CS/ECE/ISYE524: Introduction to Optimization – Convex Optimization Models

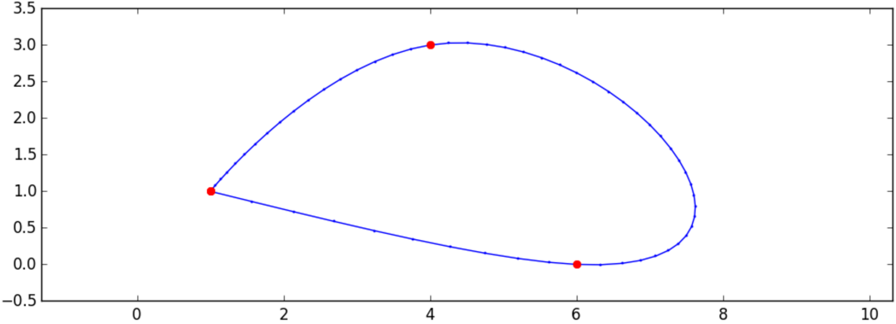
Jeff Linderoth

Department of Industrial and Systems Engineering
University of Wisconsin-Madison

March 11, 2024

Example: Anakin Goes Podracing

- Anakin Skywalker has a Pod, and he is practicing his pod-racing.
- We are given a set of k waypoint locations and times.
- The objective is to hit the waypoints at the prescribed times while minimizing fuel use.



Anakin's Dynamics

- Discretize time: $t = 0, 1, 2, \ldots, T$.
- Important variables: position x_t, velocity v_t, thrust u_t. (These are vectors in $\mathbb{R}^2$)
- Simplified model of the dynamics:

$$\begin{aligned} x_{t+1} &= x_t + v_t \\ v_{t+1} &= v_t + u_t \end{aligned} \qquad \text{for } t = 0, 1, \ldots, T-1$$

- We must choose $u_0, u_1, \ldots, u_T$.
- Initial position and velocity: $x_0 = 0$ and $v_0 = 0$.
- Waypoint constraints: $x_{t_i} = w_i$ for $i = 1, \ldots, k$.
- Minimize fuel use: $\|u_0\|^2 + \|u_1\|^2 + \cdots + \|u_T\|^2$

Anakin's First Model

First model: Hit the waypoints exactly

$$
\begin{aligned}
\underset{x_t, v_t, u_t}{\text{minimize}} \quad & \sum_{t=0}^{T} \|u_t\|^2 \\
\text{subject to:} \quad & x_{t+1} = x_t + v_t && \text{for } t = 0, 1, \ldots, T-1 \\
& v_{t+1} = v_t + u_t && \text{for } t = 0, 1, \ldots, T-1 \\
& x_0 = v_0 = 0 \\
& x_{t_i} = w_i && \text{for } i = 1, \ldots, k
\end{aligned}
$$

Julia model: Podracing.ipynb

Tradeoffs in Podracing

Second model: We are allowed to miss the waypoints (by a bit):

$$\begin{aligned} \underset{x_t, v_t, u_t}{\text{minimize}} \quad & \sum_{t=0}^{T} \|u_t\|^2 + \lambda \sum_{i=1}^{k} \|x_{t_i} - w_i\|^2 \\ \text{subject to:} \quad & x_{t+1} = x_t + v_t \quad \text{for } t = 0, 1, \ldots, T-1 \\ & v_{t+1} = v_t + u_t \quad \text{for } t = 0, 1, \ldots, T-1 \\ & x_0 = v_0 = 0 \end{aligned}$$

- λ controls the tradeoff between making u small and hitting all the waypoints.
- Tradeoff-Podracing.ipynb

Multi-objective tradeoff

- We can use a similar procedure if we have more than two costs we'd like to make small, e.g. J_1, J_2, J_3
- Choose parameters $\lambda > 0$ and $\mu > 0$. Then solve:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & J_1(x) + \lambda J_2(x) + \mu J_3(x) \\ \text{subject to:} & \text{constraints} \end{array}$$

- Each $\lambda > 0$ and $\mu > 0$ yields a solution $\hat{x}_{\lambda,\mu}$.
- Can visualize tradeoff by plotting $J_3(\hat{x}_{\lambda,\mu})$ vs $J_2(\hat{x}_{\lambda,\mu})$ vs $J_1(\hat{x}_{\lambda,\mu})$ on a 3D plot. You then obtain a *Pareto surface*.

