EE263 Prof. S. Boyd
Dec. 7–8 or Dec. 8–9, 2007.

Final exam solutions

1. *Optimal initial conditions for a bioreactor.* The dynamics of a bioreactor are given by $\dot{x}(t) = Ax(t)$, where $x(t) \in \mathbf{R}^n$ is the state, with $x_i(t)$ representing the total mass of species or component i at time t. Component i has (positive) value (or cost) c_i, so the total value (or cost) of the components at time t is $c^T x(t)$. (We ignore any extra cost that would be incurred in separating the components.) Your job is to choose the initial state, under a budget constraint, that maximizes the total value at time T. More specifically, you are to choose $x(0)$, with all entries nonnegative, that satisfies $c^T x(0) \leq B$, where B is a given positive budget. The problem data (*i.e.*, things you know) are A, c, T, and B.

   You can assume that A is such that, for any $x(0)$ with nonnegative components, $x(t)$ will also have all components nonnegative, for any $t \geq 0$. (This occurs, by the way, if and only if the off-diagonal entries of A are nonnegative.)

   (a) Explain how to solve this problem.

   (b) Carry out your method on the specific instance with data

$$A = \begin{bmatrix} 0.1 & 0.1 & 0.3 & 0 \\ 0 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0 \\ 0 & 0 & 0.2 & 0.1 \end{bmatrix}, \quad c = \begin{bmatrix} 3.5 \\ 0.6 \\ 1.1 \\ 2.0 \end{bmatrix}, \quad T = 10, \quad B = 1.$$

   Give the optimal $x(0)$, and the associated (optimal) terminal value $c^T x(T)$.

   Give us the terminal value obtained when the initial state has equal mass in each component, *i.e.*, $x(0) = \alpha \mathbf{1}$, with α adjusted so that the total initial cost is B. Compare this with the optimal terminal value.

   Also give us the terminal value obtained when the same amount, B/n, is spent on each initial state component (*i.e.*, $x(0)_i = B/(nc_i)$). Compare this with the optimal terminal value.

Solution.

(a) We have $c^T x(T) = c^T e^{tA} x(0) = b^T x(0)$, where we define $b = (e^{TA})^T c$, so our problem is to maximize $b^T x(0)$, subject to $x(0) \geq 0$ (this means all its entries are nonnegative), and $c^T x(0) \leq B$. You can think of c_i as the cost of investing in a unit of component i, and b_i as the payoff received. Thus, the gain is b_i/c_i. The

solution to this problem is to invest everything (*i.e.*, the whole budget B) in any component that has maximum gain. More formally, we choose any k for which $b_k/c_k = \max\{b_1/c_1, \ldots, b_n/c_n\}$, and then set $x(0) = Be_k$. (Recall that we assume $b_i \geq 0$ and $c_i > 0$ here.)

We didn't require a completely formal proof that this is the optimal strategy. But here is one, just so you know what one looks like. Suppose that $x(0)$ satisfies $x(0) \geq 0$, $c^T x(0) \leq B$. Then we have

$$\begin{aligned} b^T x(0) &= \sum_{i=1}^{n} (b_i/c_i)(c_i x(0)_i) \\ &\leq \left(\max_{i=1,\ldots,n} (b_i/c_i)\right) \left(\sum_{i=1}^{n} c_i x(0)_i\right) \\ &\leq B \max_{i=1,\ldots,n} (b_i/c_i). \end{aligned}$$

This shows that no feasible choice of $x(0)$ can yield terminal value $c^T x(T) = b^T x(0)$ more than $B \max_{i=1,\ldots,n}(b_i/c_i)$. But the choice described above yields this value of $b^T x(0)$, and so must be optimal.

(b) The code below solves the problem.

```
% problem data
A=[ 0.1 0.1 0.3 0;
    0   0.2 0.4 0.3;
    0.1 0.3 0.1 0;
    0   0   0.2 0.1];
c=[3.5; 0.6; 1.1; 2.0];
n=length(c);
T=10;
B=1;
b= (expm(T*A))'*c;
[g,k]=max(b./c); % get max value and index k
opt_x0=zeros(n,1);
opt_x0(k)=B/c(k);
opt_term_value=b'*opt_x0
opt_x0

% terminal value with equal mass in each initial component
x0mass=(B/sum(c))*ones(n,1);
term_value=b'*x0mass

% terminal value with equal value in each initial component
x0val=(B/n)./c;
term_value=b'*x0val
```

The optimal initial condition is $x(0) = (5/3)e_2$, which yields terminal value 1168. With equal initial mass in each component, the terminal value is 300; with equal initial investment in each component, the terminal value is 552.

