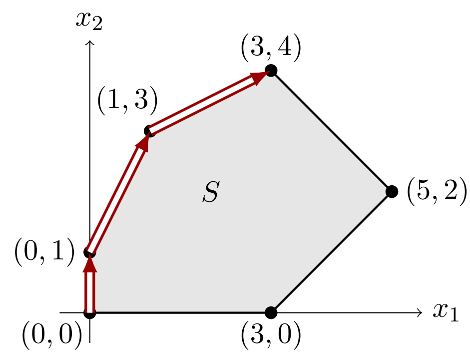
Solutions to Final Exam

F1. **(a)** The sketch below also shows the sequence of points generated by applying the simplex method (part **(c)**).



(b) It is helpful to first put the problem in form

$$\text{(1)} \qquad \begin{aligned} \text{minimize} \quad & z = c^\top x \\ \text{subject to} \quad & Ax \le b \\ & x \ge 0 \end{aligned}$$

where $b \ge 0$. To do so one must simply multiply the first and third constraints by -1.

Then I put the problem in standard form

$$\text{(2)} \qquad \begin{aligned} \text{minimize} \quad & z = c^\top x \\ \text{subject to} \quad & Ax = b \\ & x \ge 0 \end{aligned}$$

by adding slack variables x_3, x_4, x_5, x_6. The standard form has these data:

$$A = \begin{bmatrix} 1 & -1 & 1 & & & \\ -2 & 1 & & 1 & & \\ -1 & 2 & & & 1 & \\ 1 & 1 & & & & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 3 \\ 1 \\ 5 \\ 7 \end{bmatrix}, \qquad c = \begin{bmatrix} -2 \\ -3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(c) I applied the code `mysimplex.m` to this problem, using the form (2). For these problems the addition of m slack variables, i.e. one for each scalar inequality constraint, puts the problem in standard form *and* allows one to find a basic feasible solution by setting the original variables to zero; the entries of b give the values of the slack variables. The code does this internally.

Applying the code looked like this:

```
>> A = [1 -1; -2 1; -1 2; 1 1];
>> b = [3 1 5 7]';
>> c = [-2 -3]';
>> [x,z] = mysimplex(c,A,b,true)
...
```

With a fourth argument of `true`, the code print out its iterates. In the original variables:

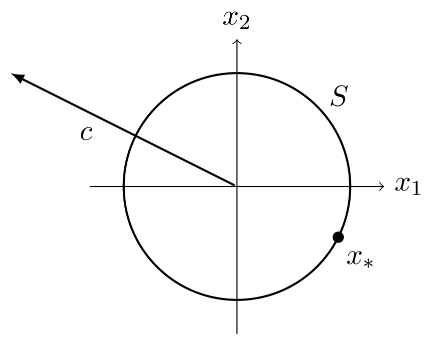
$$(0,0) \to (0,1) \to (1,3) \to (3,4)$$

These transitions are shown as double arrows in the sketch.

F2. **(a)** We are minimizing $z = c^\top x$ subject to the constraint that the length of the vector x is one: $\|x\| = 1$. That is, we are minimizing over the unit sphere S. On the other hand, the objective function is an inner product, so when $x \in S$ then

$$z = c^\top x = \|c\|\|x\| \cos\theta = \|c\| \cos\theta$$

where θ is the angle between c and x. To make the right side as small as possible we need $\theta = \pi$, that is, x should point opposite to c. Thus the solution is a unit vector opposite to c: $x_* = -c/\|c\|$. A 2D case is illustrated in the sketch below.



(b) For this problem $\lambda \in \mathbb{R}^1$; there is a single Lagrange multiplier. The Lagrangian is

$$\mathcal{L}(x, \lambda) = c^\top x - \lambda(\|x\|^2 - 1).$$

The first-order optimality conditions from Theorem 14.15 are

$$\nabla_x \mathcal{L}(x, \lambda) = c - 2\lambda x = 0 \tag{3}$$

along with feasibility $\|x\|^2 = 1$.