Minimum-norm as a regularization

- When $Ax = b$ is underdetermined (A is wide), we can resolve ambiguity by adding a cost function, e.g. *min-norm* LS:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x\|^2 \\ \text{subject to:} \quad & Ax = b \end{aligned}$$

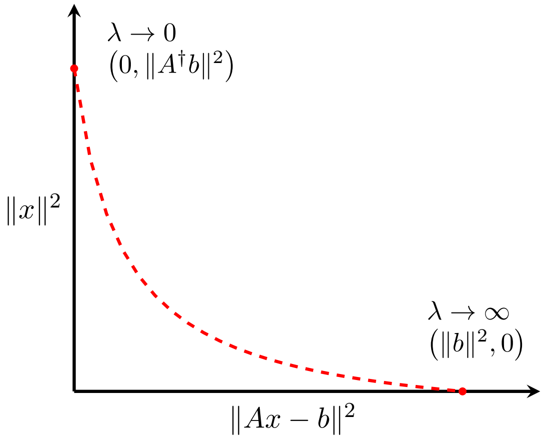
- Alternative approach: express it as a tradeoff!

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda\|x\|^2$$

- Tradeoffs of this type are called regularization and λ is called the *regularization parameter* or *regularization weight*
- If we let $\lambda \to \infty$, we just obtain $\hat{x} = 0$
- If we let $\lambda \to 0$, we obtain the minimum-norm solution!

Tradeoff visualization

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda \|x\|^2$$



Regularization

Regularization: Additional penalty term added to the cost function to encourage a solution with desirable properties.

Regularized least squares:

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda R(x)$$

- $R(x)$ is the regularizer (penalty function)
- λ is the regularization parameter
- The model has different names depending on $R(x)$.

Regularized least squares turns out to be important in many contexts!

Regularization

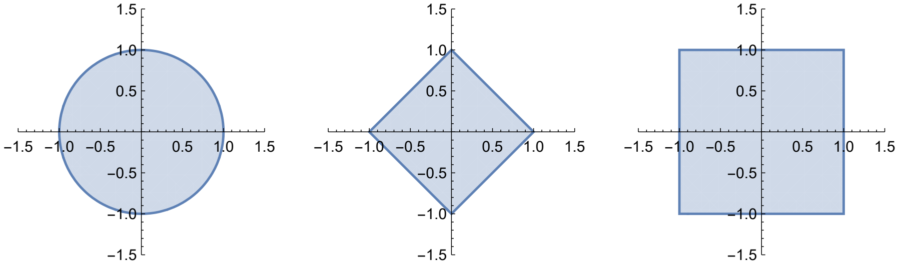
$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2 + \lambda R(x)$$

1. If $R(x) = \|x\|^2 = x_1^2 + x_2^2 + \cdots + x_n^2$
   It is called: L_2 *regularization*, *Tikhonov regularization*, or *Ridge regression* depending on the application. It has the effect of *smoothing* the solution.
2. If $R(x) = \|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$
   It is called: L_1 *regularization* or *LASSO*. It has the effect of *sparsifying* the solution ($\hat{x}$ will have few nonzero entries).
3. $R(x) = \|x\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$
   It is called L_∞ *regularization* and it has the effect of *equalizing* the solution (makes many components tie for the max absolute value).

