STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, Ferbuary 2025, pp 877–890.
Published online in International Academic Press (www.IAPress.org)

Some Direct Numerical Methods for Solving the Nonlinear Optimal Control Problem Practical in Aeronautic

Mohamed Aliane [1,*], Nacima Moussouni [2,3]

[1] *Laboratoty LaMA, Department of Mathematics and Computer Science, University of Medea, Algeria*
[2] *Faculty of Medicine, University Ammar Telidji of Laghouat, Algeria*
[3] *Laboratoty L2CSP of University Mouloud Mammeri Tizi Ouzou, Algeria*

Abstract In this study, we have implemented direct numerical methods to convert the continuous optimal control problem into a nonlinear optimization problem. We used three discretization techniques: the Euler method, the second-order Runge-Kutta method, and the fourth-order Runge-Kutta method. Subsequently, the resulting non-linear optimization problem was solved using MATLAB's fmincon function. To evaluate the efficiency and accuracy of the proposed approach, we modeled a nonlinear optimal control problem relevant to aeronautics. Our objective was to minimize the travel time of a rocket from an initial point to a final point at a specified altitude, considering aerodynamic forces and gravity, with the control variable being the rocket's heel angle. To compare the different methods, we developed a MATLAB implementation and showcased various simulation results.

Keywords Optimal Control, Euler Discretization Method, Rung kutta method, Active-Set Method, Aerodynamic

AMS 2010 subject classifications 49M25, 90C05

DOI: 10.19139/soic-2310-5070-1440

1. Introduction

Methods for solving optimal control problems existing in the literature are two types: direct and indirect methods. The direct method is based on discretization [2, 11, 14]. It gives an approximation solution. The indirect method is based on Pontryagin's maximum principle [2, 7, 8, 9, 12, 13, 14]. It gives only the necessary condition of optimality.

Recently, in [1], the authors addressed the problem of non-rectilinear motion of a rocket with variable mass using an indirect method. This approach involves numerically solving the problem via the shooting method: first, a boundary value problem is derived using Pontryagin's maximum principle, after which a non-linear system of equations is solved to obtain an approximate solution. In contrast, our work employs direct numerical methods, which are easier to implement.

In this work, we have presented a nonlinear optimal control model of a rocket moving from an initial position to a final position, it is the trajectory that the rocket must follow to take a target localized in minimal times, where the control represents the heel angle of the rocket. In order to solve the considered problem numericaly by three direct numerical methods (*Euler discretization method* (EDM)[2, 3], *second order Rung-Kutta discretization method* (RKDM2) and *fourth order Rung-Kutta discretization method* (RKDM4)). Then the obtained *non-linear programming problems* (NLPP) is solved with the fmincon function of MATLAB [11]. In order to see which method is best.

The article is organized as follows. In Section 2 we present some numerical discretization methods for solving optimal control problem. In Section 3, using the notions of physics, we modeled the problem in the form of a

*Correspondence to: Mohamed Aliane (Email: aliane. mohamed@univ-medea.dz). Department of Mathematics and Computer science, university of Medea. Pôle urbain, 26083,Medea, Algeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

nonlinear optimal control problem. We present the mathematical nonlinear optimal control model corresponding to the problem of minimal time of a rocket which moves with a nonrectilinear motion. In Section 4, we solve the optimal control problem with three direct numerical methods and give some numerical results. In Section 5, we present numerical comparison. Finally, we conclude our paper and give some perspectives.

2. Some direct numerical methods for solving optimal control problem

2.1. *Optimal control problem(OCP)*

Optimal control problem (OCP) considered in this paper has the following characteristics: A multi-input/ multi-output time variant system, a free final time with a fixed initial and final state and path contraintes [13]. Let $x(t) \in \mathbb{R}^n$ the state of a system and $u(t) \in \mathbb{R}^m$ the control in a given time interval $I = [0, T]$. A continuous from this OCP is given as follows:

$$\begin{cases} \text{Minimize} J(T, u(t)) = g(T, x(T)), \\ \dot{x}(t) = f(t, x(t), u(t)), \\ x(0) = x_0,\ \psi(x(T)) = 0, \\ u(t) \in U \subseteq \mathbb{R}^m,\ t \in I, \end{cases} \tag{1}$$

where the function $g : \mathbb{R}^n \times I \longrightarrow \mathbb{R}$, $f : \mathbb{R}^n \times I \longrightarrow \mathbb{R}^n$, $\psi : \mathbb{R}^n \longrightarrow \mathbb{R}^r$, $0 \leq r \leq n$, $x_0 \in \mathbb{R}^n$ and U is a compact subset of $\mathbb{R}^m$.

2.2. *Transformation of the OCP into a NLPP*

Direct optimization methods to solve the optimal control problem (OCP) are based on a suitable discretization of the problem (1). In this section, we describe three methods of discretization:

- Euler descritization method (EDM)
- Second order Rung-Kutta descritization method (RKDM2)
- Fourth order Rung-Kutta descritization method (RKDM4)

First, we discretize the interval $[0, T]$ into N subintervals, where N is chosen in advance. Then we get the following moments:

$$0 = t_0 < t_1 = t_0 + h < ...... < t_{N-1} = t_0 + (N-1)h < t_N = T,$$

where $h = \frac{T}{N}$.

