Teaching Students Nonlinear Programming with Computer Algebra System

Włodzimierz Wojas · Jan Krupa

Received: 1 December 2017 / Revised: 25 May 2018 / Accepted: 1 June 2018 / Published online: 6 August 2018 © The Author(s) 2018

Abstract This paper presents several didactic examples of the nonlinear programming (NLP) problems solved with Mathematica. We solved examples of Karush–Kuhn–Tucker necessary conditions, Lagrange multipliers method, convex optimization, and graphical method. We compared the hand calculation in Karush–Kuhn–Tucker method with Lagrange multipliers method. The paper contains Mathematica symbolic codes used for Karush–Kuhn–Tucker necessary conditions and the Hessian analysis in convex optimization. We present also some didactic graphs for various aspects of NLP problems using plots and dynamic plots. The use of Mathematica during teaching students about NLP by Computer Algebra System (CAS) seems to be very useful both as the calculations support (checking hand calculation) and when creating didactic graphical visualizations using dynamic plots. We did not find in available literature any similar example of NLP problems solved with CAS or the use of dynamic plots.

Keywords Higher education · Mathematical didactics · Nonlinear programming · Mathematical programming · Application of CAS · Mathematica

Mathematics Subject Classification 97B40 · 97R20 · 90C30 · 97I60

1 Introduction

Elements of NLP are taught during the following university courses: mathematical analysis, mathematical programming, operations researches, and optimizations methods. Usually, this includes Lagrange multipliers method, Karush–Kuhn–Tucker necessary conditions, and some elements of convex optimization. This paper presents some didactic examples of NLP problems solved with Mathematica [6,8]. We present Mathematica calculations as the support for NLP teaching using CAS, procedures allowing to solve the problems, and graphical visualizations.
of some aspects of used methods. This paper presents two examples of NLP problems for Karush–Kuhn–Tucker necessary conditions, one example of NLP problem for Lagrange multipliers method, three examples of NLP problems for graphical method only, and one example of NLP problem for convex optimization. We compare the hand calculation in Karush–Kuhn–Tucker necessary conditions with calculation in Lagrange multipliers method.

Introduction to NLP theory is presented in [1–5,7].

We will consider NLP problems in the following form:

\[
\begin{align*}
\text{minimize} & \quad f(x_1, x_2, \ldots, x_n) \\
\text{subject to} & \quad g_i(x_1, x_2, \ldots, x_n) \geq 0, \ i = 1, 2, \ldots, m, \\
& \quad (x_1, x_2, \ldots, x_n) \in X, \\
\end{align*}
\]

where \( n \) and \( m \) are positive integers, \( X \) is a subset of \( \mathbb{R}^n \) and \( f, g_i \) are real-valued functions on \( X \) with at least one function of \( f, g_i \) \((i = 1, 2, \ldots, m)\) being nonlinear. A feasible region of the NLP problem is defined as a set of all possible points that satisfy all the problem’s constrains.

2 Karush–Kuhn–Tucker Necessary Conditions

We start with the following theorem [1]:

**Theorem** (Karush–Kuhn–Tucker Necessary Conditions) Let \( X \) be a nonempty open set in \( \mathbb{R}^n \), and let \( f : \mathbb{R}^n \mapsto \mathbb{R} \) and \( g_i : \mathbb{R}^n \mapsto \mathbb{R} \) for \( i = 1, \ldots, m \). Consider the Problem (P) to minimize \( f(x) \) subject to \( g_i(x) \geq 0 \) for \( i = 1, \ldots, m \). Let \( \bar{x} \) be a feasible solution, and denote \( I = \{ i : g_i(\bar{x}) = 0 \} \). Suppose that \( f \) and \( g_i, \) for \( i = 1, 2, \ldots, m \) are differentiable at \( \bar{x} \). Furthermore, suppose that \( \nabla g_i(\bar{x}) \) for \( i \in I \) are linearly independent. If \( \bar{x} \) solves Problem (P) locally, there exist scalars \( \lambda_i, \) for \( i = 1, 2, \ldots, m \) such that:

\[
\begin{align*}
\nabla f(\bar{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\bar{x}) &= 0 \quad (1) \\
\lambda_i \nabla g_i(\bar{x}) &= 0 \quad \text{for } i = 1, 2, \ldots, m \quad (2) \\
\lambda_i &\geq 0 \quad \text{for } i = 1, 2, \ldots, m. \quad (3)
\end{align*}
\]

**Example 1** Using Karush–Kuhn–Tucker necessary conditions we determine global minimum and maximum function \( f(x, y) = y - x^2 \) subject to: \(-x^2 - 4y^2 + 1 \geq 0, -1 + x + 2y \geq 0\).

