Chapter 1

Convex Optimization

1.1 Convex sets

1. The half-space $\Omega_i = \{x \in \mathbb{R}^n | a_i^T x \leq b_i\}$ is convex since it is a sublevel set of an affine (and therefore convex) function $f_i(x) = a_i^T x$
2. The wedge $\{x \in \mathbb{R}^n | a_1^T x \leq b_1, a_2^T x \leq b_2\}$ is convex since it is the intersection $\Omega_1 \bigcap \Omega_2$ of the convex sets Ω_i defined above
3. The set of points closer to a point than to a set:
   $\Omega(\mathcal{S}) = \{x \in \mathbb{R}^n | \quad ||x - x_0||_2 \leq ||x - y||_2 \, \forall y \in \mathcal{S}\}$ is convex since:
   - $f(x) = ||x||_2$ is a convex function
   - so for each $y \in \mathcal{S}$ the set $\Omega_y = \{x \in \mathbb{R}^n | \quad ||x - x_0||_2 \leq ||x - y||_2\}$ is a convex set, since it is the sublevel set of a convex function
   - so the intersection $\Omega(\mathcal{S}) = \bigcap_{y \in \mathcal{S}} \Omega_y$ is also convex
4. The set of points $\Omega(\mathcal{S}, \mathcal{T}) = \{x \in \mathbb{R}^n | \quad ||x - z||_2 \leq ||x - y||_2 \, \forall y \in \mathcal{S}, \forall z \in \mathcal{T}\}$ closer to one set $\mathcal{T}$ than another $\mathcal{S}$ is not convex. Counterexample in $\mathbb{R}^2$:
   Let $\mathcal{T} = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \middle| \quad x^2 + y^2 = 2 \right\} \cup \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$ and $\mathcal{S} = \left\{ \begin{bmatrix} x \\ y \end{bmatrix}^T \middle| \quad x^2 + y^2 = 1 \right\}$ where the points closer to $\mathcal{S}$ than $\mathcal{T}$ form a ring of width 1 which is not convex.

1.2 Convex functions

1. $f(x) = -\log(x)$ is convex on $\mathbb{R}_{++}$ since:

$$\begin{aligned} f(x) &= -\log(x) = \log(x^{-1}) \\ f'(x) &= -x^{-1} \\ f''(x) &= x^{-2} = \frac{1}{x^2} \\ &\implies \forall x \in \mathbb{R}_{++} : f''(x) > 0 \end{aligned}$$

   and the domain $\mathbb{R}_{++}$ is convex.

2. $f(x) = x^3$ is not convex on $\mathbb{R}$ since:

$$\begin{aligned} f(x) &= x^3 \\ f'(x) &= 3x^2 \\ f''(x) &= 6x \\ &\implies f''(-1) < 0 \wedge f''(1) > 0 \end{aligned}$$

3. $f(x_1, x_2) = \frac{1}{x_1 x_2}$ is convex on $\mathbb{R}^2_{++}$ since:

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{x_1 x_2} = x_1^{-1} x_2^{-1} \\ \nabla f(x_1, x_2) &= \begin{bmatrix} -x_1^{-2} x_2^{-1} \\ -x_1^{-1} x_2^{-2} \end{bmatrix} \\ \nabla^2 f(x_1, x_2) &= \begin{bmatrix} 2x_1^{-3} x_2^{-1} & x_1^{-2} x_2^{-2} \\ x_1^{-2} x_2^{-2} & 2x_1^{-1} x_2^{-3} \end{bmatrix} \end{aligned}$$

where on $\forall \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2_{++}$ it holds that:

$$\begin{aligned} 2x_1^{-3} x_2^{-1} &> 0 \\ x_1^{-2} x_2^{-2} &> 0 \\ 2x_1^{-1} x_2^{-3} &> 0 \\ \implies \nabla^2 f(x_1, x_2) &\succeq 0 \end{aligned}$$

