6.079/6.975 S. Boyd & P. Parrilo
December 10–11, 2009.

Final exam

This is a 24 hour take-home final exam. Please turn it in to Professor Stephen Boyd, (Stata Center), on Friday December 11, at 5PM (or before).

You may use any books, notes, or computer programs (*e.g.*, Matlab, CVX), but you may not discuss the exam with anyone until December 11 after 5PM, after everyone has taken the exam. The only exception is that you can ask us for clarification, via email. Please address your emails to *both professors and the TA*.

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

When a problem involves computation you must give all of the following: a clear discussion and justification of exactly what you did, the Matlab source code that produces the result, and the final numerical results or plots.

Matlab files containing problem data are available on Stellar.

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

Be sure to check your email and the course web site on Stellar often during the exam, just in case we need to send out an important announcement.

And one technical comment. For problems that require you to work out a numerical solution, you are welcome to use a solution method that involves solving more than just a single convex optimization problem. (Of course, only when this is necessary.)

1. *Optimal generator dispatch.* In the *generator dispatch problem*, we schedule the electrical output power of a set of generators over some time interval, to minimize the total cost of generation while exactly meeting the (assumed known) electrical demand. One challenge in this problem is that the generators have dynamic constraints, which couple their output powers over time. For example, every generator has a maximum rate at which its power can be increased or decreased.

   We label the generators $i = 1, \ldots, n$, and the time periods $t = 1, \ldots, T$. We let $p_{i,t}$ denote the (nonnegative) power output of generator i at time interval t. The (positive) electrical demand in period t is d_t. The total generated power in each period must equal the demand:

$$\sum_{i=1}^{n} p_{i,t} = d_t, \quad t = 1, \ldots, T.$$

   Each generator has a minimum and maximum allowed output power:

$$P_i^{\min} \leq p_{i,t} \leq P_i^{\max}, \quad i = 1, \ldots, n, \quad t = 1, \ldots, T.$$

   The cost of operating generator i at power output u is $\phi_i(u)$, where ϕ_i is an increasing strictly convex function. (Assuming the cost is mostly fuel cost, convexity of ϕ_i says that the thermal efficiency of the generator decreases as its output power increases.) We will assume these cost functions are quadratic: $\phi_i(u) = \alpha_i u + \beta_i u^2$, with α_i and β_i positive.

   Each generator has a maximum ramp-rate, which limits the amount its power output can change over one time period:

$$|p_{i,t+1} - p_{i,t}| \leq R_i, \quad i = 1, \ldots, n, \quad t = 1, \ldots, T-1.$$

   In addition, changing the power output of generator i from u_t to u_{t+1} incurs an additional cost $\psi_i(u_{t+1} - u_t)$, where ψ_i is a convex function. (This cost can be a real one, due to increased fuel use during a change of power, or a fictitious one that accounts for the increased maintenance cost or decreased lifetime caused by frequent or large changes in power output.) We will use the power change cost functions $\psi_i(v) = \gamma_i |v|$, where γ_i are positive.

   Power plants with large capacity (*i.e.*, $P_i^{\max}$) are typically more efficient (*i.e.*, have smaller α_i, β_i), but have smaller ramp-rate limits, and higher costs associated with changing power levels. Small gas-turbine plants ('peakers') are less efficient, have less capacity, but their power levels can be rapidly changed.

   The total cost of operating the generators is

$$C = \sum_{i=1}^{n} \sum_{t=1}^{T} \phi_i(p_{i,t}) + \sum_{i=1}^{n} \sum_{t=1}^{T-1} \psi_i(p_{i,t+1} - p_{i,t}).$$

   Choosing the generator output schedules to minimize C, while respecting the constraints described above, is a convex optimization problem. The problem data are d_t

(the demands), the generator power limits $P_i^{\min}$ and $P_i^{\max}$, the ramp-rate limits R_i, and the cost function parameters α_i, β_i, and γ_i. We will assume that problem is feasible, and that $p_{i,t}^\star$ are the (unique) optimal output powers.

