EE364a Convex Optimization I Prof. S. Boyd
March 14–15 or March 15–16, 2008.

Final exam

You may use any books, notes, or computer programs (*e.g.*, Matlab, `cvx`), but you may not discuss the exam with anyone until March 18, after everyone has taken the exam. The only exception is that you can ask us for clarification, via the course staff email address. We've tried pretty hard to make the exam unambiguous and clear, so we're unlikely to say much.

Please make a copy of your exam before handing it in.

Please attach the cover page to the front of your exam. Assemble your solutions in order (problem 1, problem 2, problem 3, ...), starting a new page for each problem. Put everything associated with each problem (*e.g.*, text, code, plots) together; do not attach code of plots at the end of the final.

We will deduct points from long needlessly complex solutions, even if they are correct. Our solutions are not long, so if you find that your solution to a problem goes on and on for many pages, you should try to figure out a simpler one. We expect neat, legible exams from everyone, including those enrolled Cr/N.

When a problem involves computation you must give all of the following: a clear discussion and justification of exactly what you did, the Matlab (or other) source code that produces the result, and the final numerical results or plots.
To download Matlab files containing problem data, you'll have to type the whole URL given in the problem into your browser; there are no links on the course web page pointing to these files. To get a file called `filename.m`, for example, you would retrieve

`http://www.stanford.edu/class/ee364a/final-data/filename.m`

with your browser.

Please respect the honor code. Although we allow you to work on homework assignments in small groups, you cannot discuss the final with anyone, at least until everyone has taken it.

All problems have equal weight. Some are easier than they might appear at first glance.

Download the most recent version of `cvx`.

Be sure to check your email often during the exam, just in case we need to send out an important announcement.

1. *Optimal investment to fund an expense stream.* An organization (such as a municipality) knows its operating expenses over the next T periods, denoted $E_1, \ldots, E_T$. (Normally these are positive; but we can have negative E_t, which corresponds to income.) These expenses will be funded by a combination of investment income, from a mixture of bonds purchased at $t = 0$, and a cash account.

   The bonds generate investment income, denoted $I_1, \ldots, I_T$. The cash balance is denoted $B_0, \ldots, B_T$, where $B_0 \geq 0$ is the amount of the initial deposit into the cash account. We can have $B_t < 0$ for $t = 1, \ldots, T$, which represents borrowing.

   After paying for the expenses using investment income and cash, in period t, we are left with $B_t - E_t + I_t$ in cash. If this amount is positive, it earns interest at the rate $r_+ > 0$; if it is negative, we must pay interest at rate r_-, where $r_- \geq r_+$. Thus the expenses, investment income, and cash balances are linked as follows:

$$B_{t+1} = \begin{cases} (1 + r_+)(B_t - E_t + I_t) & B_t - E_t + I_t \geq 0 \\ (1 + r_-)(B_t - E_t + I_t) & B_t - E_t + I_t < 0, \end{cases}$$

   for $t = 1, \ldots, T - 1$. We take $B_1 = (1 + r_+)B_0$, and we require that $B_T - E_T + I_T = 0$, which means the final cash balance, plus income, exactly covers the final expense.

   The initial investment will be a mixture of bonds, labeled $1, \ldots, n$. Bond i has a price $P_i > 0$, a coupon payment $C_i > 0$, and a maturity $M_i \in \{1, \ldots, T\}$. Bond i generates an income stream given by

$$a_t^{(i)} = \begin{cases} C_i & t < M_i \\ C_i + 1 & t = M_i \\ 0 & t > M_i, \end{cases}$$

   for $t = 1, \ldots, T$. If x_i is the number of units of bond i purchased (at $t = 0$), the total investment cash flow is

$$I_t = x_1 a_t^{(1)} + \cdots + x_n a_t^{(n)}, \quad t = 1, \ldots, T.$$

   We will require $x_i \geq 0$. (The x_i can be fractional; they do not need to be integers.)