Norm balls

For a norm $\|\cdot\|_p$, the **norm ball** of radius r is the set:

$$B_r = \{x \in \mathbb{R}^n \mid \|x\|_p \leq r\}$$



$$\|x\|_2 \leq 1 \qquad \|x\|_1 \leq 1 \qquad \|x\|_\infty \leq 1$$

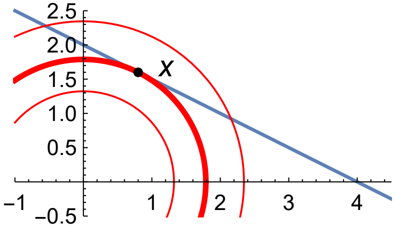
$$x^2 + y^2 \leq 1 \qquad |x| + |y| \leq 1 \qquad \max\{|x|, |y|\} \leq 1$$

Simple example

Consider the minimum-norm problem for different norms:

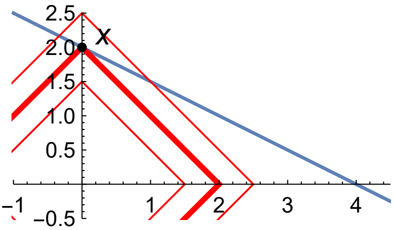
$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|x\|_p \\ \text{subject to:} & Ax = b \end{array}$$

- set of solutions to $Ax = b$ is an affine subspace
- solution is point belonging to smallest norm ball
- for $p = 2$, this occurs at the perpendicular distance

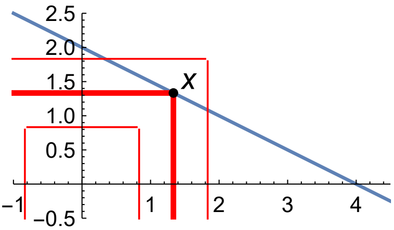


Simple example

- for $p = 1$, this occurs at one of the axes.
- *sparsifying* behavior



- for $p = \infty$, this occurs at corners or edges, where many components tie for max value
- *equalizing* behavior



Example: hovercraft revisited (simpler 1D case)

One-dimensional version of the hovercraft problem:

- Start at $x_1 = 0$ with $v_1 = 0$ (at rest at position zero)
- Finish at $x_{50} = 100$ with $v_{50} = 0$ (at rest at position 100)
- Same simple dynamics as before:

$$\begin{aligned} x_{t+1} &= x_t + v_t \\ v_{t+1} &= v_t + u_t \end{aligned} \qquad \text{for: } t = 1, 2, \ldots, 49$$

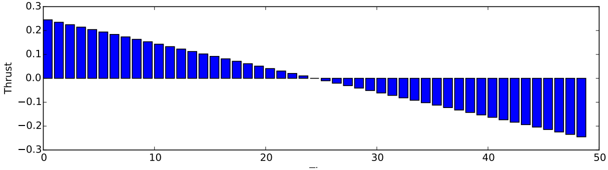
- Decide thruster inputs $u_1, u_2, \ldots, u_{49}$.
- This time: minimize $\|u\|_p$

Example: hovercraft revisited

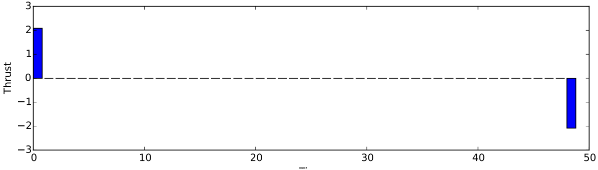
$$\begin{aligned} \underset{x_t, v_t, u_t}{\text{minimize}} \quad & \|u\|_p \\ \text{subject to:} \quad & x_{t+1} = x_t + v_t && \text{for } t = 1, \ldots, 49 \\ & v_{t+1} = v_t + u_t && \text{for } t = 1, \ldots, 49 \\ & x_1 = 0, \quad x_{50} = 100 \\ & v_1 = 0, \quad v_{50} = 0 \end{aligned}$$

- This model has 150 variables, but very easy to understand.

