Pr. 1. (sol/hs121b)

(a) $\boldsymbol{F}_M \boldsymbol{X}$ multiplies each column of $\boldsymbol{X}$ by $\boldsymbol{F}_M$, so it computes the M-point 1D DFT of each column of $\boldsymbol{X}$. Therefore, $(\boldsymbol{F}_N \boldsymbol{X}^T)^T = \boldsymbol{X}\boldsymbol{F}_N^T$ computes the N-point 1D DFT of each row of $\boldsymbol{X}$. Combining these operations yields $\boldsymbol{S}_{\boldsymbol{X}} = \boldsymbol{F}_M \boldsymbol{X} \boldsymbol{F}_N^T$. Note that we use transpose, not Hermitian transpose (even for complex-valued data). Applying the vec trick yields

$$\text{vec}(\boldsymbol{S}_{\boldsymbol{X}}) = (\boldsymbol{F}_N \otimes \boldsymbol{F}_M)\text{vec}(\boldsymbol{X}).$$

(b) Approach 1. Simply substitute in the $\boldsymbol{S}_{\boldsymbol{X}}$ expression and confirm that it results in the original $\boldsymbol{X}$:

$$\frac{1}{MN}\boldsymbol{F}_M' \boldsymbol{S}_{\boldsymbol{X}} \overline{\boldsymbol{F}}_N = \frac{1}{MN}\boldsymbol{F}_M' \left(\boldsymbol{F}_M \boldsymbol{X} \boldsymbol{F}_N^T\right) \overline{\boldsymbol{F}}_N = \underbrace{\left(\frac{1}{M}\boldsymbol{F}_M' \boldsymbol{F}_M\right)}_{\boldsymbol{I}_M} \boldsymbol{X} \underbrace{\overline{\left(\frac{1}{N}\boldsymbol{F}_N' \boldsymbol{F}_N\right)}}_{\boldsymbol{I}_N} = \boldsymbol{X}.$$

Because this expression returns the original $\boldsymbol{X}$, for any $\boldsymbol{X}$, it must correspond to the inverse transform.

Approach 2. Observe that

$$\frac{1}{MN}\boldsymbol{F}_M' \boldsymbol{S}_{\boldsymbol{X}} \overline{\boldsymbol{F}}_N = \underbrace{\left(\frac{1}{M}\boldsymbol{F}_M' \boldsymbol{F}_M\right)}_{=\boldsymbol{I}_M} \boldsymbol{X} \underbrace{\left(\frac{1}{N}\boldsymbol{F}_N^T \overline{\boldsymbol{F}}_N\right)}_{=:\boldsymbol{Y}},$$

where $\boldsymbol{Y} = \frac{1}{N}\boldsymbol{F}_N^T \overline{\boldsymbol{F}}_N = \overline{\left(\frac{1}{N}\overline{\boldsymbol{F}}_N^T \boldsymbol{F}_N\right)} = \overline{\left(\frac{1}{N}\boldsymbol{F}_N' \boldsymbol{F}_N\right)} = \overline{\boldsymbol{I}}_N = \boldsymbol{I}_N$. Applying the vec trick:

$$\text{vec}(\boldsymbol{X}) = \frac{1}{MN}(\boldsymbol{F}_N \otimes \boldsymbol{F}_M)'\text{vec}(\boldsymbol{S}_{\boldsymbol{X}}) = \frac{1}{MN}(\boldsymbol{F}_N' \otimes \boldsymbol{F}_M')\text{vec}(\boldsymbol{S}_{\boldsymbol{X}}).$$

Approach 3. Alternatively:

$$\text{vec}(\boldsymbol{X}) = (\boldsymbol{F}_N \otimes \boldsymbol{F}_M)^{-1}\text{vec}(\boldsymbol{S}_{\boldsymbol{X}}) = (\boldsymbol{F}_N^{-1} \otimes \boldsymbol{F}_M^{-1})\text{vec}(\boldsymbol{S}_{\boldsymbol{X}}) = \frac{1}{MN}(\boldsymbol{F}_N' \otimes \boldsymbol{F}_M')\text{vec}(\boldsymbol{S}_{\boldsymbol{X}}).$$

Pr. 2. (sol/hs072)

Here

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

because $\boldsymbol{A}$ is rank-1 and can be written as an outerproduct $\boldsymbol{A} = \boldsymbol{z}\boldsymbol{z}'$ where $\boldsymbol{z} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and we found the eigendecomposition of such rank-1 matrices previously. Recall that since $\boldsymbol{A}$ is symmetric, positive-semidefinite, its eigendecomposition is the same as a singular value decomposition.

Consider the minimum norm solution given by

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

Here $\text{rank}(\boldsymbol{A}) = 1 < 2$, so the set of all solutions is $\hat{\boldsymbol{x}} + \mathcal{N}(\boldsymbol{A}) = \hat{\boldsymbol{x}} + \text{span}(\boldsymbol{V}_0)$. Moreover

$$\mathcal{N}(\boldsymbol{A}) = \text{span}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right).$$

(a) Here $\mathbb{F} = \mathbb{R}$ so vectors of the following form yield the same (minimum) squared error:

$$\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} + \alpha \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \alpha \in \mathbb{R}.$$

(b) Here $\mathbb{F} = \mathbb{C}$ so we simply take $\alpha \in \mathbb{C}$ above.