   The total initial investment required to purchase the bonds, and fund the initial cash balance at $t = 0$, is $x_1 P_1 + \cdots + x_n P_n + B_0$.

   (a) Explain how to choose x and B_0 to minimize the total initial investment required to fund the expense stream.

   (b) Solve the problem instance given in `opt_funding_data.m`. Give optimal values of x and B_0. Give the optimal total initial investment, and compare it to the initial investment required if no bonds were purchased (which would mean that all the expenses were funded from the cash account). Plot the cash balance (versus period) with optimal bond investment, and with no bond investment.

2. *Utility versus latency trade-off in a network.* We consider a network with m edges, labeled $1, \dots, m$, and n flows, labeled $1, \dots, n$. Each flow has an associated nonnegative flow rate f_j; each edge or link has an associated positive capacity c_i. Each flow passes over a fixed set of links (its route); the total traffic t_i on link i is the sum of the flow rates over all flows that pass through link i. The flow routes are described by a routing matrix $R \in \mathbf{R}^{m \times n}$, defined as

$$R_{ij} = \begin{cases} 1 & \text{flow } j \text{ passes through link } i \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the vector of link traffic, $t \in \mathbf{R}^m$, is given by $t = Rf$. The link capacity constraint can be expressed as $Rf \preceq c$. The (logarithmic) network utility is defined as $U(f) = \sum_{j=1}^n \log f_j$.

The (average queuing) delay on link i is given by

$$d_i = \frac{1}{c_i - t_i}$$

(multiplied by a constant, that doesn't matter to us). We take $d_i = \infty$ for $t_i = c_i$. The delay or latency for flow j, denoted l_j, is the sum of the link delays over all links that flow j passes through. We define the maximum flow latency as

$$L = \max\{l_1, \dots, l_n\}.$$

We are given R and c; we are to choose f.

(a) How would you find the flow rates that maximize the utility U, ignoring flow latency? (In particular, we allow $L = \infty$.) We'll refer to this maximum achievable utility as $U^{\max}$.

(b) How would you find the flow rates that minimize the maximum flow latency L, ignoring utility? (In particular, we allow $U = -\infty$.) We'll refer to this minimum achievable latency as $L^{\min}$.

(c) Explain how to find the optimal trade-off between utility U (which we want to maximize) and latency L (which we want to minimize).

(d) Find $U^{\max}$, $L^{\min}$, and plot the optimal trade-off of utility versus latency for the network with data given in `net_util_data.m`, showing $L^{\min}$ and $U^{\max}$ on the same plot. Your plot should cover the range from $L = 1.1L^{\min}$ to $L = 11L^{\min}$. Plot U vertically, on a linear scale, and L horizontally, using a log scale.

Note. For parts (a), (b), and (c), your answer can involve solving one or more convex optimization problems. But if there is a simpler solution, you should say so.

3. *Optimal design of a tensile structure.* A tensile structure is modeled as a set of n masses in $\mathbf{R}^2$, some of which are fixed, connected by a set of N springs. The masses are in equilibrium, with spring forces, connection forces for the fixed masses, and gravity balanced. (This equilibrium occurs when the position of the masses minimizes the total energy, defined below.)

   We let $(x_i, y_i) \in \mathbf{R}^2$ denote the position of mass i, and $m_i > 0$ its mass value. The first p masses are fixed, which means that $x_i = x_i^{\text{fixed}}$ and $y_i = y_i^{\text{fixed}}$, for $i = 1, \ldots, p$. The gravitational potential energy of mass i is gm_iy_i, where $g \approx 9.8$ is the gravitational acceleration.

   Suppose spring j connects masses r and s. Its elastic potential energy is

$$(1/2)k_j\left((x_r - x_s)^2 + (y_r - y_s)^2\right),$$

   where $k_j \geq 0$ is the stiffness of spring j.