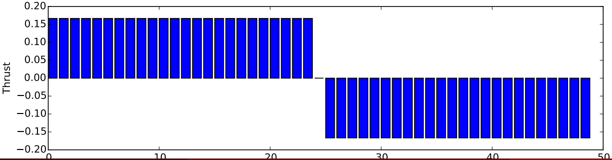
1. Minimizing $\|u\|_2^2$ (smooth)



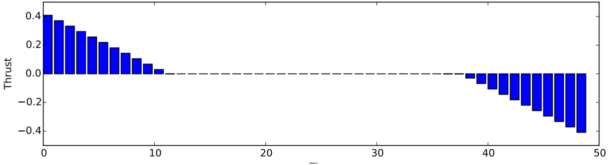
2. Minimizing $\|u\|_1$ (sparse)



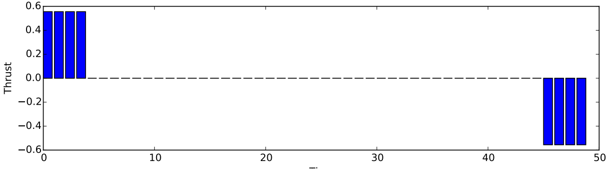
3. Minimizing $\|u\|_\infty$ (equalized)



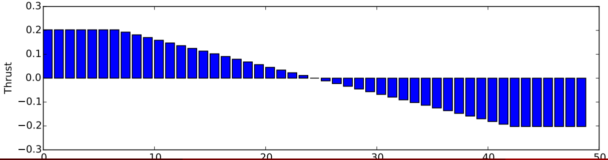
1. Minimizing $\|u\|_2^2 + \lambda\|u\|_1$ (smooth and sparse)



2. Minimizing $\|u\|_\infty + \lambda\|u\|_1$ (equalized and sparse)



3. Minimizing $\|u\|_2^2 + \lambda\|u\|_\infty$ (equalized and smooth)



Hierarchical Optimization

- One popular way for dealing with multiple objectives or goals is to combine then into objective using weights or penalties.
- This can work.
- But I don't recommend it in some cases...
  - If the quantities you are combining are in different units
  - If you have significantly more than two objectives

- Instead, often decision-makers have priorities for the individual goals
- In this case you can solve the problem in a hierarchical manner

Hierarchical Optimization

- Suppose we wish to make some decisions $x \in X$, but we have three different goals/objectives:
  1. $\min f_1(x)$
  2. $\max f_2(x)$
  3. $\min f_3(x)$
- In hierarchical optimization, we solve the objectives in the given order.
- As we move down the hierarchy, we impose the constraint that we do (almost?) as well with respect to the previous objectives.
- This of course generalizes to any number of goals that you can put in a priority order

Hierarchical Optimization

1. First, solve for the highest priority objective
$$z_1^* := \min_{x \in X} f_1(x)$$
2. Next, constrained to do (almost) as well with respect to first objective, solve for second priority:
$$z_2^* := \max_{x \in X} \{ f_2(x) : f_1(x) \leq (1+\varepsilon) z_1^* \}$$
3. Finally, solve final objective constrained to do (almost) as well with respect to first two objectives:
$$\min_{x \in X} \{ f_3(x) : f_1(x) \leq (1+\varepsilon) z_1^*, f_2(x) \geq (1-\varepsilon) z_2^* \}$$

The Upshot

Final solution is best with respect to the third objective that is within ε% of the best for the first two objectives

Annakin Again—Constrained Waypoints

- Suppose now that we only wish to make sure that we get "close enough" to the waypoints, which we measure by saying that each component of the position is within β units away

$$
\begin{aligned}
F^* := \underset{x,v,u}{\text{minimize}} \quad & \sum_{t=0}^{T} \|u_t\|^2 \\
\text{subject to:} \quad & x_{t+1} = x_t + v_t && \text{for } t = 0, 1, \ldots, T-1 \\
& v_{t+1} = v_t + u_t && \text{for } t = 0, 1, \ldots, T-1 \\
& \|x_{t_i} - w_i\|_\infty \le \beta && \text{for } i = 1, \ldots, k \\
& x_0 = v_0 = 0
\end{aligned}
$$