2.2.1. *Euler descritization method (EDM)*

The application of the Euler scheme [11] for the resolution of Cauchy's problems gives the following nonlinear programming problem (2):

$$\begin{cases} J(t_N, u(t_i)) = g(t_N, x(t_N)) \longrightarrow \min, \\ x(t_{i+1}) = x(t_i) + hf(t_i, x(t_i), u(t_i)), \\ x(t_0) = x_0,\ \psi(x(t_N)) = 0, \\ u(t_i) \in U \subseteq \mathbb{R}^m,\ t_i = t_0 + ih,\ i = 1 \ldots N-1. \end{cases} \tag{2}$$

2.2.2. Second order Rung-Kutta descritization method (RKDM2) The application of the second-order Rung-Kutta scheme [19]for the resolution of Cauchy's problems gives the following nonlinear programming problem (3):

$$\begin{cases} J(t_N, u(t_i)) = g(t_N, x(t_N)) \longrightarrow \min, \\ k_1 = f(t_i, x(t_i), u(t_i)), \\ k_2 = f(t_{i+1}, x(t_i) + hk_1, u(t_{i+1})), \\ x(t_{i+1}) = x(t_i) + \dfrac{h}{2}(k_1 + k_2), \\ x(t_0) = x_0,\ \psi(x(t_N)) = 0, \\ u(t_i) \in U \subseteq \mathbb{R}^m,\ t_i = t_0 + ih,\ i = 1 \ldots N-1. \end{cases} \tag{3}$$

2.2.3. Fourth order Rung-Kutta descritization method (RKDM4) The application of the fourth-order Rung Kutta scheme [19]for the resolution of Cauchy problems gives the following nonlinear programming problem (4):

$$\begin{cases} J(u(t_i), t_N) = g(t_N, x(t_N)), \\ k_1 = f(t_i, x(t_i), u(t_i)), \\ k_2 = f(t_i + \frac{h}{2}, x(t_i) + \frac{h}{2}k_1, \frac{1}{2}(u(t_i) + u(t_{i+1})), \\ k_3 = f(t_i + \frac{h}{2}, x(t_i) + \frac{h}{2}k_2, \frac{1}{2}(u(t_i) + u(t_{i+1}))), \\ k_4 = f(t_{i+1}, x(t_i) + hk_3, u(t_{i+1})), \\ x(t_{i+1}) = x(t_i) + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ x(t_0) = x_0,\ \psi(x(t_N)) = 0, \\ u(t_i) \in U \subseteq \mathbb{R}^m,\ t_i = t_0 + ih,\ i = 1 \ldots N-1. \end{cases} \tag{4}$$

3. Statement of the problem

In the following sections, we first build a mathematical model for the minimum time trajectory of the rocket moving from a point M_0 to a final point M_f. For simplicity, we consider the rocket as a point mass moving in a two-dimensional direction with Cartesian coordinates (x_{10}, x_{20}) and (x_{1f}, x_{2f}) respectively. We denote by $x(t) = (x_1(t), x_2(t))$ the position of the rocket at time $t \in [0, T]$, ($x_2(t)$ represents the altitude of the rocket at time t) and $v(t) = (v_1(t), v_2(t))$ his speed at time t, where $t \in [0, T]$. The motion equations are given by

$$\begin{cases} \dot{x}_1(t) = v_1(t), \\ \dot{x}_2(t) = v_2(t), \quad t \in [0, T]. \end{cases}$$

Let $T_p(t)$ and $\theta(t)$ be, respectively, the thrust and the heel angle of the rocket at time $t \in [0, T]$. We have

$$\vec{T}_p(t) = u_{\max} \begin{pmatrix} \cos(\theta(t)) \\ \sin(\theta(t)) \end{pmatrix},$$

where $u_{\max}$ is a positive real number representing the module of the force $\vec{T}_p(t)$. The earth is supposed to be flat. We assume, for simplification purposes, for aerodynamic forces $\vec{F}_D$, we also take into account the effect of air resistance and the data for the mass of the rocket, his thrust is assumed to be available in advance and the acceleration of gravity g is constant ($g = 9.80665 m.s^{-2}$).
The air resistance F_D is given via

$$F_D(x_2(t), v_1(t), v_2(t)) = \frac{1}{2}\rho v^2(t) C_D S_{ref},$$

where the two constants C_D and S_{ref} are the drag coefficient and the area of the cross-section of the rocket, respectively. Furthermore, $v(t) = \sqrt{v_1^2(t) + v_2^2(t)}$ is the total velocity, and ρ is the atmospheric density as a function of the altitude modeled in this problem by

$$\rho = \rho_0 e^{\frac{-x_2(t)}{h_s}},$$

where
ρ_0 :the reference atmospheric density corresponding to the launch point
h_s : is an altitude scale.
We have

$$\overrightarrow{F}_D(x_2(t), v_1(t), v_2(t)) = \alpha(v_1(t)^2 + v_2(t)^2)e^{-\beta x_2(t)} \begin{pmatrix} \cos(\phi(t)) \\ \sin(\phi(t)) \end{pmatrix},$$

$$\overrightarrow{g} = \begin{pmatrix} 0 \\ -g \end{pmatrix},$$

where $\alpha = \frac{1}{2}\rho_0 C_D S_{ref}$ and $\beta = \dfrac{1}{h_s}$ parametrs of the aerodynamic forces, there are such that:
α: is the product of the drag coefficient, the surface of the machine and the atmospheric density.
β: is the inverse of the altitude scale.
we have

$$\cos(\phi(t)) = \frac{v_1(t)}{\sqrt{v_1(t)^2 + v_2(t)^2}},$$

and

$$\sin(\phi(t)) = \frac{v_2(t)}{\sqrt{v_1(t)^2 + v_2(t)^2}},$$

By applying the second Newton's law:

$$\sum \overrightarrow{F}_{ext} = m\overrightarrow{a},$$

We obtain the following equation:

$$\frac{d\overrightarrow{v}(t)}{dt} = \overrightarrow{g} + \frac{\overrightarrow{T}_p(t)}{m} + \frac{\overrightarrow{F}_a(x_2(t), v_1(t), v_2(t))}{m}.$$

By projecting on the $x-$axis, we obtain

$$\frac{dv_1(t)}{dt} = \frac{u_{\max}}{m}\cos(\theta(t)) - \frac{\alpha}{m}v_1(t)\sqrt{v_1(t)^2 + v_2(t)^2}e^{-\beta x_2(t)},$$

and

$$\frac{dv_2(t)}{dt} = \frac{u_{\max}}{m}\sin(\theta(t)) - \frac{\alpha}{m}v_1(t)\sqrt{v_1(t)^2 + v_2(t)^2}e^{-\beta x_2(t)} - g.$$