2. *Simultaneously estimating student ability and exercise difficulty.* Each of n students takes an exam that contains m questions. Student j receives (nonnegative) grade G_{ij} on question i. One simple model for predicting the grades is to estimate $G_{ij} \approx \hat{G}_{ij} = a_j/d_i$, where a_j is a (nonnegative) number that gives the *ability* of student j, and d_i is a (positive) number that gives the *difficulty* of exam question i. Given a particular model, we could simultaneously scale the student abilities and the exam difficulties by any positive number, without affecting $\hat{G}_{ij}$. Thus, to ensure a unique model, we will *normalize* the exam question difficulties d_i, so that the mean exam question difficulty across the m questions is 1.

   In this problem, you are given a complete set of grades (*i.e.*, the matrix $G \in \mathbf{R}^{m\times n}$). Your task is to find a set of nonnegative student abilities, and a set of positive, normalized question difficulties, so that $G_{ij} \approx \hat{G}_{ij}$. In particular, choose your model to minimize the RMS error, J,

$$J = \left(\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(G_{ij} - \hat{G}_{ij} \right)^2 \right)^{1/2}.$$

   This can be compared to the RMS value of the grades,

$$\left(\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} G_{ij}^2 \right)^{1/2}.$$

   (a) Explain how to solve this problem, using any concepts from EE263. If your method is approximate, or not guaranteed to find the global minimum value of J, say so. If carrying out your method requires some rank or other conditions to hold, say so.

   Note: You do not have to concern yourself with the requirement that a_j are nonnegative and d_i are positive. You can just assume this works out, or is easily corrected.

   (b) Carry out your method on the data found in `grademodeldata.m`. Give the optimal value of J, and also express it as a fraction of the RMS value of the grades. Give the difficulties of the 7 problems on the exam.

   Solution. First we note that $\hat{G}$ is a rank-1 matrix, since we can write

$$\hat{G} = \begin{bmatrix} 1/d_1 \\ 1/d_2 \\ \vdots \\ 1/d_m \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}.$$

   Our problem is to find the best rank-1 approximation to G, judged by the criterion

$$mnJ^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(G_{ij} - \hat{G}_{ij} \right)^2 = \|G - \hat{G}\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. We mentioned in lecture 16 that this problem has the same solution as minimizing $\|G - \hat{G}\|$, subject to $\mathbf{Rank}(\hat{G}) = 1$, *i.e.*, the top dyad in the SVD of G. Let

$$G = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

be the SVD of G. The optimal rank-1 approximation of G is

$$\hat{G} = \sigma_1 u_1 v_1^T.$$

(The optimal rank-1 approximation is unique if $\sigma_1 > \sigma_2$.) We can manipulate this model into the required form. We take

$$a_i = \frac{m\sigma_1 v_{1i}}{\sum_{j=1}^{n}(1/u_{1j})}, \quad i = 1, \ldots, n,$$

and

$$d_i = \frac{m}{u_{1i}\sum_{j=1}^{n}(1/u_{1j})}, \quad i = 1, \ldots, m.$$

The following Matlab code solves the problem.

```
grademodeldata;
% svd
[U,Sigma,V] = svd(G, 'econ');
% extract the top dyad
u1 = U(:,1);
v1 = V(:,1);
s1 = Sigma(1,1);
% normalize d
a = m*s1*v1/sum(1./u1);
d = m./(sum(1./u1)*u1);
% compute optimal cost
Jopt = sqrt(1/(m*n))*norm(G-(1./d)*a','fro')
RMSgrades = sqrt(1/(m*n))*norm(G,'fro')
ratio=Jopt/RMSgrades
```

For the given data, we find

$$\begin{aligned}
d_1 &= 0.9429,\\
d_2 &= 1.2780,\\
d_3 &= 0.9015,\\
d_4 &= 0.9197,\\
d_5 &= 0.7729,\\
d_6 &= 1.0418,\\
d_7 &= 1.1433.
\end{aligned}$$

The optimal fitting cost is $J^\star = 5.6759$. The RMS value of the grades is 15.88, so the optimal fitting cost is 0.3574 of the RMS value of the grades.

3. *Optimal espresso cup pre-heating.* At time $t = 0$ boiling water, at 100°C, is poured into an espresso cup; after P seconds (the 'pre-heating time'), the water is poured out, and espresso, with initial temperature 95°C, is poured in. (You can assume this operation occurs instantaneously.) The espresso is then consumed exactly 15 seconds later (yes, instantaneously). The problem is to choose the pre-heating time P so as to maximize the temperature of the espresso when it is consumed.