Formally, Karush–Kuhn–Tucker necessary conditions require an independent solution of two tasks—one for minimum and one for maximum. We first solve the NLP problem for minimum and next the NLP problem for maximum.

**a)** Consider the following problem:

\[
\begin{align*}
\text{minimize} & \quad y - x^2 \\
\text{subject to} & \quad -x^2 - 4y^2 + 1 \geq 0, \\
& \quad -1 + x + 2y \geq 0. \\
\end{align*}
\]

Let \( g_1(x, y) = -x^2 - 4y^2 + 1 \) and \( g_2(x, y) = -1 + x + 2y \).

\( \nabla f(x, y) = \lambda_1 \nabla g_1(x, y) + \lambda_2 \nabla g_2(x, y) \) is equivalent to:

\( [-2x, 1] = \lambda_1 [-2x, -8y] + \lambda_2 [1, 2] \). So we must solve the following system:
\begin{align*}
-2x &= -2\lambda_1 x + \lambda_2 \\
1 &= -8\lambda_1 y + 2\lambda_2 \\
-x^2 - 4y^2 + 1 &\geq 0 \\
-1 + x + 2y &\geq 0 \\
\lambda_1(-x^2 - 4y^2 + 1) &= 0 \\
\lambda_2(-1 + x + 2y) &= 0
\end{align*}
\hfill (5)

where \( \lambda_1 \geq 0, \lambda_2 \geq 0. \)

\textbf{Case 1.} Let \( g_1(x, y) = 0 \) and \( g_2(x, y) = 0. \) Hence \( x^2 + 4y^2 = 1 \) and \( x + 2y = 1. \) So \( (x = 1 \) and \( y = 0) \) or \( (x = 0 \) and \( y = 1/2). \) If \( (x = 1 \) and \( y = 0) \) then \( 2\lambda_2 = 1, \) hence \( \lambda_2 = 1/2 \geq 0 \) and \( 2\lambda_1 = 1/2 + 2, \) hence \( \lambda_1 = 5/4 \geq 0. \) \( \nabla g_1(1, 0) = [-2, 0], \nabla g_2(1, 0) = [1, 2] \) are linearly independent. If \( (x = 0 \) and \( y = 1/2) \) then \( \lambda_2 = 0 \) and \( 1 = -4\lambda_1, \) hence \( \lambda_1 = -1/4 < 0 \) contradicts \( \lambda_1 \geq 0. \)

\textbf{Case 2.} Let \( g_1(x, y) = 0 \) and \( g_2(x, y) > 0. \) Hence \( \lambda_2 = 0, \) hence

\begin{align*}
-2x &= -2\lambda_1 x \\
1 &= -8\lambda_1 y \\
-x^2 - 4y^2 + 1 &= 0.
\end{align*}
\hfill (6)

It follows that \( 2(x(\lambda_1-1) = 0. \) Hence \( (x = 0 \) and \( y = \pm 1/2) \) or \( \lambda_1 = 1. \) If \( (x = 0 \) and \( y = 1/2) \) then \( \lambda_1 = -1/4 < 0 \) contradicts \( \lambda_1 \geq 0. \) If \( (x = 0 \) and \( y = -1/2) \) then \( \lambda_1 = 1/4 > 0 \) and \( g_2(0, -1/2) = -1 + 0 - 1 = -2 \) contradicts assumption in current \textbf{Case 2} \( g_2(x, y) > 0. \) If \( \lambda_1 = 1 \geq 0 \) then \( x \) can be any real number and \( y = -1/8 \) hence \( g_1(x, -1/8) = -x^2 - 4(-1/8)^2 + 1 = 0. \) So \( x = \pm \sqrt[4]{3} \).