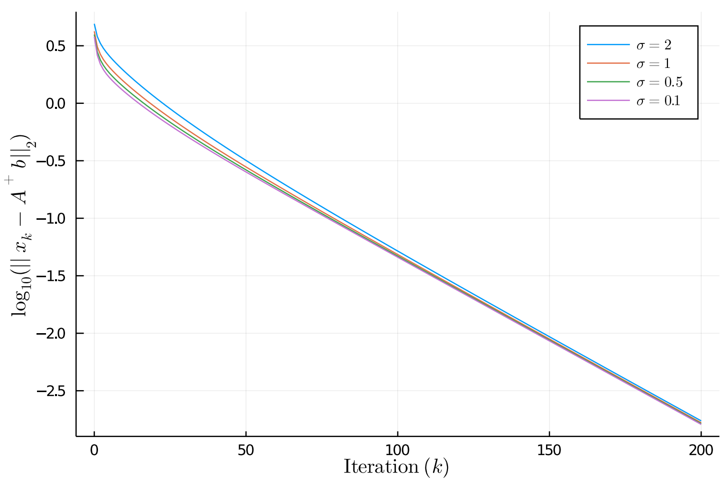
Pr. 3. (sol/hs070)
Here we have that $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ so $\boldsymbol{W}\boldsymbol{r} = \boldsymbol{W}\boldsymbol{b} - \boldsymbol{W}\boldsymbol{A}\boldsymbol{x}$. Let $\tilde{\boldsymbol{r}} = \boldsymbol{W}\boldsymbol{r}$, $\tilde{\boldsymbol{A}} = \boldsymbol{W}\boldsymbol{A}$ and $\tilde{\boldsymbol{b}} = \boldsymbol{W}\boldsymbol{b}$. Then minimizing $\|\boldsymbol{W}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{W}\boldsymbol{b}\|_2$ is equivalent to minimizing $\|\tilde{\boldsymbol{A}}\boldsymbol{x} - \tilde{\boldsymbol{b}}\|_2$. Thus $\hat{\boldsymbol{x}} = \tilde{\boldsymbol{A}}^+\tilde{\boldsymbol{b}} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^+\tilde{\boldsymbol{U}}'\boldsymbol{W}\boldsymbol{b}$, where $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}'$ is a full SVD of the weighted matrix $\tilde{\boldsymbol{A}} = \boldsymbol{W}\boldsymbol{A}$. (The SVD of $\boldsymbol{A}$ itself is *not* useful here.) Using the compact SVD of $\tilde{\boldsymbol{A}}$, our final expression is $\boxed{\hat{\boldsymbol{x}} = \tilde{\boldsymbol{V}}_r\tilde{\boldsymbol{\Sigma}}_r^{-1}\tilde{\boldsymbol{U}}_r'\boldsymbol{W}\boldsymbol{b},}$ where r is the rank of $\boldsymbol{W}\boldsymbol{A}$, and we know $r \geq 1$ because $\boldsymbol{W}\boldsymbol{A} \neq \boldsymbol{0}$.

Pr. 4. (sol/hs076)