(c) Equation (3) says that an optimal solution x_* is parallel to c, i.e. $x_* = \frac{1}{2\lambda} c$. Applying feasibility gives

$$1 = \|x_*\| = \frac{1}{|2\lambda_*|}\|c\|$$

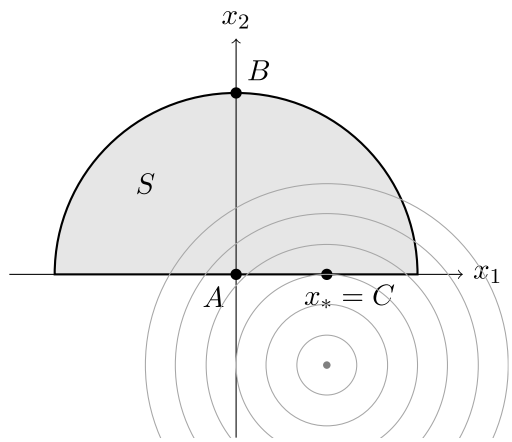
so that $|\lambda_*| = \|c\|/2$ or equivalently $\lambda_* = \pm\|c\|/2$. Thus first-order optimality gives two possible optimal points

$$(x_*, \lambda_*) = \left(+\frac{c}{\|c\|}, +\frac{\|c\|}{2}\right), \left(-\frac{c}{\|c\|}, -\frac{\|c\|}{2}\right).$$

(*Completing the second-order conditions, as follows, is* not *requested or required.*)

Note that the first (positive) solution fails the second-order necessary condition. In fact the Hessian of the Lagrangian is $\nabla^2_{xx}\mathcal{L}(x, \lambda) = -2\lambda I$. At the positive solution this is a negative multiple of the identity. Regardless of the choice of null-space matrix $Z(x_*)$, the matrix $Z(x_*)^\top \nabla^2_{xx}\mathcal{L}(x, \lambda) Z(x_*)$ is not positive semi-definite. However, at the negative solution the Hessian is a positive multiple of the identity, and thus the second-order sufficient conditions hold (regardless of which null space basis matrix $Z(x_*)$ is chosen).

F3. **(a)** I made the following sketch which includes the feasible set S, the contours of f, the location of the unconstrained minimum at $(1,-1)^\top$, the solution $x_* = (1,0)^\top$, and the points A, B, C considered in part **(b)**.



Informally, the solution is at x_* because the gradient of f is parallel to the gradient of the active constraint there. One cannot descend further because the "fence" of the constraint $g_2(x) \ge 0$ stops steepest-descent motion toward the unconstrained minimum. Moving along the boundary also does not allow decrease.

(*Formally, from* **(b)** *below, at x_* we have $(\lambda_*)_1 = 0$ and $(\lambda_*)_2 = 2$, so $\nabla f(x_*) = 2\nabla g_2(x_*)$ and the gradients are parallel and pointed in the same direction.*)

(b) The problem has two constraints "$g_i(x) \ge 0$": $g_1(x) = 4 - x_1^2 - x_2^2$ and $g_2(x) = x_2$. The Lagrangian is

$$\mathcal{L}(x,\lambda) = f(x) - \lambda^\top g(x) = (x_1-1)^2 + (x_2+1)^2 - \lambda_1(4 - x_1^2 - x_2^2) - \lambda_2 x_2$$

with x-gradient

$$\nabla_x \mathcal{L}(x,\lambda) = \begin{bmatrix} 2(x_1-1) + 2\lambda_1 x_1 \\ 2(x_2+1) + 2\lambda_1 x_2 - \lambda_2 \end{bmatrix}$$

For each point we will determine whether we can solve the first-order optimality conditions from Theorem 14.18, namely

$$\begin{aligned} \nabla_x \mathcal{L}(x,\lambda) &= 0 \\ \lambda &\ge 0 \\ \lambda^\top g(x) &= 0 \end{aligned}$$