   To describe the topology, *i.e.*, which springs connect which masses, we will use the incidence matrx $A \in \mathbf{R}^{n \times N}$, defined as

$$A_{ij} = \begin{cases} 1 & \text{head of spring } j \text{ connects to mass } i \\ -1 & \text{tail of spring } j \text{ connects to mass } i \\ 0 & \text{otherwise.} \end{cases}$$

   Here we arbitrarily choose a head and tail for each spring, but in fact the springs are completely symmetric, and the choice can be reversed without any effect. (Hopefully you will discover why it is convenient to use the incidence matrix A to specify the topology of the system.)

   The total energy is the sum of the gravitational energies, over all the masses, plus the sum of the elastic energies, over all springs. The equilibrium positions of the masses is the point that minimizes the total energy, subject to the constraints that the first p positions are fixed. (In the equilibrium positions, the total force on each mass is zero.) We let E_{min} denote the total energy of the system, in its equilibrium position. (We assume the energy is bounded below; this occurs if and only if each mass is connected, through some set of springs with positive stiffness, to a fixed mass.)

   The total energy E_{min} is a measure of the stiffness of the structure, with larger E_{min} corresponding to stiffer. (We can think of $E_{\text{min}} = -\infty$ as an infinitely unstiff structure; in this case, at least one mass is not even supported against gravity.)

   (a) Suppose we know the fixed positions $x_1^{\text{fixed}}, \ldots, x_p^{\text{fixed}}$, $y_1^{\text{fixed}}, \ldots, y_p^{\text{fixed}}$, the mass values $m_1, \ldots, m_n$, the spring topology A, and the constant g. You are to choose nonnegative $k_1, \ldots, k_N$, subject to a budget constraint $\mathbf{1}^T k = k_1 + \cdots + k_N = k^{\text{tot}}$, where k^{tot} is given. Your goal is to maximize E_{min}.

   Explain how to do this using convex optimization.

(b) Carry out your method for the problem data given in `tens_struct_data.m`. This file defines all the needed data, and also plots the equilibrium configuration when the stiffness is evenly distributed across the springs (*i.e.*, $k = (k^{\text{tot}}/N)\mathbf{1}$).

Report the optimal value of E_{min}. Plot the optimized equilibrium configuration, and compare it to the equilibrium configuration with evenly distributed stiffness. (The code for doing this is in the file `tens_struct_data.m`, but commented out.)

4. *Identifying a sparse linear dynamical system.* A linear dynamical system has the form

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad t = 1, \ldots, T-1,$$

where $x(t) \in \mathbf{R}^n$ is the state, $u(t) \in \mathbf{R}^m$ is the input signal, and $w(t) \in \mathbf{R}^n$ is the process noise, at time t. We assume the process noises are IID $\mathcal{N}(0, W)$, where $W \succ 0$ is the covariance matrix. The matrix $A \in \mathbf{R}^{n \times n}$ is called the dynamics matrix or the state transition matrix, and the matrix $B \in \mathbf{R}^{n \times m}$ is called the input matrix.

You are given accurate measurements of the state and input signal, *i.e.*, $x(1), \ldots, x(T)$, $u(1), \ldots, u(T-1)$, and W is known. Your job is to find a state transition matrix $\hat{A}$ and input matrix $\hat{B}$ from these data, that are plausible, and in addition are sparse, *i.e.*, have many zero entries. (The sparser the better.)

By doing this, you are effectively estimating the structure of the dynamical system, *i.e.*, you are determining which components of $x(t)$ and $u(t)$ affect which components of $x(t+1)$. In some applications, this structure might be more interesting than the actual values of the (nonzero) coefficients in $\hat{A}$ and $\hat{B}$.

By plausible, we mean that

$$\sum_{t=1}^{T-1} \left\| W^{-1/2} \left(x(t+1) - \hat{A}x(t) - \hat{B}u(t) \right) \right\|_2^2 \in n(T-1) \pm 2\sqrt{2n(T-1)},$$

where $a \pm b$ means the interval $[a-b, a+b]$. (You can just take this as our definition of plausible. But to explain this choice, we note that when $\hat{A} = A$ and $\hat{B} = B$, the left-hand side is χ^2, with $n(T-1)$ degrees of freedom, and so has mean $n(T-1)$ and standard deviation $\sqrt{2n(T-1)}$.)

