# Optimization with MATLAB

Sofia Almirante, 1527718

Pau Reig, 1527816

January 2023

## Exercise 1

For this exercise, we are going to name $x_1$ and $x_2$ the number of ceramics and copper to produce, respectively. Since the maximum items the workshop can make every day is caped to 80, we deduce that $x_1 + x_2 \leq 80$. They also require that the total number of ceramic items must not exceed the total number of copper items by more than 30, which we can translate to $x_1 - x_2 \leq 30$. The workshop has only 160h available (among all people and kinds of items), and an item of ceramics is made in 1h, while a copper item is made in 4h. This can be translated to $x_1 + 4x_2 \leq 160$. We obviously assume also that the number of items produced must be a positive number, so $x_1, x_2 \geq 0$. Lastly, we know every ceramic item is sold for 200 € and every copper item is sold for 600 €, so the benefit of the workshop will be given by the expression $B = 200x_1 + 600x_2$. We have to convert this into a minimization problem in order to properly use the MATLAB tool. For this, we just need to change the sign of the expression of $B$ to obtain the objective function $f$. In summary, the proposed optimization problem can be written mathematically as follows:

$$
\begin{aligned}
\min_{x_1, x_2} \quad & -200x_1 - 600x_2 \\
\text{s.t.} \quad & x_1 + x_2 \leq 80, \\
& x_1 - x_2 \leq 30, \\
& x_1 + 4x_2 \leq 160, \\
& x_1, x_2 \geq 0
\end{aligned} \tag{1}
$$

We can see this is a constrained linear multidimensional optimization problem, which is why we are going to solve it with the *linprog* MATLAB function. The code used to solve this program is the following one:

```
f = [-200 -600];
A = [1 1; 1 -1 ; 1 4];
b = [80 30 160];
lb = zeros(1,2);
[xsol , Fsol] = linprog(f, A, b, [], [], lb)
```

This provides the solutions $x_1 = 53.33$ and $x_2 = 26.67$. It can make sense to talk about decimal items produced if we have the option to leave an uncompleted item at the end of the day and finish it the next one (although it could be difficult to measure). In this case, if the building process can be divided in three steps this steps, be a good solution that maximize the benefits of the workshop when considering the benefits over time.

On the other hand, if we want to maximize the benefits produced in a day, or it is not possible to create an item in different days, we can check the benefits obtained with the two nearest full numbers. Choosing the values $x_1 = 53$ and $x_2 = 27$ provides a benefit $B = 26800$, while $x_1 = 54$ and $x_2 = 26$ provides a benefit of $B = 26400$. Despite the first solution shows a bigger benefit, it does not satisfy all the constraint conditions because $x_1 + 4x_2 = 161 > 160$. For this reason, we conclude the solution for this optimization problem is:

$$\boxed{x_1 = 54, x_2 = 26}$$

In Fig. 1, we have represented the constraints of this optimization problem in a way that the painted surface includes all the points that satisfy the constraint conditions at the same time. As we know from linear programming, the optimal value is a vertex of this region. In this case we can check that the maximum correspond to the intersection of the lines in purple and blue. We can then conclude our results are coherent with the expected results and satisfy all the constraint conditions.