The goal is to bring the rocket to an orbit altitude chosen in advance with minimizing the final time. Here, the control represents the heel angle of the rocket. The problem is therefore to determine the optimal trajectory of this rocket. This problem can be modeled with the following optimal control problem (OCP):

$$\begin{cases} \text{Minimize } J(\theta(t), T) = T, \\ \dot{x}_1(t) = v_1(t), \\ \dot{x}_2(t) = v_2(t), \\ \dot{v}_1(t) = \dfrac{u_{\max}}{m} \cos(\theta(t)) - \dfrac{\alpha v_1(t) \sqrt{v_1(t)^2 + v_2(t)^2}}{m} e^{-\beta x_2(t)}, \\ \dot{v}_2(t) = \dfrac{u_{\max}}{m} \sin(\theta(t)) - \dfrac{\alpha v_2(t) \sqrt{v_1(t)^2 + v_2(t)^2}}{m} e^{-\beta x_2(t)} - g, \\ x_1(0) = x_{10}, x_2(0) = x_{20},\ v_1(0) = v_{10},\ v_2(0) = v_{20}, \\ x_2(T) = x_{2f},\ v_1(T) = v_{1f},\ v_2(T) = v_{2f}, \\ \theta(t) \in \mathbb{R},\ t \in [0, T]. \end{cases} \tag{5}$$

The hard knots for this problem are given by the boundary conditions and the initial conditions were arbitrarily chosen to be [6]:

$$x_{10} = 0km, x_{20} = 0.005km,\ v_{10} = 0km.s^{-1},\ v_{20} = 0.01km.s^{-1}.$$

While the final conditions are given by the soft landing requirement:

$$x_{2f} = 180km,\ v_{1f} = 7.905km.s^{-1},\ v_{2f} = 0km.s^{-1}.$$

The constants parameters for this problem were arbitrarily chosen as:

$$\alpha = 2.164,\ \beta = 0.113,\ u_{\max} = 1900kN,\ m = 60528kg.$$

4. Resolution of the problem

4.1. Numerical resolution by the EDM

First, we discretize the interval $[0, T]$ into N subintervals, where N is chosen in advance. Then we get the following moments:

$$0 = t_0 < t_1 = t_0 + h < \ldots\ldots < t_{N-1} = t_0 + (N-1)h < t_N = T,$$

where $h = \frac{T}{N}$. The application of the Euler scheme for the resolution of Cauchy's problems gives the following nonlinear programming problem (NLPP):

$$\begin{cases} \text{Minimize } J(\theta(t_i), t_N) = t_N, \\ x_1(t_{i+1}) = x_1(t_i) + h v_1(t_i), \\ x_2(t_{i+1}) = x_2(t_i) + h v_2(t_i), \\ v_1(t_{i+1}) = v_1(t_i) + h \dfrac{u_{\max}}{m} \cos(\theta(t_i)) - h \dfrac{\alpha v_1(t_i) \sqrt{v_1(t_i)^2 + v_2(t_i)^2}}{m} e^{-\beta x_2(t_i)}, \\ v_2(t_{i+1}) = v_2(t_i) + h \left(\dfrac{u_{\max}}{m} \sin(\theta(t_i)) - g \right) - h \dfrac{\alpha v_2(t_i) \sqrt{v_1(t_i)^2 + v_2(t_i)^2}}{m} e^{-\beta x_2(t_i)}, \\ x_1(0) = x_{10}, x_2(0) = x_{20}, v_1(0) = v_{10}, v_2(0) = v_{20}, x_2(t_N) = x_{2f}, \\ v_1(t_N) = v_{1f},\ v_2(t_N) = v_{2f}, \theta(t_i) \in \mathbb{R},\ t_i = t_0 + ih,\ i = 0, 1, \ldots, N-1. \end{cases} \tag{6}$$

Then the solution of this nonlinear optimization program is an approximate solution for the minimal time of the original continuous nonlinear optimal control problem.

4.1.1. Code of MATLAB To illustrate the numerical implementation of the proposed method, we present below an example of MATLAB code. This code solves the problem by applying the principles described earlier. The algorithm has been designed to be both efficient and easy to use, thereby facilitating the reproduction of results and experimentation with different parameters.

Algorithm EDM

```
function EDM
clear all; close all; clc;
tic;
n = 200;
unit = rand(n, 1);
tfinit = 280; xinit = [unit; tfinit];
[rep, Fval, exitflag] = fmincon(@finaltime, xinit, [ ], [ ], [ ], [ ], [ ], @cond);
tf = rep(end); x(1) = 0; y(1) = 0.005; z(1) = 0; w(1) = 0.01;
u = 1900; m = 60528; g = 0.00980665; a = 2.164; b = 0.113;
for i = 1 : n
   x(i + 1) = x(i) + tf/n * z(i);
   y(i + 1) = y(i) + tf/n * w(i);
   z(i + 1) = z(i) + tf/n * (u/m * cos(rep(i)) - a/mz(i) * sqrt(z(i)^2 + w(i)^2) * exp(-b * y(i); )
   w(i + 1) = w(i) + tf/n * (u/m * sin(rep(i)) - g - a/m * z(i) * sqrt(z(i)^2 + w(i)^2) * exp(-b * y(i));
end
subplot(231); plot(linspace(0, tf, n + 1), x(1 : n + 1));
title('x_1(t)'), xlabel('t'), ylabel('x_1(t)'),
subplot(232); plot(linspace(0, tf, n), y(1 : n));
title('x_2(t)'), xlabel('t'), ylabel('x_2(t)'),
subplot(233); plot(linspace(0, tf, n + 1), z(1 : n + 1));
title('v_1(t)'), xlabel('t'), ylabel('v_1(t)'),
subplot(234); plot(linspace(0, tf, n), w(1 : n));
title('v_2(t)'), xlabel('t'), ylabel('v_2(t)'),
subplot(235); plot(linspace(0, tf, n), rep(1 : n));
title('Controle'), xlabel('t'), ylabel('θ(t)'), grid
subplot(236); plot(x(1 : n), y(1 : n));
title('Optimaltrajectory'), xlabel('x_1(t)'), ylabel('x_2(t)')
Time = toc
function [c, ceq] = cond(x)
n = length(x) - 1;
c = 0; tf = x(end); xf = 0; yf = 0.005; zf = 0; wf = 0.01;
u = 1900; m = 60528; g = 0.00980665; a = 2.164; b = 0.113;
for i = 1 : n
   xf = xf + tf/n * zf;
   yf = yf + tf/n * wf;
   zf = zf + tf/n * (u/m) * cos(x(i)) - (tf/n) * (a/m) * zf * sqrt(zf^2 + wf^2) * exp(-b * yf);
   wf = wf + tf/n * [(u/m) * sin(x(i)) - g] - (tf/n) * (a/m) * wf * sqrt(zf^2 + wf^2) * exp(-b * yf);
end
ceq = [xf - []; yf - 180; zf - 7.905; wf];
function val = finaltime(x)
val = x(end);
```

