QP Solver

We now write our own solver for the problem

$$\begin{aligned} \min \quad & \frac{1}{2}\Delta z^T\, H\, \Delta z + \Delta z^T h \\ \text{s.t.} \quad & E\, \Delta z = e. \end{aligned}$$

This problem can be easily solved by writing the Lagrangian

$$L = \frac{1}{2}\Delta z^T\, H\, \Delta z + \Delta z^T h + \lambda^T (E\, \Delta z - e)$$

and identifying the saddlepoint

$$\nabla L = 0.$$

Closed-Form Solution

By computing the gradient of the lagrangian, we find the condition

$$\nabla L = \begin{bmatrix} \nabla_x L \\ \nabla_\lambda L \end{bmatrix} = \begin{bmatrix} H & E^T \\ E & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \lambda \end{bmatrix} + \begin{bmatrix} h \\ -e \end{bmatrix} = 0$$

which is solved by

$$\begin{bmatrix} \Delta z \\ \lambda \end{bmatrix} = \begin{bmatrix} H & E^T \\ E & 0 \end{bmatrix}^{-1} \begin{bmatrix} -h \\ e \end{bmatrix}$$

```
function [dz,l] = solve_QP(H,h,E,e,C,c)
% % Generate an appropriately sized matrix of zeros
% nz = size(H,1);
% nl = size(E,1);
% O = zeros(nl);
%
% % Turn the problem into a linear systems of equations
% A = [H E'
%      E O];
% b = [-h
%       e];
%
% % Solve linear system of equations
% y = A\b;
%
% % Extract dz and l from y
% % dz = y(1:nz);
%  l = y(nz+(1:nl));
[dz,~,~,~,lag] = quadprog(H,h,C,c,E,e,[],[],
[],optimoptions('quadprog','Display','none'));
l = lag.eqlin;
end
```

Note that the same result can be obtained using

```
dz = quadprog(H,h,[],[],E,e)
```