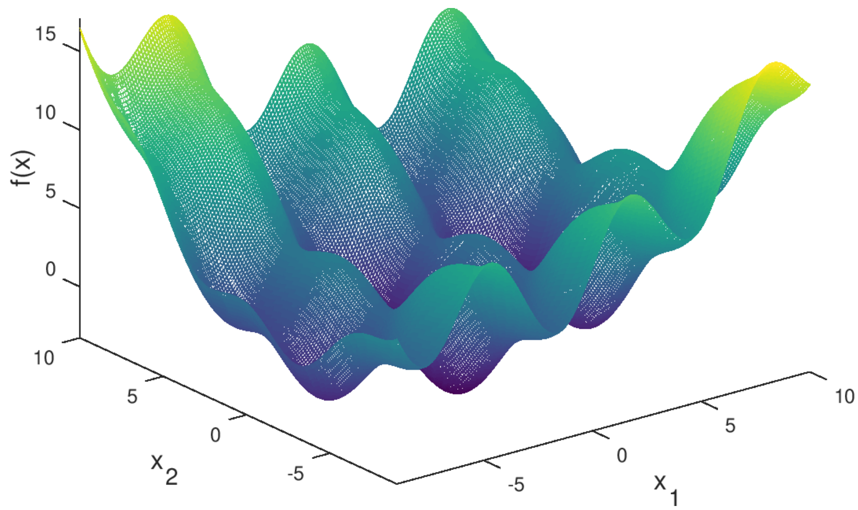
Note that at each point we have values for x_1, x_2 so the unknowns are λ_1, λ_2. We have three scalar equations to solve, namely the first and last of the above conditions.

$A = (0,0)^\top$: At A only the second constraint is active. The equations we want to solve simplify to $-2 = 0$ and $2 - \lambda_2 = 0$ and $4\lambda_1 + 0\lambda_2 = 0$. There is no solution because $-2 \ne 0$. (*At A the gradient of f is not parallel to the gradient of the active (second) constraint.*)

$B = (0,2)^\top$: At B only the first constraint is active. The equations we want to solve simplify to $-2 = 0$, $6 + 4\lambda_1 - \lambda_2 = 0$, and $0\lambda_1 + 2\lambda_2 = 0$. Again there is no solution because $-2 \ne 0$. (*At B the gradient of f is not parallel to the gradient of the active (first) constraint.*)

$C = (1,0)^\top$: At C only the second constraint is active. The equations we want to solve simplify to $0 + 2\lambda_1 = 0$ and $2 - \lambda_2 = 0$ and $3\lambda_1 + 0\lambda_2 = 0$. A solution is $\lambda_1 = 0$ and $\lambda_2 = 2$. Note $\lambda \ge 0$. Thus $C = x_*$ satisfies the necessary conditions in Theorem 14.18.

F4. **(a)** I generated the figure below. (See the last part of `gridsearch.m`.) By rotating the figure around it seems the global minimum can be roughly estimated: $f(x_*) \approx -3$.



(b) I wrote two codes. The first evaluates the objective function and its gradient. We need a function like this when we apply `sdbt.m`:

f4fcn.m

```
function [f,df] = f4fcn(x)
% F4FCN  The objective function, and its gradient, for F4 on the Final Exam

f = 3.0 * sin(x(1)) + cos(x(2)) + 0.05 * (x(1)^2 - x(1)*x(2) + 2.0*x(2)^2);
if nargout > 1
    df = [3.0*cos(x(1)) + 0.05*(2.0*x(1)-x(2));
          -sin(x(2)) + 0.05*(-x(1)+4.0*x(2))];
end
```

The second code does the grid search. It requires `sdbt.m` to be on the current path. (*Your code does not have to be a general 2D grid search like this; it can be special to this particular problem.*)

gridsearch.m

```
function gridsearch(f,x1,x2,tol)
% GRIDSEARCH  Solve 2D global minimization problems by combining a grid search
% with steepest descent (SDBT).
% Usage:
%    gridsearch(f,x1,x2,tol)
% where
%    f   = handle for function which returns f and gradient of f (see SDBT)
%    x1  = list of x_1 coordinates for initial points
%    x2  = ...      x_2 ...
%    tol = tolerance for SDBT [default: 1.0e-6]
% Example:  problem F4 does
% >> gridsearch(@f4fcn,-9:10,-9:10,1.0e-6)
% Requires:  SDBT

if nargin < 4,  tol = 1.0e-6;  end

% search, saving all f-values, and running minimum of f(x), and best x
```