   We now give the thermal model used. We take the temperature of the liquid in the cup (water or espresso) as one state; for the cup we use an n-state finite element model. The vector $x(t) \in \mathbf{R}^{n+1}$ gives the temperature distribution at time t: $x_1(t)$ is the liquid (water or espresso) temperature at time t, and $x_2(t), \ldots, x_{n+1}(t)$ are the temperatures of the elements in the cup. All of these are in degrees C, with t in seconds. The dynamics are

$$\frac{d}{dt}(x(t) - 20 \cdot \mathbf{1}) = A(x(t) - 20 \cdot \mathbf{1}),$$

   where $A \in \mathbf{R}^{(n+1)\times(n+1)}$. (The vector $20 \cdot \mathbf{1}$, with all components 20, represents the ambient temperature.) The initial temperature distribution is

$$x(0) = \begin{bmatrix} 100 \\ 20 \\ \vdots \\ 20 \end{bmatrix}.$$

   At $t = P$, the liquid temperature changes instantly from whatever value it has, to 95; the other states do not change. Note that the dynamics of the system are the same before and after pre-heating (because we assume that water and espresso behave in the same way, thermally speaking).

   We have *very generously* derived the matrix A for you. You will find it in `espressodata.m`. In addition to `A`, the file also defines `n`, and, respectively, the ambient, espresso and preheat water temperatures `Ta` (which is 20), `Te` (95), and `Tl` (100).

   Explain your method, submit your code, and give final answers, which must include the optimal value of P and the resulting optimal espresso temperature when it is consumed. Give both to an accuracy of one decimal place, as in

   '$P = 23.5$ s, which gives an espresso temperature at consumption of 62.3°C.'

   (This is not the correct answer, of course.)

   Solution. After P seconds of pre-heating, we will have

$$x(P) - 20 \cdot \mathbf{1} = e^{PA}(x(0) - 20 \cdot \mathbf{1}).$$

   Define a new vector $\tilde{x}(P)$ with $\tilde{x}_i(P) = x_i(P)$ for $i = 2, \ldots, n+1$, and $\tilde{x}_1(P) = 95$. (Thus, $\tilde{x}(P)$ is the state immediately after the water is replaced with espresso.) The temperature distribution at time $P + 15$ will be

$$x(P + 15) - 20 \cdot \mathbf{1} = e^{15A}(\tilde{x}(P) - 20 \cdot \mathbf{1}).$$

We now have a method for calculating the temperature of the espresso at the instant of consumption for a given P:

$$T(P) - 20 = e_1^T x(P + 15) = e_1^T e^{15A}(\tilde{x}(P) - 20 \cdot \mathbf{1}),$$

where e_1 is the first unit vector. Thus, we have

$$T(P) = e_1^T e^{15A}(\tilde{x}(P) - 20 \cdot \mathbf{1}) + 20.$$

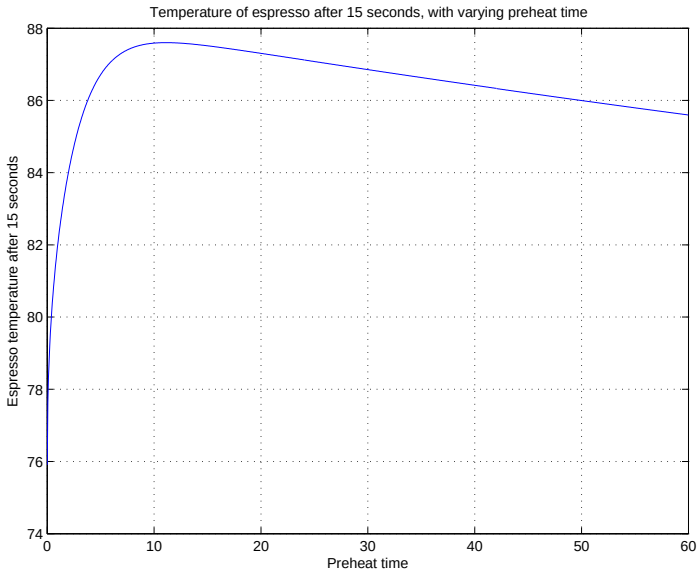
To find the optimal value of P we use a simple search method, by calculating $T(P)$ over a finely-sampled range of values of P, and selecting the maximum value.

The optimal preheating time for this example is 11.1 seconds. This will give an espresso temperature of 87.6°C.

