Pr. 1. (sol/hs019)
The software experiment suggests that `eigvals(A)` are equal to `1 ./ eigvals(B)`, possibly reordered.
Given coefficients $\boldsymbol{a} = (a_0, a_1, \ldots, a_n)$, the characteristic equation of the corresponding companion matrix $\boldsymbol{A}$ is

$$p(z) = z^n + \frac{a_{n-1}}{a_n} z^{n-1} + \cdots + \frac{a_1}{a_n} z + \frac{a_0}{a_n} = 0.$$

Because `b = reverse(a)`, the characteristic equation of the corresponding companion matrix $\boldsymbol{B}$ is

$$q(z) = z^n + \frac{b_{n-1}}{b_n} z^{n-1} + \cdots + \frac{b_1}{b_n} z + \frac{b_0}{b_n} = z^n + \frac{a_1}{a_0} z^{n-1} + \cdots + \frac{a_{n-1}}{a_0} z + \frac{a_n}{a_0} = 0.$$

Now make the following transformation of variables:

$$\tilde{q}(y) \triangleq q\left(\frac{1}{y}\right) = \left(\frac{1}{y}\right)^n + \frac{a_1}{a_0}\left(\frac{1}{y}\right)^{n-1} + \frac{a_2}{a_0}\left(\frac{1}{y}\right)^{n-2} + \cdots + \frac{a_{n-1}}{a_0}\left(\frac{1}{y}\right) + \frac{a_n}{a_0} = 0.$$

Multiplying throughout by $\frac{a_0}{a_n} y^n$ yields

$$\tilde{q}(y) \frac{a_0}{a_n} y^n = \frac{a_0}{a_n} + \frac{a_1}{a_n} y + \frac{a_2}{a_n} y^2 + \cdots + \frac{a_{n-1}}{a_n} y^n + y^n = 0,$$

which has the same roots as $p(z)$. But the roots of $p(z)$ are the eigenvalues of $\boldsymbol{A}$, and the roots of $\tilde{q}(y)$ are the reciprocals of the eigenvalues of $\boldsymbol{B}$, due to the transformation $z = \frac{1}{y}$, confirming the numerical observation.

A subtle point here is that the analysis above assumes $a_0 \neq 0$ and $a_n \neq 0$. If either are zero, then there is a root at 0 that must be considered separately. The code uses `randn` so the probability that any coefficient is zero is essentially zero.

Pr. 2. (sol/hsj7f)
The roots of $z^N = 1$ are $\mathrm{e}^{-\imath 2\pi n/N}$ for $n = 0, \ldots, N-1$ so the corresponding $N \times N$ Vandermonde matrix is just a variation of the DFT matrix having orthogonal columns and rows, for which $\boldsymbol{A}\boldsymbol{A}' = \boldsymbol{A}'\boldsymbol{A} = N\boldsymbol{I}_N$. Element k, n of $\boldsymbol{A}$ is $\left(\mathrm{e}^{-\imath 2\pi n/N}\right)^k = \mathrm{e}^{-\imath 2\pi kn/N}$, for $k, n = 0, \ldots, N-1$.
The $N \times N$ Vandermonde matrix $\boldsymbol{B}$ corresponding to $\left\{\mathrm{e}^{-\imath 2\pi(n+1/2)/N} : n = 0, \ldots, N-1\right\}$ also has orthogonal columns and rows, for which $\boldsymbol{B}\boldsymbol{B}' = \boldsymbol{B}'\boldsymbol{B} = N\boldsymbol{I}_N$.

To see this, let $z_n = \mathrm{e}^{-\imath 2\pi(n+1/2)/N}$ and define the nth column of $\boldsymbol{B}$ to be $\boldsymbol{b}_n = \begin{bmatrix} 1 \\ z_n \\ \vdots \\ z_n^{N-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \mathrm{e}^{-\imath 2\pi(n+1/2)/N} \\ \vdots \\ \mathrm{e}^{-\imath 2\pi(n+1/2)/(N-1)/N} \end{bmatrix}$.