(a) A possible `Julia` implementation is

```
"""
    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 × 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)

    if (mu == 0) # use the following default value:
        mu = 1. / (maximum(sum(abs.(A),dims=1)) * maximum(sum(abs.(A),dims=2)))
    end

    # 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) The figure below shows $\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|$ 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 $\hat{\boldsymbol{x}} = \boldsymbol{A}^+ \boldsymbol{b}$.



Pr. 5. (sol/hs082)

(a) Note that $\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\delta^2\|\boldsymbol{x}\|_2^2 = \frac{1}{2}\| \underbrace{\begin{bmatrix}\boldsymbol{A}\\ \delta\boldsymbol{I}\end{bmatrix}}_{\triangleq\widetilde{\boldsymbol{A}}} \boldsymbol{x} - \underbrace{\begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{0}\end{bmatrix}}_{\triangleq\widetilde{\boldsymbol{b}}} \|_2^2$. Thus we can use the standard least-squares result to get the solution

$$\hat{\boldsymbol{x}} = \widetilde{\boldsymbol{A}}^+\widetilde{\boldsymbol{b}} = (\boldsymbol{A}'\boldsymbol{A}+\delta^2\boldsymbol{I})^{-1}\boldsymbol{A}'\boldsymbol{b} = \boldsymbol{V}(\boldsymbol{\Sigma}'\boldsymbol{\Sigma}+\delta^2\boldsymbol{I})^{-1}\boldsymbol{\Sigma}'\boldsymbol{U}'\boldsymbol{b} = \boldsymbol{V}_r(\boldsymbol{\Sigma}_r^2+\delta^2\boldsymbol{I})^{-1}\boldsymbol{\Sigma}_r\boldsymbol{U}_r'\boldsymbol{b},$$

where we use the fact that $(\boldsymbol{A}'\boldsymbol{A}+\delta\boldsymbol{I})$ is always invertible whenever $\delta > 0$.
Grader: accept either full SVD or compact SVD form above.

(b) As $\delta\to\infty$ we have $\hat{\boldsymbol{x}}(\delta)\to\boldsymbol{0}$ which makes sense because the $\delta^2\|\boldsymbol{x}\|_2^2$ term dominates the cost function.

(c) The iteration $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mu\widetilde{\boldsymbol{A}}'(\widetilde{\boldsymbol{A}}\boldsymbol{x}_k-\widetilde{\boldsymbol{b}}) = \boldsymbol{x}_k - \mu\left(\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b})+\delta^2\boldsymbol{x}_k\right)$, will converge to $\hat{\boldsymbol{x}}$ whenever $0<\mu<2/\sigma_1(\widetilde{\boldsymbol{A}})^2$.

(d) Because $\widetilde{\boldsymbol{A}}'\widetilde{\boldsymbol{A}} = \boldsymbol{A}'\boldsymbol{A}+\delta^2\boldsymbol{I} = \boldsymbol{V}_r(\boldsymbol{\Sigma}_r^2+\delta^2\boldsymbol{I})\boldsymbol{V}_r'$, it follows that

$$\sigma_1^2(\widetilde{\boldsymbol{A}}) = \sigma_1^2(\boldsymbol{A})+\delta^2.$$

We use this equality with a previous HW problem to determine the allowable step sizes: $0<\mu<\frac{2}{\sigma_1^2(\boldsymbol{A})+\delta^2}$.

Pr. 6. (sol/hs071)
Here $\boldsymbol{A} = \begin{bmatrix}1&0\\0&0\end{bmatrix}$ and $\boldsymbol{A}^+ = \begin{bmatrix}1&0\\0&0\end{bmatrix}$ and the optimal least-squares estimate is: $\boldsymbol{x}^* = \boldsymbol{A}^+\boldsymbol{b} = \begin{bmatrix}1&0\\0&0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}$.

(a) Consider $\boldsymbol{A}_1 = \begin{bmatrix}1&\delta\\0&0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}1&\delta\end{bmatrix}$ so that by inspection an SVD of $\boldsymbol{A}_1$ is $\boldsymbol{A}_1 = \sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^T$ where $\sigma_1 = \sqrt{1+\delta^2}$, $\boldsymbol{u}_1 = \begin{bmatrix}1\\0\end{bmatrix}$ and $\boldsymbol{v}_1 = \begin{bmatrix}1/\sqrt{1+\delta^2}\\ \delta/\sqrt{1+\delta^2}\end{bmatrix}$. We do not need to compute the other singular vectors because the optimal (minimum-norm) least-squares solution is simply: $\boldsymbol{z}^* = \boldsymbol{A}_1^+\boldsymbol{b} = \frac{1}{\sigma_1}\boldsymbol{v}_1\boldsymbol{u}_1^T\boldsymbol{b} = \frac{1}{1+\delta^2}\begin{bmatrix}1\\ \delta\end{bmatrix}$,
for which $\boldsymbol{z}^*\to\begin{bmatrix}1\\0\end{bmatrix}$ as $\delta\to 0$, so $\|\boldsymbol{z}^*-\boldsymbol{x}^*\|_2\to 0$.

(b) Consider now the case where $\boldsymbol{A}_2 = \begin{bmatrix}1&0\\0&\delta\end{bmatrix}$, which has $\operatorname{rank}(\boldsymbol{A}) = 2$ for $\delta>0$. Here the optimal least-squares solution is: $\boldsymbol{z}^* = \boldsymbol{A}_2^+\boldsymbol{b} = \begin{bmatrix}1\\ 1/\delta\end{bmatrix}$. In this case, $\|\boldsymbol{x}^*-\boldsymbol{z}^*\|_2\to\infty$ as $\delta\to 0$.

This exercise illustrates that the manner in which error appears can impact a solution negligibly or dramatically!

Pr. 7. (sol/hs051)

(a) Every point $\begin{bmatrix}x\\y\\z\end{bmatrix}$ on the plane satisfies the equation $\begin{bmatrix}a&b&c\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix} = 0$. Thus every point on the plane must satisfy $\begin{bmatrix}x\\y\\z\end{bmatrix}\in\mathcal{N}(\begin{bmatrix}a&b&c\end{bmatrix})$. Clearly: $\begin{bmatrix}a&b&c\end{bmatrix} = \underbrace{1}_{=\boldsymbol{u}_1}\underbrace{\sqrt{a^2+b^2+c^2}}_{=\sigma_1}\underbrace{\frac{\begin{bmatrix}a&b&c\end{bmatrix}}{\sqrt{a^2+b^2+c^2}}}_{=\boldsymbol{v}_1^T}$. Performing a full SVD of the 1×3 matrix $\boldsymbol{A} = \begin{bmatrix}a&b&c\end{bmatrix}$ will provide a 3×3 matrix $\boldsymbol{V}$ where $\mathcal{N}(\begin{bmatrix}a&b&c\end{bmatrix}) = \operatorname{span}(\{\boldsymbol{v}_2,\boldsymbol{v}_3\}) = \mathcal{R}(\boldsymbol{V}_0)$ because $\boldsymbol{A}$ is a rank-1 matrix. Thus $\{\boldsymbol{v}_2,\boldsymbol{v}_3\}$ form 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

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

(c) Code: `v1 = [1, 2, 3] / sqrt(14);` `(I - v1*v1') * [4, 5, 6]`
yields $(1.714, 0.429, -0.857)$

Pr. 8. (sol/hs073)
For any given pixel at location (x, y), the LS solution is

$$\hat{\rho}(x, y) = (\boldsymbol{L}^T)^+ \boldsymbol{g}(x, y).$$

Transposing both sides yields

$$\hat{\rho}(x, y)^T = \boldsymbol{g}(x, y)^T \boldsymbol{L}^+.$$

This form is convenient for applying a matrix version:

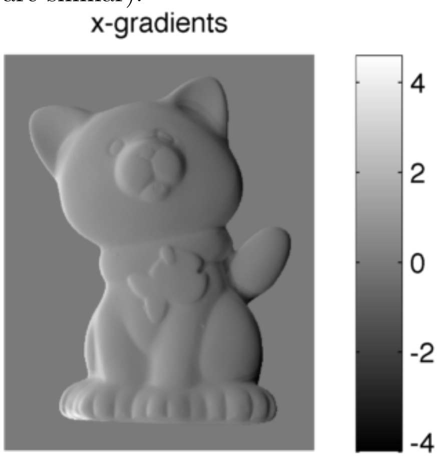
$$\hat{\rho} = \begin{bmatrix}\hat{\rho}(1,1)^T\\ \vdots\\ \hat{\rho}(m,n)^T\end{bmatrix} = \begin{bmatrix}\boldsymbol{g}(1,1)^T\\ \vdots\\ \boldsymbol{g}(m,n)^T\end{bmatrix} \boldsymbol{L}^+.$$

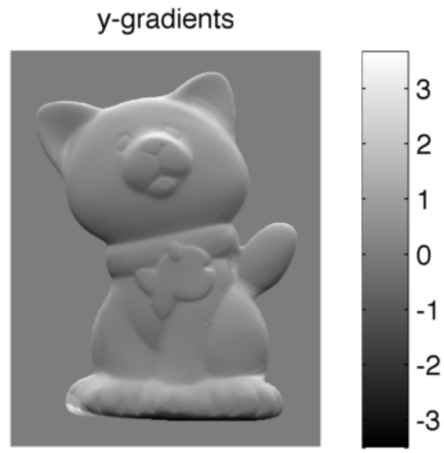
A possible `Julia` implementation is

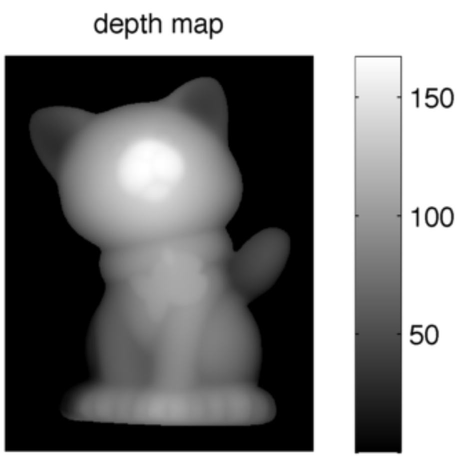
```
using LinearAlgebra: normalize, pinv