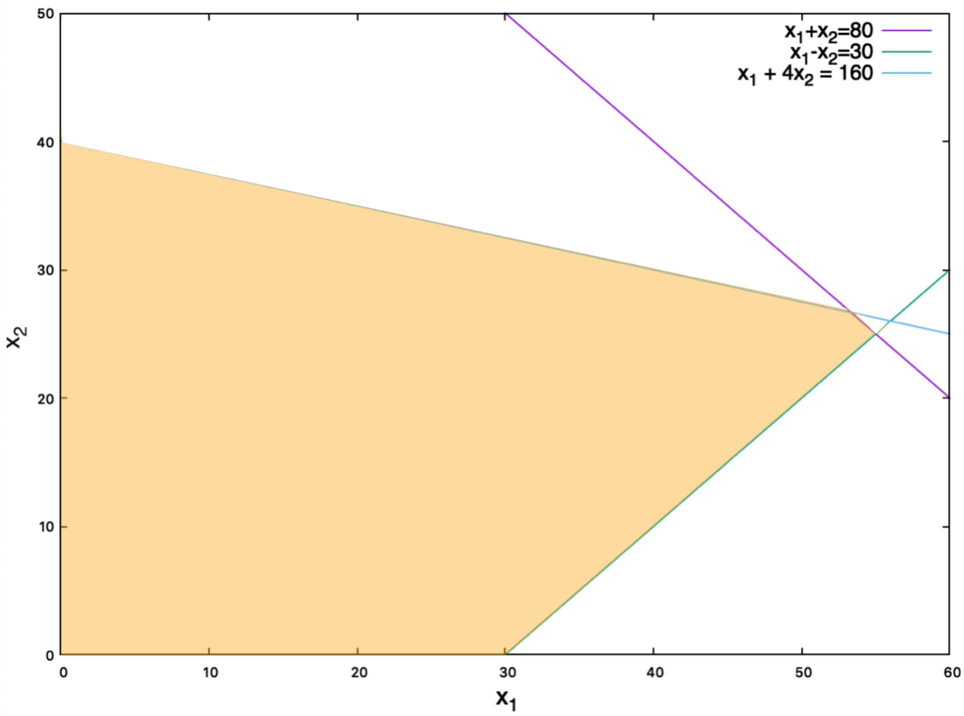


Figure 1: Graphical constraints of the optimization problem.

# Exercise 2

In this problem, we are asked to optimize the inflation rate given by the function

$$I = xM_o e^{-\alpha x},$$

with the value $\alpha = 0.2$. Since the value of $M_0$ is unknown, we are going to study in a first place how the function behaves in terms of it.
We know the derivative of the function in the singular points is equal to 0. Using this, we can show that the singular point will not depend on the $M_0$ value. In general, considering $I = M_0 f(x)$:

$$\frac{dI}{dx} = M_0 f'(x) = 0 \Longrightarrow f'(x) = 0 \qquad \forall M_0,$$

For this reason, we are going to proceed assuming $M_0 = 1$.

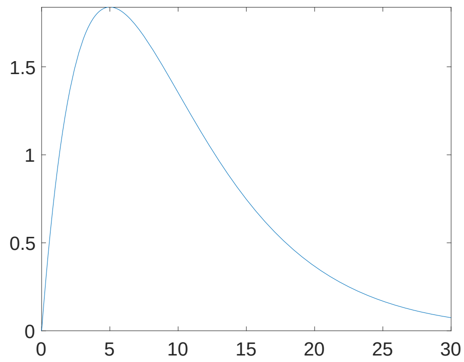


Figure 2: I(x) with $M_0 = 1$.

We can see that the optimum point of the function is a maximum. That is why we are going to change the sign of the function in order to treat it as a minimization problem in order to use the corresponding MATLAB tool. Overall, the objective function results in:

$$f = -xe^{-0.2x}. \tag{2}$$

We can see that this is an unconstrained monodimensional problem, so we are going to use the *fminbnd* MATLAB function within the fixed interval [0,100] to solve it as follows:

```
f = @(x) -x*exp(-0.2*x);
[xsol2, ysol2] = fminbnd(f,0,100)
```

The results provided after executing this code have been $x = 5.0$, $f(x) = -1.8394$, which are compatible with the behavior of the function shown in Fig. 2.
So finally, we can give a general solution of the original problem as:

$$\boxed{x^* = 5.0 \qquad f(x^*) = 1.8394 M_0}$$

## Exercise 3

In this exercise, we want to maximize the benefit of three companies. The benefit of a company that sells $x$ number of products is always calculated by the formula

$$\text{Benefit} = x[\text{Price}(x)] - \text{Cost}(x)), \tag{3}$$

where we have assumed that the Price is the value per unit and the cost is the cost of the production for $x$ products.

We can use this idea to find the fitness function of our problem. In our case, the total benefit of the three companies will be represented by $B$, that will be the sum of the three benefits of each company ($b_j$).

$$B = \sum_{j=1}^{3} b_j.$$

Each benefit will be computed using Eq. 3 where, in our case, the number of products sold will be $p_j$, the price (per unit) of the product for each company will be $V_j$ and the total cost $C_j$. Therefore, the total benefit can be computed as

$$B = \sum_{j=1}^{3} (p_j V_j - C_j).$$

Now, we can apply the information given to relate $V_j$ and $C_j$ with the independent variables.

