imath, -5)$ but cannot have eigenvalues $(3, 4, 4\imath, -5)$.
Prove that every right singular vector of $\boldsymbol{A}$ is also an eigenvector of $\boldsymbol{A}$.
This problem finishes the story on relating **SVD** and **eigendecomposition**.

Pr. 9.
This problem concerns matrices associated with **projective transformations** used in computer graphics and computer vision. For $\boldsymbol{H} \in \mathbb{R}^{3\times 3}$, suppose $\boldsymbol{\beta}^T = \begin{bmatrix}\beta_1 & \beta_2 & \beta_3\end{bmatrix} = \begin{bmatrix}x & y & 1\end{bmatrix} \boldsymbol{H}$. (To match computer vision conventions we will do row vector times matrix but we will still denote vectors as columns vectors using the transpose symbol as needed.) As an additional assumption, suppose that $\beta_3 \neq 0$.

(a) (Easy) Show that we can write any such $\boldsymbol{\beta}$ with $\beta_3 \neq 0$ as $\begin{bmatrix}\beta_3\widetilde{x} & \beta_3\widetilde{y} & \beta_3\end{bmatrix}$, by expressing $\widetilde{x}$ and $\widetilde{y}$ in terms of β_1, β_2 and β_3.

(b) Summarizing our notation so far

$$\boldsymbol{\beta}^T = \begin{bmatrix}\beta_3\widetilde{x} & \beta_3\widetilde{y} & \beta_3\end{bmatrix} = \begin{bmatrix}x & y & 1\end{bmatrix} \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}.$$

Use the equations $\beta_1 - \widetilde{x}\beta_3 = 0$ and $\beta_2 - \widetilde{y}\beta_3 = 0$ to show that $\boldsymbol{h} = \operatorname{vec}(\boldsymbol{H}) \in \mathbb{R}^9$ satisfies an equation of the form

$$\boldsymbol{A}\boldsymbol{h} = \boldsymbol{0},$$

where $\boldsymbol{A} \in \mathbb{R}^{2\times 9}$. Explicitly specify the $\boldsymbol{A}$ matrix in this equation; it should only involve x, y, $\widetilde{x}$ and $\widetilde{y}$.

Hint: Remember that $\operatorname{vec}(\cdot)$ stacks the columns of $\boldsymbol{H}$. $\boldsymbol{A}$ has two rows. The first row of $\boldsymbol{A}$ uses x, y, $\widetilde{x}$ and the second row uses x, y, $\widetilde{y}$.

(Further hint: The first row of $\boldsymbol{A}$ uses the first and third columns of $\boldsymbol{H}$, while the second row of $\boldsymbol{A}$ uses the second and third columns of $\boldsymbol{H}$.)

If you have not done so already, write your final answer compactly by letting $\boldsymbol{\alpha}^T = \begin{bmatrix}x & y & 1\end{bmatrix}$ and $\boldsymbol{0}^T = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$. Your answer should involve only $\boldsymbol{\alpha}$, $\boldsymbol{0}$, $\widetilde{x}$ and $\widetilde{y}$.

(c) In which of the **four fundamental subspaces** associated with the matrix $\boldsymbol{A}$ does the vector $\boldsymbol{h}$ lie?

Pr. 10.
Suppose that $\boldsymbol{X} = \boldsymbol{U}_x \boldsymbol{\Sigma}_x \boldsymbol{V}_x'$ and $\boldsymbol{Y} = \boldsymbol{U}_y \boldsymbol{\Sigma}_y \boldsymbol{V}_y'$ denote **SVD** s of the matrices $\boldsymbol{X} \in \mathbb{C}^{m \times n}$ and $\boldsymbol{Y} \in \mathbb{C}^{p \times q}$.

(a) Show that

$$\boldsymbol{X} \otimes \boldsymbol{Y} = (\boldsymbol{U}_x \otimes \boldsymbol{U}_y)(\boldsymbol{\Sigma}_x \otimes \boldsymbol{\Sigma}_y)(\boldsymbol{V}_x \otimes \boldsymbol{V}_y)'$$

is an SVD of $\boldsymbol{X} \otimes \boldsymbol{Y}$ (up to a permutation of the singular values). Note that the matrix $\boldsymbol{\Sigma}_x \otimes \boldsymbol{\Sigma}_y$ is something like a diagonal matrix containing the pairwise *products* of the singular values of $\boldsymbol{X}$ and $\boldsymbol{Y}$!