"""
    N = compute_normals(data, L)

In:
- `data` `m × n × d` matrix whose `d` slices contain `m × n` images
of a common scene under different lighting conditions

- `L` `3 × d` matrix whose columns are the lighting direction vectors
for the images in data, with `d ≥ 3`

Out:
- `N` `m × n × 3` matrix containing the unit-norm surface normal vectors
for each pixel in the scene
"""
function compute_normals(data, L)

    m, n, d = size(data) # Parse inputs
    L = mapslices(normalize, L, dims=1) # Normalize lighting direction vectors

    # Solve least-squares problem
    # Here, using pinv() is efficient because L is a small matrix
    # and we apply pinv(L) to many (mn) pixels.
    data = reshape(data, m * n, d) # now [mn × d]
    N = data * pinv(L)
    N = reshape(N, m, n, 3) # now [m × n × 3]

    # alternative one-line "Julia way" that avoids reshape:
    # N = mapslices((v) -> pinv(L)'*v, data, dims=3) # apply pinv to each pixel
    # However, it ran slower than the above, per timing test below.

    N = mapslices(normalize, N, dims=3) # normalize normal vectors
    return N
end


#= timing test
    using Random; Random.seed!(0)
    using BenchmarkTools
    m = 100; n = 200; d = 15
```

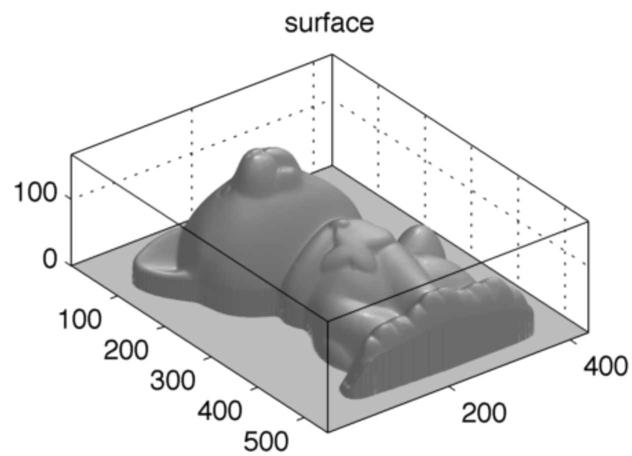
```
    data = randn(m,n,d)
    L = randn(3,d)
    pL = pinv(L)
    pLt = pL'
    way0 = (data, L) -> compute_normals(data, L)
    way1 = (data, pL) -> reshape(reshape(data, m * n, d) * pL, m, n, 3)
    way2 = (data, pLt) -> mapslices((v) -> pLt*v, data, dims=3)
    @assert isapprox(way1(data,pL), way2(data,pLt)) # check equivalence
    @btime way0($data, $L)
    @btime way1($data, $pL) # much faster than way2
    @btime way2($data, $pLt)
=#
```

Pr. 9. (sol/hs075)
Here is the photometric stereo reconstruction of depth and surface (from an older version of `Julia`, but the results are similar).





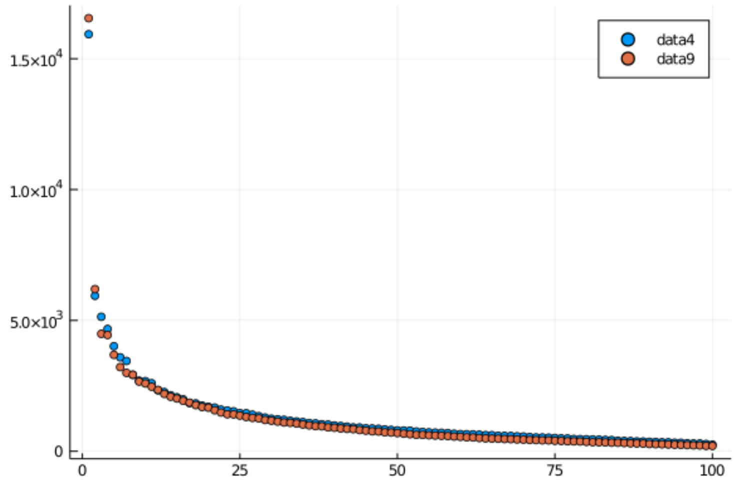




Grader: the student answers should be in color. Report any grayscale solutions like those above to the professor.

Pr. 10. (sol/hsjt2)

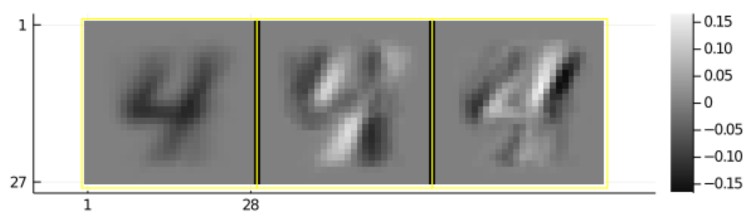
Part 1 Scree plot:

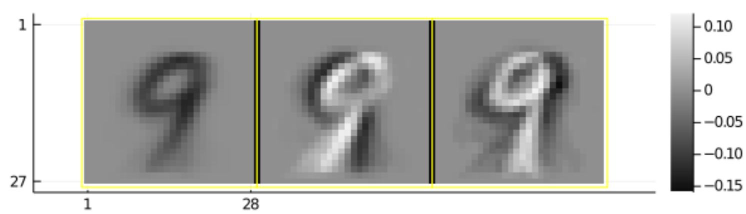


Where do you think the knee is in the scree plot? (1-2 sentences. You will get full credit for many different numbers here as long as you show you understand what the knee is.)

The knee is where the scree plot levels out. Here, you could easily claim there is a knee after 4, 5, 7, or around 20 singular values. Even though there is a large drop after the first singular value, note that one would not be a good answer, as there are still relatively large differences between the second and third singular values. Graders: still give full credit to students who say 1 as long as their explanation shows understanding of what the knee is.

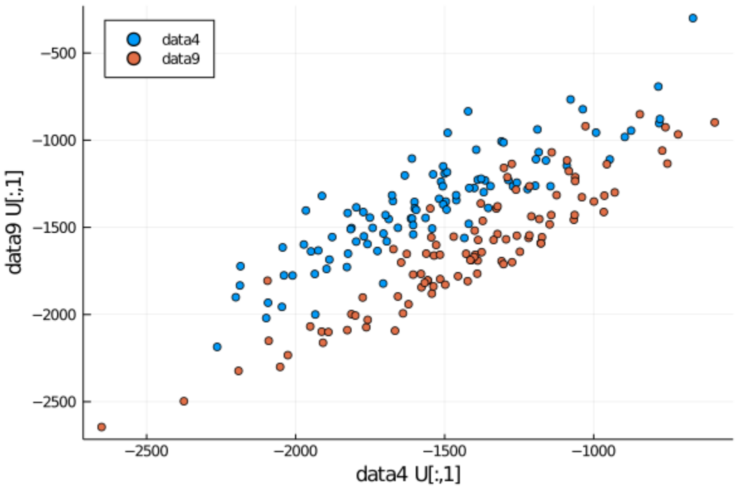
Part 2 Images of the three left singular vectors for both classes:





Do the left singular vectors look sensible? Why? (1-3 sentences) The vectors look sensible as a basis because they resemble the digits. You can imagine using a linear combination of the basis vectors to approximate many different 4s and 9s.

Part 3 Plot of separation based on first left singular vector:



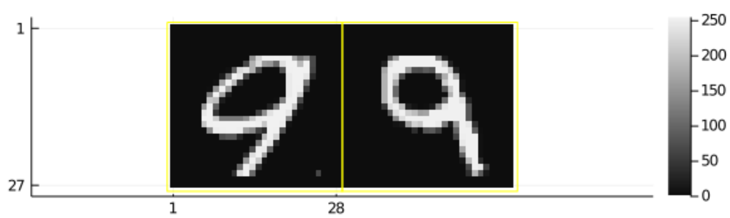
Explanation of what the regress function returns (1-3 sentences): The regress function returns the linear coefficient, $\boldsymbol{x}$, that minimizes $\arg\min_{x \in \mathbb{R}} \|\boldsymbol{y} - \boldsymbol{U}x\|_2$. Note that x is a scalar here because we are only using a single basis vector in this part.

Part 4 Classification accuracy:

Classification accuracy for the 4 digits: 77.2%
Classification accuracy for the 9 digits: 93.1%

Comment on how good/bad the classifier is:
The classification accuracy is worse than our previous classifiers.
However, classifying 4s and 9s is harder than classifying 0s and 1s, so we would need to re-run the classifiers on the same digits to make a fair comparison. We could likely make a better subspace classifier by picking the number of subspace basis vectors more carefully.

Part 5 Pictures of the first two 4's and 9's that are misclassified:



Non-graded problem(s) below

Pr. 11. (sol/hsj3u)

(a) Proof. If $\mathcal{S} \subset \mathcal{T}$, then $\mathcal{S} \cup \mathcal{T} = \mathcal{T}$ which is a subspace.
Similarly by symmetry if $\mathcal{T}$ is a subset of $\mathcal{S}$.

(b) Now we show the "only if" part.