In other words, element k, n of $\boldsymbol{B}$ is $z_n^k = \left(\mathrm{e}^{-\imath 2\pi(n+1/2)/N}\right)^k = \mathrm{e}^{-\imath 2\pi(n+1/2)k/N}$, for $k, n = 0, \ldots, N-1$.
Now examine the inner-product between the mth and nth column of $\boldsymbol{B}$:

$$\boldsymbol{b}_m' \boldsymbol{b}_n = \sum_{k=0}^{N-1} (z_m^k)^* z_n^k = \sum_{k=0}^{N-1} \left(\mathrm{e}^{\imath 2\pi(m+1/2)/N}\right)^k \left(\mathrm{e}^{-\imath 2\pi(n+1/2)/N}\right)^k = \sum_{k=0}^{N-1} \mathrm{e}^{\imath 2\pi(m-n)k/N} = N\,\mathbb{I}_{n=m}.$$

(a) Thus $\boldsymbol{X}\boldsymbol{X}' = |a|^2 \boldsymbol{A}\boldsymbol{A}' + |b|^2 \boldsymbol{B}\boldsymbol{B}' = |a|^2 N\boldsymbol{I}_N + |b|^2 N\boldsymbol{I}_N = (|a|^2 + |b|^2)N\boldsymbol{I}_N$, so $\boldsymbol{X}$ is a **tight frame**.

(b) Its **frame bound** is $\alpha = \beta = (|a|^2 + |b|^2)N$.

(c) Because $\boldsymbol{X}$ has full row rank, the MNLS solution is

$$\hat{\boldsymbol{x}} = \boldsymbol{X}^+ \boldsymbol{y} = (\boldsymbol{X}\boldsymbol{X}')^{-1}\boldsymbol{X}'\boldsymbol{y} = \left((|a|^2 + |b|^2)N\boldsymbol{I}_N\right)^{-1} \begin{bmatrix} a^* \boldsymbol{A}' \\ b^* \boldsymbol{B}' \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \frac{a^*}{(|a|^2+|b|^2)} \left(\frac{1}{N}\boldsymbol{A}'\boldsymbol{y}\right) \\ \frac{b^*}{(|a|^2+|b|^2)} \left(\frac{1}{N}\boldsymbol{B}'\boldsymbol{y}\right) \end{bmatrix}. \tag{1}$$

Now we need to compute the matrix-vector products $(1/N)\boldsymbol{A}'\boldsymbol{y}$ and $(1/N)\boldsymbol{B}'\boldsymbol{y}$ as efficiently as possible.

The problem statement says "a" Vandermonde matrix, so we chose above the most convenient form corresponding to the DFT matrix. For that choice, $(1/N)\boldsymbol{A}'\boldsymbol{y}$ is the vector with nth element $(1/N)\sum_{k=0}^{N-1} \mathrm{e}^{\imath 2\pi kn/N} y_k$, for $n = 0, \ldots, N-1$, which is equivalent to the code `ifft(y)`.

Similarly, $(1/N)\boldsymbol{B}'\boldsymbol{y}$ is the vector with nth element $(1/N)\sum_{k=0}^{N-1} \mathrm{e}^{\imath 2\pi k(n+1/2)/N} y_k = (1/N)\sum_{k=0}^{N-1} \mathrm{e}^{\imath 2\pi kn/N} \left(\mathrm{e}^{\imath \pi k/N} y_k\right)$, for $n = 0, \ldots, N-1$, which is equivalent to the code `ifft(y .* exp.((1im * pi / N) * (0:N-1)))` where `N = length(y)`.

Combining, the final `Julia` code is as follows.

```
using FFTW: ifft
function solver(a::Number, b::Number, y::AbstractVector)
    N = length(y)
    c = abs2(a) + abs2(b)
    phase = exp.((1im * pi / N) * (0:N-1))
    return [
        (conj(a) / c) * ifft(y);
        (conj(b) / c) * ifft(y .* phase)
    ]
end
```

Grader: also accept solutions that construct the $N \times N$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ that look like (1) that require $O(N^2)$ computation. Using `fft` is recommended but not required.