Hint: First show that the formula is correct, then argue that it is an SVD by showing that $\boldsymbol{U}_x \otimes \boldsymbol{U}_y$ and $\boldsymbol{V}_x \otimes \boldsymbol{V}_y$ are unitary matrices.

Hint: You may find the following properties of **Kronecker product** useful:

- $(\boldsymbol{X} \otimes \boldsymbol{Y})' = \boldsymbol{X}' \otimes \boldsymbol{Y}'$
- $(\boldsymbol{X} \otimes \boldsymbol{Y})(\boldsymbol{W} \otimes \boldsymbol{Z}) = (\boldsymbol{X}\boldsymbol{W}) \otimes (\boldsymbol{Y}\boldsymbol{Z})$

(b) Now suppose that $\boldsymbol{A} = \boldsymbol{Q}_a \boldsymbol{\Lambda}_a \boldsymbol{Q}_a'$ and $\boldsymbol{B} = \boldsymbol{Q}_b \boldsymbol{\Lambda}_b \boldsymbol{Q}_b'$ are **eigendecompositions** of the (square Hermitian) matrices $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{C}^{m \times m}$.

Show that

$$\boldsymbol{A} \otimes \boldsymbol{B} = (\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \otimes \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)'$$

is an eigendecomposition of $\boldsymbol{A} \otimes \boldsymbol{B}$. Note that the matrix $\boldsymbol{\Lambda}_a \otimes \boldsymbol{\Lambda}_b$ is a diagonal matrix whose diagonal contains the pairwise *products* of the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$!

Hint: First show that the formula is correct, then argue that it is an eigendecomposition by showing that $\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b$ is a unitary matrix.

(c) Continuing (b), now show that

$$\boldsymbol{A} \oplus \boldsymbol{B} = (\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)(\boldsymbol{\Lambda}_a \oplus \boldsymbol{\Lambda}_b)(\boldsymbol{Q}_a \otimes \boldsymbol{Q}_b)'$$

is an eigendecomposition of the **Kronecker sum** $\boldsymbol{A} \oplus \boldsymbol{B} \triangleq (\boldsymbol{A} \otimes \boldsymbol{I}) + (\boldsymbol{I} \otimes \boldsymbol{B})$, where you must determine the size(s) of the "$\boldsymbol{I}$" in that expression. Note that the matrix $\boldsymbol{\Lambda}_a \oplus \boldsymbol{\Lambda}_b$ is a diagonal matrix whose diagonal contains the pairwise *sums* of the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$.

Hint: Start by writing $\boldsymbol{I}_n = \boldsymbol{Q}_a \boldsymbol{I}_n \boldsymbol{Q}_a'$ and $\boldsymbol{I}_m = \boldsymbol{Q}_b \boldsymbol{I}_m \boldsymbol{Q}_b'$ in the definition of $\boldsymbol{A} \oplus \boldsymbol{B}$, and then apply (b).

Pr. 11.
Let $\boldsymbol{A}$ be an $m \times n$ matrix with SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'$. Let $\boldsymbol{u}_i$ denote the ith column of $\boldsymbol{U}$, and $\boldsymbol{v}_i$ denote the ith column of $\boldsymbol{V}$. Let σ_i is the ith diagonal element of $\boldsymbol{\Sigma}$, and let $\boldsymbol{A}$ have rank r. Let $\boldsymbol{I}_d$ denote the $d \times d$ identity matrix.
Simplify the following **pseudo-inverse** expressions as much as possible.

(a) (1) $(\boldsymbol{u}_i)^+ =$
(2) $(\boldsymbol{u}_i \boldsymbol{u}_i')^+ =$
(3) $(\boldsymbol{u}_i \boldsymbol{v}_i')^+ =$
(4) $(\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i')^+ =$

(b) (1) $\boldsymbol{U}^+ =$
(2) $\boldsymbol{V}^+ =$
(3) $\boldsymbol{U}\boldsymbol{U}^+ =$
(4) $\boldsymbol{V}\boldsymbol{V}^+ =$
(5) $\boldsymbol{\Sigma}^+ =$
(6) $(\boldsymbol{U}\boldsymbol{\Sigma})^+ =$
(7) $(\boldsymbol{\Sigma}\boldsymbol{V}')^+ =$
(8) $\boldsymbol{A}^+ = (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}')^+ =$
(9) $(\boldsymbol{U}\boldsymbol{V}')^+ =$