4.1.2. Numerical results for the EDM We have solved the nonlinear program (6) with the interior-point method implemented in MATLAB2009b for $N \in \{200, 500, 800, 1000, 1500, 2000, 3000\}$. The obtained numerical results (running time $CPUT$, the minimal time T) are presented in Table 1. Then, we find the results plotted in Figure 1

Table 1. Numerical simulation of the EDM.

N	T	$CPUT$ **(s)**	$x_2(T)$	$v_1(T)$	$v_2(T)$
200	276.6994	1.2557	180	7.905	$-3.6921E-10$
500	276.7002	3.2572	180	7.905	$-9.7575E-10$
800	276.7004	10.4867	180	7.905	$-2.6796E-09$
1000	276.7005	17.4522	180	7.905	$-2.4122E-09$
1500	276.7006	51.4660	180	7.905	$-2.4753E-09$
2000	276.7006	108.1496	180	7.905	$-1.2215E-09$
3000	276.7007	396.4375	180	7.905	$-2.0787E-09$

for $N = 3000$.

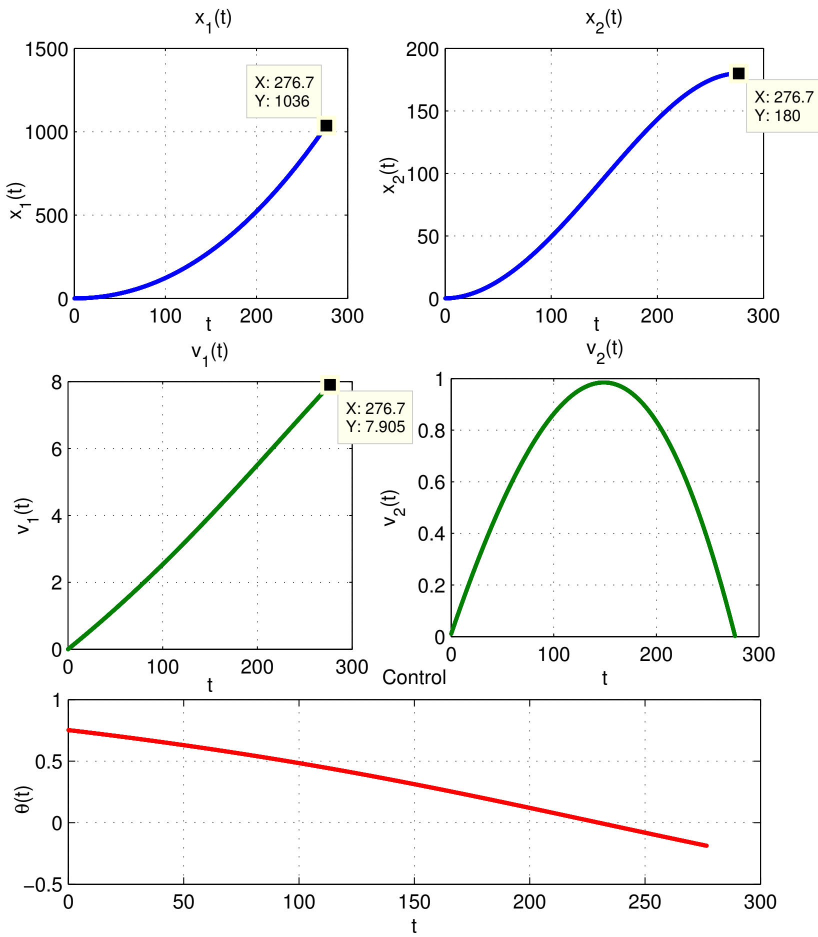

Figure 1. Optimal trajectories obtained by the EDM

4.2. Numerical resolution by the second order Rung-Kutta discretization method

First, we discretize the interval $[0,T]$ into N subintervals, where N is chosen in advance. Then we get the following moments:

$$0 = t_0 < t_1 = t_0 + h < ...... < t_{N-1} = t_0 + (N-1)h < t_N = T,$$

where $h = \frac{T}{N}$. The application of the second order Rung Kutta scheme for the resolution of Cauchy's problems gives the following nonlinear programming problem:

$$\begin{cases} \text{Minimize } J(\theta(t_i), t_N) = t_N, \\ x_1(t_{i+1}) = x_1(t_i) + \dfrac{h}{2}(k_{1,1} + k_{2,1}), \\ x_2(t_{i+1}) = x_2(t_i) + \dfrac{h}{2}(k_{1,2} + k_{2,2}), \\ v_1(t_{i+1}) = v_1(t_i) + \dfrac{h}{2}(k_{1,3} + k_{2,3}), \\ v_2(t_{i+1}) = v_2(t_i) + \dfrac{h}{2}(k_{1,4} + k_{2,4}), \\ x_1(0) = x_{10}, x_2(0) = x_{20}, v_1(0) = v_{10}, v_2(0) = v_{20}, \\ x_2(t_N) = x_{2f},\ v_1(t_N) = v_{1f},\ v_2(t_N) = v_{2f}, \\ \theta(t_i) \in \mathbb{R},\ t_i = t_0 + ih,\ i = 0, 1, \ldots, N-1, \end{cases} \tag{7}$$

