EE364a: Convex Optimization I S. Boyd
March 13–14 or March 14–15, 2015

Final Exam

This is a 24 hour take-home final. Please turn it in at Bytes Cafe in the Packard building, 24 hours after you pick it up.

You may use any books, notes, or computer programs, but you may not discuss the exam with anyone until March 16, 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, or scan it, 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 or 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 source code that produces the result, and the final numerical results or plots.

Files containing problem data can be found in the usual place,

`http://www.stanford.edu/~boyd/cvxbook/cvxbook_additional_exercises/`

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 easy. Others, not so much.

Be sure you are using the most recent version of CVX, CVXPY, or Convex,jl. Check your email often during the exam, just in case we need to send out an important announcement.

Some problems involve applications. But you do not need to know *anything* about the problem area to solve the problem; the problem statement contains everything you need.

1. *Optimal evacuation planning.* We consider the problem of evacuating people from a dangerous area in a way that minimizes risk exposure. We model the area as a connected graph with n nodes and m edges; people can assemble or collect at the nodes, and travel between nodes (in either direction) over the edges. We let $q_t \in \mathbf{R}^n_+$ denote the vector of the numbers of people at the nodes, in time period t, for $t = 1, \ldots, T$, where T is the number of periods we consider. (We will consider the entries of q_t as real numbers, not integers.) The initial population distribution q_1 is given. The nodes have capacity constraints, given by $q_t \preceq Q$, where $Q \in \mathbf{R}^n_+$ is the vector of node capacities.

   We use the incidence matrix $A \in \mathbf{R}^{n\times m}$ to describe the graph. We assign an arbitrary reference direction to each edge, and take

$$A_{ij} = \begin{cases} +1 & \text{if edge } j \text{ enters node } i \\ -1 & \text{if edge } j \text{ exits node } i \\ 0 & \text{otherwise.} \end{cases}$$

   The population dynamics are given by $q_{t+1} = Af_t + q_t$, $t = 1, \ldots, T-1$ where $f_t \in \mathbf{R}^m$ is the vector of population movement (flow) across the edges, for $t = 1, \ldots, T-1$. A positive flow denotes movement in the direction of the edge; negative flow denotes population flow in the reverse direction. Each edge has a capacity, *i.e.*, $|f_t| \preceq F$, where $F \in \mathbf{R}^m_+$ is the vector of edge capacities, and $|f_t|$ denotes the elementwise absolute value of f_t.

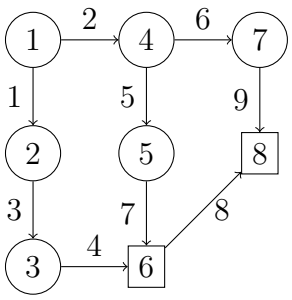
   An *evacuation plan* is a sequence $q_1, q_2, \ldots, q_T$ and $f_1, f_2, \ldots, f_{T-1}$ obeying the constraints above. The goal is to find an evacuation plan that minimizes the total risk exposure, defined as

$$R_{\text{tot}} = \sum_{t=1}^{T} \left(r^T q_t + s^T q_t^2\right) + \sum_{t=1}^{T-1} \left(\tilde{r}^T |f_t| + \tilde{s}^T f_t^2\right),$$

   where $r, s \in \mathbf{R}^n_+$ are given vectors of risk exposure coefficients associated with the nodes, and $\tilde{r}, \tilde{s} \in \mathbf{R}^m_+$ are given vectors of risk exposure coefficients associated with the edges. The notation q_t^2 and f_t^2 refers to elementwise squares of the vectors. Roughly speaking, the risk exposure is a quadratic function of the occupancy of a node, or the (absolute value of the) flow of people along an edge. The linear terms can be interpreted as the risk exposure per person; the quadratic terms can be interpreted as the additional risk associated with crowding.

   A subset of nodes have zero risk ($r_i = s_i = 0$), and are designated as *safe nodes*. The population is considered *evacuated* at time t if $r^T q_t + s^T q_t^2 = 0$. The *evacuation time* t_{evac} of an evacuation plan is the smallest such t. We will assume that T is sufficiently large and that the total capacity of the safe nodes exceeds the total initial population, so evacuation is possible.