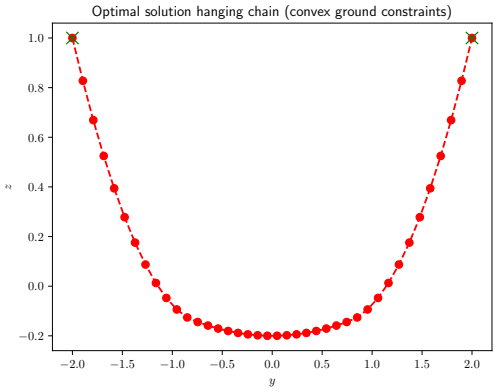
4. $f(x_1, x_2) = \frac{x_1}{x_2}$ is not convex on $\mathbb{R}^2_{++}$ since:

$$\begin{aligned} f(x_1, x_2) &= \frac{x_1}{x_2} = x_1 x_2^{-1} \\ \nabla f(x_1, x_2) &= \begin{bmatrix} x_2^{-1} \\ -x_1 x_2^{-2} \end{bmatrix} \\ \nabla^2 f(x_1, x_2) &= \begin{bmatrix} 0 & -x_2^{-2} \\ -x_2^{-2} & 2x_1 x_2^{-3} \end{bmatrix} \\ \begin{bmatrix} z_1 & z_2 \end{bmatrix} \begin{bmatrix} 0 & -x_2^{-2} \\ -x_2^{-2} & 2x_1 x_2^{-3} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} &= \begin{bmatrix} -x_2^{-2} z_2 & -x_2^{-2} z_1 + 2x_1 x_2^{-3} z_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ &= -\frac{z_1 z_2}{x_2^2} - \frac{z_1 z_2}{x_2^2} + 2\frac{x_1 z_2^2}{x_2^3} \\ &= -2\frac{z_1 z_2}{x_2^2} + 2\frac{x_1 z_2^2}{x_2^3} \end{aligned}$$

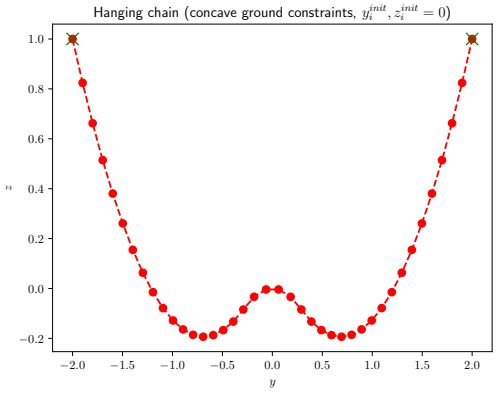
where $-2\frac{z_1 z_2}{x_2^2} + 2\frac{x_1 z_2^2}{x_2^3}$ is neither always non-positive nor always non-negative for arbitrary z_1, z_2 and all x_1, x_2

1.3 Hanging chain, revisited

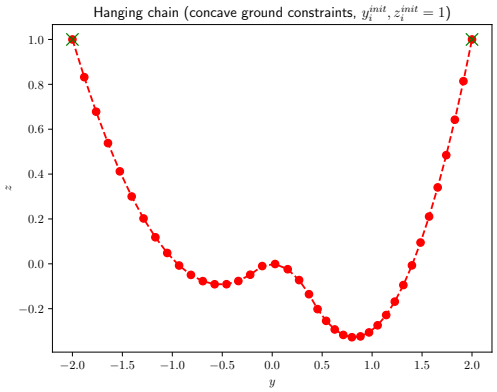
1. Since the energy function is convex quadratic and the ground constraints are convex quadtratic, this would be a convex QCQP
2. I expect concave quadratic constraints to make the problem non-convex
3. I expect the former problem to have a unique global and local minimum due to strict convexity, while the latter could have several local minima (intuitively the chain could hang down on either side of the 'hill' defined by the concave parabola)
4. See Figure 1.1
5. See Figure 1.1



