Constrained and unconstrained optimisation

Questions 1–4 should be done by hand (i.e., using pen and paper). Their purpose is to familiarise yourself with the vocabulary and notation of optimisation problems, and to provide some insight into higher level concepts.

1. For each of the following scenarios, formulate the optimisation problem mathematically. You must clearly define each decision variable in the context of the problem. **You are not required to solve these problems.**

   (a) The Reddy Mikks company produces both interior and exterior paints from two raw materials, M_1 and M_2. The following table provides some basic information.

	Tons of raw material per ton of		Maximum daily availability (tons)
	exterior paint	interior paint	
Raw material, M_1	6	4	24
Raw material, M_2	1	2	6
Profit per ton ($1000)	5	4	

   Furthermore, a market survey indicates that the daily demand for interior paint does not exceed that for exterior paint by more than 1 ton. Also, the maximum daily demand for interior paint is 2 tons. Reddy Mikks wants to determine the optimum product mix of interior and exterior paints that maximizes the total daily profit.

   (b) The Reddy Mikks company is also in the process of designing a new paint can, which consists of an open-top cylindrical container and a circular lid. The manufacturing cost for the cylindrical section is 1.5 cents per square centimetre of material, while the cost for the lid is 4 cents per square centimetre of material. To accomodate the paint, the can is required to have a minimum capacity of 3.5 litres (which is 3,500 cubic centimetres). For storage reasons, the radius of the can should not exceed 10cm. Reddy Mikks must determine the size of the paint can to minimise material cost.

   (c) A furniture company sources their construction materials from two separate companies: *Wood You Believe It* and *Wood If I Could*. Each company has a maximum weekly output: 200 units of material for Would You Believe It, and 150 units for Wood If I Could. Wood You Believe It charges $50 per unit and asserts that at most 5% of their material comes with a manufacturing defect. Wood If I Could charges $70 per unit and asserts that at most 1% of their material comes with a manufacturing defect. The furniture company has a weekly budget of $15,000 and can tolerate at most 9 manufacturing defects. The furniture company wishes to maximise the amount of manufacturing materials available.

   (d) Each month, an office supply manufacturer rosters up to 300 hours of employee labour on its manufacturing lines. They are well known for their high quality branded pens and notepads and, due to overwhelming popularity, the manufacturer knows that every pen and notepad they make will be sold. The manufacturer sells packs, each containing two notepads and three pens. They are contracted to build at least 1,500 packs per month. Notepads cannot be purchased individually, but pens can be. The profit per pen is $2 and the profit per notepad is $3. It takes 2.5 minutes to produce a single pen and 1 minute to produce a single notepad. Naturally, the goal is to maximise profit.

   (e) Find the point(s) on the parabola with equation $y = x^2 - 1$ which are closest to the point $\mathbf{p} = (0, 1)^T$.

2. Consider the following optimisation problem:

$$\begin{aligned} \text{maximise} \quad & z = x_1 - x_2 \\ \text{subject to} \quad & x_1 + x_2 \leqslant 3 \\ & x_2 \leqslant 2 \\ & \mathbf{x} \geqslant \mathbf{0}. \end{aligned}$$

   Sketch the feasible region. Then, on the same diagram, draw the level sets of the objective function with $z = 0$, $z = 1$ and $z = 2$. Can you find or estimate the optimal solution using these level sets?

   This is an example of a linear programming problem, which we will see in more detail in Week 4.

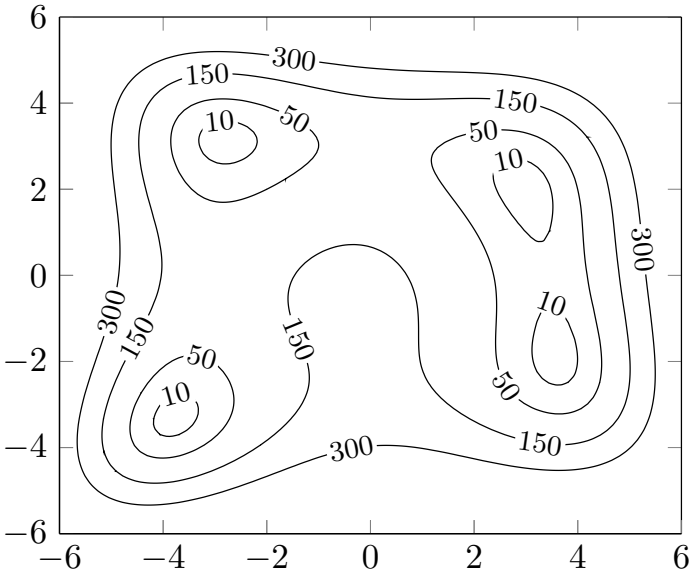
3. Consider the following optimisation problem:

$$\begin{aligned} \text{maximise} \quad & z = \tfrac{4x_1}{x_2 - 3} \\ \text{subject to} \quad & 0 \leqslant x_2 \leqslant 2 - x_1^2. \end{aligned}$$

Sketch the feasible region. Then, on the same diagram, draw three level sets of the objective function with $z = -1, z = 0$ and $z = 1$. Can you find or estimate the optimal solution using these level sets?

This is an example of a nonlinear optimisation problem with inequality constraints, which we will learn more about in Week 6.