   Use CVX* to find an optimal evacuation plan for the problem instance with data given in `opt_evac_data.*`. (We display the graph below, with safe nodes denoted as squares.)



Report the associated optimal risk exposure $R^\star_{\text{tot}}$. Plot the time period risk

$$R_t = r^T q_t + s^T q_t^2 + \tilde{r}^T |f_t| + \tilde{s}^T f_t^2$$

versus time. (For $t = T$, you can take the edge risk to be zero.) Plot the node occupancies q_t, and edge flows f_t versus time. Briefly comment on the results you see. Give the evacuation time t_{evac} (considering any $r^T q_t + s^T q_t^2 \leq 10^{-4}$ to be zero).

Hint. With CVXPY, use the ECOS solver with `p.solve(solver=cvxpy.ECOS)`.

2. *Convexity of products of powers.* This problem concerns the product of powers function $f : \mathbf{R}^n_{++} \to \mathbf{R}$ given by $f(x) = x_1^{\theta_1} \cdots x_n^{\theta_n}$, where $\theta \in \mathbf{R}^n$ is a vector of powers. We are interested in finding values of θ for which f is convex or concave. You already know a few, for example when $n = 2$ and $\theta = (2, -1)$, f is convex (the quadratic-over-linear function), and when $\theta = (1/n)\mathbf{1}$, f is concave (geometric mean). Of course, if $n = 1$, f is convex when $\theta \geq 1$ or $\theta \leq 0$, and concave when $0 \leq \theta \leq 1$.

   Show each of the statements below. We will not read long or complicated proofs, or ones that involve Hessians. We are looking for short, snappy ones, that (where possible) use composition rules, perspective, partial minimization, or other operations, together with known convex or concave functions, such as the ones listed in the previous paragraph. Feel free to use the results of earlier statements in later ones.

   (a) When $n = 2$, $\theta \succeq 0$, and $\mathbf{1}^T\theta = 1$, f is concave.

   (b) When $\theta \succeq 0$ and $\mathbf{1}^T\theta = 1$, f is concave. (This is the same as part (a), but here it is for general n.)

   (c) When $\theta \succeq 0$ and $\mathbf{1}^T\theta \leq 1$, f is concave.

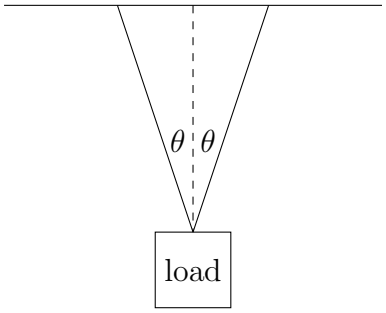
   (d) When $\theta \preceq 0$, f is convex.

   (e) When $\mathbf{1}^T\theta = 1$ and exactly *one* of the elements of θ is positive, f is convex.

   (f) When $\mathbf{1}^T\theta \geq 1$ and exactly *one* of the elements of θ is positive, f is convex.

   Remark. Parts (c), (d), and (f) exactly characterize the cases when f is either convex or concave. That is, if none of these conditions on θ hold, f is neither convex nor concave. Your teaching staff has, however, kindly refrained from asking you to show this.

3. *Minimum time maneuver for a crane.* A crane manipulates a load with mass $m > 0$ in two dimensions using two cables attached to the load. The cables maintain angles $\pm\theta$ with respect to vertical, as shown below.



The (scalar) tensions T^{left} and T^{right} in the two cables are independently controllable, from 0 up to a given maximum tension $T^{\max}$. The total force on the load is

$$F = T^{\text{left}} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} + T^{\text{right}} \begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} + mg,$$

where $g = (0, -9.8)$ is the acceleration due to gravity. The acceleration of the load is then F/m.