Tradeoff-Podracing-Constraint.ipynb

Hierarchical—Finish Slow

- Now suppose that for a fixed value of β, we have a secondary objective to be going as slow as possible at the end of the race
- But we also want to make sure that we won't use too much more fuel than we do in an optimal fuel plan
- Tradeoff-Podracing-Hierarchical.ipynb

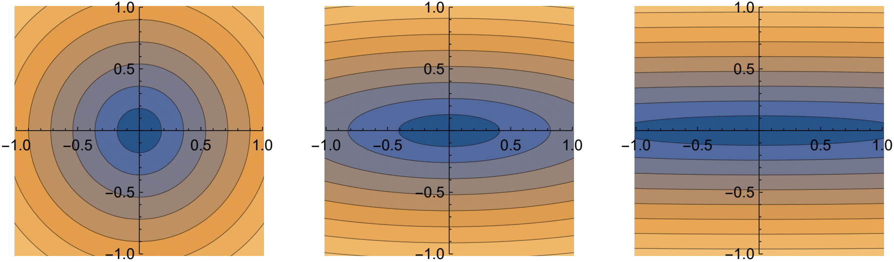
$$
\begin{array}{lll}
\underset{x,v,u}{\text{minimize}} & \|v_T\|^2 & \\
\text{subject to:} & x_{t+1} = x_t + v_t & \text{for } t = 0, 1, \dots, T-1 \\
& v_{t+1} = v_t + u_t & \text{for } t = 0, 1, \dots, T-1 \\
& \|x_{t_i} - w_i\|_\infty \leq \beta & \text{for } i = 1, \dots, k \\
& x_0 = v_0 = 0 & \\
& \displaystyle\sum_{t=1}^{T} \|u_t\|^2 \leq (1+\varepsilon) F^* &
\end{array}
$$

Ellipsoids

- For linear constraints, the set of x satisfying $c^\mathsf{T} x = b$ is a *hyperplane* and the set $c^\mathsf{T} x \leq b$ is a *halfspace*.
- For quadratic constraints, the set of x satisfying $x^\mathsf{T} Q x \leq b$ is an *ellipsoid* if $Q \succ 0$.
- If $Q \succ 0$, then $x^\mathsf{T} Q x \leq b \iff \|Q^{1/2} x\|^2 \leq b$.
- (Recall that if we write the eigenvalue decomposition $Q = U \Lambda U^T$, then $Q^{1/2} = U \Lambda^{1/2} U^T$, where $\Lambda^{1/2}$ is the diagonal matrix whose diagonal entries are the square roots of the diagonals of Λ.)

Degenerate Ellipsoids

Ellipsoid axes have length $\frac{1}{\sqrt{\lambda_i}}$. When an eigenvalue is close to zero, contours are stretched in that direction.



$x^2 + y^2$ $\qquad$ $\frac{1}{10}x^2 + y^2$ $\qquad$ $\frac{1}{100}x^2 + y^2$

- Warmer colors $=$ larger values
- If $\lambda_i = 0$, then $Q \succeq 0$. The ellipsoid $x^\top Qx \leq 1$ is *degenerate* (stretches out to infinity (is constant) in direction u_i).