(a) *Price decomposition.* Show that there are power prices $Q_1, \ldots, Q_T$ for which the following holds: For each i, $p_{i,t}^\star$ solves the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{t=1}^T \left(\phi_i(p_{i,t}) - Q_t p_{i,t}\right) + \sum_{t=1}^{T-1} \psi_i(p_{i,t+1} - p_{i,t}) \\ \text{subject to} & P_i^{\min} \leq p_{i,t} \leq P_i^{\max}, \quad t = 1, \ldots, T \\ & |p_{i,t+1} - p_{i,t}| \leq R_i, \quad t = 1, \ldots, T-1. \end{array}$$

The objective here is the portion of the objective for generator i, minus the revenue generated by the sale of power at the prices Q_t. Note that this problem involves *only* generator i; it can be solved independently of the other generators (once the prices are known). How would you find the prices Q_t?

You do not have to give a full formal proof; but you must explain your argument fully. You are welcome to use results from the text book.

(b) Solve the generator dispatch problem with the data given in `gen_dispatch_data.m`, which gives (fake, but not unreasonable) demand data for 2 days, at 15 minute intervals. This file includes code to plot the demand, optimal generator powers, and prices. (You must replace these variables with their correct values.) Comment on anything you see in your solution that might at first seem odd. Using the prices found, solve the problems in part (a) for the generators separately, to be sure they give the optimal powers (up to some small numerical errors).

Remark. While beyond the scope of this course, we mention that there are very simple price update mechanisms that adjust the prices in such a way that when the generators independently schedule themselves using the prices (as described above), we end up with the total power generated in each period matching the demand, *i.e.*, the optimal solution of the whole (coupled) problem. This gives a decentralized method for generator dispatch.

2. *Internal rate of return for cash streams with a single initial investment.* We use the notation of example 3.34 in the textbook. Let $x \in \mathbf{R}^{n+1}$ be a cash flow over n periods, with x indexed from 0 to n, where the index denotes period number. We assume that $x_0 < 0$, $x_j \geq 0$ for $j = 1, \ldots, n$, and $x_0 + \cdots + x_n > 0$. This means that there is an initial positive investment; thereafter, only payments are made, with the total of the payments exceeding the initial investment. (In the more general setting of example 3.34, we allow additional investments to be made after the initial investment.)

   (a) Show that $\mathrm{IRR}(x)$ is quasilinear in this case.

   (b) *Blending initial investment only streams.* Use the result in part (a) to show the following. Let $x^{(i)} \in \mathbf{R}^{n+1}$, $i = 1, \ldots, k$, be a set of k cash flows over n periods, each of which satisfies the conditions above. Let $w \in \mathbf{R}^k_+$, with $\mathbf{1}^T w = 1$, and consider the blended cash flow given by $x = w_1 x^{(1)} + \cdots + w_k x^{(k)}$. (We can think of this as investing a fraction w_i in cash flow i.) Show that $\mathrm{IRR}(x) \leq \max_i \mathrm{IRR}(x^{(i)})$. Thus, blending a set of cash flows (with initial investment only) will not improve the IRR over the best individual IRR of the cash flows.

3. *Infimal convolution.* Let $f_1, \ldots, f_m$ be convex functions on $\mathbf{R}^n$. Their *infimal convolution*, denoted $g = f_1 \diamond \cdots \diamond f_m$ (several other notations are also used), is defined as

$$g(x) = \inf\{f_1(x_1) + \cdots + f_m(x_m) \mid x_1 + \cdots + x_m = x\},$$

   with the natural domain (*i.e.*, defined by $g(x) < \infty$). In one simple interpretation, $f_i(x_i)$ is the cost for the ith firm to produce a mix of products given by x_i; $g(x)$ is then the optimal cost obtained if the firms can freely exchange products to produce, all together, the mix given by x. (The name 'convolution' presumably comes from the observation that if we replace the sum above with the product, and the infimum above with integration, then we obtain the normal convolution.)

   (a) Show that g is convex.

   (b) Show that $g^* = f_1^* + \cdots + f_m^*$. In other words, the conjugate of the infimal convolution is the sum of the conjugates.

   (c) Verify the identity in part (b) for the specific case of two strictly convex quadratic functions, $f_i(x) = (1/2)x^T P_i x$, with $P_i \in \mathbf{S}^n_{++}$, $i = 1, 2$.

   Hint: Depending on how you work out the conjugates, you might find the matrix identity $(X + Y)^{-1} Y = X^{-1}(X^{-1} + Y^{-1})^{-1}$ useful.