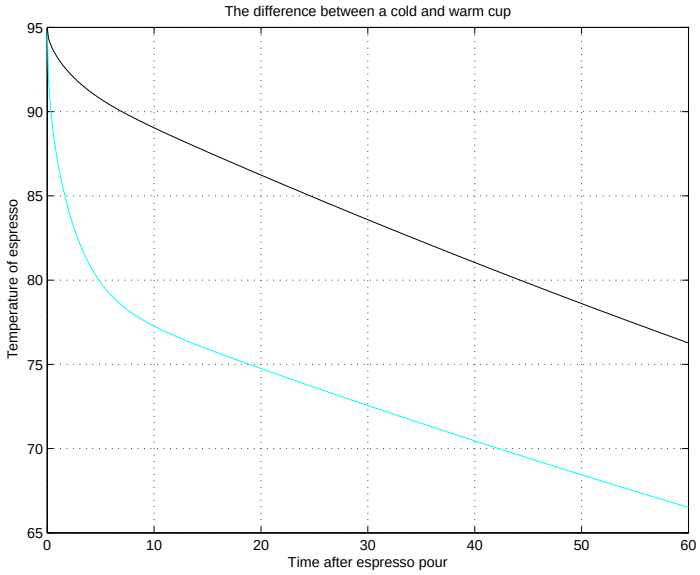
Matlab code to calculate the answers appears below.

```
% load data.
espressodata;

% Test a range of preheating times up to a minute.
Tphs = linspace(0, 60, 1000);
% Condition at instant when preheating liquid is added.
% Note change of coordinates by subtracting Ta (and elsewhere).
p0 = [Tl; Ta*ones(n,1)] - Ta;

y = zeros(size(Tphs));
for i = 1:length(Tphs)
    Tph = Tphs(i);

    % Find state after preheating by propagating forward.
    xph = expm(Tph*A)*p0;
    % Instantaneously add espresso, changing only the liquid portion of the
    % state.
    xph(1) = Te - Ta;

    % Record temperature at time 15.
    z = expm(15*A)*xph;
    y(i) = z(1);
end

[Tmax, i] = max(y+Ta);
```

The graph below shows how preheat time affects the drinking temperature.



The next graph shows the temperature of the espresso over a 5 minute period, with and without preheating.



4. *Optimal dynamic purchasing.* You are to complete a large order to buy a certain number, B, of shares in some company. You are to do this over T time periods. (Depending on the circumstances, a single time period could be between tens of milliseconds and minutes.) We will let b_t denote the number of shares bought in time period t, for $t = 1, \ldots, T$, so we have $b_1 + \cdots + b_T = B$. (The quantities $B, b_1, \ldots, b_T$ can all be any real number; $b_t < 0$, for example, means we *sold* shares in the period t. We also don't require b_t to be integers.) We let p_t denote the price per share in period t, so the total cost of purchasing the B shares is $C = p_1 b_1 + \cdots + p_T b_T$.

   The amounts we purchase are large enough to have a noticeable effect on the price of the shares. The prices change according to the following equations:

$$p_1 = \bar{p} + \alpha b_1, \qquad p_t = \theta p_{t-1} + (1-\theta)\bar{p} + \alpha b_t, \quad t = 2, \ldots, T.$$

   Here $\bar{p}$ is the base price of the shares and α and θ are parameters that determine how purchases affect the prices. The parameter α, which is positive, tells us how much the price goes up in the current period when we buy one share. The parameter θ, which lies between 0 and 1, measures the *memory*: If $\theta = 0$ the share price has no memory, and the purchase made in period t only affects the price in that period; if θ is 0.5 (say), the effect a purchase has on the price decays by a factor of two between periods. If $\theta = 1$, the price has perfect memory and the price change will persist for all future periods.

   If purchases didn't increase the price, the cost of purchasing the shares would always be $\bar{p}B$. The difference between the total cost and this cost, $C - \bar{p}B$, is called the *transaction cost*.

   Find the purchase quantities $b_1, \ldots, b_T$ that minimize the transaction cost $C - \bar{p}B$, for the particular problem instance with

$$B = 10000, \quad T = 10, \quad \bar{p} = 10, \quad \theta = 0.8, \quad \alpha = 0.00015.$$

   Give the optimal transaction cost. Also give the transaction cost if all the shares were purchased in the first period, and the transaction cost if the purchases were evenly spread over the periods (*i.e.*, if 1000 shares were purchased in each period). Compare these three quantities.

   You must explain your method clearly, using any concepts from this class, such as least-squares, pseudo-inverses, eigenvalues, singular values, etc. If your method requires that some rank or other conditions to hold, say so. You must also check, in your Matlab code, that these conditions are satisfied for the given problem instance.