Ellipsoids with linear terms

If $Q \succ 0$, then the quadratic form with extra linear term:

$$x^{\mathsf{T}}Qx + r^{\mathsf{T}}x + s$$

defines an *shifted* ellipsoid, whose center is not at 0. To see why, complete the square!
For scalars (ellipsoids in $\mathbb{R}^1$ are not very interesting), we have:

$$qx^2 + rx + s = q\left(x + \frac{r}{2q}\right)^2 + \left(s - \frac{r^2}{4q}\right)$$

In the matrix case, we have:

$$x^{\mathsf{T}}Qx + r^{\mathsf{T}}x + s = \left(x + \tfrac{1}{2}Q^{-1}r\right)^{\mathsf{T}} Q \left(x + \tfrac{1}{2}Q^{-1}r\right) + \left(s - \tfrac{1}{4}r^{\mathsf{T}}Q^{-1}r\right)$$

Ellipsoids with linear terms

Therefore, the inequality $x^\intercal Q x + r^\intercal x + s \leq b$ is equivalent to:

$$\left(x + \tfrac{1}{2}Q^{-1}r\right)^\intercal Q \left(x + \tfrac{1}{2}Q^{-1}r\right) \leq \left(b - s + \tfrac{1}{4}r^\intercal Q^{-1} r\right)$$

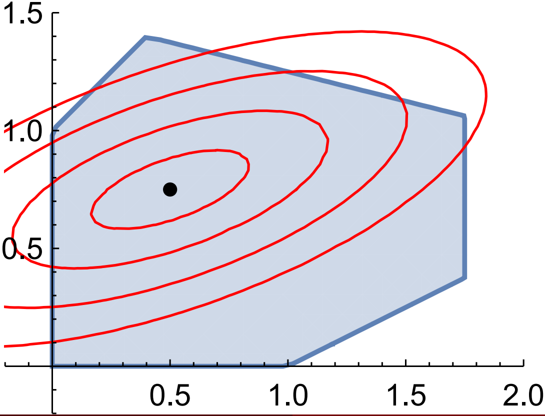
This is an ellipse centered at $-\frac{1}{2}Q^{-1}r$ — but its shape is still defined by the matrix Q.

Writing this using the matrix square root, we have:

$$\left\|Q^{1/2}x + \tfrac{1}{2}Q^{-1/2}r\right\|^2 \leq \left(b - s + \tfrac{1}{4}r^\intercal Q^{-1} r\right)$$

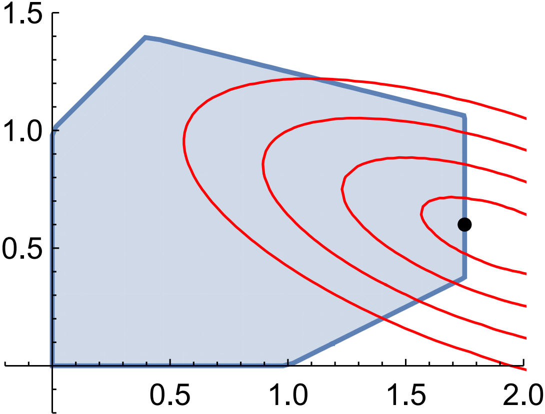
Quadratic programs

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\top P x + q^\top x + r \\ \text{subject to:} & Ax \leq b \end{array}$$



Quadratic programs

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\mathsf{T} P x + q^\mathsf{T} x + r \\ \text{subject to:} & Ax \leq b \end{array}$$



QCQPs

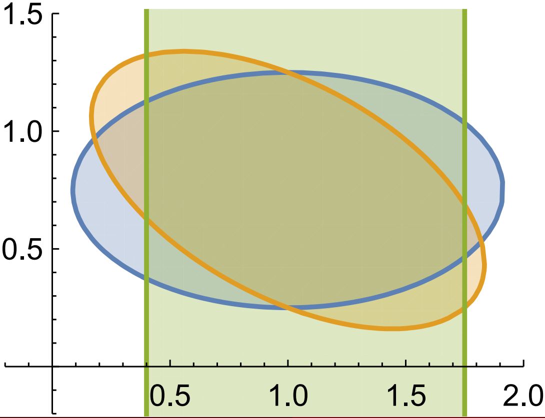
Quadratically constrained quadratic program (QCQP) has both a quadratic cost and quadratic constraints:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^\mathsf{T} P_0 x + q_0^\mathsf{T} x + r_0 \\ \text{subject to:} \quad & x^\mathsf{T} P_i x + q_i^\mathsf{T} x + r_i \le 0 \quad \text{for } i = 1, \dots, m \end{aligned}$$