We approximate the motion of the load using

$$p_{i+1} = p_i + hv_i, \quad v_{i+1} = v_i + (h/m)F_i, \quad i = 1, 2, \ldots,$$

where $p_i \in \mathbf{R}^2$ is the position of the load, $v_i \in \mathbf{R}^2$ is the velocity of the load, and $F_i \in \mathbf{R}^2$ is the force on the load, at time $t = ih$. Here $h > 0$ is a small (given) time step.

The goal is to move the load, which is initially at rest at position p^{init} to the position p^{des}, also at rest, in minimum time. In other words, we seek the smallest k for which

$$p_1 = p^{\text{init}}, \quad p_k = p^{\text{des}}, \quad v_1 = v_k = (0, 0)$$

is possible, subject to the constraints described above.

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

(b) Carry out the method of part (a) for the problem instance with

$$m = 0.1, \quad \theta = 15^\circ, \quad T^{\max} = 2, \quad p^{\text{init}} = (0, 0), \quad p^{\text{des}} = (10, 2),$$

with time step $h = 0.1$. Report the minimum time $k^\star$. Plot the tensions versus time, and the load trajectory, *i.e.*, the points $p_1, \ldots, p_k$ in $\mathbf{R}^2$. Does the load move along the line segment between p^{init} and p^{des} (*i.e.*, the shortest path from p^{init} and p^{des})? Comment briefly.

4. *Portfolio rebalancing.* We consider the problem of rebalancing a portfolio of assets over multiple periods. We let $h_t \in \mathbf{R}^n$ denote the vector of our dollar value holdings in n assets, at the beginning of period t, for $t = 1, \ldots, T$, with negative entries meaning short positions. We will work with the portfolio weight vector, defined as $w_t = h_t/(\mathbf{1}^T h_t)$, where we assume that $\mathbf{1}^T h_t > 0$, *i.e.*, the total portfolio value is positive.

   The *target portfolio weight vector* $w^\star$ is defined as the solution of the problem

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

   where $w \in \mathbf{R}^n$ is the variable, μ is the mean return, $\Sigma \in \mathbf{S}^n_{++}$ is the return covariance, and $\gamma > 0$ is the risk aversion parameter. The data μ, Σ, and γ are given. In words, the target weights maximize the risk-adjusted expected return.

   At the beginning of each period t we are allowed to rebalance the portfolio by buying and selling assets. We call the post-trade portfolio weights $\tilde{w}_t$. They are found by solving the (rebalancing) problem

$$\begin{array}{ll} \text{maximize} & \mu^T w - \frac{\gamma}{2} w^T \Sigma w - \kappa^T |w - w_t| \\ \text{subject to} & \mathbf{1}^T w = 1, \end{array}$$

   with variable $w \in \mathbf{R}^n$, where $\kappa \in \mathbf{R}^n_+$ is the vector of (so-called linear) transaction costs for the assets. (For example, these could model bid/ask spread.) Thus, we choose the post-trade weights to maximize the risk-adjusted expected return, minus the transactions costs associated with rebalancing the portfolio. Note that the pre-trade weight vector w_t is known at the time we solve the problem. If we have $\tilde{w}_t = w_t$, it means that no rebalancing is done at the beginning of period t; we simply hold our current portfolio. (This happens if $w_t = w^\star$, for example.)

   After holding the rebalanced portfolio over the investment period, the dollar value of our portfolio becomes $h_{t+1} = \mathbf{diag}(r_t)\tilde{h}_t$, where $r_t \in \mathbf{R}^n_{++}$ is the (random) vector of asset returns over period t, and $\tilde{h}_t$ is the post-trade portfolio given in dollar values (which you do not need to know). The next weight vector is then given by

$$w_{t+1} = \frac{\mathbf{diag}\,(r_t)\tilde{w}_t}{r_t^T \tilde{w}_t}.$$

   (If $r_t^T \tilde{w}_t \leq 0$, which means our portfolio has negative value after the investment period, we have gone bust, and all trading stops.) The standard model is that r_t are IID random variables with mean and covariance μ and Σ, but this is not relevant in this problem.