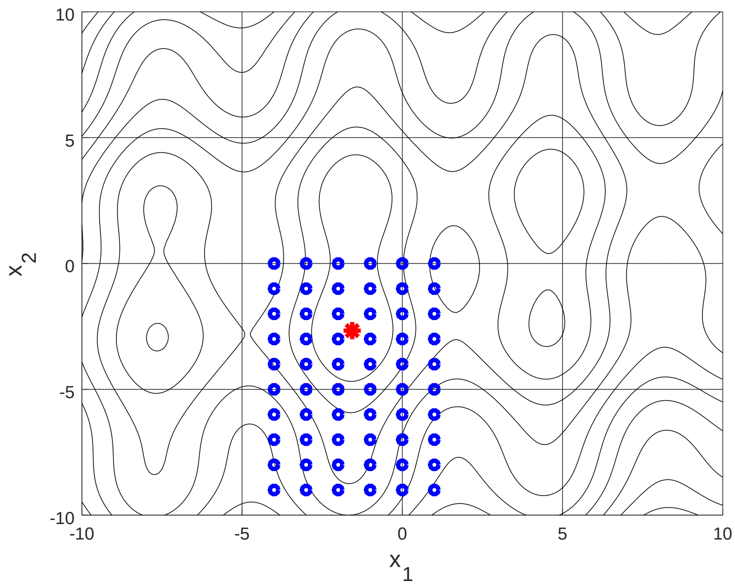
```
fprintf('searching using a grid of %d initial points x_0 ...\n',...
        length(x1)*length(x2))
fval = zeros(20,20);
fstar = 1.0e6;  % larger than max
for j = 1:length(x1)
    for k = 1:length(x2)
        x0 = [x1(j); x2(k)];
        xk = sdbt(x0,f,tol,100);   % set maxiters = 100 (for speed)
        fval(j,k) = f(xk);
        if fval(j,k) < fstar
            fstar = fval(j,k);
            xstar = xk;
        end
    end
end
fprintf('global minimum:  f(%.6f,%.6f) = %.6f\n',xstar(1),xstar(2),fstar)

% first draw contour map of solution
figure(1),  clf,  hold on
N = 201;
x1f = linspace(min(x1),max(x1),N);   % finer grid for contour/mesh plotting
x2f = linspace(min(x2),max(x2),N);
zz = zeros(N,N);
for j = 1:N
    for k = 1:N
        zz(j,k) = f([x1f(j); x2f(k)]);
    end
end
contour(x1f,x2f,zz','k')
% next show xstar and those x0 that yielded xstar
plot(xstar(1),xstar(2),'r*','markersize',12)
for j = 1:length(x1)
    for k = 1:length(x2)
        if fval(j,k) <= fstar + tol
            plot(x1(j),x2(k),'bo','markersize',6)
        end
    end
end
axis([-10.5 10.5 -10.5 10.5]),  axis tight,  grid on
xlabel('x_1','fontsize',16),  ylabel('x_2','fontsize',16)

% draw surface (mesh) plot of objective function
figure(2),  clf,  hold on
mesh(x1f,x2f,zz')
view(3),  axis tight
xlabel('x_1','fontsize',16),  ylabel('x_2','fontsize',16)
zlabel('f(x)','fontsize',16)
```

Running this code looks like this:

```
>> gridsearch(@f4fcn,-9:10,-9:10,5.0e-7)
searching using a grid of 400 initial points x_0 ...
global minimum:  f(-1.563167,-2.668592) = -3.264378
```

Note that the second and third arguments are lists of x_1 and x_2 coordinates, respectively, of the initial points x_0. For `tol` you can choose any value around 10^{-6} to get about 6 digit accuracy.

The code produces two figures, namely the surface plot above and the contour map below which shows the computed global minimum x_* (large star) and the x_0 values (small circles) which lead to x_*. In this case we see that a rectangle of 60 values of x_0 lead to the solution x_*.



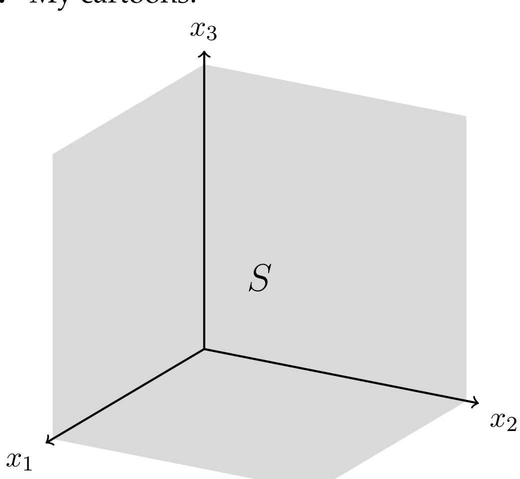
One might observe that a coarser grid of x_0 will do just fine. This is true! For example

```
>> gridsearch(@f4fcn,-10:2:10,-10:2:10,5.0e-7)
```