Here is (optional) code for checking the derivations above.

```
using LinearAlgebra
using FFTW
using SpecialMatrices: Vandermonde
include("hsj7f.jl") # solver(a, b, y)

a = randn(ComplexF64)
b = randn(ComplexF64)
N = rand(3:80)
y = randn(ComplexF64, N)

VMa(N) = [exp(-1im*2*pi*k*n/N) for n=0:N-1, k=0:N-1]
VMb(N) = [exp(-1im*2*pi*(k+1/2)*n/N) for n=0:N-1, k=0:N-1]
@assert VMa(N) * y ≈ fft(y)

k = 0:N-1
Va = transpose(Vandermonde(exp.((-1im*2*pi/N)*k)))
Vb = transpose(Vandermonde(exp.((-1im*2*pi/N)*(k .+ 1/2))))
@assert Va ≈ VMa(N)
@assert Vb ≈ VMb(N)

A = VMa(N)
B = VMb(N)
X = [a*A b*B]

@assert A'*y ≈ N * ifft(y)
@assert B'*y ≈ N * ifft(y .* exp.((1im * pi / N) * (0:N-1)))

x1 = pinv(X) * y # slow
x2 = solver(a, b, y) # fast

@show norm(X*x1 - y), norm(X*x2 - y) # should both be ≈ 0
@show norm(x1), norm(x2) # should be similar
@assert x1 ≈ x2
```

Pr. 3. (sol/hsj7c)

(a) If $\boldsymbol{A}$ and $\boldsymbol{B}$ are both circulant matrices of the same size, then they both are diagonalized by the same DFT matrix, *i.e.*, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}_A\boldsymbol{Q}'$ and $\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{\Lambda}_B\boldsymbol{Q}'$. Thus using the fact that diagonal matrices commute:

$$\boldsymbol{A}\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{\Lambda}_B\boldsymbol{Q}'\boldsymbol{Q}\boldsymbol{\Lambda}_A\boldsymbol{Q}' = \boldsymbol{Q}\boldsymbol{\Lambda}_B\boldsymbol{\Lambda}_A\boldsymbol{Q}' = \boldsymbol{Q}\boldsymbol{\Lambda}_A\boldsymbol{\Lambda}_B\boldsymbol{Q}' = \boldsymbol{Q}\boldsymbol{\Lambda}_A\boldsymbol{Q}'\boldsymbol{Q}\boldsymbol{\Lambda}_B\boldsymbol{Q}' = \boldsymbol{B}\boldsymbol{A}.$$

(b) If $\boldsymbol{C}$ is a circulant matrix, then $\boldsymbol{C}'$ is also a circulant matrix so $\boldsymbol{C}$ and $\boldsymbol{C}'$ commute by the previous problem. Thus $\boldsymbol{C}\boldsymbol{C}' = \boldsymbol{C}'\boldsymbol{C}$ so $\boldsymbol{C}$ is normal.

(c) The first column is $(-1, 0, \ldots, 1)$ for which the DFT is $-1 + \mathrm{e}^{\imath 2\pi k/N}$ for $k = 0, \ldots, N-1$

(d) For $N = 4$ we have $-1 + \mathrm{e}^{\imath 2\pi k/4} = -1 + \mathrm{e}^{\imath \pi k/2} = (0, -1+\imath, -2, -1-\imath)$

```
using LinearAlgebra
N = 4
C = -I + [i==mod(j-1,N) for i=0:N-1, j=0:N-1]
eigvals(C)
```

returns:

```
4-element Array{Complex{Float64},1}:
     -2.0000000000000013 + 0.0im
     -1.0000000000000002 + 1.0000000000000002im
     -1.0000000000000002 - 1.0000000000000002im
 -1.8732355726879183e-16 + 0.0im
```

This is the same to within numerical precision but in a different order.

(e) Because all circulant matrices are normal matrices, the singular values are simply the (sorted) absolute values of the eigenvalues. Thus the nuclear norm is simply:

```
using FFTW
```

```
nucnorm_circulant = (C) -> sum(abs.(fft(C[:,1])))
```

(f) Optional. Suppose $\boldsymbol{C}_1$ is $N_1 \times N_1$ and $\boldsymbol{C}_2$ is $N_2 \times N_2$.

Let $\lambda_n,\ n = 0, \ldots, N_1 - 1$ denote the DFT of the first column of $\boldsymbol{C}_1$.

Let $\omega_n,\ n = 0, \ldots, N_2 - 1$ denote the DFT of the first column of $\boldsymbol{C}_2$.

