

Solutions to Part B of Problem Sheet 1

Solution (1.4)

- (a) The unit circle with respect to the ∞ -norm is the square with corners $(\pm 1, \pm 1)^\top$.
 (b) The trick in transforming the unconstrained problem

$$\text{minimize} \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_\infty \quad (5)$$

into a constrained linear programming problem is to characterise the ∞ -norm as the solution of a minimization problem. In fact, for any set of numbers x_1, \dots, x_n ,

$$\max_{1 \leq i \leq n} |x_i| = \min_{\forall i: |x_i| \leq t} t.$$

Put simply, the *maximum* of a set of non-negative numbers is the *smallest* upper bound on these numbers. We can further replace the condition $|x_i| \leq t$ by $-t \leq x_i \leq t$, so that the problem (??) becomes

$$\begin{aligned} &\underset{(\mathbf{x}, t)}{\text{minimize}} && t \\ &\text{subject to} && -t \leq \mathbf{a}_1^\top \mathbf{x} \leq t \\ & && \dots \\ & && -t \leq \mathbf{a}_m^\top \mathbf{x} \leq t, \end{aligned} \quad (6)$$

where \mathbf{a}_i^\top are the rows of the matrix \mathbf{A} . This problem can be brought into *standard form* by replacing each condition with the pair of conditions

$$\begin{aligned} \mathbf{a}_i^\top \mathbf{x} - t &\leq 0 \\ -\mathbf{a}_i^\top \mathbf{x} - t &\leq 0. \end{aligned}$$

The solution \mathbf{x} of Problem (??) can be read off the solution (\mathbf{x}, t) of Problem (??).

Solution (1.5)

- (a) This function is not convex. There are various ways of deriving this, for example, Theorem 2.4(2), where one verifies that the Hessian, or second derivative, is $-1/x^2$, which is not positive semidefinite.

Alternatively, one can also prove the statement using a pedestrian approach. We have to show that there are points $y \neq x$ and $\lambda \in [0, 1]$ such that

$$\log(\lambda x + (1 - \lambda)y) > \lambda \log(x) + (1 - \lambda) \log(y).$$

Let's choose $y = 0$. Then what needs to be shown is that for the points $\mathbf{p}_1 = (1, 0)$ and $\mathbf{p}_2 = (x, \log(x))$, the line joining \mathbf{p}_1 and \mathbf{p}_2 lies *below* the curve $(t, \log(t))$ between 1 and x . The line is given by the equation

$$\ell(t) = \frac{\log(x)}{x - 1}(t - 1).$$

Evaluating this, for example, at $x = 2$ and $t = 1.5$, one sees that $\ell(t) > \log(t)$, which is enough evidence that $\log(t)$ is not convex. With a little more effort one can deduce that the function is actually concave.

- (b) The function $f(x) = x^4$ is convex, as we will verify using Theorem 2.4. First, note that the derivative $4x^3$ is an increasing function with x . Given two points (x, x^4) and (y, y^4) with $y > x$, the line connecting them has slope $(y^4 - x^4)/(y - x)$. By the mean value theorem, there exists a $z \in (x, y)$ such that

$$\frac{y^4 - x^4}{y - x} = f'(z) = 4z^3 \geq 4x^3.$$

Rearranging this inequality, we get

$$f(y) - f(x) = y^4 - x^4 \geq 4x^3(y - x) = f'(x)(y - x),$$

which is precisely the criterium for convexity in Theorem 2.4(1).

- (c) Using Theorem 2.4(2), we compute the Hessian as

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

This matrix is positive semidefinite on \mathbb{R}_{++}^2 , since for all $\mathbf{x} \in \mathbb{R}_{++}^2$ we have

$$\mathbf{x}^\top \nabla^2 f(\mathbf{x}) \mathbf{x} = 2x_1x_2 > 0.$$

It follows that the function $f(\mathbf{x}) = x_1x_2$ is convex.

- (d) The Hessian matrix of $f(\mathbf{x}) = x_1/x_2$ is

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 0 & -\frac{1}{x_2^2} \\ -\frac{1}{x_2^2} & 2\frac{x_1}{x_2^3} \end{pmatrix}.$$

This matrix is not positive semidefinite for all valid values of \mathbf{x} (take for example $\mathbf{x} = (1, 1)^\top$, which leads to a negative eigenvalue).

- (e) The function $e^x - 1$ is convex, as is easily seen using Theorem 2.4(2) by computing the second derivative.
- (f) The function $f(\mathbf{x}) = \max_i x_i$ is convex. Here, we can't use the criteria from Theorem 2.4 since the function is not differentiable, so we have to verify convexity directly:

$$\max_i \lambda x_i + (1 - \lambda)y_i \leq \lambda \max_i x_i + (1 - \lambda) \max_i y_i.$$

Solution (1.6) After defining A and b , we can use the following code to generate a $2 \times s$ matrix, where s is the number of steps it takes to reach a certain accuracy:

```
function xout = graddescplot(A,b,x,tol)
    r = A'*(b-A*x);
    xout = [];
    while norm(r,2)>tol
        hold on;
        Ar = A*r;
        alpha = r'*r/(Ar'*Ar);
        x = x+alpha*r;
        xout = [xx,x];
        r = r-alpha*A'*Ar;
    end
end
```

We can then plot the trajectory as

```
plot(xx(1,:),xx(2,:), 'LineWidth',3);
```

which gives the plot