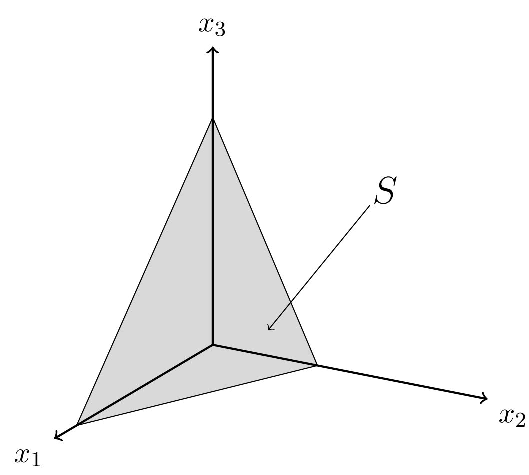
finds the same minimum to the same accuracy using 1/4 the work. However, too coarse and one misses the global minimum:

```
>> gridsearch(@f4fcn,-10:12:14,-10:12:14,5.0e-7)
searching using a grid of 9 initial points x_0 ...
global minimum:  f(4.605414,2.804601) = -2.725351
```

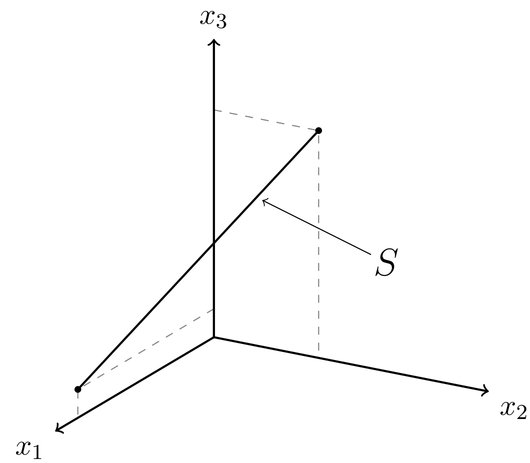
F5. My cartoons:



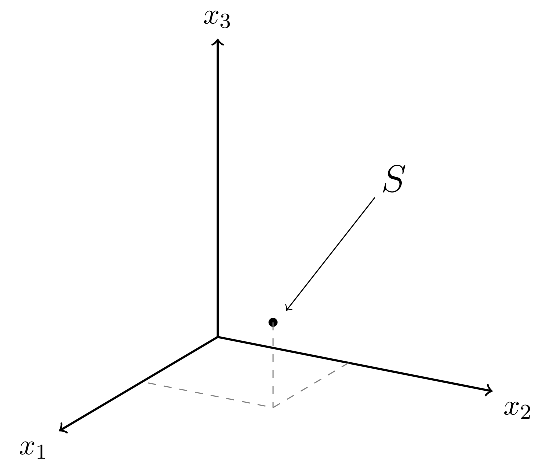
$m = 0$



$m = 1$



$m = 2$



$m = 3$

F6. **(a)** Given $x \in \mathbb{R}^n$, as usual we define $g(x)$ to be a column vector formed from the numbers $g_1(x), \dots, g_\ell(x)$. For a feasible point x also define $h(x)$ to be the column vector formed from all constraints $h_i(x)$ and $\tilde{h}(x)$ to be the column vector formed from the constraints $h_i(x)$ which are active at x, i.e. for which $h_i(x) = 0$. In these terms we can define

Definition. x_* is a *regular point of the constraints* if the matrix

$$\begin{bmatrix} \nabla g(x_*) & \nabla \tilde{h}(x_*) \end{bmatrix}$$

has linearly independent columns.

Just as in the textbook's definition on page 503, the inactive inequality constraints are not relevant in this definition.

The Lagrangian for this problem is

$$\begin{aligned} \mathcal{L}(x, \lambda, \mu) &= f(x) - \sum_{i=1}^{\ell} \lambda_i g_i(x) - \sum_{j=1}^{m} h_j(x) \\ &= f(x) - \lambda^\top g(x) - \mu^\top h(x) \end{aligned}$$

where $\lambda \in \mathbb{R}^\ell$ and $\mu \in \mathbb{R}^m$. Note that $h(x)$, not $\tilde{h}(x)$ is used here.