In our case we know that

$$V_1 = 12 - p_1, \qquad V_2 = 20 - 1.5p_2, \qquad V_3 = 28 - 2.5p_3.$$

And that:

$$C_1 = 3p_1^{1.3}, \qquad C_2 = 5p_2^{1.2}, \qquad C_3 = 6p_3^{1.15}.$$

We also have some constraints. First of all, the amount of water accessible is restricted by the river flow $Q$ and the minimum water that has to be kept on the river $R$, such that

$$x_1 + x_2 + x_3 + R \leq Q,$$

where $x_j$ is the amount of water allocated to each company. Another constraint is the total amount of products that can be produced for each company which is given by the expressions $P_j(x_j)$, such that $p_j \leq P_j(x_j)$. And finally we have to consider (obviously) only positive values for $p_j$ and $x_j$ therefore $p_j, x_j \geq 0$.

With all of this, we can write our maximization problem as:

$$\begin{aligned}
\max_{x_1, x_2, x_3, p_1, p_2, p_3} \quad & p_1[12 - p_1] - 3p_1^{1.3} + p_2[20 - 1.5p_2] - 5p_2^{1.2} + p_3[28 - 2.5p_3] - 6p_3^{1.15} \\
\text{s.t.} \quad & x_1 + x_2 + x_3 \leq Q - R, \\
& p_1 - 0.4x_1^{0.9} \leq 0, \\
& p2 - 0.5x_2^{0.8} \leq 0, \\
& p_3 - 0.6x_3^{0.7} \leq 0, \\
& x_j, p_j \geq 0 \quad j = 1, 2, 3
\end{aligned} \tag{4}$$

### Matlab

Since this is a constrained nonlinear multidimensional problem, we have used the *fmincon* MATLAB function. The code used is the following:

```
X0 = [0.2 0.4 0.4 0.1 0.15 0.2];
QR = 10;
A = [1 1 1 0 0 0];
B = QR;
LB = zeros(1,6);
NONLCON = @ConstraintsD4_3;
FUN = @FunctionD4_3;
X = fmincon(FUN,X0,A,B,[],[],LB,[],NONLCON)
B = -FunctionD4_3(X)
```

The function has been defined in a separate file as:

```
function[fx] = FUN(xv)
    p1 = xv(4);
    p2 = xv(5);
    p3 = xv(6);
    C1 = 3*(p1)^1.3;
    C2 = 5*(p2)^1.2;
    C3 = 6*(p3)^1.15;
    V1 = 12-p1;
    V2 = 20-1.5*p2;
    V3 = 28-2.5*p3;

    fx = - p1*(V1) - p2*(V2) - p3*(V3) + C1 + C2 + C3;
end
```

Lastly, the constraint function has also been created:

```
function[C,Ceq] = ConstraintsD4_3(xc)
    C(1) = - 0.4*xc(1)^(0.9) + xc(4);
    C(2) = - 0.5*xc(2)^(0.8) + xc(5);
    C(3) = - 0.6*xc(3)^(0.7) + xc(6);

    Ceq = [];
end
```

After the execution of the code, the solution provided (using $QR = 10$) a benefit of $B = 53.47$ with the solutions:

| | | |
|---|---|---|
| $p_1 = 0.4564$ | $p_2 = 1.4358$ | $p_3 = 1.8780$ |
| $x_1 = 1.1578$ | $x_2 = 3.7383$ | $x_3 = 5.1039$ |

## Lagrange multipliers

For using the Lagrange multipliers method, we have to modify the problem assuming that water is a binding constraint and all the variables are positive. Considering this, our problem now looks like this:

$$\begin{aligned} \max_{x_1, x_2, x_3, p_1, p_2, p_3} \quad & p_1[12 - p_1] - 3p_1^{1.3} + p_2[20 - 1.5p_2] - 5p_2^{1.2} + p_3[28 - 2.5p_3] - 6p_3^{1.15} \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + R - Q = 0, \\ & p_1 - 0.4x_1^{0.9} = 0, \\ & p_2 - 0.5x_2^{0.8} = 0, \\ & p_3 - 0.6x_3^{0.7} = 0 \end{aligned} \tag{5}$$