   (a) *No-trade condition.* Show that $\tilde{w}_t = w_t$ is optimal in the rebalancing problem if

$$\gamma\,|\Sigma(w_t - w^\star)| \preceq \kappa$$

   holds, where the absolute value on the left is elementwise.

Interpretation. The lefthand side measures the deviation of w_t from the target portfolio $w^\star$; when this deviation is smaller than the cost of trading, you do not rebalance.

Hint. Find dual variables, that with $w = w_t$ satisfy the KKT conditions for the rebalancing problem.

(b) Starting from $w_1 = w^\star$, compute a sequence of portfolio weights $\tilde{w}_t$ for $t = 1, \ldots, T$. For each t, find $\tilde{w}_t$ by solving the rebalancing problem (with w_t a known constant); then generate a vector of returns r_t (using our supplied function) to compute w_{t+1} (The sequence of weights is random, so the results won't be the same each time you run your script. But they should look similar.)

Report the fraction of periods in which the no-trade condition holds and the fraction of periods in which the solution has only zero (or negligible) trades, defined as $\|\tilde{w}_t - w_t\|_\infty \leq 10^{-3}$. Plot the sequence $\tilde{w}_t$ for $t = 1, 2, \ldots, T$.

The file `portf_weight_rebalance_data.*` provides the data, a function to generate a (random) vector r_t of market returns, and the code to plot the sequence $\tilde{w}_t$. (The plotting code also draws a dot for every non-negligible trade.)

Carry this out for two values of κ, $\kappa = \kappa_1$ and $\kappa = \kappa_2$. Briefly comment on what you observe.

Hint. In CVXPY we recommend using the solver ECOS. But if you use SCS you should increase the default accuracy, by passing `eps=1e-4` to the `cvxpy.Problem.solve()` method.

5. *Solving nonlinear circuit equations using convex optimization.* An electrical circuit consists of b two-terminal devices (or branches) connected to n nodes, plus a so-called ground node. The goal is to compute several sets of physical quantities that characterize the circuit operation. The vector of *branch voltages* is $v \in \mathbf{R}^b$, where v_j is the voltage appearing across device j. The vector of *branch currents* is $i \in \mathbf{R}^b$, where i_j is the current flowing through device j. (The symbol i, which is often used to denote an index, is unfortunately the standard symbol used to denote current.) The vector of *node potentials* is $e \in \mathbf{R}^n$, where e_k is the potential of node k with respect to the ground node. (The ground node has potential zero by definition.)

   The circuit variables v, i, and e satisfy several physical laws. Kirchhoff's current law (KCL) can be expressed as $Ai = 0$, and Kirchhoff's voltage law (KVL) can be expressed as $v = A^T e$, where $A \in \mathbf{R}^{n\times b}$ is the reduced incidence matrix, which describes the circuit topology:

$$A_{kj} = \begin{cases} +1 & \text{branch } j \text{ enters node } k \\ -1 & \text{branch } j \text{ leaves node } k \\ 0 & \text{otherwise,} \end{cases}$$

   for $k = 1, \ldots, n$, $j = 1, \ldots, b$. (KCL states that current is conserved at each node, and KVL states that the voltage across each branch is the difference of the potentials of the nodes it is connected to.)

   The branch voltages and currents are related by

$$v_j = \phi_j(i_j), \quad j = 1, \ldots, b,$$

   where ϕ_j is a given function that depends on the *type* of device j. We will assume that these functions are continuous and nondecreasing. We give a few examples. If device j is a resistor with resistance $R_j > 0$, we have $\phi_j(i_j) = R_j i_j$ (which is called Ohm's law). If device j is a voltage source with voltage V_j and internal resistance $r_j > 0$, we have $\phi_j(i_j) = V_j + r_j i_j$. And for a more interesting example, if device j is a diode, we have $\phi_j(i_j) = V_T \log(1 + i_j/I_S)$, where I_S and V_T are known positive constants.