Then the eigenvalues of the $N_1N_2 \times N_1N_2$ matrix $\boldsymbol{B} = \boldsymbol{C}_2 \otimes \boldsymbol{C}_1$ are the N_1N_2 values: $\{\lambda_n\omega_m : n = 0, \ldots, N_1 - 1,\ m = 0, \ldots, N_2 - 1\}$.

Challenge. We can use the vec trick and an earlier HW problem about the 2D DFT here. Take the first column of $\boldsymbol{B}$ and reshape it into a 2D array $\boldsymbol{X}$ of size $N_1 \times N_2$. Then the eigenvalues of $\boldsymbol{B}$ are the 2D DFT of that array: $\boldsymbol{Q}_{N_1}\boldsymbol{X}\boldsymbol{Q}_{N_2}^\top$ where $\boldsymbol{Q}_N$ denotes the $N \times N$ DFT matrix

Pr. 4. (sol/hsj68)

(a) Because the Schatten p-norm is unitarily invariant:

$$\hat{\boldsymbol{X}} = \boldsymbol{U}_r \hat{\boldsymbol{\Sigma}}_r \boldsymbol{V}_r', \quad \hat{\boldsymbol{\Sigma}}_r = \underset{\boldsymbol{S}=\mathrm{Diag}(s_1,\ldots,s_r)}{\arg\min} \frac{1}{2} \|\boldsymbol{\Sigma}_r - \boldsymbol{S}\|_{\mathrm{F}}^2 + \beta \sum_{k=1}^{r} |s_k|^{1/2},$$

as described in the course notes. So here we must solve

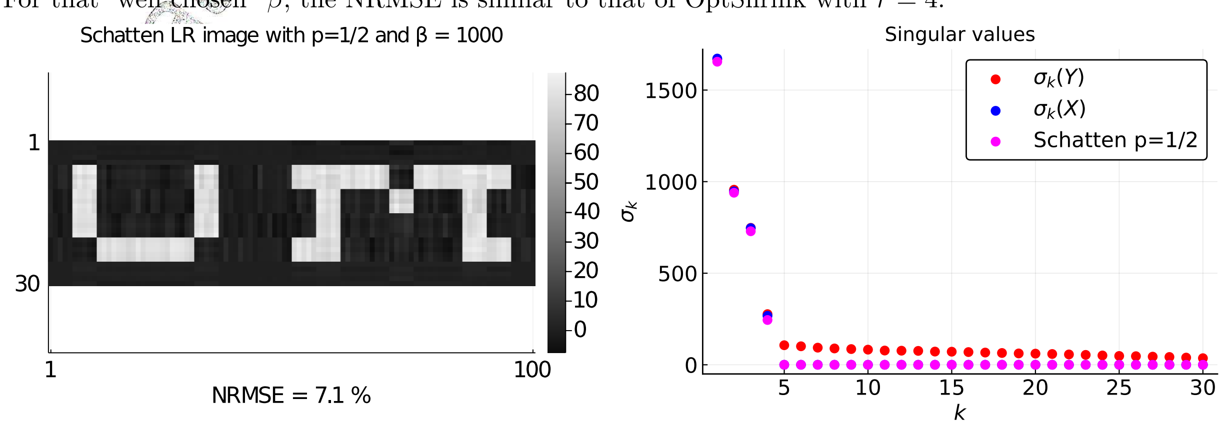
$$\underset{s_1,\ldots,s_r \geq 0}{\arg\min} \sum_{k=1}^{r} \left(\frac{1}{2} (\sigma_k - s_k)^2 + \beta s_k^{1/2} \right), \; i.e., \; \hat{\sigma}_k = \underset{s \geq 0}{\arg\min} \frac{1}{2} (\sigma_k - s)^2 + \beta s^{1/2}.$$

This is exactly what a previous problem solved, so we use that solution.