   Solution. We first derive a compact expression for p, the vector of prices, in terms of b, the vector of purchase amounts. By iterating the price process, we get

$$\begin{aligned} p_1 &= \bar{p} + \alpha b_1 \\ p_2 &= \bar{p} + \alpha b_2 + \alpha\theta b_1 \end{aligned}$$

$$\begin{aligned} p_3 &= \bar{p} + \alpha b_3 + \alpha\theta b_2 + \alpha\theta^2 b_1 \\ &\vdots \\ p_T &= \bar{p} + \alpha b_T + \alpha\theta b_{T-1} + \cdots + \alpha\theta^{T-1} b_1. \end{aligned}$$

We write this as $p = \bar{p}\mathbf{1} + Ab$, where $A \in \mathbf{R}^{T\times T}$ is the (lower triangular Toeplitz) matrix with

$$A_{ij} = \begin{cases} \alpha\theta^{i-j} & i \geq j \\ 0 & \text{otherwise.} \end{cases}$$

The cost is

$$C = b^T p = b^T(\bar{p}\mathbf{1} + Ab) = \bar{p}B + b^T Ab,$$

since $b^T\mathbf{1} = B$. The first term, $\bar{p}B$, is just the total cost if the purchases did not increase the share price. The second term, $b^T Ab$, is the transaction cost. So we see now that the transaction cost is a quadratic form in b. The matrix A is not symmetric, which can lead to trouble, so we'll write the transaction cost as $(1/2)b^T(A + A^T)b$.

For our problem instance, we can check that A is indeed positive definite. Intuitively, this must be the case; otherwise C can be made arbitrarily negative (*i.e.*, we can make an arbitrary profit).

To find the optimal purchase quantities b, we must solve the following optimization problem:

$$\begin{array}{ll} \text{minimize} & (1/2)b^T(A + A^T)b \\ \text{subject to} & \mathbf{1}^T b = B, \end{array}$$

with variable b. This not exactly a problem we've solved before, but we can solve it using methods from the notes. (We'll also see below how to convert it to a problem we have solved before.)

The Lagrangian is

$$L(b, \lambda) = (1/2)b^T(A + A^T)b + \lambda\left(\mathbf{1}^T b - B\right),$$

and the optimality conditions are

$$\nabla_b L(b, \lambda) = (A + A^T)b + \lambda\mathbf{1} = 0, \quad \nabla_\lambda L(b, \lambda) = \mathbf{1}^T b - B = 0.$$

This can be written as

$$\begin{bmatrix} A + A^T & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} b \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ B \end{bmatrix},$$

and so

$$\begin{bmatrix} b \\ \lambda \end{bmatrix} = \begin{bmatrix} A + A^T & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ B \end{bmatrix},$$

assuming that the block matrix is invertible. (It is, for our problem instance.)

Alternatively, we can solve the optimality conditions by block elimination. First we consider $\nabla_b L(b,\lambda) = 0$, which gives $b = -\lambda(A + A^T)^{-1}\mathbf{1}$. The Lagrange multiplier λ is chosen to satisfy $\nabla_\lambda L(b,\lambda) = \mathbf{1}^T b - B = 0$. This gives

$$\mathbf{1}^T b = -\lambda \mathbf{1}^T (A + A^T)^{-1}\mathbf{1} = B,$$

so $\lambda = -B/\mathbf{1}^T(A + A^T)^{-1}\mathbf{1}$, and we get

$$b = \frac{B}{\mathbf{1}^T(A + A^T)^{-1}\mathbf{1}}(A + A^T)^{-1}\mathbf{1}.$$

The associated transaction cost is

$$\begin{aligned}(1/2)b^T(A + A^T)b &= \frac{B^2}{2\left(\mathbf{1}^T(A + A^T)^{-1}\mathbf{1}\right)^2}\mathbf{1}^T(A + A^T)^{-1}(A + A^T)(A + A^T)^{-1}\mathbf{1} \\ &= \frac{B^2}{2\left(\mathbf{1}^T(A + A^T)^{-1}\mathbf{1}\right)}.\end{aligned}$$

Another way of solving the optimization problem is to rearrange it into a general norm minimization problem with equality constraints, and refer directly to the solution given in lecture slides 8-13 to 8-15. To do this, we note that $A + A^T$ is positive definite, so we can find a symmetric matrix $F \in \mathbf{R}^{T\times T}$ for which $F^2 = A + A^T$. (Indeed, we can take $F = (A + A^T)^{1/2}$.) This gives the optimization problem

$$\begin{array}{ll}\text{minimize} & (1/2)\|Fb\|^2 \\ \text{subject to} & \mathbf{1}^T b = B.\end{array}$$

The solution then follows from lecture 8, and is, of course, exactly the same as the one given above.