(c) (1) For $1 \leq k \leq m$, $\boldsymbol{U}[:, 1:k]^+ =$

(2) For $1 \le k \le n$, $\boldsymbol{V}[:, 1:k]^+ =$

(3) For $1 \le k \le \min\{m, n\}$. $\boldsymbol{\Sigma}[1:k, 1:k]^+ =$

(4) For $1 \le k \le m$, $\boldsymbol{U}[:, 1:k]\boldsymbol{U}[:, 1:k]^+ =$

(5) For $1 \le k \le n$, $\boldsymbol{V}[:, 1:k]\boldsymbol{V}[:, 1:k]^+ =$

(6) For $1 \le k \le m$, $\boldsymbol{U}[:, 1:k]^+\boldsymbol{U}[:, 1:k] =$

(7) For $1 \le k \le n$, $\boldsymbol{V}[:, 1:k]^+\boldsymbol{V}[:, 1:k] =$

(d) (1) For $1 \le k \le m$, $\boldsymbol{I}_m - \boldsymbol{U}[:, 1:k]\boldsymbol{U}[:, 1:k]^+ =$

(2) For $1 \le k \le n$, $\boldsymbol{I}_n - \boldsymbol{V}[:, 1:k]\boldsymbol{V}[:, 1:k]^+ =$

(3) For $1 \le k \le m$, $\boldsymbol{I}_k - \boldsymbol{U}[:, 1:k]^+\boldsymbol{U}[:, 1:k] =$

(4) For $1 \le k \le n$, $\boldsymbol{I}_k - \boldsymbol{V}[:, 1:k]^+\boldsymbol{V}[:, 1:k] =$

(e) (1) For $1 \le k \le r$, $(\boldsymbol{U}[:, 1:k]\boldsymbol{\Sigma}[1:k, 1:k]\boldsymbol{V}[:, 1:k]')^+ =$

(2) $(\boldsymbol{A}')^+ =$

(3) $(\boldsymbol{A}^+)' =$

(4) For $\alpha \neq 0$, $(\alpha\boldsymbol{A})^+ =$

(5) $(\boldsymbol{A}\boldsymbol{A}^+)' =$

(6) $(\boldsymbol{A}^+\boldsymbol{A})' =$

(f) (1) $\boldsymbol{A}\boldsymbol{A}^+ =$

(2) $\boldsymbol{A}^+\boldsymbol{A} =$

(3) $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} =$

(4) $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ =$

(g) (1) $(\boldsymbol{A}'\boldsymbol{A})^+\boldsymbol{A}' =$

(2) $\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{A}')^+ =$

(3) If $r = n$, then $\boldsymbol{A}^+\boldsymbol{A} =$

(4) If $r = m$, then $\boldsymbol{A}\boldsymbol{A}^+ =$

(5) If $r = n$, then $(\boldsymbol{A}'\boldsymbol{A})^{-1}\boldsymbol{A}' =$

(6) If $r = m$, then $\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{A}')^{-1} =$

Pr. 12.

(a) Show that if the functions $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$ are all convex functions of $\boldsymbol{x}$, then the following are also convex:

(1) $\sum_{i=1}^n f_i(\boldsymbol{x})$.

(2) $\sum_{i=1}^n w_i f_i(\boldsymbol{x})$ for scalars $w_i \ge 0$.

(3) Provide an example wherein w_i being negative breaks the convexity.

(4) $\max\{f_1, f_2, \ldots, f_n\}$.

(b) Which of the following loss functions are convex functions of $\boldsymbol{x}$? Assume that $\boldsymbol{A}$ and $\boldsymbol{D}$ are matrices and $\boldsymbol{b}$ is a vector with compatible dimensions. Justify your answer: using the results from a previous part may help.

(1) $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \|\boldsymbol{D}\boldsymbol{x}\|_2^2$.

(2) $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \|\boldsymbol{D}\boldsymbol{x}\|_1$.

(3) $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 - \|\boldsymbol{D}\boldsymbol{x}\|_2^2$.

(4) $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 - \|\boldsymbol{D}\boldsymbol{x}\|_1$.