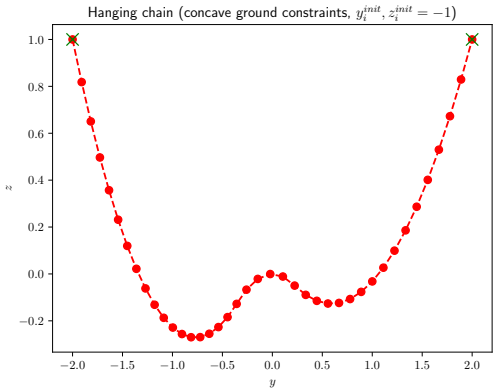
(a) Convex ground constraint $z_i \geq -0.2 + 0.1y_i^2$



(b) Concave ground constraint $z_i \geq -y_i^2$ with solver initialized at $y_i^{init} = z_i^{init} = 0$



(c) Concave ground constraint $z_i \geq -y_i^2$ with solver initialized at $y_i^{init} = z_i^{init} = 1$



(d) Concave ground constraint $z_i \geq -y_i^2$ with solver initialized at $y_i^{init} = z_i^{init} = -1$

Figure 1.1: Results for hanging chain energy minimization with differing inequality constraints, which form either a strictly convex problem with a unique solution (Figure 1.1a) or a problem with concave constraints that have multiple local minima, which can be explored by altering the initial values of the decision variables in the solver (Figure 1.1b, Figure 1.1c, Figure 1.1d)

Chapter 2

Minimum of a coercive function

ZZ: Let $f : \mathbb{R}^n \to \mathbb{R} \in C^0$ be a continuous, coercive function ($\lim_{||x|| \to \infty} f(x) = +\infty$), then there exists (at least) one global minimum.

- Choose any point $\tilde{x}$, then choose any $y > f(\tilde{x})$
- Let $\Omega := \{x \in \mathbb{R}^n | f(x) \leq y\}$ be the sublevel set of f with respect to y, which is non-empty by construction since $\tilde{x} \in \Omega$
- Ω is bounded since f is coercive and $\lim_{||x|| \to \infty} f(x) = +\infty$, there always exists a sphere of some radius r around the origin $\vec{\mathbf{0}}$ such that $\forall x \in \{x| \, ||x|| \geq r\} : f(x) \geq y$, so the sphere bounds all points with function values lower than some given value, here y, and therefore the sublevel set Ω
- Ω is closed since the sublevel set is defined through the inequality $\leq y$ (and not $< y$) and f is continuous (so there is no open region due to discontinuities)
- Ω is closed, bounded and non-empty - therefore, by the Weierstrass theorem, there exists a minimizer $x^* \in \Omega$ of f on Ω
- By construction, x^* is a minimizer of f not only on Ω but on all of $\mathbb{R}^n$, since Ω is a sublevel set and therefore $\forall x \in \mathbb{R}^n \backslash \Omega : f(x) \geq y$ ■

LAGRANGE DUALITY AND DUAL PROBLEMS

3.1 Logarithmic barrier

$$\begin{aligned} p^* = \min_{x \in \mathbb{R}^n} \quad & c^T x - \sum_{j=1}^{n} \log x_j \\ \text{s.t.} \quad & a^T x = b \end{aligned}$$

1.