4. *Robust minimum volume covering ellipsoid.* Suppose z is a point in $\mathbf{R}^n$ and $\mathcal{E}$ is an ellipsoid in $\mathbf{R}^n$ with center c. The *Mahalanobis distance* of the point to the ellipsoid center is defined as

$$M(z, \mathcal{E}) = \inf\{t \geq 0 \mid z \in c + t(\mathcal{E} - c)\},$$

which is the factor by which we need to scale the ellipsoid about its center so that z is on its boundary. We have $z \in \mathcal{E}$ if and only if $M(z, \mathcal{E}) \leq 1$. We can use $(M(z, \mathcal{E}) - 1)_+$ as a measure of the Mahalanobis distance of the point z to the ellipsoid $\mathcal{E}$.

Now we can describe the problem. We are given m points $x_1, \ldots, x_m \in \mathbf{R}^n$. The goal is to find the optimal trade-off between the volume of the ellipsoid $\mathcal{E}$ and the total Mahalanobis distance of the points to the ellipsoid, *i.e.*,

$$\sum_{i=1}^{m} (M(z, \mathcal{E}) - 1)_+ .$$

Note that this can be considered a robust version of finding the smallest volume ellipsoid that covers a set of points, since here we allow one or more points to be outside the ellipsoid.

(a) Explain how to solve this problem. You must say clearly what your variables are, what problem you solve, and why the problem is convex.

(b) Carry out your method on the data given in `rob_min_vol_ellips_data.m`. Plot the optimal trade-off curve of ellipsoid volume versus total Mahalanobis distance. For some selected points on the trade-off curve, plot the ellipsoid and the points (which are in $\mathbf{R}^2$). We are only interested in the region of the curve where the ellipsoid volume is within a factor of ten (say) of the minimum volume ellipsoid that covers all the points.

Important. Depending on how you formulate the problem, you might encounter problems that are unbounded below, or where CVX encounters numerical difficulty. Just avoid these by appropriate choice of parameter.

Very important. If you use Matlab version 7.0 (which is filled with bugs) you might find that functions involving determinants don't work in CVX. If you use this version of Matlab, then you must download the file `blkdiag.m` on the course website and put it in your Matlab path before the default version (which has a bug).

5. *Fitting a vector field to given directions.* This problem concerns a vector field on $\mathbf{R}^n$, *i.e.*, a function $F : \mathbf{R}^n \to \mathbf{R}^n$. We are given the *direction* of the vector field at points $x^{(1)}, \ldots, x^{(N)} \in \mathbf{R}^n$,

$$q^{(i)} = \frac{1}{\|F(x^{(i)})\|_2} F(x^{(i)}), \quad i = 1, \ldots, N.$$

(These directions might be obtained, for example, from samples of trajectories of the differential equation $\dot{z} = F(z)$.) The goal is to fit these samples with a vector field of the form

$$\hat{F} = \alpha_1 F_1 + \cdots + \alpha_m F_m,$$

where $F_1, \ldots, F_m : \mathbf{R}^n \to \mathbf{R}^n$ are given (basis) functions, and $\alpha \in \mathbf{R}^m$ is a set of coefficients that we will choose.

We will measure the fit using the maximum angle error,

$$J = \max_{i=1,\ldots,N} \left| \angle(q^{(i)}, \hat{F}(x^{(i)})) \right|,$$

where $\angle(z, w) = \cos^{-1}((z^T w)/\|z\|_2 \|w\|_2)$ denotes the angle between nonzero vectors z and w. We are only interested in the case when J is smaller than $\pi/2$.

(a) Explain how to choose α so as to minimize J using convex optimization. Your method can involve solving multiple convex problems. Be sure to explain how you handle the constraints $\hat{F}(x^{(i)}) \neq 0$.

(b) Use your method to solve the problem instance with data given in `vfield_fit_data.m`, with an affine vector field fit, *i.e.*, $\hat{F}(z) = Az + b$. (The matrix A and vector b are the parameters α above.) Give your answer to the nearest degree, as in '$20^\circ < J^\star \leq 21^\circ$'.

This file also contains code that plots the vector field directions, and also (but commented out) the directions of the vector field fit, $\hat{F}(x^{(i)})/\|\hat{F}(x^{(i)})\|_2$. Create this plot, with your fitted vector field.