- If $P_i \succeq 0$ for $i = 0, 1, \dots, m$, it is a *convex QCQP*
  - feasible set is convex
  - solution can be on boundary or in the interior
  - relatively easy to solve
- If any $P_i \not\succeq 0$, the QCQP becomes **very hard** to solve.

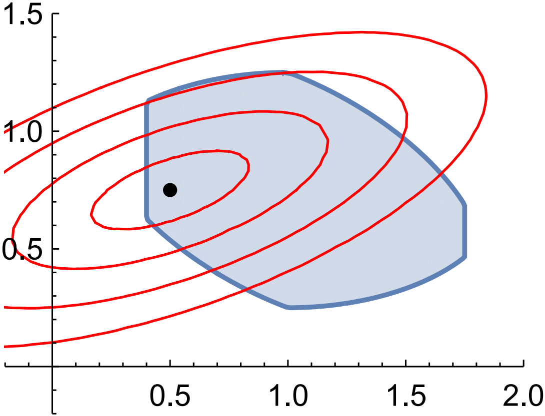
QCQPs

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\top P_0 x + q_0^\top x + r_0 \\ \text{subject to:} & x^\top P_i x + q_i^\top x + r_i \leq 0 \quad \text{for } i = 1, \ldots, m \end{array}$$



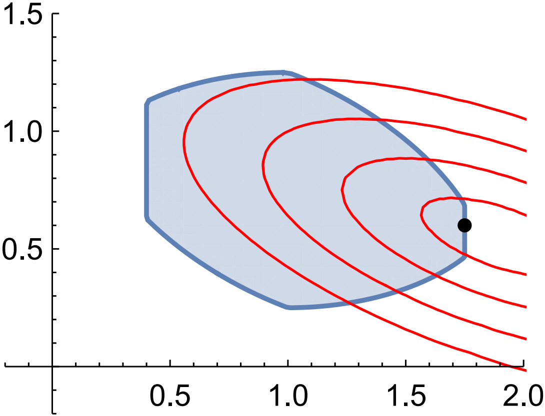
QCQPs

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^\mathsf{T} P_0 x + q_0^\mathsf{T} x + r_0 \\ \text{subject to:} \quad & x^\mathsf{T} P_i x + q_i^\mathsf{T} x + r_i \le 0 \quad \text{for } i = 1, \dots, m \end{aligned}$$



QCQPs

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^{\mathsf{T}} P_0 x + q_0^{\mathsf{T}} x + r_0 \\ \text{subject to:} & x^{\mathsf{T}} P_i x + q_i^{\mathsf{T}} x + r_i \leq 0 \quad \text{for } i = 1, \ldots, m \end{array}$$

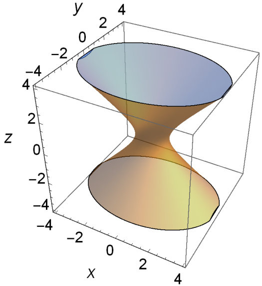


Difficult quadratic constraints

The following types of quadratic constraints make a problem nonconvex and generally difficult to solve (but not always).

Indefinite quadratic constraints.

- Example: $x^2 + 2y^2 - z^2 \leq 1$ corresponds to the nonconvex region on the right.
- **Note:** Be mindful of $\leq$ vs $\geq$! e.g. $x^2 + y^2 \geq 1$ is nonconvex.



Quadratic equalities.

- Using quadratic equalities, you can encode Boolean constraints. Example: $x^2 = 1$ is equivalent to $x \in \{-1, 1\}$. (There are many interesting problems with these kinds of variables!)