(a) Describe a method for finding $\hat{A}$ and $\hat{B}$, based on convex optimization.

We are looking for a *very simple* method, that involves solving *one* convex optimization problem. (There are many extensions of this basic method, that would improve the simple method, *i.e.*, yield sparser $\hat{A}$ and $\hat{B}$ that are still plausible. We're not asking you to describe or implement any of these.)

(b) Carry out your method on the data found in `sparse_lds_data.m`. Give the values of $\hat{A}$ and $\hat{B}$ that you find, and verify that they are plausible.

In the data file, we give you the true values of A and B, so you can evaluate the performance of your method. (Needless to say, you are not allowed to use these values when forming $\hat{A}$ and $\hat{B}$.) Using these true values, give the number of false positives and false negatives in both $\hat{A}$ and $\hat{B}$. A false positive in $\hat{A}$, for example, is an entry that is nonzero, while the corresponding entry in A is zero. A false negative is an entry of $\hat{A}$ that is zero, while the corresponding entry of A is nonzero. To judge whether an entry of $\hat{A}$ (or $\hat{B}$) is nonzero, you can use the test $|\hat{A}_{ij}| \geq 0.01$ (or $|\hat{B}_{ij}| \geq 0.01$).

5. *Minimum energy processor speed scheduling.* A single processor can adjust its speed in each of T time periods, labeled $1, \ldots, T$. Its speed in period t will be denoted s_t, $t = 1, \ldots, T$. The speeds must lie between given (positive) minimum and maximum values, $S^{\min}$ and $S^{\max}$, respectively, and must satisfy a slew-rate limit, $|s_{t+1} - s_t| \leq R$, $t = 1, \ldots, T-1$. (That is, R is the maximum allowed period-to-period change in speed.) The energy consumed by the processor in period t is given by $\phi(s_t)$, where $\phi : \mathbf{R} \to \mathbf{R}$ is increasing and convex. The total energy consumed over all the periods is $E = \sum_{t=1}^{T} \phi(s_t)$.

   The processor must handle n jobs, labeled $1, \ldots, n$. Each job has an availability time $A_i \in \{1, \ldots, T\}$, and a deadline $D_i \in \{1, \ldots, T\}$, with $D_i \geq A_i$. The processor cannot start work on job i until period $t = A_i$, and must complete the job by the end of period D_i. Job i involves a (nonnegative) total work W_i. You can assume that in each time period, there is at least one job available, *i.e.*, for each t, there is at least one i with $A_i \leq t$ and $D_i \geq t$.

   In period t, the processor allocates its effort across the n jobs as θ_t, where $\mathbf{1}^T \theta_t = 1$, $\theta_t \succeq 0$. Here θ_{ti} (the ith component of θ_t) gives the fraction of the processor effort devoted to job i in period t. Respecting the availability and deadline constraints requires that $\theta_{ti} = 0$ for $t < A_i$ or $t > D_i$. To complete the jobs we must have

$$\sum_{t=A_i}^{D_i} \theta_{ti} s_t \geq W_i, \quad i = 1, \ldots, n.$$

   (a) Formulate the problem of choosing the speeds $s_1, \ldots, s_T$, and the allocations $\theta_1, \ldots, \theta_T$, in order to minimize the total energy E, as a convex optimization problem. The problem data are $S^{\min}$, $S^{\max}$, R, ϕ, and the job data, A_i, D_i, W_i, $i = 1, \ldots, n$. Be sure to justify any change of variables, or introduction of new variables, that you use in your formulation.

   (b) Carry out your method on the problem instance described in `proc_sched_data.m`, with quadratic energy function $\phi(s_t) = \alpha + \beta s_t + \gamma s_t^2$. (The parameters α, β, and γ are given in the data file.) Executing this file will also give a plot showing the availability times and deadlines for the jobs.