4. The function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by $f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ is known as Himmelblau's function, and is a function that is commonly used for testing optimisation methods. Some level sets of f are shown below.



How many local minimisers does f appear to have? Use the diagram to estimate their coordinates.

Do you think inspecting level sets is the best way to solve optimisation problems?

Introduction to MATLAB

Questions 5 and 6 continue the previous question by illustrating some tools in MATLAB that can be useful in certain cases. You do not need to fully understand the syntax at this stage; we will learn more about that next week. For now, copy and paste the code into MATLAB.

5. The following MATLAB code can be used to plot Himmelblau's function:

```
func = @(x1, x2) (x1.^2 + x2 - 11).^2 + (x1 + x2.^2 - 7).^2;
[X,Y] = meshgrid(-6:1/10:6, -6:1/10:6);
Z = func(X,Y);
surf(X,Y,Z)
```

Run this code and investigate the plot. Does your answer to Question 4 change?

6. One way to perform optimisation in MATLAB uses the `fminsearch` function. In the code below, running `fminsearch(f2, [0,0])` will start at an initial point $(x_1, x_2) = (0, 0)$ and find an estimate for a local minimiser based on that starting point.

```
% fminsearch requires a function of a particular form.
func = @(x1, x2) (x1.^2 + x2 - 11).^2 + (x1 + x2.^2 - 7).^2;
f = @(x) func(x(1), x(2));
fminsearch(f, [0,0])
```

By adjusting the starting point, find all local minimisers of Himmelblau's function.

The `fminsearch` *function applies a technique called the downhill simplex method, which we will learn more about in Week 3.*

The remaining exercises are designed to familiarise yourself with the syntax and structure of MATLAB code. While the questions do not directly pertain to optimisation, working through them will introduce you to some of MATLAB's features that may be unconventional, yet highly advantageous, when compared to other programming languages.

7. Write MATLAB commands to do all of the following *without using any loops.*

   (a) Generate a random array `X` with a random size between 10 and 20 elements that contains integer values between -5 and 5.

   (b) Calculate the sum of all elements in `X`.

   (c) Calculate the product of the first and last elements of `X`, without referring to the length of `X`.

   (d) Calculate the sum of all elements in `X` that are divisible by 3.

   (e) Create an array that contains all elements of `X` but with any copies of the maximum element of `X` removed.

   (f) Create an array containing the elements in `X` that are greater than the mean of all elements in `X`.

   (g) Create an array that is obtained by repeating each element of `X` twice. For example, if `X = [1, 2, 3]` then the result should be `[1, 1, 2, 2, 3, 3]`.
   Hint: create an array of suitable length containing only zeros by using the `zeros` *function, then set the odd elements to* `X`*, then set the even elements to* `X`*.*

8. When solving a quadratic equation $ax^2 + bx + c = 0$ using the quadratic formula,

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

   the term $\Delta = b^2 - 4ac$ is called the *discriminant.* Write a MATLAB function `solveQuadratic` with input arguments `a`, `b` and `c` and two outputs, `roots` and `discriminant` in that order. The output variable `discriminant` contains the discriminant, and the output variable `roots` is a list containing two elements with the following conditions:

   - if both solutions are real (which occurs when $\Delta > 0$), then `roots` contains the two solutions sorted from smallest to largest,
   - if the solution is repeated (which occurs when $\Delta = 0$), then `roots` includes that solution twice,
   - if the solutions are complex conjugates (which occurs when $\Delta < 0$), then the first element of `roots` is the solution with negative imaginary part and the second element of `roots` is the solution with positive imaginary part.

   If needed, the functions `real` and `imag` output the real and imaginary parts of a complex number, but it is possible to complete this exercise without them.

9. Rewrite the function in Question 8 so that, instead of separate arguments `a`, `b` and `c`, there is just one input argument, `coefficients`, which is a list containing a in the first element, b in the second, and c in the third.

10. Use MATLAB to perform the following.

    (a) Generate two random arrays `A1` and `A2` of the same random size (between 5 and 10 elements), that contain integer values between -10 and 10.

    (b) Create a new array `A` by concatenating `A1` with `A2`.

    (c) Without using any loops, create a new array that is obtained by interleaving the elements of `A1` and `A2`. For example, if `A1 = [1,2,3,4,5]` and `A2 = [6,7,8,9,10]`, then the result is `[1,6,2,7,3,8,4,9,5,10]`.

    (d) Use a loop to create an array that contains the elements of `A` that appear exactly twice in `A`. The resulting array should include each of those elements twice, so if `A = [1,1,2,3,5,2,9]`, then the result is `[1,1,2,2]`.
    Hint: what does `sum([1, 2, 3, 3, 4, 5] == 3)` *tell you?*

    (e) Use a loop to create an array that is a copy of `A` with any duplicates removed.
    For example, if `A = [1,1,7,4,4,1]` then the result is `[1,7,4]`.
    Hint: the command `ismember(x,a)` *will return true if* `x` *is an element of the list* `a`.

    (f) Use a loop to create an array `counts` where `counts(i)` is the number of times `A(i)` appears in `A`.

    (g) Repeat parts (d) and (e) by using the result of part (f) and no further loops.

    (h) **Challenge**: can you do part (f) without using any loops?