Seeking a contradiction, suppose that $\mathcal{S}$ is not a subset of $\mathcal{T}$ and $\mathcal{T}$ is not a subset of $\mathcal{S}$. Then ther $\exists \boldsymbol{s} \in \mathcal{S}$ for which $\boldsymbol{s} \notin \mathcal{T}$ and likewise $\exists \boldsymbol{t} \in \mathcal{T}$ for which $\boldsymbol{t} \notin \mathcal{S}$. Now suppose that the union $\mathcal{U} = \mathcal{S} \cup \mathcal{T}$ is a subspace. Because $\boldsymbol{s}, \boldsymbol{t} \in \mathcal{U}$, we have that the sum vector also is in the union: $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{t} \in \mathcal{U}$. Thus either $\boldsymbol{x} \in \mathcal{S}$ or $\boldsymbol{x} \in \mathcal{T}$ (or both). If $\boldsymbol{x} \in \mathcal{S}$ then it follows that $\boldsymbol{t} = \boldsymbol{x} - \boldsymbol{s} \in \mathcal{S}$, because $\mathcal{S}$ is a subspace, contradicting the choice of $\boldsymbol{t} \notin \mathcal{S}$. Similarly if $\boldsymbol{x} \in \mathcal{T}$.

Pr. 12. (sol/hsj41)

(a) From a previous problem, the general solution for $\delta > 0$ is $\hat{\boldsymbol{x}} = (\boldsymbol{\Phi}'\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1} \boldsymbol{\Phi}' \boldsymbol{b}$.

Using the push-through identity, $\hat{\boldsymbol{x}} = \boldsymbol{\Phi}'(\boldsymbol{\Phi}\boldsymbol{\Phi}' + \delta \boldsymbol{I})^{-1} \boldsymbol{b}$.

Here, because $\boldsymbol{\Phi}$ is a Parseval tight frame and $\boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{z}$, we have $\boldsymbol{\Phi}\boldsymbol{\Phi}' = \boldsymbol{I}$ so $\hat{\boldsymbol{x}} = \boldsymbol{\Phi}'(\boldsymbol{I} + \delta \boldsymbol{I})^{-1} \boldsymbol{\Phi} \boldsymbol{z} = \frac{1}{1+\delta} \boldsymbol{\Phi}' \boldsymbol{\Phi} \boldsymbol{z}$.

In general, no further simplification seems possible.

When $\delta = 0$, the problem solution is not unique in general, so this is a MNLS solution.

(b) The solution is unique when $\delta > 0$ or when $\boldsymbol{\Phi}$ is also square (and thus unitary).

(c) If $\delta = 0$ then $\hat{\boldsymbol{x}} = \boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{z}$, and $\boldsymbol{\Phi}\hat{\boldsymbol{x}} = \boldsymbol{\Phi}\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{z}$, so both $\hat{\boldsymbol{x}}$ and $\boldsymbol{z}$ here provide equivalent signal representations

(d) The singular values of $\boldsymbol{\Phi}$ are the square roots of the eigenvalues of $\boldsymbol{\Phi}\boldsymbol{\Phi}' = \boldsymbol{I}$ which are all 1, so $\kappa(\boldsymbol{\Phi}) = 1$, which is the best possible condition number.

Nevertheless, typically $\boldsymbol{\Phi}$ is a wide matrix so the LS problem is under-determined and hence some form of regularization may be beneficial.

Pr. 13. (sol/hsj46)

(a) When $\boldsymbol{\Phi}$ is a tight frame with frame bound α, the tight frame condition implies that all the singular values equal $\sqrt{\alpha}$. Thus any SVD of $\boldsymbol{\Phi}$ looks like $\boldsymbol{\Phi} = \boldsymbol{U} \begin{bmatrix} \sqrt{\alpha}\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$, for which $\alpha \boldsymbol{I} - \boldsymbol{\Phi}'\boldsymbol{\Phi} = \alpha \boldsymbol{I} - \boldsymbol{V} \begin{bmatrix} \sqrt{\alpha}\boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{U}'\boldsymbol{U} \begin{bmatrix} \sqrt{\alpha}\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}' = \alpha \boldsymbol{V}\boldsymbol{V}' - \boldsymbol{V} \begin{bmatrix} \alpha \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}' = \boldsymbol{V} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \alpha \boldsymbol{I} \end{bmatrix} \boldsymbol{V}' \succeq \boldsymbol{0}$. Thus $\boldsymbol{\Phi}'\boldsymbol{\Phi} \preceq \alpha \boldsymbol{I}$.

(b) $\alpha \boldsymbol{I} - \boldsymbol{\Phi}\boldsymbol{\Phi}' = \alpha \boldsymbol{I} - \alpha \boldsymbol{I} = \boldsymbol{0}$

Pr. 14. (sol/hsj45)
Repeating for reference the Venn diagram for projections:

$\boldsymbol{A}_1$
Square matrices

$\boldsymbol{A}_2$
Idempotent
Projection matrix
$\boldsymbol{P}^2 = \boldsymbol{P}$
diagonalizable
eigenvalues $\{0, 1\}$

$\boldsymbol{A}_3$
Orthogonal
projection matrix
$\boldsymbol{P} = \boldsymbol{P}'$
normal

$\boldsymbol{A}_4$
Nonzero orthogonal
projection matrix
$\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}'$
where $\boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I}$

$\boldsymbol{A}_5$
Diagonal
with $\{0, 1\}$
elements

$\boldsymbol{I}$

$\boldsymbol{A}_1 = [2]$ is square but not idempotent

$\boldsymbol{A}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ is idempotent but not an orthogonal projection matrix

$\boldsymbol{A}_3 = \begin{bmatrix} 0 \end{bmatrix}$ an orthogonal projection matrix but not nonzero

$\boldsymbol{A}_4 = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) \end{bmatrix}$ is a nonzero orthogonal projection matrix but not diagonal for any ϕ where $\cos(\phi)\sin(\phi) \neq 0$

$\boldsymbol{A}_5 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ is a diagonal nonzero orthogonal projection matrix but not $\boldsymbol{I}$