But \( g_2(\sqrt[4]{3}/4, -1/8) = -1 + \sqrt[4]{3}/4 - 1/4 = \sqrt[4]{3}-5/4 < 0 \) contradicts \( g_2(x, y) > 0 \) and \( g_2(-\sqrt[4]{3}/4, -1/8) = -1 - \sqrt[4]{3}/4 - 1/4 < 0 \) contradicts \( g_2(x, y) > 0. \)

\textbf{Case 3.} Let \( g_1(x, y) > 0 \) and \( g_2(x, y) = 0. \) Hence \( \lambda_1 = 0, \) hence \(-2x = \lambda_2 \) and \( 1 = 2\lambda_2. \) So \( \lambda_2 = 1/2 \geq 0 \) and \( x = -1/4 \) hence \( g_2(-1/4, x) = -1 - 1/4 + 2y = 0, \) hence \( x = -1/4 \) and \( y = 5/8. \) But \( g_1(-1/4, 5/8) = -(1/4)^2 - 4(5/8)^2 + 1 = -1/16 - 25/16 + 1 = -5/8 < 0 \) contradicts assumption in current \textbf{Case 3} \( g_1(-1/4, 5/8) > 0. \)

\textbf{Case 4.} Let \( g_1(x, y) > 0 \) and \( g_2(x, y) > 0. \) Hence \( \lambda_1 = \lambda_2 = 0, \) hence \(-2x = 0 \) and \( 1 = 0 \) a contradiction.

From \textbf{Cases 1, 2, 3, 4} it follows that the only point which satisfies Karush-Kuhn-Tucker conditions is \((1, 0), \lambda_1 = 5/4, \lambda_2 = 1/2.\)

\textit{b)} Consider the following problem:

\[
\begin{align*}
\text{maximize} & \quad y - x^2 \\
\text{subject to:} & \quad -x^2 - 4y^2 + 1 \geq 0, \\
& \quad -1 + x + 2y \geq 0.
\end{align*}
\hfill (7)
\]

Let \( g_1(x, y) = -x^2 - 4y^2 + 1 \) and \( g_2(x, y) = -1 + x + 2y. \)
\[ -\nabla f(x, y) = \lambda_1 \nabla g_1(x, y) + \lambda_2 \nabla g_2(x, y) \]
is equivalent to:

\[
[2x, -1] = \lambda_1[-2x, -8y] + \lambda_2[1, 2].
\]

So we must solve the following system:

\begin{align*}
2x &= -2\lambda_1 x + \lambda_2 \\
-1 &= -8\lambda_1 y + 2\lambda_2 \\
-x^2 - 4y^2 + 1 &\geq 0 \\
-1 + x + 2y &\geq 0 \\
\lambda_1(-x^2 - 4y^2 + 1) &= 0 \\
\lambda_2(-1 + x + 2y) &= 0
\end{align*}
\hfill (8)

where \( \lambda_1 \geq 0, \lambda_2 \geq 0. \)
Case 1. Let \( g_1(x, y) = 0 \) and \( g_2(x, y) = 0 \). Hence \( x^2 + 4y^2 = 1 \) and \( x + 2y = 1 \). So \((x = 1 \text{ and } y = 0) \) or \((x = 0 \text{ and } y = 1/2)\).

If \((x = 1 \text{ and } y = 0)\) then \( 2\lambda_2 = -1 \), hence \( \lambda_2 = -1/2 < 0 \) contradicts \( \lambda_2 \geq 0 \).

If \((x = 0 \text{ and } y = 1/2)\) then \( \lambda_2 = 0 \) and \( -1 = -4\lambda_1 \), hence \( \lambda_1 = 1/4 \geq 0 \). \( \nabla g_1(0, 1/2) = [0, -4], \nabla g_2(0, 1/2) = [1, 2] \) are linearly independent.

Case 2. Let \( g_1(x, y) = 0 \) and \( g_2(x, y) > 0 \). Hence \( \lambda_2 = 0 \), hence

\[
\begin{align*}
2x &= -2\lambda_1 x \\
-1 &= -8\lambda_1 y \\
-x^2 - 4y^2 + 1 &= 0.
\end{align*}
\] (9)

It follows that \( 2x(\lambda_1 + 1) = 0 \). Hence \((x = 0 \text{ and } y = \pm 1/2) \) or \( \lambda_1 = -1 \).

If \((x = 0 \text{ and } y = 1/2)\) then \( \lambda_1 = 1/4 \geq 0 \). But \( g_2(0, 1/2) = -1 + 0 + 1 = 0 \) contradicts assumption in current

Case 2 \( g_2(x, y) > 0 \).

If \((x = 0 \text{ and } y = -1/2)\) then \( \lambda_1 = -1/4 < 0 \) contradicts \( \lambda_1 \geq 0 \).