$$\begin{aligned} \mathcal{L}(x, \lambda) &= c^T x - \sum_{j=1}^{n} \log x_j - \lambda^T \left(a^T x - b\right) \\ q(\lambda) &= \inf_{x \in \mathbb{R}^n} \quad L(x, \lambda) \\ &= \inf_{x \in \mathbb{R}^n} c^T x - \sum_{j=1}^{n} \log x_j - \lambda^T \left(a^T x - b\right) \\ &= \inf_{x \in \mathbb{R}^n} - \sum_{j=1}^{n} \log x_j + c^T x - \lambda^T a^T x + \lambda^T b \\ &= \lambda^T b + \inf_{x \in \mathbb{R}^n} - \sum_{j=1}^{n} \log x_j + (c - \lambda a)^T x \\ &= \lambda^T b + \inf_{x \in \mathbb{R}^n} \underbrace{\sum_{j=1}^{n} ((c - \lambda a)_j x_j - \log x_j)}_{=:k(x)} \\ \frac{\partial k}{\partial x_j} &= (c_j - \lambda_j a_j) - \frac{1}{x_j} \stackrel{!}{=} 0 \\ \implies x_j^* &\stackrel{!}{=} \frac{1}{c_j - \lambda_j a_j} \qquad \text{if } c_j - \lambda_j a_j > 0 \text{ and } k(x) \text{ convex} \\ q(\lambda) = \mathcal{L}(x^*, \lambda) &= \lambda^T b + \sum_{j=1}^{n} (1 + \log(c_j - \lambda_j a_j)) \end{aligned}$$

so the dual problem is

$$\begin{aligned} d^* = \max_{\lambda \in \mathbb{R}^n} \quad & n + \lambda^T b + \sum_{j=1}^{n} \log(c_j - \lambda_j a_j) \\ \text{s.t.} \quad & c - \lambda^T a > 0 \end{aligned}$$

2. $d^* = p^*$ due to strong duality, since Slater's condition holds (affine equality constraints) and the

primal problem is convex on $\mathbb{R}^n_{++}$ (where solutions are feasible):

$$\frac{\partial}{\partial x_j}(c_j x_j - \log x_j) = c_j - \frac{1}{x_j}$$
$$\frac{\partial^2}{\partial x_j^2}(c_j x_j - \log x_j) = \frac{1}{x_j^2}$$
$$\forall x_j \in \mathbb{R}_{++} : \frac{1}{x_j^2} > 0$$

3.2 Linear programming

1.

$$\begin{aligned} \min_{x \in \{0,1\}^n} \quad & -c^T x \\ \text{s.t.} \quad & Ax \geq b \end{aligned}$$

is equivalent to:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & -c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x - x \otimes x \end{aligned}$$

where $\otimes$ is component-wise multiplication and the equality constraint is equivalent to $\forall i \in \{1, \ldots, n\}$: $x_i(1 - x_i) = 0 \implies x \in \{0, 1\}^n$

2. The reformulation is not convex, since the feasible set due to the equality constraints is still $\{0, 1\}^n$ (intersected with some half-space defined by $Ax - b \geq 0$), which is not a convex set.

3. $\mathcal{L}(x, \lambda, \mu) = -c^T x - \lambda^T (x - x \otimes x) - \mu^T (Ax - b)$

4.

$$\frac{\partial}{\partial x}\mathcal{L}_i(x, \lambda, \mu) = -c_i + \lambda_i(2x_i - 1) - (A^T \mu)_i \stackrel{!}{=} 0$$
$$x_i^* = \frac{c_i + (A^T \mu)_i}{2\lambda_i} + \frac{1}{2}$$

for $\lambda_i \neq 0$, so the dual problem is:

$$\begin{aligned} \max_{\lambda,\mu} \left(\inf_x \mathcal{L}(x, \lambda, \mu) \right) &= \max_{\lambda,\mu} q(\lambda, \mu) \\ &= \max_{\lambda,\mu} = -c^T x^* - \lambda^T (x^* - x^* \otimes x^*) - \mu^T (Ax^* - b) \\ &\quad \text{s.t. } \mu \geq 0 \\ &\quad \forall i : \lambda_i \neq 0 \end{aligned}$$

5. The dual problem is always convex (since $q(\lambda, \mu)$ is always concave by Theorem 4.3)

6. Maximizing $q(\lambda, \mu)$ yields a lower bound to the ILP (Lemma 4.2), which could be used as the solution to the relaxed problem in a branch-and-bound approach to minimize the primal objective.