   (a) Find a method to solve the circuit equations, *i.e.*, find v, i, and e that satisfy KCL, KVL, and the branch equations, that relies on convex optimization. State the optimization problem clearly, indicating what the variables are. Be sure to explain how solving the convex optimization problem you propose leads to choices of the circuit variables that satisfy all of the circuit equations. You can assume that no pathologies occur in the problem that you propose, for example, it is feasible, a suitable constraint qualification holds, and so on.

   Hint. You might find the function $\psi : \mathbf{R}^b \to \mathbf{R}$,

$$\psi(i_1, \ldots, i_b) = \sum_{j=1}^{b} \int_0^{i_j} \phi_j(u_j)\, du_j,$$

   useful.

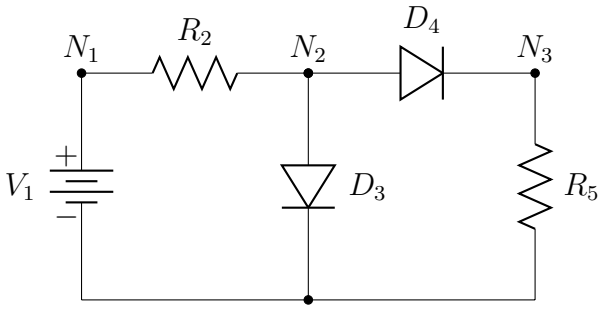
(b) Consider the circuit shown in the diagram below. Device 1 is a voltage source with parameters $V_1 = 1000$, $r_1 = 1$. Devices 2 and 5 are resistors with resistance $R_2 = 1000$, and $R_5 = 100$ respectively. Devices 3 and 4 are identical diodes with parameters $V_T = 26$, $I_S = 1$. (The units are mV, mA, and Ω.)

The nodes are labeled N_1, N_2, and N_3; the ground node is at the bottom. The incidence matrix A is

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

(The reference direction for each edge is down or to the right.)

Use the method in part (a) to compute v, i, and e. Verify that all the circuit equations hold.



6. *Optimal material blending.* A standard industrial operation is to blend or mix raw materials (typically fluids such as different grades of crude oil) to create blended materials or products. This problem addresses optimizing the blending operation. We produce n blended materials from m raw materials. Each raw and blended material is characterized by a vector that gives the concentration of each of q constituents (such as different octane hydrocarbons). Let $c_1, \ldots, c_m \in \mathbf{R}^q_+$ and $\tilde{c}_1, \ldots, \tilde{c}_n \in \mathbf{R}^q_+$ be the concentration vectors of the raw materials and the blended materials, respectively. We have $\mathbf{1}^T c_j = \mathbf{1}^T \tilde{c}_i = 1$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The raw material concentrations are given; the blended product concentrations must lie between some given bounds, $\tilde{c}_i^{\min} \preceq \tilde{c}_i \preceq \tilde{c}_i^{\max}$.

   Each blended material is created by pumping raw materials (continuously) into a vat or container where they are mixed to produce the blended material (which continuously flows out of the mixing vat). Let $f_{ij} \geq 0$ denote the flow of raw material j (say, in kg/s) into the vat for product i, for $i = 1, \ldots, n$, $j = 1, \ldots, m$. These flows are limited by the total availability of each raw material: $\sum_{i=1}^n f_{ij} \leq F_j$, $j = 1, \ldots, m$, where $F_j > 0$ is the maximum total flow of raw material j available. Let $\tilde{f}_i \geq 0$ denote the flow rates of the blended materials. These also have limits: $\tilde{f}_i \leq \tilde{F}_i$, $i = 1, \ldots, n$.

   The raw and blended material flows are related by the (mass conservation) equations

$$\sum_{j=1}^m f_{ij} c_j = \tilde{f}_i \tilde{c}_i, \quad i = 1, \ldots, n.$$

   (The lefthand side is the vector of incoming constituent mass flows and the righthand side is the vector of outgoing constituent mass flows.)