(b) A possible `Julia` implementation is

```
using LinearAlgebra: svd, Diagonal
#include("shrink_p_1_2_sol.jl")

"""
    lr_schatten(Y, reg::Real)

Compute the regularized low-rank matrix approximation as the minimizer over `X`
of `1/2 \\|Y - X\\|^2 + reg R(x)`
where `R(X)` is the Schatten p-norm of `X` raised to the pth power, for `p=1/2`,
i.e., `R(X) = \\sum_k (\\sigma_k(X))^{1/2}`

In:
- `Y` : `M × N` matrix
- `reg` regularization parameter

Out:
- `Xh` : `M × N` solution to above minimization problem
"""
function lr_schatten(Y, reg::Real)
    (U,s,V) = svd(Y)
    sh = shrink_p_1_2(s, reg)
    return U * Diagonal(sh) * V'
end


function shrink_p_1_2(y, reg::Real)
    xh = zeros(size(y))
    fun = (y) -> 4/3 * y * cos(1/3 * acos(- (3^(3/2)*reg) / (4*y^(3/2))))^2
    big = y .> 3/2 * reg^(2/3)
    xh[big] = fun.(y[big])
    return xh
end
```

(c) For $\beta = 1000$ the NRMSE of the Schatten p-norm LR method is 7.4 % and the resulting image $\hat{\boldsymbol{X}}$ and singular values are shown below. The denoised image $\hat{\boldsymbol{X}}$ has singular values that closely match those of the latent matrix $\boldsymbol{X}$. For that "well chosen" β, the NRMSE is similar to that of OptShrink with $\hat{r} = 4$.



Pr. 5. (sol/hs011)

(a) Let $\boldsymbol{A}$ and $\boldsymbol{B}$ denote companion matrices associated with degree m polynomial $p(z) = \alpha_0 + \alpha_1 z + \cdots + \alpha_{m-1} z^{m-1} + z^m$ and degree n polynomial $q(z) = \beta_0 + \beta_1 z + \cdots + \beta_{n-1} z^{n-1} + z^n$, respectively.

Consider the matrix:

$$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{I}_m + \boldsymbol{I}_n \otimes (-\boldsymbol{B}) = \boldsymbol{A} \otimes \boldsymbol{I}_m - \boldsymbol{I}_n \otimes \boldsymbol{B}.$$

The eigenvalues of $\boldsymbol{C}$ are $\lambda_i(\boldsymbol{A}) - \lambda_j(\boldsymbol{B})$, so that if the polynomials corresponding to $\boldsymbol{A}$ and $\boldsymbol{B}$ have common roots, then at least one of the eigenvalues of $\boldsymbol{C}$ will be identically zero. Thus if $\det(\boldsymbol{C}) = 0$, then we can declare that the polynomials associated with the companion matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ have common roots.

If $n = 0$ or $m = 0$, then the set of roots for at least one of the two polynomials is the empty set, so there are no common roots.

(b) A possible `Julia` implementation is

```julia
using LinearAlgebra: I, det


"""
    haveCommonRoot = common_root(a, b ; atol)

Determine if the polynomials described by input coefficient vectors `a`
and `b` share a common root, to within an absolute tolerance parameter `atol`.
Assume leading coefficients `a[end]` and `b[end]` are nonzero.

In:
- `a` : vector of length `m + 1` with `a[m+1] != 0` and `m ≥ 0`
defining a degree `m` polynomial of the form:
`p(z) = a[m+1] z^m + a[m] z^(m - 1) + ... + a[2] z + a[1]`

- `b` : vector of length `n + 1` with `b[n+1] != 0` and `n ≥ 0`
defining a degree `n` polynomial of the form:
`q(z) = b[n+1] z^n + b[n] z^(n - 1) + ... + b[1] z + b[1]`

Option:
- `atol::Real` absolute tolerance for calling `isapprox`

Out:
- `haveCommonRoot` = `true` when `p` and `q` share a common root, else `false`
"""
function common_root(a::AbstractVector, b::AbstractVector ; atol::Real=1e-6)

    compan(c) = [-transpose(reverse(c)); [I zeros(length(c)-1)]] # from notes

    # If either polynomial is a constant (m=0 or n=0), then no common roots
    ((length(a) == 1) || (length(b) == 1)) && return false

    # Construct companion matrices
    A = compan(a[1:end-1] / a[end])
    B = compan(b[1:end-1] / b[end])

    # Compute Kronecker sum of A and -B
    C = kron(A, I(size(B,1))) - kron(I(size(A,1)), B)

    # Check for common roots by seeing if determinant ≈ 0
    return isapprox(det(C), 0 ; atol)
end
```