If \( \lambda_1 = -1 < 0 \) contradicts \( \lambda_1 \geq 0 \).

Case 3. Let \( g_1(x, y) > 0 \) and \( g_2(x, y) = 0 \). Hence \( \lambda_1 = 0 \), hence \( 2x = \lambda_2 \) and \( -1 = 2\lambda_2 \). So \( \lambda_2 = -1/2 < 0 \) contradicts \( \lambda_2 \geq 0 \).

Case 4. Let \( g_1(x, y) > 0 \) and \( g_2(x, y) > 0 \). Hence \( \lambda_1 = \lambda_2 = 0 \), hence \( 2x = 0 \) and \(-1 = 0 \) a contradiction.

From Cases 1, 2, 3, 4 it follows that the only point which satisfies Karush-Kuhn-Tucker conditions is \((0, 1/2), \lambda_1 = 1/4, \lambda_2 = 0 \). \( \nabla g_1(0, 1/2) = [0, -4], \nabla g_2(0, 1/2) = [1, 2] \) are linearly independent.

The set \( D \) is compact and \( f \) is continuous on it so finally we get: the greatest value \( 1/2 \) and smallest value \(-1 \) of \( f \) on \( D \) are attained corresponding at points \((0, 1/2) \) and \((1, 0) \) respectively.

In Fig. 1 we present graphical interpretation of Karush–Kuhn–Tucker necessary conditions for the above problem.

In the Fig. 1 we see that: \( \text{grad } f(1, 0) = [-2, 1] = \frac{5}{4} \text{ grad } g_1(1, 0) + \frac{1}{2} \text{ grad } g_2(1, 0) = \frac{5}{4} [-2, 0] + \frac{1}{2} [1, 2] \)

We solve this problem also using Mathematica procedures. We present the solution below:

Listing 1  Mathematica code for point a):

\[
\text{List1 := \{f \rightarrow x - x^2, g1 \rightarrow -x^2 - 4y^2 + 1, g2 \rightarrow -1 + x + 2y, F \rightarrow -\lambda_1 * g1 - \lambda_2 * g21;}
\]

\[
\text{List2 := \{F1 := D[F, x], F2 := D[F, y]\}}
\]

\[
\text{Out2 := \{-\lambda_2 - 2x + 2\lambda_1 x, 1 - 2\lambda_2 + 8\lambda_1 y\}}
\]

\[
\text{Out3 := Solve[\{F1 \rightarrow 0, F2 \rightarrow 0, \lambda_1 \rightarrow g1 \rightarrow 0, \lambda_2 \rightarrow g2 \rightarrow 0, \lambda_1 \rightarrow g1 \rightarrow 0, \lambda_2 \rightarrow g2 \rightarrow 0, g1 \rightarrow 0, g2 \rightarrow 0\},}
\]

\[
\{x, y, \lambda_1, \lambda_2\}, \text{Reals}\}
\]

\[
\text{Out4 := \{\{x \rightarrow 1, y \rightarrow 0, \lambda_1 \rightarrow 5/4, \lambda_2 \rightarrow 1/2\}\}}
\]

\[
\text{Out4 := Table[f/. Take[r2[[i]], {1, 3}], {i, 1, 1}]}\]

\[
\text{Out4 := \{-1\}}
\]

\[
\text{Out5 := Minimize[\{-x^2, 0 <= x <= -x^2 - 4y^2 + 1, -1 + x + 2y >= 0\}, \{x, y\}]}\]

\[
\text{Out5 := \{-1, \{x \rightarrow 1, y \rightarrow 0\}\}}
\]
**Example 2** Let us solve the following NLP problem:

\[
\begin{align*}
\text{maximize} & \quad x^2 + y^2 - z \\
\text{subject to:} & \quad x - y^2 \geq 0, \\
& \quad -2x - 3z + 9 \geq 0, \\
& \quad -x - z + 4 \geq 0, \\
& \quad z \geq 0.
\end{align*}
\] (10)
Using Lagrange multipliers method we determine global minimum and maximum function problems simultaneously. It does not require an independent examination of two tasks—for minimum and for maximum. We can solve both values on Lagrange function \( F \).

From the above listing we get that the greatest value 20 of \( f \) on feasible set is attained at points \((4, -2, 0)\) and \((4, 2, 0)\).

In Fig. 2 we present graphical interpretation of Karush–Kuhn–Tucker necessary conditions for the above problem. The dynamic versions of the Fig. 2 can be found in Electronic supplementary material.