   Each raw and blended material has a (positive) price, p_j, $j = 1, \ldots, m$ (for the raw materials), and $\tilde{p}_i$, $i = 1, \ldots, n$ (for the blended materials). We pay for the raw materials, and get paid for the blended materials. The total profit for the blending process is

$$-\sum_{i=1}^n \sum_{j=1}^m f_{ij} p_j + \sum_{i=1}^n \tilde{f}_i \tilde{p}_i.$$

   The goal is to choose the variables f_{ij}, $\tilde{f}_i$, and $\tilde{c}_i$ so as to maximize the profit, subject to the constraints. The problem data are c_j, $\tilde{c}_i^{\min}$, $\tilde{c}_i^{\max}$, F_j, $\tilde{F}_i$, p_j, and $\tilde{p}_j$.

   (a) Explain how to solve this problem using convex or quasi-convex optimization. You must justify any change of variables or problem transformation, and explain how you recover the solution of the blending problem from the solution of your proposed problem.

   (b) Carry out the method of part (a) on the problem instance given in `material_blending_data.*`. Report the optimal profit, and the associated values of f_{ij}, $\tilde{f}_i$, and $\tilde{c}_i$.

7. *Graph isomorphism via linear programming.* An (undirected) graph with n vertices can be described by its adjacency matrix $A \in \mathbf{S}^n$, given by

$$A_{ij} = \begin{cases} 1 & \text{there is an edge between vertices } i \text{ and } j \\ 0 & \text{otherwise.} \end{cases}$$

Two (undirected) graphs are *isomorphic* if we can permute the vertices of one so it is the same as the other (*i.e.*, the same pairs of vertices are connected by edges). If we describe them by their adjacency matrices A and B, isomorphism is equivalent to the existence of a permutation matrix $P \in \mathbf{R}^{n\times n}$ such that $PAP^T = B$. (Recall that a matrix P is a permutation matrix if each row and column has exactly one entry 1, and all other entries 0.) Determining if two graphs are isomorphic, and if so, finding a suitable permutation matrix P, is called the *graph isomorphism problem.*

Remarks (not needed to solve the problem). It is not currently known if the graph isomorphism problem is NP-complete or solvable in polynomial time. The graph isomorphism problem comes up in several applications, such as determining if two descriptions of a molecule are the same, or whether the physical layout of an electronic circuit correctly reflects the given circuit schematic diagram.

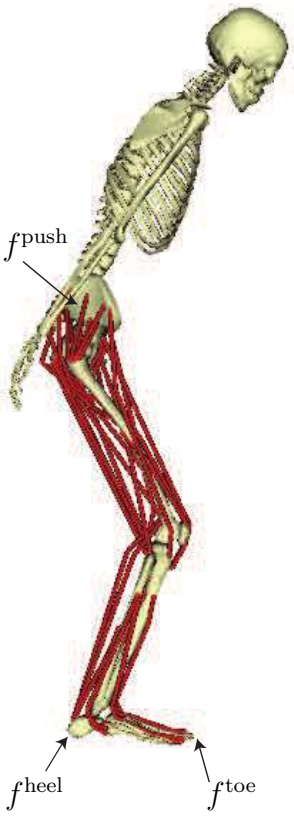
(a) Find a set of linear equalities and inequalities on $P \in \mathbf{R}^{n\times n}$, that together with the Boolean constraint $P_{ij} \in \{0, 1\}$, are necessary and sufficient for P to be a permutation matrix satisfying $PAP^T = B$. Thus, the graph isomorphism problem is equivalent to a Boolean feasibility LP.

(b) Consider the relaxed version of the Boolean feasibility LP found in part (a), *i.e.*, the LP that results when the constraints $P_{ij} \in \{0, 1\}$ are replaced with $P_{ij} \in [0, 1]$. When this LP is infeasible, we can be sure that the two graphs are not isomorphic. If a solution of the LP is found that satisfies $P_{ij} \in \{0, 1\}$, then the graphs are isomporphic and we have solved the graph isomorphism problem. This of course does not always happen, even if the graphs are isomorphic.