(b) The following theorem, which is the full Karush-Kuhn-Tucker theorem,[1] states the first-order necessary conditions.

Theorem. Suppose x_* is a regular point of the constraints and a local minimizer. Then there exist vectors $\lambda_* \in \mathbb{R}^\ell$ and $\mu_* \in \mathbb{R}^m$ so that

$g(x_*) = 0$	primal feasibility: equality constraints
$h(x_*) \geq 0$	primal feasibility: inequality constraints
$\nabla_x \mathcal{L}(x_*, \lambda_*, \mu_*) = 0$	stationarity
$\mu_* \geq 0$	dual feasibility
$\mu_*^\top h(x_*) = 0$	complementary slackness

[1] en.wikipedia.org/wiki/Karush-Kuhn-Tucker_conditions

You were not asked to prove this, just to state it. Note that the stationarity condition, i.e. the first-order condition itself, can be written

$$\nabla f(x_*) = \nabla g(x_*)\lambda_* + \nabla h(x_*)\mu_*$$

That is, the gradient of f at the solution can be written as a linear combination of the gradients of the constraints. When there are inequality constraints, however, by complementary slackness we expect some of their corresponding multipliers μ_i to be zero.

F7. (*I have written the solution as a theorem, but this is just a stylistic choice. Because the feasible set is convex and the objective function is both strictly-convex and coercive ($f(x) \to \infty$ as $\|x\| \to \infty$), this minimization problem has a unique global solution, which we find. I used Lagrange-multiplier techniques (e.g. equation (14.2)), but if you use a null-space matrix then it is easiest to go to section 3.3 and find that you can choose $Z = I - A^\top(AA^\top)^{-1}A$. Finally, note that* MATLAB *computes this x_* if you do "*`A \ b`*" when A is $m \times n$ and $m < n$, and $b \in \mathbb{R}^m$; try it out!*)

Theorem. Suppose $A \in \mathbb{R}^{m\times n}$ has full row rank, so $m \le n$, and suppose $b \in \mathbb{R}^m$. The point

$$x_* = A^\top(AA^\top)^{-1}b \tag{4}$$

solves the problem

$$\begin{aligned} &\text{minimize} && f(x) = \tfrac{1}{2}x^\top x \\ &\text{subject to} && Ax = b \end{aligned} \tag{5}$$

Proof. Suppose $\tilde{x}$ is a point which satisfies $A\tilde{x} = b$ and which is a local minimizer of $f(x)$. Note $\nabla f(x) = x$. By Lemma 14.2, but using the Lagrange multipliers form in equation (14.2), there is a vector $\tilde{\lambda} \in \mathbb{R}^m$ so that the local minimizer $\tilde{x}$ satisfies

$$\tilde{x} = A^\top\tilde{\lambda}.$$

(*The Lagrangian for this problem is $\mathcal{L}(x,\lambda) = \frac{1}{2}x^\top x - \lambda^\top(Ax - b)$ so $\nabla_x\mathcal{L}(x,\lambda) = x - A^\top\lambda$.*)

That is, we have this system of equations describing $\tilde{x}$ and $\tilde{\lambda}$:

$$\begin{aligned} \tilde{x} - A^\top\tilde{\lambda} &= 0 \\ A\tilde{x} &= b \end{aligned} \tag{6}$$

Solving the first equation for $\tilde{x}$ and substituting into the second equation gives

$$AA^\top\tilde{\lambda} = b.$$

Because A has full row rank it follows that $AA^\top$ is positive definite, thus invertible,[2] so

$$\tilde{\lambda} = (AA^\top)^{-1}b.$$

From the first equation in (6) we now get $\tilde{x} = A^\top(AA^\top)^{-1}b$. Thus $\tilde{x}$ is the point x_* given by (4).

On the other hand, x_* satisfies the sufficient conditions in Lemma 14.3. Note that $\nabla^2 f(x) = I$. Suppose Z is a null space basis matrix for A, so Z has full column rank, and then $Z^\top Z$ is positive definite.[3] But then $Z^\top\nabla^2 f(x_*)Z = Z^\top IZ = Z^\top Z$ is positive definite. Thus x_* is a strict local minimizer of the problem. □

[2] This fact was made explicit in class, and it is Exercise 3.4 in section 3.3.

[3] This is really the same fact again, but it is Exercise 3.5 in section 3.3.