In the Fig. 2 we see that: \( v = -\text{grad } f(4, -2, 0) \) is a linear combination of the vectors \( w_1 = \text{grad } g_1(4, -2, 0) \), \( w_3 = \text{grad } g_3(4, -2, 0) \), \( w_4 = \text{grad } g_4(4, -2, 0) \) with coefficients \( \lambda_1 = 1, \lambda_3 = 9, \lambda_4 = 10, (\lambda_2 = 0) \).

### 3 Lagrange Multipliers Method

Now, we solve the problem from Example 1 by hand using Lagrange multipliers method. The Lagrange method does not require an independent examination of two tasks—for minimum and for maximum. We can solve both problems simultaneously.

**Example 3** Using Lagrange multipliers method we determine global minimum and maximum function \( f(x, y) = y - x^2 \) subject to: \(-x^2 - 4y^2 + 1 \geq 0, -1 + x + 2y \geq 0\)

Let \( g_1(x, y) = -x^2 - 4y^2 + 1 \) and \( g_2(x, y) = -1 + x + 2y \).

Let \( D \) be a set of all points in \( \mathbb{R}^2 \) which satisfy all the constraints in the above problem. The set \( D \) is compact and the function \( f \) is continuous on it, so by Weierstrass theorem \( f \) attains infimum (smallest) and supremum (greatest) values on \( D \).

The function \( f \) can achieve the smallest and greatest value in interior or on boundary of \( D \). If the global optimum is achieved at a point in the interior of \( D \) the point has to be stationary point of \( f \). But \( \frac{\partial f}{\partial x}(x, y) = -2x \) and \( \frac{\partial f}{\partial y}(x, y) = 1 \). So there is no stationary point for \( f \) in the interior of \( D \).

If the global optimum is achieved at a point in the boundary of \( D \) the point has to be stationary point of Lagrange function \( F_1(x, y) = f(x, y) - \lambda_1 g_1(x, y) \) on boundary \( g_1(x, y) = 0, x \in [0, 1], y \in [0, 1/2] \) or \( F_2(x, y) = f(x, y) - \lambda_2 g_2(x, y) \) on boundary \( g_2(x, y) = 0, x \in [0, 1], y \in [0, 1/2] \).

#### Listing 3 Mathematica code:

```mathematica
In[1]:= (* max *) \( f = -z + x^2 + y^2 \), \( g_1 = x - y^2 \), \( g_2 = -2x - 3z + 9 \), \( g_3 = -x - z + 4 \), \( g_4 = z \).
\( F = -f - \lambda_1 * g_1 - \lambda_2 * g_2 - \lambda_3 * g_3 - \lambda_4 * g_4; \)
In[2]:= \( \{ F = \text{D}[F, x], F_2 = \text{D}[F, y], F_3 = \text{D}[F, z] \} \)
Out[2]= \{-\lambda_1 + 2x_2 + \lambda_3 - 2x, -2y + 2x_1 y, 1 + 3\lambda_2 + \lambda_3 - \lambda_4 \}
In[3]:= \( r = 2 \) \( \text{Solve} \{ F_1 == 0, F_2 == 0, F_3 == 0, \lambda_1 * g_1 == 0, \lambda_2 * g_2 == 0, \lambda_3 * g_3 == 0, \lambda_4 * g_4 == 0, \lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0, g_1 > 0, g_2 > 0, g_3 > 0, g_4 > 0, \} \{ x, y, z, \lambda_1, \lambda_2, \lambda_3, \lambda_4 \} \)
Out[3]= \{ \{ x > 0, y > 0, z > 0, \lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0 \}, \{ x > 0, y > 0, z > 0, \lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0 \}, \{ x > 0, y > 0, z > 0, \lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0 \}, \{ x > 0, y > 0, z > 0, \lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0 \} \}
In[4]:= \( n = \text{Length}[r] \)
Out[4]= 4
In[5]:= \( \text{Table}[ f / . \text{Take}[r[[i]], \{ 1, 3 \}], \{ i, 1, n \} \) 
Out[5]= \{ 0, 20, 16, 20 \}
In[6]:= \( \text{Maximize} \{ x_1^2 + x_2^2 - x_3, 0 <= -2x_1 - 3x_3 + 9, x_1 - x_2^2 >= 0, -x_1 - x_3 + 4 >= 0, x_3 >= 0 \} \{ x_1, x_2, x_3 \} \)
Out[6]= \{ 20, \{ x_1 > 4, x_2 > -2, x_3 > 0 \} \}
```
Consider $F_1(x, y) = y - x^2 - \lambda_1(-x^2 - 4y^2 + 1)$ on boundary $-x^2 - 4y^2 + 1 = 0$, $x \in [0, 1]$, $y \in [0, 1/2]$ and solve the following system:

\[
\begin{cases}
-2x + 2\lambda_1 x = 0 \\
1 + 8\lambda_1 y = 0 \\
-x^2 - 4y^2 + 1 = 0 \\
x \in [0, 1], y \in [0, 1/2]
\end{cases}
\]

(11)

From the first equation we have $x(\lambda_1 - 1) = 0$ so $x = 0$ or $\lambda_1 = 1$.

When $x = 0$ then from the third equation we have $y = \pm 1/2$ and from the second equation $\lambda_1 = \mp 1/4$.

When $\lambda_1 = 1$ then from the second equation we have $y = -1/8 \notin [0, 1/2]$.

So the only stationary point of $F_1$ on boundary $g_1(x, y) = 0$ is $(0, 1/2), \lambda_1 = -1/4 (-1/2 \notin [0, 1/2])$. But we also have to consider the ends of the curve $-x^2 - 4y^2 + 1 = 0$, $x \in [0, 1]$, $y \in [0, 1/2]$ which are $(1, 0)$ and $(0, 1/2)$.
Example 5 We solve the problem from Example 2 using graphical method. In the Fig. 4 we present Mathematica dynamic plot for the Example 2 in two versions: for global minimum and global maximum. The level sets is a family of paraboloids $x^2 + y^2 - z = c$.

Listing 4 Mathematica code for Lagrange multipliers method:

```mathematica
In[1] := \{f = y - x^2, g1 = -x^2 - 4 y^2 + 1, g2 = -1 + x + 2 y, F1 = f - \lambda_1 g1, F2 = f - \lambda_2 g2\}
Out[1] := \{-x^2 + y, 1 - x^2 - 4 y^2, -1 + x + 2 y, -x^2 + y - \lambda_1 (1 - x^2 - 4 y^2), -x^2 + y - \lambda_2 (-1 + x + 2 y)\}
In[2] := \{F1 x == D[F1, x], F1 y == D[F1, y]\}
Out[2] := \{-2x + 2\lambda_1 x, 1 + 8\lambda_1 y\}
In[3] := r1 = Solve[\{F1 x == 0, F1 y == 0, g1 == 0, 0 <= x <= 1, 0 <= y <= 1/2\}, \{x, y, \lambda_1\}]
Out[3] := \{\{x -> 0, y -> 1/2, \lambda_1 -> -1/4\}\}
```

Consider $F_2(x, y) = y - x^2 - \lambda_1 (x + 2y - 1)$ on boundary $x + 2y - 1 = 0, x \in [0, 1], y \in [0, 1/2]$ and solve the following system:

$$
\begin{cases}
-2x - \lambda_2 = 0 \\
1 - 2\lambda_2 = 0 \\
-1 + x + 2y = 0 \\
x \in [0, 1], y \in [0, 1/2]
\end{cases}
$$

From the second equation we have $\lambda_2 = 1/2$ so from the first equation we have $x = -1/4 \notin [0, 1]$. So there are no stationary points $F_2$ on boundary $g_2(x, y) = 0, x \in [0, 1], y \in [0, 1/2]$. But we also have to consider the ends of the curve $x + 2y - 1 = 0, x \in [0, 1], y \in [0, 1/2]$ which are $(1, 0)$ and $(0, 1/2)$.

Listing 5 Mathematica code Lagrange multipliers method:

```mathematica
In[4] := \{F2 x = D[F2, x], F2 y = D[F2, y]\}
Out[4] := \{-\lambda_2 - 2x, 1 - 2\lambda_2\}
In[5] := r1 = Solve[\{F2 x == 0, F2 y == 0, g2 == 0, 0 <= x <= 1, 0 <= y <= 1/2\}, \{x, y, \lambda_2\}]
Out[5] := \{\}\nIn[6] := f/.\{\{x -> 1, y -> 0\}, \{x -> 0, y -> 1/2\}\}
Out[6] := \{-1, 1/2\}
```

$f(1, 0) = -1, f(0, 1/2) = 1/2$.