With this, we can construct the Lagrangian function adding a Lagrange multiplier $\lambda_i$ for each constraint:

$$\mathcal{L}(p_j, x_j, \lambda_i) = p_1[12 - p_1] - 3p_1^{1.3} + p_2[20 - 1.5p_2] - 5p_2^{1.2} + p_3[28 - 2.5p_3] - 6p_3^{1.15} -$$

$$-\lambda_1[x_1 + x_2 + x_3 + R - Q] - \lambda_2[p_1 - 0.4x_1^{0.9}] - \lambda_3[p2 - 0.5x_2^{0.8}] - \lambda_4[p_3 - 0.6x_3^{0.7}]$$

And now we have to compute $\nabla\mathcal{L} = 0$:

$$\begin{aligned} &\tfrac{\partial\mathcal{L}}{\partial p_1} = 12 - 2p_1 - 1.3 \cdot 3p_1^{0.3} - \lambda_2 = 0 && \tfrac{\partial\mathcal{L}}{\partial p_2} = 20 - 3p_2 - 1.2 \cdot 5p_2^{0.2} - \lambda_3 = 0 \\ &\tfrac{\partial\mathcal{L}}{\partial p_3} = 28 - 5p_3 - 1.15 \cdot 6p_3^{0.15} - \lambda_4 = 0 && \tfrac{\partial\mathcal{L}}{\partial x_1} = 0.9 \cdot 0.4x_1^{-0.1}\lambda_2 - \lambda_1 = 0 \\ &\tfrac{\partial\mathcal{L}}{\partial x_2} = 0.8 \cdot 0.5x_2^{-0.2}\lambda_3 - \lambda_1 = 0 && \tfrac{\partial\mathcal{L}}{\partial x_3} = 0.7 \cdot 0.6x_3^{-0.3}\lambda_4 - \lambda_1 = 0 \\ &\tfrac{\partial\mathcal{L}}{\partial \lambda_1} = x_1 + x_2 + x_3 + R - Q = 0 && \tfrac{\partial\mathcal{L}}{\partial \lambda_2} = p1 - 0.4x_1^{0.9} = 0 \\ &\tfrac{\partial\mathcal{L}}{\partial \lambda_3} = p_2 - 0.5x_2^{0.8} = 0 && \tfrac{\partial\mathcal{L}}{\partial \lambda_4} = p_3 - 0.6x_3^{0.7} = 0 \end{aligned}$$

We can use the matlab tool *fsolve* to solve this system with the following code:

```
x0 = [1 1 1 1 1 1 1 1 1 1];
RQ = 10;
fun = @(x) Lagrange(x,RQ);
xSol = fsolve(fun, x0)
function [L] = Lagrange(x,RQ)
    L(1) = 12 - 2*x(1) - 1.3*3*x(1)^(0.3) - x(8);
    L(2) = 20 - 3*x(2) - 1.2*5*x(2)^(0.2) - x(9);
    L(3) = 28 - 5*x(3) - 1.15*6*x(3)^(0.15) - x(10);
```

```
    L(4) = x(8)*0.9*0.4*x(4)^(-0.1) - x(7);
    L(5) = x(9)*0.8*0.5*x(5)^(-0.2) - x(7);
    L(6) = x(10)*0.7*0.6*x(6)^(-0.3) - x(7);
    L(7) = x(1) - 0.4*x(4)^(0.9);
    L(8) = x(2) - 0.5*x(5)^(0.8);
    L(9) = x(3) - 0.6*x(6)^(0.7);
    L(10) = x(4) + x(5) + x(6) - RQ;
end
```

Where we have defined $QR$ as $Q - R$

This give us the following solution for our problem (using $QR = 10$):

| | | |
|---|---|---|
| $p_1 = 0.4564$ | $p_2 = 1.4358$ | $p_3 = 1.8780$ |
| $x_1 = 1.1578$ | $x_2 = 3.7383$ | $x_3 = 5.1039$ |

That we can see that fits perfect with the solution obtained with *fmincon*.