A standard trick to encourage the entries of P to take on the values 0 and 1 is to add a random linear objective to the relaxed feasibility LP. (This doesn't change whether the problem is feasible or not.) In other words, we minimize $\sum_{i,j} W_{ij}P_{ij}$, where W_{ij} are chosen randomly (say, from $\mathcal{N}(0, 1)$). (This can be repeated with different choices of W.)

Carry out this scheme for the two isomorphic graphs with adjacency matrices A and B given in `graph_isomorphism_data.*` to find a permutation matrix P that satisfies $PAP^T = B$. Report the permutation vector, given by the matrix-vector product Pv, where $v = (1, 2, \ldots, n)$. Verify that all the required conditions on P hold. To check that the entries of the solution of the LP are (close to) $\{0, 1\}$, report $\max_{i,j} P_{ij}(1 - P_{ij})$. And yes, you might have to try more than one instance of the randomized method described above before you find a permutation that establishes isomorphism of the two graphs.

8. *Maintaining static balance.* In this problem we study a human's ability to maintain balance against an applied external force. We will use a planar (two-dimensional) model to characterize the set of push forces a human can sustain before he or she is unable to maintain balance. We model the human as a linkage of 4 body segments, which we consider to be rigid bodies: the foot, lower leg, upper leg, and pelvis (into which we lump the upper body). The pose is given by the joint angles, but this won't matter in this problem, since we consider a fixed pose. A set of 40 muscles act on the body segments; each of these develops a (scalar) tension t_i that satisfies $0 \leq t_i \leq T_i^{\max}$, where $T_i^{\max}$ is the maximum possible tension for muscle i. (The maximum muscle tensions depend on the pose, and the person, but here they are known constants.) An external pushing force $f^{\text{push}} \in \mathbf{R}^2$ acts on the pelvis. Two (ground contact) forces act on the foot: $f^{\text{heel}} \in \mathbf{R}^2$ and $f^{\text{toe}} \in \mathbf{R}^2$. (These are shown at right.) These must satisfy



$$|f_1^{\text{heel}}| \leq \mu f_2^{\text{heel}}, \quad |f_1^{\text{toe}}| \leq \mu f_2^{\text{toe}},$$

where $\mu > 0$ is the coefficient of friction of the ground. There are also joint forces that act at the joints between the body segments, and gravity forces for each body segment, but we won't need them explicitly in this problem.

To maintain balance, the net force and torque on each each body segment must be satisfied. These equations can be written out from the geometry of the body (*e.g.*, attachment points for the muscles) and the pose. They can be reduced to a set of 6 linear equations:

$$A^{\text{musc}} t + A^{\text{toe}} f^{\text{toe}} + A^{\text{heel}} f^{\text{heel}} + A^{\text{push}} f^{\text{push}} = b,$$

where $t \in \mathbf{R}^{40}$ is the vector of muscle tensions, and A^{musc}, A^{toe}, A^{heel}, and A^{push} are known matrices and $b \in \mathbf{R}^6$ is a known vector. These data depend on the pose, body weight and dimensions, and muscle lines of action. Fortunately for you, our biomechanics expert Apoorva has worked them out; you will find them in `static_balance_data.*` (along with $T^{\max}$ and μ).

We say that the push force f^{push} can be *resisted* if there exist muscle tensions and ground contact forces that satisfy the constraints above. (This raises a philosophical question: Does a person solve an optimization to decide whether he or she should lose their balance? In any case, this approach makes good predictions.)

Find $\mathcal{F}^{\text{res}} \subset \mathbf{R}^2$, the set of push forces that can be resisted. Plot it as a shaded region.

Hints. Show that $\mathcal{F}^{\text{res}}$ is a convex set. For the given data, $0 \in \mathcal{F}^{\text{res}}$. Then for $\theta = 1^\circ, 2^\circ, \ldots, 360^\circ$, determine the maximum push force, applied in the direction θ, that can be resisted. To make a filled region on a plot, you can use the command `fill()`

in Matlab. For Python and Julia, `fill()` is also available through PyPlot. In Julia, make sure to use the ECOS solver with `solver = ECOSSolver(verbose=false)`.

Remark. A person can resist a much larger force applied to the hip than you might think.