where

$$k_1 = \begin{pmatrix} k_{1,1} \\ k_{1,2} \\ k_{1,3} \\ k_{1,4} \end{pmatrix} = \begin{pmatrix} v_1(t_i) \\ v_2(t_i) \\ \dfrac{u_{\max}}{m}\cos(\theta(t_i)) - h\dfrac{\alpha}{m}v_1(t_i)\sqrt{v_1(t_i)^2 + v_2(t_i)^2}e^{-bx_2(t_i)} \\ \dfrac{u_{\max}}{m}sin(\theta(t_i)) - g - h\dfrac{\alpha}{m}v_2(t_i)\sqrt{v_1(t_i)^2 + v_2(t_i)^2}e^{-bx_2(t_i)} \end{pmatrix},$$

and

$$k_2 = \begin{pmatrix} k_{2,1} \\ k_{2,2} \\ k_{2,3} \\ k_{2,4} \end{pmatrix} = \begin{pmatrix} k_{1,1} + hk_{1,3} \\ k_{1,2} + hk_{1,4} \\ \dfrac{u_{\max}}{m}\cos(\theta(t_{i+1})) - \dfrac{\alpha}{m}k_{2,1}\sqrt{k_{2,1}^2 + k_{2,2}^2}e^{-b(x_2(t_i)+hk_{1,2})} \\ \dfrac{u_{\max}}{m}\sin(\theta(t_{i+1})) - g - \dfrac{\alpha}{m}k_{2,2}\sqrt{k_{2,1}^2 + k_{2,2}^2}e^{-b(x_2(t_i)+hk_{1,2})} \end{pmatrix}.$$

Then the solution of this nonlinear optimization program is an approximate solution for the minimal time of the original continuous nonlinear optimal control problem.

4.2.1. Numerical results by the second order Rung-Kutta discretization method We have solved the nonlinear program (7) with the interior-point method implemented in MATLAB2009b for $N \in \{200, 500, 800, 1000, 1500, 2000, 3000\}$. The obtained numerical results (running time $CPUT$, the minimal time $T_{\min}$, $x_1(t_N)$, $x_2(t_N)$, $v_1(t_N)$ and $v_2(t_N)$) are presented in Table 2.
Then, we find the results plotted in Figure 2 for $N = 3000$.

Table 2. Numerical simulation of the second order Rung-Kutta discretization technique

N	T_{min}	$CPUT$ **(s)**	$x_2(t_N)$	$v_1(t_N)$	$v_2(t_N)$
200	276.7850	1.3349	180	7.905	$8.1826E-08$
500	276.7273	2.9154	180	7.905	$3.2798E-08$
800	276.7166	7.8917	180	7.905	$1.8457E-08$
1000	276.7133	26.2517	180	7.905	$1.6497E-08$
1500	276.7089	75.6885	180	7.905	$8.2617E-09$
2000	276.7068	148.7676	180	7.905	$7.6327E-09$
3000	276.7048	546.2568	180	7.905	$2.7238E-09$

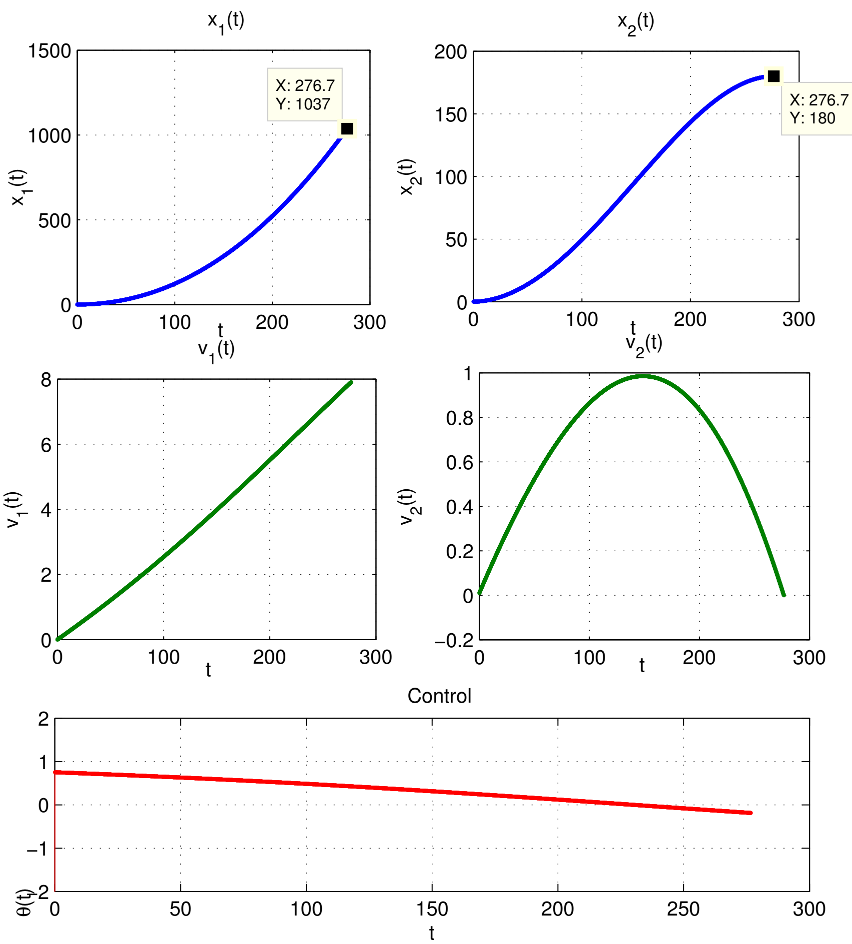

Figure 2. Optimal trajectories obtained by the RKDM2

4.3. Numerical resolution by the fourth order Rung-Kutta discretization method

First, we discretize the interval $[0, T]$ into N subintervals, where N is chosen in advance. Then we get the following moments:

$$0 = t_0 < t_1 = t_0 + h < ...... < t_{N-1} = t_0 + (N-1)h < t_N = T,$$

where $h = \frac{T}{N}$. The application of the fourth order Rung Kutta scheme for the resolution of Cauchy's problems gives the following nonlinear programming problem (8):

$$\begin{cases} \text{Minimize } J(\theta(t_i), t_N) = t_N, \\ x_1(t_{i+1}) = x_1(t_i) + \dfrac{h}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}), \\ x_2(t_{i+1}) = x_2(t_i) + \dfrac{h}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}), \\ v_1(t_{i+1}) = v_1(t_i) + \dfrac{h}{6}(k_{1,3} + 2k_{2,3} + 2k_{3,3} + k_{4,3}), \\ v_2(t_{i+1}) = v_2(t_i) + \dfrac{h}{6}(k_{1,4} + 2k_{2,4} + 2k_{3,4} + k_{4,4}), \\ x_1(0) = x_{10}, x_2(0) = x_{20}, v_1(0) = v_{10}, v_2(0) = v_{20}, \\ x_2(t_N) = x_{2f},\ v_1(t_N) = v_{1f},\ v_2(t_N) = v_{2f}, \\ \theta(t_i) \in \mathbb{R},\ t_i = t_0 + ih,\ i = 0, 1, \dots, N-1. \end{cases} \tag{8}$$

where

$$\begin{cases} k_{1,1} = v_1(t_i), \\ k_{1,2} = v_2(t_i), \\ k_{1,3} = \dfrac{u_{\max}}{m}\cos(\theta(t_i)) - h\dfrac{\alpha}{m}v_1(t_i)\sqrt{v_1(t_i)^2 + v_2(t_i)^2}e^{-bx_2(t_i)}, \\ k_{1,4} = \dfrac{u_{\max}}{m}sin(\theta(t_i)) - g - h\dfrac{\alpha}{m}v_2(t_i)\sqrt{v_1(t_i)^2 + v_2(t_i)^2}e^{-bx_2(t_i)} \end{cases}$$

$$\begin{cases} k_{2,1} = k_{1,1} + hk_{1,3}, \\ k_{2,2} = k_{1,2} + hk_{1,4}, \\ k_{2,3} = \dfrac{u_{\max}}{2m}\left(\cos(\theta(t_i)) + \cos(\theta(t_{i+1}))\right) - \dfrac{\alpha}{m}k_{2,1}\sqrt{k_{2,1}^2 + k_{2,2}^2}e^{-b(x_2(t_i)+hk_{1,2})}, \\ k_{2,4} = \dfrac{u_{\max}}{2m}\left(\sin(\theta(t_i)) + \sin(\theta(t_{i+1}))\right) - g - \dfrac{\alpha}{m}k_{2,2}\sqrt{k_{2,1}^2 + k_{2,2}^2}e^{-b(x_2(t_i)+hk_{1,2})}, \end{cases}$$

$$\begin{cases} k_{3,1} = k_{1,1} + hk_{2,3}, \\ k_{3,2} = k_{1,2} + hk_{2,4}, \\ k_{3,3} = \dfrac{u_{\max}}{2m}\left(\cos(\theta(t_i)) + \cos(\theta(t_{i+1}))\right) - \dfrac{\alpha}{m}k_{3,1}\sqrt{k_{3,1}^2 + k_{3,2}^2}e^{-b(x_2(i)+hk_{2,2})}, \\ k_{3,4} = \dfrac{u_{\max}}{2m}\left(\sin(\theta(t_i)) + \sin(\theta(t_{i+1}))\right) - g - \dfrac{\alpha}{m}k_{3,2}\sqrt{k_{3,1}^2 + k_{3,2}^2}e^{-b(x_2(i)+hk_{2,2})}, \end{cases}$$

$$\begin{cases} k_{4,1} = k_{1,1} + hk_{3,3}, \\ k_{4,2} = k_{1,2} + hk_{3,4}, \\ k_{4,3} = \dfrac{u_{\max}}{m}\cos(\theta(t_{i+1})) - \dfrac{\alpha}{m}k_{4,1}\sqrt{k_{4,1}^2 + k_{4,2}^2}e^{-b(x_2(i)+hk_{3,2})}, \\ k_{4,4} = \dfrac{u_{\max}}{m}\sin(\theta(t_{i+1})) - g - \dfrac{\alpha}{m}k_{4,2}\sqrt{k_{4,1}^2 + k_{4,2}^2}e^{-b(x_2(i)+hk_{3,2})}, \end{cases}$$

Then the solution of this nonlinear optimization program is an approximate solution for the optimal lateral offset of the original continuous nonlinear optimal control problem.

Table 3. Numerical simulation of the fourth order Rung-Kutta discretization method

N	T_{min}	$CPUT$ **(s)**	$x_2(t_N)$	$v_1(t_N)$	$v_2(t_N)$
200	276.7226	2.1492	180	7.905	$2.8226E-08$
500	276.7089	6.8505	180	7.905	$1.1105E-08$
800	276.7058	21.1028	180	7.905	$5.8503E-09$
1000	276.7048	37.0808	180	7.905	$4.6093E-09$
1500	276.7034	103.3506	180	7.905	$1.9948E-09$
2000	276.7028	195.9854	180	7.905	$3.5510E-10$
3000	276.7021	915.2891	180	7.905	$-1.2305E-10$

4.3.1. Numerical results by the fourth order Rung-Kutta discretization method We have solved the nonlinear program (8) with the interior-point method implemented in MATLAB2009b for $N \in \{200, 500, 800, 1000, 1500, 2000, 3000\}$. The obtained numerical results (running time $CPUT$, minimal time T, $x_1(t_N)$, $x_2(t_N)$, $v_1(t_N)$ and $v_2(t_N)$ are presented in Table 3. Then, we find the results plotted in Figure 3 for $N = 3000$.