Non-graded problem(s) below

Pr. 6. (sol/hsj7b)
It is most natural here to use the Frobenius inner product. For $k \neq l$, $\boldsymbol{G}_N^k$ and $\boldsymbol{G}_N^l$ are orthogonal because $\langle \boldsymbol{G}_N^k, \boldsymbol{G}_N^l \rangle = \langle \text{vec}(\boldsymbol{G}_N^k), \text{vec}(\boldsymbol{G}_N^l) \rangle = 0$ because the two matrices have 1's in different locations.
However, $\left\| \boldsymbol{G}_N^k \right\|_{\text{F}} = \langle \boldsymbol{G}_N^k, \boldsymbol{G}_N^k \rangle = N$, so the basis matrices are not unit norm, so the set is not an orthonormal basis with respect to the usual Frobenius inner product.
However, if one defined a scaled Frobenius inner product that has a $1/\sqrt{N}$ factor, then the set *is* an orthonormal basis with respect to that inner product.
Grader: conceivably a student may choose a different inner product for which the answers can be "no." Please let me know if you see such an answer so I can see what inner product was used.

Pr. 7. (sol/hsj22)
To construct a counter-example it suffices to consider a 2×2 matrix having distinct eigenvalues that are each a kth root of unity but whose eigenvectors are not orthogonal. Let $\boldsymbol{T} = \boldsymbol{V} \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath 2\pi/k} \end{bmatrix} \boldsymbol{V}^{-1}$ where $\boldsymbol{V} = \begin{bmatrix} 1 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix}$.
Then by design $\boldsymbol{T}^k = \boldsymbol{V}\boldsymbol{I}\boldsymbol{V}^{-1} = \boldsymbol{I}$ but $\boldsymbol{T}$ is not normal.

Pr. 8. (sol/hs104)
Note that

$$\boldsymbol{y}(t_k) = \underbrace{\begin{bmatrix} \boldsymbol{b}_1 & \dots & \boldsymbol{b}_r \end{bmatrix}}_{\triangleq \boldsymbol{B}} \underbrace{\begin{bmatrix} e^{\imath \omega_1 t_k} \\ \vdots \\ e^{\imath \omega_r t_k} \end{bmatrix}}_{\triangleq \boldsymbol{x}}. \tag{2}$$

Similarly, letting $\Delta = t_{k+1} - t_k$:

$$\boldsymbol{y}(t_{k+1}) = \begin{bmatrix} \boldsymbol{b}_1 & \dots & \boldsymbol{b}_r \end{bmatrix} \begin{bmatrix} e^{\imath \omega_1 t_{k+1}} \\ \vdots \\ e^{\imath \omega_r t_{k+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 & \dots & \boldsymbol{b}_r \end{bmatrix} \underbrace{\begin{bmatrix} e^{\imath \omega_1 \Delta} & 0 & \dots & 0 \\ 0 & e^{\imath \omega_2 \Delta} & \dots & 0 \\ 0 & 0 & \dots & e^{\imath \omega_r \Delta} \end{bmatrix}}_{\triangleq \boldsymbol{D}} \begin{bmatrix} e^{\imath \omega_1 t_k} \\ \vdots \\ e^{\imath \omega_r t_k} \end{bmatrix}. \tag{3}$$

Because there is no noise in this model and $\boldsymbol{B}$ has full column rank, we can "solve" for $\boldsymbol{x}$ in (2) using linear least squares:

$$\boldsymbol{x} = \boldsymbol{B}^+ \boldsymbol{y}(t_k). \tag{4}$$

(a) Substituting (4) in (3) yields:

$$\boldsymbol{y}(t_{k+1}) = \underbrace{\boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^+}_{\triangleq \boldsymbol{A}} \boldsymbol{y}(t_k).$$

Therefore the required matrix is

$$\boldsymbol{A} = \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^+.$$

(b) The r columns of $\boldsymbol{B}$ are linearly independent, and all the diagonal elements of $\boldsymbol{D}$ are distinct, since ω_i are distinct. From the hint, we know that $\boldsymbol{B}^+\boldsymbol{B} = \boldsymbol{I}_r$, so

$$\boldsymbol{A}\boldsymbol{b}_i = \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^+\boldsymbol{b}_i = \boldsymbol{B}\boldsymbol{D}\boldsymbol{e}_i = d_{ii}\boldsymbol{b}_i = e^{\imath \omega_i \Delta}\boldsymbol{b}_i.$$

Therefore, the r (non-zero) eigenvalues of $\boldsymbol{A}$ are $e^{\imath \omega_1 \Delta}, \dots, e^{\imath \omega_r \Delta}$ and the corresponding eigenvectors are $\boldsymbol{b}_1, \dots, \boldsymbol{b}_r$.