Pr. 15. (sol/hs133)
Throughout, let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' = \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' \in \mathbb{C}^{m\times n}$ be an SVD of $\boldsymbol{A}$ and let r denote the rank of $\boldsymbol{A}$.
Let $\boldsymbol{U}_r = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_r]$ and $\boldsymbol{U}_0 = [\boldsymbol{u}_{r+1}, \boldsymbol{u}_{r+2}, \dots, \boldsymbol{u}_m]$, so that $\boldsymbol{U} = [\boldsymbol{U}_r | \boldsymbol{U}_0]$, and similarly for $\boldsymbol{V}_r$ and $\boldsymbol{V}_0$ so that $\boldsymbol{V} = [\boldsymbol{V}_r | \boldsymbol{V}_0]$.

(a) (1) $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{U}_r \boldsymbol{U}_r' = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{u}_i'$
(2) $\boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{A})} = \boldsymbol{I}_m - \boldsymbol{U}_r \boldsymbol{U}_r' = \boldsymbol{U}_0 \boldsymbol{U}_0' = \sum_{i=r+1}^m \boldsymbol{u}_i \boldsymbol{u}_i'$
(3) $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A}')} = \boldsymbol{V}_r \boldsymbol{V}_r' = \sum_{i=1}^r \boldsymbol{v}_i \boldsymbol{v}_i'$
(4) $\boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{A}')} = \boldsymbol{I}_n - \boldsymbol{V}_r \boldsymbol{V}_r' = \boldsymbol{V}_0 \boldsymbol{V}_0' = \sum_{i=r+1}^n \boldsymbol{v}_i \boldsymbol{v}_i'$
(5) $\boldsymbol{P}_{\mathcal{N}(\boldsymbol{A})} = \boldsymbol{V}_0 \boldsymbol{V}_0' = \sum_{i=r+1}^n \boldsymbol{v}_i \boldsymbol{v}_i'$
(6) $\boldsymbol{P}_{\mathcal{N}^\perp(\boldsymbol{A})} = \boldsymbol{V}_r \boldsymbol{V}_r' = \sum_{i=1}^r \boldsymbol{v}_i \boldsymbol{v}_i'$
(7) $\boldsymbol{P}_{\mathcal{N}(\boldsymbol{A}')} = \boldsymbol{U}_0 \boldsymbol{U}_0' = \sum_{i=r+1}^m \boldsymbol{u}_i \boldsymbol{u}_i'$
(8) $\boldsymbol{P}_{\mathcal{N}^\perp(\boldsymbol{A}')} = \boldsymbol{I} - \boldsymbol{U}_0 \boldsymbol{U}_0' = \boldsymbol{U}_r \boldsymbol{U}_r' = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{u}_i'$

(b) (1) $\boldsymbol{P}^2_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{U}_r \boldsymbol{U}_r' = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{u}_i'$
(2) $\boldsymbol{P}^3_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{U}_r \boldsymbol{U}_r' = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{u}_i'$
(3) $\boldsymbol{P}^k_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{U}_r \boldsymbol{U}_r' = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{u}_i'$ for a positive integer k
(4) $(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{A})}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{0}_{m\times m}$
(5) $(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})^k = \boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{A})} = \boldsymbol{U}_0 \boldsymbol{U}_0' = \sum_{i=r+1}^m \boldsymbol{u}_i \boldsymbol{u}_i'$
(6) $(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})^k \boldsymbol{P}^j_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{A})}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{0}_{m\times m}$
(7) $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} - \boldsymbol{P}'_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{0}_{m\times m}$
(8) $\operatorname{rank} \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = r$
(9) $\operatorname{rank} \boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{A})} = m - r$
(10) $\operatorname{rank}(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}) = m - r$

(c) (1) $\|\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}\|_F = \sqrt{r}$
(2) $\|\boldsymbol{P}^k_{\mathcal{R}(\boldsymbol{A})}\|_F = \sqrt{r}$
(3) $\|\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}\|_F = \sqrt{m-r}$?
(4) $\|(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})^k\|_F = \sqrt{m-r}$
(5) $\|\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})}(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})\|_F = 0$
(6) $\prod_{i=1}^k (\boldsymbol{I} - \boldsymbol{P}_{\mathcal{R}(\boldsymbol{u}_i)}) - \boldsymbol{P}_{\mathcal{R}^\perp(\boldsymbol{u}_1,\dots,\boldsymbol{u}_k)} = \boldsymbol{0}_{m\times m}$ for any positive integer k

(d) (1) $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \sum_{i=1}^r \boldsymbol{u}_i \boldsymbol{u}_i' = \boldsymbol{U}_r \boldsymbol{U}_r' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{I}_r & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{U}'$
(2) $\boldsymbol{P}^\perp_{\mathcal{R}(\boldsymbol{A})} = \sum_{i=r+1}^m \boldsymbol{u}_i \boldsymbol{u}_i' = \boldsymbol{U}_0 \boldsymbol{U}_0' = [\boldsymbol{U}_0 | \boldsymbol{U}_r] \begin{bmatrix} \boldsymbol{I}_{m-r} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} [\boldsymbol{U}_0 | \boldsymbol{U}_r]'$