CHAPTER 4

FITTING PROBLEMS

4.1 Affine L_2 fitting

1. In matrix form, the least square problem can be denoted using $J \in \mathbb{R}^{n\times 2}$:

$$J = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{bmatrix}$$

2.

$$\begin{aligned}
\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} &= J^+ y = (J^T J)^{-1} J^T \\
&= \left(\begin{bmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \\
&= \begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{bmatrix} \\
&= \left(n \begin{bmatrix} \overline{x^2} & \overline{x} \\ \overline{x} & 1 \end{bmatrix} \right)^{-1} n \begin{bmatrix} \overline{xy} \\ \overline{y} \end{bmatrix} \\
&= \frac{n}{n^2 \left(\overline{x^2} - \overline{x}^2 \right)} \begin{bmatrix} 1 & -\overline{x} \\ -\overline{x} & \overline{x^2} \end{bmatrix} n \begin{bmatrix} \overline{xy} \\ \overline{y} \end{bmatrix} \\
&= \frac{1}{\overline{x^2} - \overline{x}^2} \begin{bmatrix} 1 & -\overline{x} \\ -\overline{x} & \overline{x^2} \end{bmatrix} \begin{bmatrix} \overline{xy} \\ \overline{y} \end{bmatrix} \\
&= \frac{1}{\overline{x^2} - \overline{x}^2} \begin{bmatrix} \overline{xy} - \overline{x} \cdot \overline{y} \\ -\overline{x} \cdot \overline{xy} + \overline{x^2} \cdot \overline{y} \end{bmatrix} \\
\hat{a} &= \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \\
\hat{b} &= \frac{\overline{x^2} \cdot \overline{y} - \overline{x} \cdot \overline{xy}}{\overline{x^2} - \overline{x}^2}
\end{aligned}$$

3. For fitting results see Figure 4.1a and fit with outliers in Figure 4.1b

4.2 Affine L_1 fitting

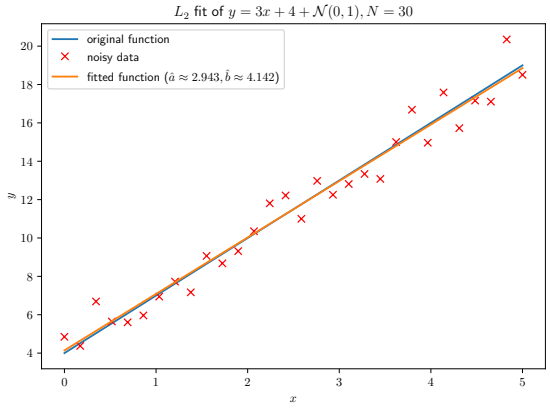
1.

$$\begin{aligned}
\min_{a,b\in\mathbb{R}, s\in\mathbb{R}^n} & \sum_{i=1}^{N} s_i \\
\text{s.t.} \quad & s_i + (ax_i + b - y_i) \geq 0 \quad \forall i \in \mathbb{N}_1^N \\
& s_i - (ax_i + b - y_i) \geq 0 \quad \forall i \in \mathbb{N}_1^N
\end{aligned}$$

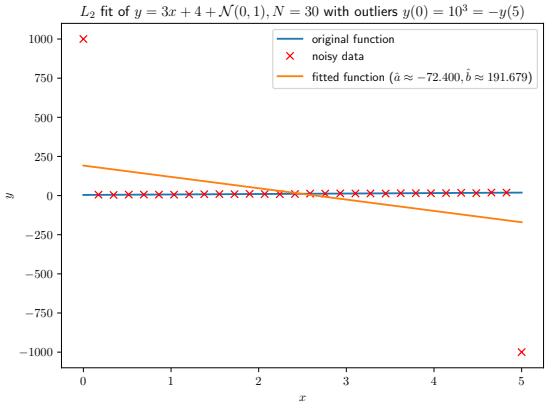
since the constraints imply $s_i \geq |ax_i + b - y_i|$ and therefore each s_i is an upper bound to the problem. As the slack variables are minimized, so is the L1 norm of the residuals.