Pr. 9. (sol/hsj5g)

(a) The definition of a symmetric gauge function directly ensures the nonnegative, positive, and homogeneity properties hold for the matrix norm.

Proving the triangle inequality takes more work; see p. 91 of Rajendra Bhatia, "Matrix analysis" at `http://doi.org/10.1007/978-1-4612-0653-8`

(b) Any proper matrix norm that is defined solely in terms of the singular values of its matrix argument is a unitarily invariant norm because $\boldsymbol{A}$ and $\boldsymbol{UAV}$ have the same singular values when $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices of the correct size.

Pr. 10. (sol/hsj5o)

This problem explores extensions of **unitary invariance** of norms. Throughout this problem, let $\boldsymbol{A} \in \mathbb{F}^{M \times N}$ and assume $\boldsymbol{X}$ is a $K \times M$ matrix with orthonormal columns and $\boldsymbol{Y}$ is a $N \times L$ matrix with orthonormal rows. (Do not assume $\boldsymbol{X}$ or $\boldsymbol{Y}$ are square.)

(a) $\|\boldsymbol{XAY}\|_{\mathrm{F}}^2 = \operatorname{trace}(\boldsymbol{XAYY'A'X'}) = \operatorname{trace}(\boldsymbol{AA'X'X}) = \operatorname{trace}(\boldsymbol{AA'})\ \|\boldsymbol{A}\|_{\mathrm{F}}^2$

(b) $f(\boldsymbol{B}) = J\,\|\boldsymbol{B}\|_* \Rightarrow f(\boldsymbol{UAV}) = J\,\|\boldsymbol{UAV}\|_* = J\,\|\boldsymbol{A}\|_* = f(\boldsymbol{A})$, for any $M \times M$ unitary matrix $\boldsymbol{U}$ and any $N \times N$ unitary matrix $\boldsymbol{V}$ because multiplying by $\boldsymbol{U}$ and $\boldsymbol{V}$ does not change the singular values of $\boldsymbol{A}$, so the nuclear norm remains unchanged

(c) Consider $\boldsymbol{A} = [1] = \boldsymbol{Y}$ and $\boldsymbol{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} / \sqrt{2}$. $1 = f(\boldsymbol{A}) \neq f(\boldsymbol{XAY}) = f\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} / \sqrt{2}\right) = 2$ because $J = K = 2$ here ($\boldsymbol{B}$ has two rows). So the norm $f(\cdot)$ is unitarily invariant, but it is not "orthonormal invariant" because of the way it depends on the matrix size.

(d) Consider the following "unified" matrix norm defined in Ch. 5: $\|\boldsymbol{A}\|_{K,p} \triangleq \left(\sum_{k=1}^K \sigma_k^p\right)^{1/p}$ where $K \in \mathbb{N}$ and $1 \leq p < \infty$.

(e) Consider the compact SVD $\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r'$. Now define $\boldsymbol{B} = \boldsymbol{XAY} = \boldsymbol{XU}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r' \boldsymbol{Y} = \tilde{\boldsymbol{U}}_r \boldsymbol{\Sigma}_r \tilde{\boldsymbol{V}}_r{}'$ where $\tilde{\boldsymbol{U}}_r = \boldsymbol{XU}_r$ and $\tilde{\boldsymbol{V}}_r = \boldsymbol{Y}'\boldsymbol{V}_r$ both have orthonormal columns.

Thus this is a valid compact SVD $\boldsymbol{B}$ so $\|\boldsymbol{B}\|_{K,p} = \|\boldsymbol{XAY}\|_{K,p} = \|\boldsymbol{\Sigma}_r\|_{K,p} = \|\boldsymbol{A}\|_{K,p}$.

This general family of unitarily invariant norms has "orthonormal invariance."