The set $D$ is compact and $f$ is continuous on it so the greatest value $1/2$ and the smallest value $-1$ of $f$ on $D$ are attained respectively at points $(0, 1/2)$ and $(1, 0)$.

4 Graphical Method

In the case of two or three decision variables, we can try to solve the NLP problem using graphical method. In this method we first plot the feasible region. Next, we consider the family of level sets of the objective function. We try to select level sets from this family that have at least one common point with the feasible region and on which the minimum value of the objective function is attained—when the task is for minimum or maximum when the task is for maximum.

Example 4 We solve the problem from Example 1 using graphical method. In the Fig. 3 we present Mathematica dynamic plot for the Example 1. The level sets is a family of parabolas $y - x^2 = c$.

The dynamic version of the Fig. 3 can be found in Electronic supplementary material.

Global maximum of the function $f(x, y) = y - x^2$ at point $(0, 1/2)$ we can determine graphically from the level set: $y - x^2 = 1/2$. Similarly global minimum of the function $f$ at point $(1, 0)$ we can determine graphically from the level set: $y - x^2 = -1$.

Example 5 We solve the problem from Example 2 using graphical method. In the Fig. 4 we present Mathematica dynamic plot for the Example 2 in two versions: for global minimum and global maximum. The level sets is a family of paraboloids $x^2 + y^2 - z = c$. 
Teaching Students Nonlinear Programming

\[ f(x, y) = y - x^2 = c = \frac{1}{2}, \quad P = (1, 0), \quad Q = (0, 1/2) \]

Fig. 3 Graphical method for Example 1 using dynamic plots

\[ f(x, y, z) = x^2 + y^2 - z = c = -3, \quad P = (0, 0, 3), \]

Fig. 4 Graphical method for Example 2 determining global minimum using dynamic plots
The dynamic version of the Fig. 4 can be found in Electronic supplementary material.

We can determine graphically a global minimum of the function \( f(x, y, z) = x^2 + y^2 - z \) at point \((0, 0, 3)\) from the level set: \( x^2 + y^2 - z = -3 \).

The dynamic version of the Fig. 5 can be found in Electronic supplementary material.

We can determine graphically a global maximum of the function \( f(x, y) \) at points \((4, -2, 0)\), \((4, 2, 0)\) from the level set: \( x^2 + y^2 - z = 20 \).

Example 6 Consider the following nonlinear integer problem:

\[
\text{minimize/} \maximize_{(x,y) \in \mathbb{R}^2} x + y \\
\text{subject to:} \quad y \leq \sqrt{x}, \\
-2x + 2y + 5y \geq 0, \\
x \geq 0, \\
(x, y) \in \mathbb{Z}^2. 
\]

The dynamic version of the Fig. 6 can be found in Electronic supplementary material.

From the dynamic plot in Fig. 6 we can find that the global minimum of the integer problem of function \( f(x, y) = x + y \) is attained at point \((0, -2)\) (on the level set \( x + y = -2 \)), \( f(0, -2) = -2 \). Similarly we can find that the global maximum of the function \( f \) is attained at point \((4, 2)\) (on the level set \( x + y = 6 \)), \( f(4, 2) = 6 \).

We can check these results using Mathematica build-in function Maximize and Minimize as presented below:

```
Listing 6 Mathematica code:
In[1]:= Maximize[{x + y, 0 <= -2x + 2y + 5, y <= Sqrt[x] && x >= 0 && Element[x | y, Integers]}, {x, y}]
Out[1]= {6, {x -> 4, y -> 2}}
In[2]:= Minimize[{x + y, 0 <= -2x + 2y + 5, y <= Sqrt[x] && x >= 0 && Element[x | y, Integers]}, {x, y}]
Out[2]= {-2, {x -> 0, y -> -2}}
```

5 Convex Optimization

This approach based on the fact that for convex function defined on convex set, local minimum is a global minimum (Figs. 7 and 8).
Example 7 Consider the following NLP problem. Find the global minimum value of the function \( f(x_1, x_2) = (x_1 + x_2^2)e^{x_1} \) on the convex set \( D = \{(x_1, x_2) \in \mathbb{R}^2 : -2 \leq x_1 \leq 0, -\sqrt{x_1 + 2} \leq x_2 \leq \sqrt{x_1 + 2}\} \).

Using Mathematica we can check that the function \( f \) is convex on the convex set \( D \) and find the stationary points of \( f \).