       Give the energy obtained by your speed profile and allocations. Plot these using the command `bar((s*ones(1,n)).*theta,1,'stacked')`, where s is the $T \times 1$ vector of speeds, and θ is the $T \times n$ matrix of allocations with components θ_{ti}. This will show, at each time period, how much effective speed is allocated to each job. The top of the plot will show the speed s_t. (You don't need to turn in a color version of this plot; B&W is fine.)

6. *Planning production with uncertain demand.* You must order (nonnegative) amounts $r_1, \ldots, r_m$ of raw materials, which are needed to manufacture (nonnegative) quantities $q_1, \ldots, q_n$ of n different products. To manufacture one unit of product j requires at least A_{ij} units of raw material i, so we must have $r \succeq Aq$. (We will assume that A_{ij} are nonnegative.) The per-unit cost of the raw materials is given by $c \in \mathbf{R}^m_+$, so the total raw material cost is $c^T r$.

   The (nonnegative) demand for product j is denoted d_j; the number of units of product j sold is $s_j = \min\{q_j, d_j\}$. (When $q_j > d_j$, $q_j - d_j$ is the amount of product j produced, but not sold; when $d_j > q_j$, $d_j - q_j$ is the amount of unmet demand.) The revenue from selling the products is $p^T s$, where $p \in \mathbf{R}^n_+$ is the vector of product prices. The profit is $p^T s - c^T r$. (Both d and q are real vectors; their entries need not be integers.)

   You are given A, c, and p. The product demand, however, is not known. Instead, a set of K possible demand vectors, $d^{(1)}, \ldots, d^{(K)}$, with associated probabilities $\pi_1, \ldots, \pi_K$, is given. (These satisfy $\mathbf{1}^T \pi = 1$, $\pi \succeq 0$.)

   You will explore two different optimization problems that arise in choosing r and q (the variables).

   I. Choose r and q ahead of time. You must choose r and q, knowing only the data listed above. (In other words, you must order the raw materials, and commit to producing the chosen quantities of products, before you know the product demand.) The objective is to maximize the expected profit.

   II. Choose r ahead of time, and q after d is known. You must choose r, knowing only the data listed above. Some time after you have chosen r, the demand will become known to you. This means that you will find out which of the K demand vectors is the true demand. Once you know this, you must choose the quantities to be manufactured. (In other words, you must order the raw materials before the product demand is known; but you can choose the mix of products to manufacture after you have learned the true product demand.) The objective is to maximize the expected profit.

   (a) Explain how to formulate each of these problems as a convex optimization problem. Clearly state what the variables are in the problem, what the constraints are, and describe the roles of any auxiliary variables or constraints you introduce.

   (b) Carry out the methods from part (a) on the problem instance with numerical data given in `planning_data.m`. This file will define `A`, `D`, `K`, `c`, `m`, `n`, `p` and `pi`. The K columns of D are the possible demand vectors. For both of the problems described above, give the optimal value of r, and the expected profit.

7. *Dual of exponential cone.* The exponential cone $K_{\text{exp}} \subseteq \mathbf{R}^3$ is defined as

$$K_{\text{exp}} = \{(x, y, z) \mid y > 0,\ ye^{x/y} \leq z\}.$$

Express the dual cone in the form

$$K_{\text{exp}}^* = \{(u, v, w) \mid \ldots \ \textit{your conditions here} \ \ldots\}.$$

Your conditions must be short (certainly no more than one line) and cannot involve any variables other than u, v, and w.

We are not worried here about the fine details of what happens on the boundaries of these cones, so you really needn't worry about it. But we make some comments here for those who do care about such things.

The cone K_{exp} as defined above is not closed. To obtain its closure, we need to add the points

$$\{(x, y, z) \mid x \leq 0,\ y = 0,\ z \geq 0\}.$$

(This makes no difference, since the dual of a cone is equal to the dual of its closure.)