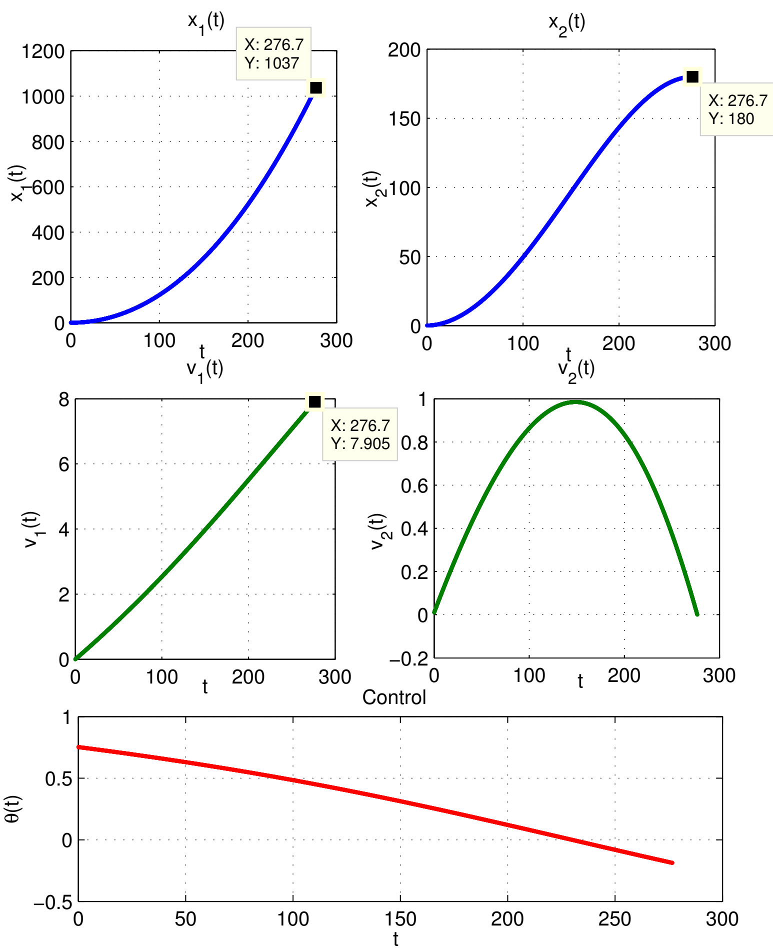

Figure 3. Optimal trajectories obtained by the RKDM4

From Figure 3, we note the angle $\theta(t)$ decreases as a function of time, furthermore, we see that $x_2(t)$ reches the altitude target $180km$, the velocity $v_1(t)$ is also increasing with time it varies from $v_1(0) = v_{10}$ to $v_1(T) = v_{1f} = 7.905km.s^{-1}$ set in Table 3 and $v_2(t)$ grows from $v_2(0) = v_{20}$ to $v_2(t_c)$ then decreases from $v_2(t_c)$ to $v_2(T)$.
Finally, we note that the running time of the fourth order Rung-Kutta discretization method necessary to find the optimal solution for $N = 3000$ is $915.2891s$.

5. Numerical comparaison

The minimal time found by the three methods are almost similar, however the execution times of these methods are quite different. Note For $N = 3000$ that the Euler discretization method gave an minimal time $T_{\min} = 276.7007s$ and required a very short execution time($CPU = 396.4375s$). The second order Rung-Kutta discretization method gave teh minimal time $T_{\min} = 276.7048s$ with an execution time ($CPU = 546.2568s$). The fourth order Rung-Kutta discretization method gave the minimal time $T_{\min} = 276.7021s$ with an execution time of $CPU = 915.2891s$,

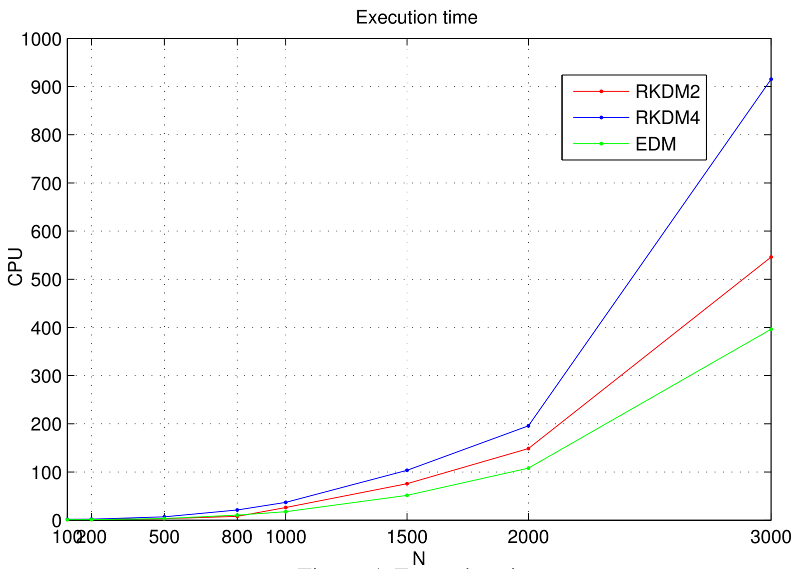

Figure 4. Execution time

Table 4. Numerical comparaison error of the $v_2(T)$

N	Error of EDM	Error of RKDM2	Error of RKDM4
200	$-3.6921E-10$	$8.1826E-08$	$2.8226E-08$
500	$-9.7575E-10$	$3.2798E-08$	$1.1105E-08$
800	$-2.6796E-09$	$1.8457E-08$	$5.8503E-09$
1000	$-2.4122E-09$	$1.6497E-08$	$4.6093E-09$
1500	$-2.4753E-09$	$8.2617E-09$	$1.9948E-09$
2000	$-1.2215E-09$	$7.6327E-09$	$3.5510E-10$
3000	$-2.0787E-09$	$2.7238E-09$	$-1.2305E-10$