2. For fitting results see Figure 4.1c and fit with outliers in Figure 4.1d

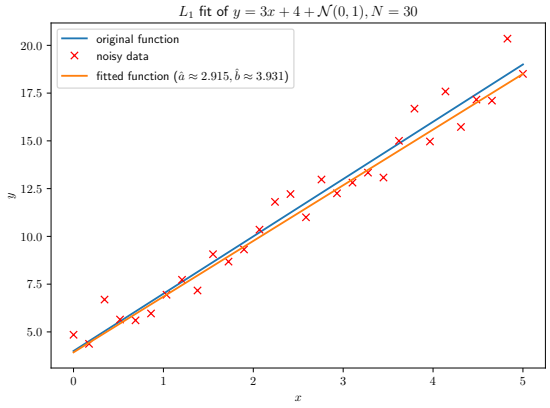
3. While the L2 norm seems to perform better in the absence of outliers, giving a more accurate estimation of the true function parameters, the L1 fit is much more robust to outliers.



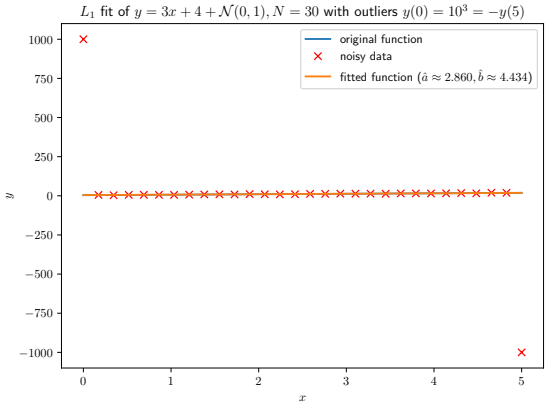
(a) L_2 fit to noisy data



(b) L_2 fit with outliers



(c) L_1 fit to noisy data



(d) L_1 fit with outliers

4.3 Regularized linear least squares

1.

$$
\begin{aligned}
f(x) &= \frac{1}{2}\,||y - Jx||_2^2 + \frac{\alpha}{2} x^T Q x \\
\nabla f(x) &= J^T J x - J^T \eta + \alpha Q x \\
\nabla^2 f(x) &= J^T J + \alpha Q
\end{aligned}
$$

so since $Q \succ 0$ the solution is unique ($\nabla^2 f(x) \succ 0$) if either $\alpha > 0$ or J is full column-rank and therefore the Gram matrix has $J^T J \succ 0$. If any of the two matrices $J^T J$ and αQ is positive definite, then since the other is at least positive semidefinite, the result of the summation must be positive definite.

2.

$$\begin{aligned}\nabla f(x) = J^T Jx - J^T\eta + \alpha Qx &\stackrel{!}{=} 0\\ (J^T J + \alpha Q)x &= J^T\eta\\ x^*(\alpha) &= (J^T J + \alpha Q)^{-1} J^T\eta\end{aligned}$$

3.

$$\begin{aligned}x^T Qx &\stackrel{!}{=} ||\tilde{x}||_2^2\\ &= x^T BBx && \text{let } B = \sqrt{Q} \succ 0 \text{ the unique principal square root}\\ &= x^T B^T Bx && B = B^T \text{since } Q, B \text{ symmetric}\\ &= (Bx)^T Bx\\ &= ||Bx||_2^2\\ \implies \tilde{x} &= Bx\end{aligned}$$

and therefore:

$$\begin{aligned}\tilde{J}\tilde{x} &\stackrel{!}{=} Jx\\ \tilde{J}Bx &= Jx && \text{insert } \tilde{x} = Bx\\ \tilde{J}B &= J\\ \implies \tilde{J} &= J\left(B^{-1}\right) && B^{-1} \text{ exists since } Q \succ 0 \implies BB \succ 0 \implies B \succ 0\end{aligned}$$

4. $x^*(\alpha) = \left(\tilde{J}^T\tilde{J} + \epsilon\mathbb{1}\right)^{-1}\tilde{J}^T y$ as shown in the previous exercise, so by Lemma 6.1 it follows that for $\alpha \to 0$ it holds that $x^*(\alpha) = \tilde{J}^+ y$ where $\tilde{J}^+$ is the Moore-Penrose inverse. Since that expression has no dependence on α, it follows trivially that $\lim_{\alpha\to 0} x^*(\alpha) = \tilde{J}^+ y$, meaning x^* converges to the vector $\tilde{J}^+ y$.