6. *Efficient solution of basic portfolio optimization problem.* This problem concerns the simplest possible portfolio optimization problem:

$$\begin{array}{ll} \text{maximize} & \mu^T w - (\lambda/2) w^T \Sigma w \\ \text{subject to} & \mathbf{1}^T w = 1, \end{array}$$

with variable $w \in \mathbf{R}^n$ (the normalized portfolio, with negative entries meaning short positions), and data μ (mean return), $\Sigma \in \mathbf{S}^n_{++}$ (return covariance), and $\lambda > 0$ (the risk aversion parameter). The return covariance has the factor form $\Sigma = FQF^T + D$, where $F \in \mathbf{R}^{n \times k}$ (with rank K) is the *factor loading matrix*, $Q \in \mathbf{S}^k_{++}$ is the factor covariance matrix, and D is a diagonal matrix with positive entries, called the *idiosyncratic risk* (since it describes the risk of each asset that is independent of the factors). This form for Σ is referred to as a 'k-factor risk model'. Some typical dimensions are $n = 2500$ (assets) and $k = 30$ (factors).

(a) What is the flop count for computing the optimal portfolio, if the low-rank plus diagonal structure of Σ is *not* exploited? You can assume that $\lambda = 1$ (which can be arranged by absorbing it into Σ).

(b) Explain how to compute the optimal portfolio more efficiently, and give the flop count for your method. You can assume that $k \ll n$. You do not have to give the best method; any method that has linear complexity in n is fine. You can assume that $\lambda = 1$.

Hints. You may want to introduce a new variable $y = F^T w$ (which is called the vector of factor exposures). You may want to work with the matrix

$$G = \begin{bmatrix} \mathbf{1} & F \\ 0 & -I \end{bmatrix} \in \mathbf{R}^{(n+k)\times(1+k)},$$

treating it as dense, ignoring the (little) exploitable structure in it.

(c) Carry out your method from part (b) on some randomly generated data with dimensions $n = 2500$, $k = 30$. For comparison (and as a check on your method), compute the optimal portfolio using the method of part (a) as well. Give the (approximate) CPU time for each method, using `tic` and `toc`. *Hints.* After you generate D and Q randomly, you might want to add a positive multiple of the identity to each, to avoid any issues related to poor conditioning. Also, to be able to invert a block diagonal matrix efficiently, you'll need to recast it as sparse.

(d) *Risk return trade-off curve.* Now suppose we want to compute the optimal portfolio for M values of the risk aversion parameter λ. Explain how to do this efficiently, and give the complexity in terms of M, n, and k. Compare to the complexity of using the method of part (b) M times. *Hint.* Show that the optimal portfolio is an affine function of $1/\lambda$.

7. *Optimizing the inertia matrix of a 2D mass distribution.* An object has density $\rho(z)$ at the point $z = (x, y) \in \mathbf{R}^2$, over some region $\mathcal{R} \subset \mathbf{R}^2$. Its mass $m \in \mathbf{R}$ and center of gravity $c \in \mathbf{R}^2$ are given by

$$m = \int_{\mathcal{R}} \rho(z)\, dxdy, \qquad c = \frac{1}{m}\int_{\mathcal{R}} \rho(z) z\, dxdy,$$

and its inertia matrix $M \in \mathbf{R}^{2\times 2}$ is

$$M = \int_{\mathcal{R}} \rho(z)(z-c)(z-c)^T\, dxdy.$$

(You do not need to know the mechanics interpretation of M to solve this problem, but here it is, for those interested. Suppose we rotate the mass distribution around a line passing through the center of gravity in the direction $q \in \mathbf{R}^2$ that lies in the plane where the mass distribution is, at angular rate ω. Then the total kinetic energy is $(\omega^2/2)q^T M q$.)

The goal is to choose the density ρ, subject to $0 \leq \rho(z) \leq \rho^{\max}$ for all $z \in \mathcal{R}$, and a fixed total mass $m = m^{\text{given}}$, in order to maximize $\lambda_{\min}(M)$.

To solve this problem numerically, we will discretize $\mathcal{R}$ into N pixels each of area a, with pixel i having constant density ρ_i and location (say, of its center) $z_i \in \mathbf{R}^2$. We will assume that the integrands above don't vary too much over the pixels, and from now on use instead the expressions

$$m = a\sum_{i=1}^N \rho_i, \quad c = \frac{a}{m}\sum_{i=1}^N \rho_i z_i, \quad M = a\sum_{i=1}^N \rho_i (z_i - c)(z_i - c)^T.$$

The problem below refers to these discretized expressions.

(a) Explain how to solve the problem using convex (or quasiconvex) optimization.

(b) Carry out your method on the problem instance with data in `inertia_dens_data.m`. This file includes code that plots a density. Give the optimal inertia matrix and its eigenvalues, and plot the optimal density.