(3) $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{u}_1)} \boldsymbol{A} = \boldsymbol{u}_1 \boldsymbol{u}_1' \boldsymbol{A} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1' = \boldsymbol{U} \begin{bmatrix} \sigma_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$
(4) For $1 \le k \le r$: $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{u}_1,\dots,\boldsymbol{u}_k)} \boldsymbol{A} = \sum_{i=1}^k \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$
(5) For $r+1 \le k \le \min(m,n)$: $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{u}_1,\dots,\boldsymbol{u}_k)} \boldsymbol{A} = \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r' = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'$
(6) $\boldsymbol{A}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{v}_1)} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1' = \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1 = \boldsymbol{U} \begin{bmatrix} \sigma_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$

(7) For $1 \le k \le r$: $\boldsymbol{A}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k)} = \sum_{i=1}^k \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$

(8) For $1 \le k \le \min(m,n)$: $\boldsymbol{A}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k)} = \sum_{i=1}^{\min(r,k)} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' = \boldsymbol{U}_{\min(r,k)} \boldsymbol{\Sigma}_{\min(r,k)} \boldsymbol{V}_{\min(r,k)}' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Sigma}_{\min(r,k)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$

(9) For $1 \le k \le r$: $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k)} \boldsymbol{A}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k)} = \sum_{i=1}^k \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$

(10) For $1 \le j,k \le \min(m,n)$, let $t \triangleq \min(j,k,r)$. Then: $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_j)} \boldsymbol{A}\boldsymbol{P}_{\mathcal{R}(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k)} = \sum_{i=1}^t \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i' = \boldsymbol{U}_t \boldsymbol{\Sigma}_t \boldsymbol{V}_t' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Sigma}_t & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$

Pr. 16. (sol/hs135)

(a) (1) $(\boldsymbol{u}_i)^+ = \boldsymbol{u}_i'$

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

(3) $(\boldsymbol{u}_i \boldsymbol{v}_i')^+ = \boldsymbol{v}_i \boldsymbol{u}_i'$

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

(b) (1) $\boldsymbol{U}^+ = \boldsymbol{U}'$

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

(3) $\boldsymbol{U}\boldsymbol{U}^+ = \boldsymbol{I}_m$

(4) $\boldsymbol{V}\boldsymbol{V}^+ = \boldsymbol{I}_n$

(5) $\boldsymbol{\Sigma}^+ = \operatorname{Diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right)$

(6) $(\boldsymbol{U}\boldsymbol{\Sigma})^+ = \boldsymbol{\Sigma}^+ \boldsymbol{U}'$, where $\boldsymbol{\Sigma}^+$ is defined above

(7) $(\boldsymbol{\Sigma}\boldsymbol{V}')^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+$, where $\boldsymbol{\Sigma}^+$ is defined above

(8) $\boldsymbol{A}^+ = (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}')^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}'$

(9) $(\boldsymbol{U}\boldsymbol{V}')^+$: This is well defined only if $\boldsymbol{U}$ and $\boldsymbol{V}$ are a reduced (economy) or truncated (compact) SVDs, or if both are square. In those cases, the answer is $\boldsymbol{V}\boldsymbol{U}'$.

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

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

(3) For $1 \le k \le \min\{m,n\}$, $\boldsymbol{\Sigma}[1:k,1:k]^+ = \operatorname{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots 0\right) \in \mathbb{R}^{k \times k}$ if $r < k$, and $\operatorname{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_k}\right)$ otherwise.

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

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

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

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

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

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

(3) For $1 \le k \le m$, $\boldsymbol{I}_k - \boldsymbol{U}[:,1:k]^+\boldsymbol{U}[:,1:k] = \boldsymbol{0}$

(4) For $1 \le k \le n$, $\boldsymbol{I}_k - \boldsymbol{V}[:,1:k]^+\boldsymbol{V}[:,1:k] = \boldsymbol{0}$

(e) (1) For $1 \le k \le r$, $(\boldsymbol{U}[:,1:k]\boldsymbol{\Sigma}[1:k,1:k]\boldsymbol{V}[:,1:k]')^+ = \boldsymbol{V}[:,1:k]\boldsymbol{\Sigma}[1:k,1:k]^+\boldsymbol{U}[:,1:k]'$, where $\boldsymbol{\Sigma}[1:k,1:k]^+ = \operatorname{Diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_k}\right))$