5. Since the Moore-Penrose pseudo inverse yields the unique solution to the problem:

$$\begin{aligned}x^* = \tilde{J}^+ y = J^+ y = \min_{x\in\mathbb{R}^n} &\frac{1}{2}||x||_2^2\\ \text{s.t.} \quad x &\in \mathcal{S}^*\end{aligned}$$

where $S^* = \left\{x \,\middle|\, \nabla\left(\frac{1}{2}||y - Jx||_2^2\right) = \vec{\mathbf{0}}\right\}$ then due to the constraint the solution x^* yielded by $x^* = \tilde{J}^+ y$ is an element of the solution set $\mathcal{S}^*$ of the unregularized minimization problem (namely that of least L_2 magnitude). Note that $\lim_{\alpha\to 0}\tilde{J} = J$.

Chapter 5

Source Code

5.1 Hanging Chain

```python
import casadi as ca
import matplotlib as mpl
import matplotlib.pyplot as plt

plt.rcParams.update({"text.usetex": True})

# PARAMETERS
N = 40              # number of masses
m = 4/N             # mass
Di = (70/40)*N      # spring constant
g0 = 9.81           # gravity

def create_opti():
    # create empty optimization problem
    opti = ca.Opti()
    y = opti.variable(N)
    z = opti.variable(N)
    # define the objective
    Vchain = 0
    for i in range(N):
        if i<N-1:
            Vchain += 1/2 * Di * ((y[i] - y[i+1])**2 + (z[i] - z[i+1])**2)
        Vchain += m*g0*z[i]
    # pass the objective to opti
    opti.minimize(Vchain)
    # add equality constraints
    opti.subject_to( y[0] == -2 )
    opti.subject_to( z[0] == 1 )
    opti.subject_to( y[N-1] == 2 )
    opti.subject_to( z[N-1] == 1 )
    return opti, y, z

def plot_solution(title, opti, y, z, init=0, save=""):
    # set solver to ipopt and solve the problem:
    opti.solver('ipopt')
    opti.set_initial(y,init)
    opti.set_initial(z,init)
    # opti.solver('sqpmethod')
    sol = opti.solve()
    # get solution and plot results
    Y = sol.value(y)
    Z = sol.value(z)
    fig, ax = plt.subplots()
    ax.plot(Y, Z, '--or')
    ax.plot(-2, 1, 'xg', markersize=10)
    ax.plot(2, 1, 'xg', markersize=10)
```

```
    ax.set_xlabel(r'$y$')
    ax.set_ylabel(r'$z$')
    ax.set_title(title)
    ## show plots
    plt.show()
    if save != "":
        fig.savefig(save+".pdf", dpi=fig.dpi)

# no ground constraints:
opti, y, z = create_opti()
plot_solution(r"Optimal solution hanging chain (without extra constraints)",opti, y, z, save="unconst

# linear ground constraints:
opti, y, z = create_opti()
opti.subject_to( z >= 0.5 )
opti.subject_to (z - 0.1*y >= 0.5)
plot_solution(r"Optimal solution hanging chain (linear ground constraints)", opti, y, z, save="linear

# convex quadratic ground constraints:
opti, y, z = create_opti()
opti.subject_to( z >= -0.2 + 0.1*y**2 )
plot_solution(r"Optimal solution hanging chain (convex ground constraints)", opti, y, z, save="convex

# concave quadratic ground constraints:
opti, y, z = create_opti()
opti.subject_to( z >= -y**2 )
plot_solution(r"Hanging chain (concave ground constraints, $y_i^{init},z_i^{init}=0$)", opti, y, z, i
plot_solution(r"Hanging chain (concave ground constraints, $y_i^{init},z_i^{init}=1$)", opti, y, z, i
plot_solution(r"Hanging chain (concave ground constraints, $y_i^{init},z_i^{init}=-1$)", opti, y, z,
```