The following Matlab code solves the problem.

```
% problem instance
B = 10000; T = 10; pbar = 10; theta = 0.8; alpha = 0.00015;
% generate A matrix
A = zeros(T,T);
for i = 1:T for j = 1:i A(i,j) = alpha*theta^(i-j); end; end;
% check that A is positive definite
min(eig(A+A'))
% nominal cost (evenly spread purchases)
cnom = ((B/T)^2)*ones(T,1)'*A*ones(T,1);
% one period cost (all shares purchased in the first period)
conep = B^2*A(1,1);
% optimal cost
blam = [A+A',ones(T,1);ones(T,1)',0]\[zeros(T,1);B];
bopt = blam(1:T);
copt = bopt'*A*bopt;
```

For our problem instance, the transaction cost incurred if all the shares are purchased in any single period (including the first) is 15000. If the purchases are evenly spread (1000 per period), we incur a transaction cost of 4822.12. If we employ the optimal strategy, we have a transaction cost of 4688.35.

5. *Angle between two subspaces.* The angle between two nonzero vectors v and w in $\mathbf{R}^n$ is defined as

$$\angle(v,w) = \cos^{-1}\left(\frac{v^T w}{\|v\|\|w\|}\right),$$

where we take $\cos^{-1}(a)$ as being between 0 and π. We define the angle between a nonzero vector $v \in \mathbf{R}^n$ and a (nonzero) subspace $\mathcal{W} \subseteq \mathbf{R}^n$ as

$$\angle(v,\mathcal{W}) = \min_{w\in\mathcal{W},\, w\neq 0} \angle(v,w).$$

Thus, $\angle(v,\mathcal{W}) = 10^\circ$ means that the smallest angle between v and any vector in $\mathcal{W}$ is 10°. If $v \in \mathcal{W}$, we have $\angle(v,\mathcal{W}) = 0$.

Finally, we define the angle between two nonzero subspaces $\mathcal{V}$ and $\mathcal{W}$ as

$$\angle(\mathcal{V},\mathcal{W}) = \max\left\{\max_{v\in\mathcal{V},\, v\neq 0} \angle(v,\mathcal{W}), \max_{w\in\mathcal{W},\, w\neq 0} \angle(w,\mathcal{V})\right\}.$$

This angle is zero if and only if the two subspaces are equal. If $\angle(\mathcal{V},\mathcal{W}) = 10^\circ$, say, it means that either there is a vector in $\mathcal{V}$ whose minimum angle to any vector of $\mathcal{W}$ is 10°, or there is a vector in $\mathcal{W}$ whose minimum angle to any vector of $\mathcal{V}$ is 10°.

(a) Suppose you are given two matrices $A \in \mathbf{R}^{n\times r}$, $B \in \mathbf{R}^{n\times r}$, each of rank r. Let $\mathcal{V} = \mathrm{range}(A)$ and $\mathcal{W} = \mathrm{range}(B)$. Explain how you could find or compute $\angle(\mathcal{V},\mathcal{W})$. You can use any of the concepts in the class, *e.g.*, least-squares, QR factorization, pseudo-inverse, norm, SVD, Jordan form, etc.

(b) Carry out your method for the matrices found in `angsubdata.m`. Give the numerical value for $\angle(\mathrm{range}(A), \mathrm{range}(B))$.

Solution. We can write

$$\cos\angle(\mathcal{V},\mathcal{W}) = \min\left\{\min_{v\in\mathcal{V},v\neq 0} \cos\angle(v,\mathcal{W}), \min_{w\in\mathcal{W},\, w\neq 0} \cos\angle(w,\mathcal{V})\right\},$$

since $\cos(\theta)$ is strictly decreasing as θ varies between 0 and π.

Taking the first part of this expression, we have

$$\min_{v\in\mathcal{V},\, v\neq 0} \cos\angle(v,\mathcal{W}) = \min_{v\in\mathcal{V},\, v\neq 0} \max_{w\in\mathcal{W},\, w\neq 0} \frac{v^T w}{\|v\|\|w\|} = \min_{v\in\mathcal{V},\, \|v\|=1} \max_{w\in\mathcal{W},\, \|w\|=1} v^T w.$$

Let V and W be matrices whose columns form orthonormal bases for $\mathcal{V}$ and $\mathcal{W}$ (*i.e.*, $\mathrm{range}(V) = \mathcal{V}$, $\mathrm{range}(W) = \mathcal{W}$, $V^T V = I$, and $W^T W = I$). This implies

$$\|Vx\| = 1 \Leftrightarrow \|x\| = 1, \qquad \|Wx\| = 1 \Leftrightarrow \|x\| = 1.$$