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

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

(4) For $\alpha \ne 0$, $(\alpha\boldsymbol{A})^+ = \frac{1}{\alpha}\boldsymbol{A}^+$

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

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

(f) (1) $\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{U}_r\boldsymbol{U}_r'$

(2) $\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{P}_{\mathcal{N}(\boldsymbol{A})^\perp} = \boldsymbol{V}_r\boldsymbol{V}_r'$

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

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

(5) $(\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})})^+ = (\boldsymbol{U}_r\boldsymbol{U}_r')^+ = \boldsymbol{U}_r\boldsymbol{U}_r'$

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

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

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

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

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

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

Pr. 17. (sol/hs137)

(a) Let $c \in [0, 1]$.

(1) For f_i with $i = 1, 2, \ldots, n$, we are given that

$$f_i(c\boldsymbol{x} + (1-c)\boldsymbol{y}) \le cf_i(\boldsymbol{x}) + (1-c)f_i(\boldsymbol{y}).$$

Adding the first 2 such equations, we have that

$$\begin{aligned}(f_1 + f_2)(c\boldsymbol{x} + (1-c)\boldsymbol{y}) &\le cf_1(\boldsymbol{x}) + (1-c)f_1(\boldsymbol{y}) + cf_2(\boldsymbol{x}) + (1-c)f_2(\boldsymbol{y}) \\ &= c(f_1 + f_2)(\boldsymbol{x}) + (1-c)(f_1 + f_2)(\boldsymbol{y}).\end{aligned}$$

We may inductively iterate this argument: we have shown that the claim is true for $n = 2$. Then, given an arbitrary k, we have a convex function $f(\boldsymbol{x}) = \sum_{i=1}^k f_i(\boldsymbol{x})$. Then, the sum of the functions $f(\boldsymbol{x}) + f_{k+1}(\boldsymbol{x})$ is also convex, by the above argument. Hence, $f(\boldsymbol{x}) = \sum_{i=1}^n f_i(\boldsymbol{x})$ is convex.
Alternatively, just add up all n inequalities without using induction.

(2) It is sufficient to show that $wf(\boldsymbol{x})$ is a convex function if w is non-negative and $f(\boldsymbol{x})$ is convex, because then we may apply the result of the previous part. We have that

$$f(c\boldsymbol{x} + (1-c)\boldsymbol{y}) \le cf(\boldsymbol{x}) + (1-c)f(\boldsymbol{y});$$

multiplying both sides by w does not change the inequality, as $w \ge 0$:

$$wf(c\boldsymbol{x} + (1-c)\boldsymbol{y}) \le cwf(\boldsymbol{x}) + (1-c)wf(\boldsymbol{y}).$$

Hence, $wf(\boldsymbol{x})$ is convex, and we have the desired result by invoking the previous part.

(3) A simple example of $w_i < 0$ breaking convexity as follows: consider $g(x) = x^2 - x^4$. Both $f_1(x) = x^2$ and $f_2(x) = x^4$ are convex, but the function $g(x) = 1 \cdot f_1(x) + (-1) \cdot f_2(x)$ is not.

(4) Again, we are given that

$$f_i(c\boldsymbol{x} + (1-c)\boldsymbol{y}) \le cf_i(\boldsymbol{x}) + (1-c)f_i(\boldsymbol{y}),$$

for all i. Taking the maximum over i, we have that

$$\begin{aligned}\max_i f_i(c\boldsymbol{x} + (1-c)\boldsymbol{y}) &\le \max_i \{cf_i(\boldsymbol{x}) + (1-c)f_i(\boldsymbol{y})\} \\ &\le \max_i \{cf_i(\boldsymbol{x})\} + \max_i \{(1-c)f_i(\boldsymbol{y})\} \\ &= c\max_i \{f_i(\boldsymbol{x})\} + (1-c)\max_i \{f_i(\boldsymbol{y})\}.\end{aligned}$$

(b) From a previous problem, we know that the functions $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$, $\|\boldsymbol{D}\boldsymbol{x}\|_2^2$, and $\|\boldsymbol{D}\boldsymbol{x}\|_1$ are convex functions of $\boldsymbol{x}$.

(1) Yes: The sum of two convex functions is convex.

(2) Yes: The sum of two convex functions is convex.

(3) No: In general a linear combination of convex functions is convex if all coefficients are non-negative, but not necessarily otherwise, and here -1 is negative.

A simple example: let $\boldsymbol{A} = \boldsymbol{I}$, $\boldsymbol{D} = 2\boldsymbol{I}$ and $\boldsymbol{b} = \boldsymbol{0}$.

However, for any case where $\boldsymbol{A}'\boldsymbol{A} \succeq \boldsymbol{D}'\boldsymbol{D}$ this function is convex! Nonnegativity of the w_i scalars is a sufficient condition, not a necessary condition.

(4) Again, not in general.

Of course if $\boldsymbol{D} = \boldsymbol{0}$ then the function is convex.

But in general it is non-convex. A simple example: let $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{D} = \boldsymbol{I}$ and $\boldsymbol{b} = \boldsymbol{0}$.

In fact in this case if $\boldsymbol{D}$ is nonzero then I believe that this function is always non-convex. Challenge: prove or disprove this conjecture.