5.2 Regression

```
import numpy as np
import casadi as ca
import matplotlib.pyplot as plt
plt.rcParams.update({"text.usetex": True,})

# setup
N = 30
np.random.seed(1312) # seed RNG for reproducible results

# generate noisy data
x = np.linspace(0.,5.,N)
y_ideal = 3*x+4
y = 3*x+4+np.random.randn(N)

# generate data with outliers
y_out = 3*x+4+np.random.randn(N)
y_out[0]=1e3
y_out[-1]=-1e3

# ordinary least squares regression
def reg(y):
    bar = lambda x: np.sum(x)/N
    a_hat = (bar(x*y) - bar(x)*bar(y)) / (bar(x**2) - bar(x)**2)
    b_hat = (bar(x**2)*bar(y) - bar(x) * bar(x*y)) / (bar(x**2) - bar(x)**2)
    return(a_hat, b_hat, a_hat*x+b_hat)
```

```
def reg_cas(y):
    opti = ca.Opti()
    s = opti.variable(N)
    a = opti.variable()
    b = opti.variable()
    # define objective function
    objective = 0
    for i in range(N):
        objective += s[i]
    opti.minimize(objective)
    # define constraints
    for i in range(N):
        opti.subject_to( s[i] + (a*x[i] + b - y[i])  >= 0)
        opti.subject_to( s[i] - (a*x[i] + b - y[i])  >= 0)
    # solve the problem
    opti.solver('ipopt')
    sol = opti.solve()
    a_hat = sol.value(a)
    b_hat = sol.value(b)
    return(a_hat, b_hat, a_hat*x+b_hat)


# plot results
def plot(title, filename, a_hat, b_hat, y):
    fig, ax = plt.subplots()
    ax.plot(x, y_ideal, label="original function")
    ax.plot(x, y, "rx", label="noisy data")
    ax.plot(x, y_reg, label="fitted function ($\\hat{{a}}\\approx{a:.3f}, \\hat{{b}}\\approx{b:.3f}$)
    ax.set_xlabel(r'$x$')
    ax.set_ylabel(r'$y$')
    ax.set_title(title)
    ax.legend()
    fig.tight_layout()
    ## show plots
    plt.show()
    fig.savefig(filename, dpi=fig.dpi)


# L2 FITS ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# perform regular regression
a_hat, b_hat, y_reg = reg(y)
plot(
    "$L_2$ fit of $y=3x+4+\\mathcal{{N}}(0,1), N={n}$".format(n=N),
    "linreg.pdf",
    a_hat,
    b_hat,
    y,
)

# introduce outliers
a_hat, b_hat, y_reg = reg(y_out)
plot(
    "$L_2$ fit of $y=3x+4+\\mathcal{{N}}(0,1), N={n}$ with outliers $y(0)=10^3=-y(5)$".format(n=N),
    "linregout.pdf",
    a_hat,
    b_hat,
    y_out,
)
```

```
# L1 FITS ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
a_hat, b_hat, y_reg = reg_cas(y)
plot(
    "$L_1$ fit of $y=3x+4+\\mathcal{{N}}(0,1), N={n}$".format(n=N),
    "linregl1.pdf",
    a_hat,
    b_hat,
    y,
)

a_hat, b_hat, y_reg = reg_cas(y_out)
plot(
    "$L_1$ fit of $y=3x+4+\\mathcal{{N}}(0,1), N={n}$ with outliers $y(0)=10^3=-y(5)$".format(n=N),
    "linregoutl1.pdf",
    a_hat,
    b_hat,
    y_out,
)
```