Now we can write

$$\min_{v\in\mathcal{V},\, v\neq 0} \cos\angle(v,\mathcal{W}) = \min_{\|y\|=1} \max_{\|x\|=1} y^T V^T W x = \min_{\|y\|=1} \|W^T V y\| = \sigma_{\min}(W^T V).$$

Similarly, we have

$$\min_{w\in\mathcal{W},\, w\neq 0} \cos\angle(w,\mathcal{V}) = \sigma_{\min}(V^TW) = \sigma_{\min}(W^TV) = \min_{v\in\mathcal{V},\, v\neq 0} \cos\angle(v,\mathcal{W}).$$

Therefore,

$$\begin{aligned}\cos\angle(\mathcal{V},\mathcal{W}) &= \min\left\{\min_{v\in\mathcal{V},\, v\neq 0} \cos\angle(v,\mathcal{W}), \min_{w\in\mathcal{W},\, w\neq 0} \cos\angle(w,\mathcal{V})\right\} \\ &= \sigma_{\min}(W^TV) = \sigma_{\min}(V^TW),\end{aligned}$$

and so,

$$\angle(\mathcal{V},\mathcal{W}) = \cos^{-1}\left(\sigma_{\min}(V^TW)\right).$$

Given two matrices A and B, we can find the angle between range(A) and range(B) in Matlab by writing,

```
angle=acos(min(svd(orth(A)'*orth(B))))
```

For the particular A and B given in `angsubdata.m`, $\angle(\text{range}(A), \text{range}(B)) = 72.83°$.

6. *Extracting the faintest signal.* An n-vector valued signal, $x(t) \in \mathbf{R}^n$, is defined for $t = 1, \ldots, T$. We'll refer to its ith component, $x_i(t)$, for $t = 1, \ldots, T$, as the ith scalar signal. The scalar signals $x_1, \ldots, x_{n-1}$ have an RMS value substantially larger than x_n. In other words, x_n is the faintest scalar signal. It is also the signal of interest for this problem. We will assume that the scalar signals $x_1, \ldots, x_n$ are unrelated to each other, and so are nearly uncorrelated (*i.e.*, nearly orthogonal).

   We aren't given the vector signal $x(t)$, but we are given a linear transformation of it,

   $$y(t) = Ax(t), \quad t = 1, \ldots, T,$$

   where $A \in \mathbf{R}^{n \times n}$ is invertible. If we knew A, we could easily recover the original signal (and therefore also the faintest scalar signal $x_n(t)$), using $x(t) = A^{-1}y(t)$, $t = 1, \ldots, T$. But, sadly, we don't know A.

   Here is a heuristic method for guessing $x_n(t)$. We will form our estimate as

   $$\hat{x}_n(t) = w^T y(t), \quad t = 1, \ldots, T,$$

   where $w \in \mathbf{R}^n$ is a vector of weights. Note that if w were chosen so that $w^T A = \alpha e_n^T$, with $\alpha \neq 0$ a constant, then we would have $\hat{x}_n(t) = \alpha x_n(t)$, *i.e.*, a perfect reconstruction except for the scale factor α.

   Now, the important part of our heuristic: we choose w to minimize the RMS value of $\hat{x}_n$, subject to $\|w\| = 1$. *Very roughly*, one idea behind the heuristic is that, in general, $w^T y$ is a linear combination of the scalar signals $x_1, \ldots, x_n$. If the linear combination has a small norm, that's because the linear combination is 'rich in x_n', and has only a small amount of energy contributed by $x_1, \ldots, x_{n-1}$. That, in fact, is exactly what we want. In any case, you don't need to worry about why the heuristic works (or doesn't work)—it's the method you are going to use in this problem.

   (a) Explain how to find a w that minimizes the RMS value of $\hat{x}_n$, using concepts from the class (*e.g.*, range, rank, least-squares, QR factorization, eigenvalues, singular values, and so on).

   (b) Carry out your method on the problem instance with $n = 4$, $T = 26000$, described in the Matlab file `faintestdata.m`. This file will define an $n \times T$ matrix Y, where the tth column of Y is the vector $y(t)$. The file will also define n and T. Submit your code, and give us the optimal weight vector $w \in \mathbf{R}^4$ you find, along with the associated RMS value of $\hat{x}_n$.

   The following is not needed to solve the problem. The signals are actually audio tracks, each 3.25 seconds long and sampled at 8 kHz. The Matlab file `faintestaudio.m` contains commands to generate wave files of the linear combinations $y_1, \ldots, y_4$, and a wave file of your estimate $\hat{x}_n$. You are welcome to generate and listen to these files.