In the last listing \( M \) is Hessian of the function \( f \). Using the Mathematica build-in function Reduce we can see that the Hessian of the function \( f \) is positive defined on the set \( D \). So the function \( f \) is convex on the convex set \( D \).

Using Mathematica we can find that local minimum of \( f \) is attained at point \((-1, 0), f(-1, 0) = -1/e\). So the global minimum of \( f \) on the set \( D \) is attained at point \((-1, 0)\).

The dynamic version of the Fig. 6 can be found in Electronic supplementary material.

6 Summary and Conclusions

This paper presents some didactic examples of NLP problems solved with Mathematica. In general, teaching students about NLP, for the first time, is not an easy task without computer support. Examining formal conditions
Fig. 7 Graph of the convex set $D$ in Example 7

Fig. 8 Graph of the convex function $f$ on the convex set $D$ in Example 7, using dynamic plot

by hand (Karush–Kuhn–Tucker necessary conditions or Lagrange multiple constrains) in case of the more difficult problems can be quite laborious. Also, graphical interpretations are not easy to present without the support of computer programs such as CAS. In Example 1 (Karush–Kuhn–Tucker necessary conditions) with two constrains we get four main cases to consider (with two subcases) what is possible to do by hand with students. But, in Example 2 with four constrains we get sixteen main cases with some subcases, what is rather not appropriate for hand calculations during classes, but it’s possible to do with support of computer programs such as CAS. In Example 3 we use Lagrange multiple constrains method for Example 1 to compare Lagrange method with its generalization, the Karush–Kuhn–Tucker method. In Example 3 we finally get only four cases. Using Lagrange method for this problem seems to be significantly easier than Karush–Kuhn–Tucker method. In general, CAS programs give us a possibility to support hand calculations for NLP problems with more constrains. We can also
check the convexity condition of the function in convex optimization problem and determine local extremes using
CAS functions (Example 7). The possibility to visualize some aspects of NLP problems with CAS is also important.
In Examples 1 and 2 we present graphical interpretation of Karush–Kuhn–Tucker necessary conditions. Using CAS
dynamic plot we can see normal vectors to the surfaces in Example 2 from different sides. In Examples 4, 5, and 6
we solve the problems using graphical method with dynamic plots. The possibility of graphical visualization and
the dynamic plots use are very important in this approach. In case of the graphical method, we analyze the family
of level sets of the optimized function in order to determine optimal level set during NLP problems solving. We try
to translate the initial level set in the right direction to the optimal position. The above translation is to increase
the function values if the problem is for maximum or decrease the function values if the problem is for minimum. The
method itself has a dynamic character. Therefore, the CAS dynamic plots seems to be an appropriate tool for visual
presentation. This allows students to follow the method steps and provides deeper understanding consequently. In
case of two variables and not so many constrains we can draw feasible region and level-sets by hand of course, but
in case of three variables (Example 5) it is much more difficult, so computer graphics support seems to be very
useful. In case of determining the global minimum of the convex function (Example 7), the use of CAS dynamic
plots allows students to dynamically view the surface being a graph of the optimized function and observe the global
minimum point on the surface. Generally, the use of Mathematica during teaching students about NLP by CAS
programs seems to be very useful both as the hand calculations support and when creating graphical visualizations
of various aspects of NLP problems. We did not find any analogical approach to the NLP problems solving using
CAS in available literature.

Open Access  This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Bazaraa, S.M., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 2nd edn. Wiley, New York (1979)
2. Griva, I., Nash, S.G., Sofer, A.: Linear and Nonlinear Optimization. SIAM, Philadelphia (2009)
3. Luenberger, D.G.: Linear and Nonlinear Programming, 3rd edn. Springer, Berlin (2010)
4. Ostaszewski, A.: Advanced Mathematical Methods (London School of Economics), 1st edn. Cambridge University Press, Cambridge (1991)
5. Peressini, A.L., Sullivan Jr., F.E., Uhl, J.J.: The Mathematics of Nonlinear Programming. Springer, New York (1988)
6. Ruskeepaa, H.: Mathematica Navigator: Graphics and Methods of Applied Mathematics. Academic Press, Boston (2005)
7. Ruszczyński, A.P.: Nonlinear Optimization. Princeton University Press, Princeton (2006)
8. Wolfram, S.: The Mathematica Book. Cambridge University Press, Cambridge (1996)