Solution. First, we pack $y(t)$ into an $n \times T$ matrix Y, where the tth column of Y is the vector $y(t)$. Then the RMS value of our estimate $\hat{x}_n(t)$ is given by $(1/\sqrt{T})\|w^T Y\|$. This expression is minimized over unit-length w when w is a left singular vector corresponding to the smallest singular value of Y.

The RMS value of our estimate $\hat{x}_n(t)$, with this choice of w, will be $(1/\sqrt{T})\sigma_{\min}(Y)$.

In Matlab, this is simple. The code below shows a solution. Note the use of the flag `'econ'` to make the problem soluble.

```
[U, E, V] = svd(Y, 'econ');
w = U(:,4)
xhatn = w'*Y;
sqrt(mean(xhatn.^2))
```

The optimal RMS value is 0.0061, when $w = \pm[-0.35\ 0.37\ 0.58\ 0.63]^T$.

7. *Some true-false questions.* In the following statements, $A \in \mathbf{R}^{n\times n}$, $\sigma_{\min}$ refers to σ_n (the nth largest singular value), and κ refers to the condition number. Tell us whether each statement is true or false. 'True' means that the statement holds for any matrix $A \in \mathbf{R}^{n\times n}$, for any n. 'False' means that the statement is not true. The only answers we will read are 'True', 'False', and 'My attorney has advised me to not answer this question at this time'. (This last choice will receive partial credit.) If you write anything else, you will receive no credit for that statement. In particular, do not write justification for any answer, or provide any counter-examples.

   (a) $\|e^A\| \leq e^{\|A\|}$.

   (b) $\sigma_{\min}(e^A) \geq e^{\sigma_{\min}(A)}$.

   (c) $\kappa(e^A) \leq e^{\kappa(A)}$.

   (d) $\kappa(e^A) \leq e^{2\|A\|}$.

   (e) $\mathbf{Rank}(e^A) \geq \mathbf{Rank}(A)$.

   (f) $\mathbf{Rank}(e^A - I) \leq \mathbf{Rank}(A)$.

   Solution.

   (a) $\|e^A\| \leq e^{\|A\|}$. This one is **true**. To see this we use the power series expansion:

$$\begin{aligned} \|e^A\| &= \|I + A + (1/2!)A^2 + \cdots\| \\ &\leq \|I\| + \|A\| + \|(1/2!)A^2\| + \cdots \\ &= 1 + \|A\| + (1/2!)\|A\|^2 + \cdots \\ &= e^{\|A\|}. \end{aligned}$$

   (b) $\sigma_{\min}(e^A) \geq e^{\sigma_{\min}(A)}$. This is **false**. To see this we take

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

   so that

$$e^A = \begin{bmatrix} e & 0 \\ 0 & 1/e \end{bmatrix}.$$

   So $\sigma_{\min}(e^A) = 1/e < e = e^{\sigma_{\min}(A)}$.

   (c) $\kappa(e^A) \leq e^{\kappa(A)}$. This is **false**. We can take the same counterexample as part (b). This gives

$$\kappa(e^A) = e^2 > e = e^{\kappa(A)}.$$

   (d) $\kappa(e^A) \leq e^{2\|A\|}$. This one is **true**. To see why, we note that $\|e^A\| \leq e^{\|A\|}$, by part (a). We also have

$$\frac{1}{\sigma_{\min}(e^A)} = \left\|(e^A)^{-1}\right\| = \|e^{-A}\| \leq e^{\|-A\|} = e^{\|A\|}.$$

It follows that $\sigma_{\min}(e^A) \geq e^{-\|A\|}$. Therefore we have

$$\kappa(e^A) = \|e^A\|\|(e^A)^{-1}\| \leq e^{2\|A\|}.$$

(e) $\mathbf{Rank}(e^A) \geq \mathbf{Rank}(A)$. This is **true**. In fact, e^A is nonsingular, no matter what A is, so it has rank n.

(f) $\mathbf{Rank}(e^A - I) \leq \mathbf{Rank}(A)$. This is **true**. If $Av = 0$, then $A^k v = 0$ for any $k \geq 1$. it follows that $(e^A - I)v = 0$, since $e^A - I$ has a power series that has no constant term. Thus, we have $\mathcal{N}(A) \subseteq \mathcal{N}(e^A - I)$, and so $\dim \mathcal{N}(A) \leq \dim \mathcal{N}(e^A - I)$. We write this as

$$n - \mathbf{Rank}(A) \leq n - \mathbf{Rank}(e^A - I),$$

which gives the given inequality.