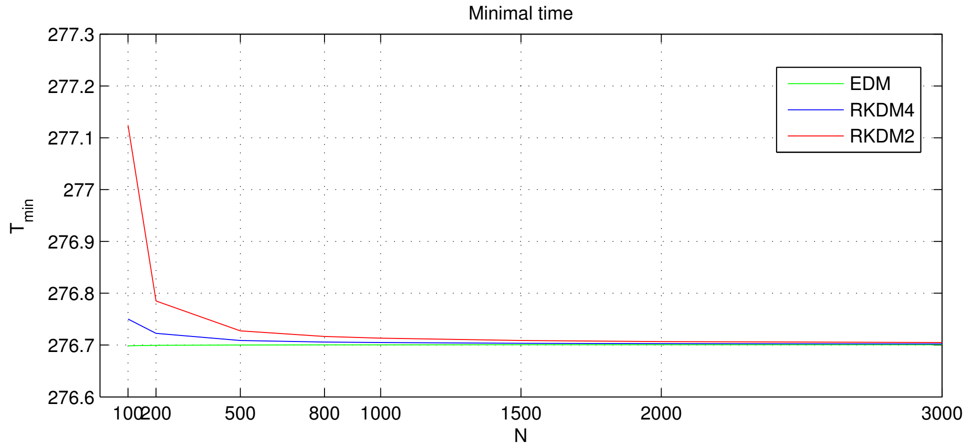

Figure 5. Minimal time

- The convergence is fast and the computational time is small.
- The Euler discretization method (EDM) is very fast, but less accurate than the other methods.
- The second order Rung-Kutta discretization method (RKDM2) is highly accurate and stable, but it is slow compared to EDM and fast compared to RKDM4.
- The fourth order Rung-Kutta discretization method (RKDM4) is highly accurate and stable, but it is slow compared to EDM and RKDM2.
- From [1] the indirect method is faster than the EDM, but is very difficult to apply.

6. Conclusion

In this work, we have solved a practical problem that arises in the aerospace field by formulating it as a nonlinear optimal control problem. For the numerical resolution, the considered problem is solved by three techniques of discretization(technique using the Euler formula, technique using the second-order Rung-Kutta formula and the one using the fourth-order Rung-Kutta formula). The optimal solutions obtained are almost similar, but a large difference in the execution time of the three numerical methods has been observed. In the future, we will work on developing this method so that we can apply it in general in other fields.

REFERENCES

1. M. Aliane, N. Moussouni, K. Louadj, N. Boizot *Indirect method for solving non-linear optimal control of a non-rectilinear motion of a rocket with variable mass* , Boletim da Sociedade Paranaense de Matematica, vol. 42, pp. 1–10, 2024.

2. M. Aliane, N. Moussouni, M. Bentobache, *Optimal control of a rectilinear motion of a rocket*, Statistics, Optimization & Information Computing, vol. 8, pp. 281–295, 2020.

3. M. Aliane, N. Moussouni and M. Bentobache, *Nonlinear optimal control of the heel angle of a rocket*, 6th International Conference on Control, Decision and Information Technologies (CODIT'19); Paris, France, April 23-26; 2019.

4. N. Moussouni, M. Aliane and L. Dehbi, *Optimalcontrol: White flies and their predators in two and three species*, 10th International Conference on Control, Decision and Information Technologies (CODIT'24); Valetta, Malta, July 01-04; 2024.

5. M. Aliane, M. Bentobache, N. Moussouni, P. Marthon, *Direct method to solve linear-quadratic optimal control problems* Numerical Algebra, Control & Optimization, 2021.

6. Tsai, Hung-i Bruce, Optimal trajectory designs and systems engineering analyses of reusable launch vehicles, 2003.

7. K. Louadj, P. Spiteri, F. Demim, M. Aidene, A. Nemra and F. Messine *Application Optimal Control for a Problem Aircraft Flight*, Engineering Science and Technology Review, vol. 11, pp.156–164, 2018.

8. N. Moussouni, and M. Aidene, *An Algorithm for Optimization of Cereal Output*, Acta Applicandae Mathematicae, vol. 11, no. 9, pp. 113–127, 2011.

9. N. Moussouni, and M. Aidene, *Optimization of cereal output in presence of locusts*, An International Journal of Optimization and Control: Theories & Applications, vol. 6, no. 1, pp. 1–10, 2016.

10. N. Khimoum, M.O. Bibi, *Primal-dual method for solving a linear-quadratic multi-input optimal control problem*, Optimization Letters, vol. 14, pp. 653–669, 2019.

11. O. Oukacha, *Direct method for the optimization of optimal control problems*, PhD. Dissertation, University of Tizi-Ouzou, Algeria, 2016 (in french).

12. L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko, *The mathematical theory of optimal processes*, New York: Intersciences Publisher, 1962.

13. E. Trélat, *Optimal control: theory and applications*, Paris: Vuibert, Concrete mathematics collection, 2005 (in french).

14. J.N.C. Gonçalves, H.S. Rodrigues, and M.T.T. Monteiro, *On the dynamics of a viral marketing model with optimal control using indirect and direct methods*, Statistics, Optimization & Information Computing, vol. 6, pp. 633–644, 2018.

15. J. T. BETTS, *Survey of numerical methods for trajectory optimization* J. Guid. Contr. Dynam., vol. 21, pp. 193–207, 1998.

16. W. ROH, Y. KIM, *Trajectory optimization for a multi-stage launch vehicle using time finite element and direct collocation methods* Eng. Optim., vol. 34, 15–32, 2002.

17. P. Williams, *Jacobi pseudospectral method for solving optimal control problems*, J. Guid. Contr. Dynam., vol. 27, pp. 293–297, 2004.

18. I. Chryssoverghi, A. Bacopoulos, *Discrete approximation of relaxed optimal control problems*, Journal of Optimization Theory and Applications, vol. 65, pp.395–407, 1990.

19. E. Süli, D. Mayers, An Introduction to Numerical Analysis New York, NY: Cambridge University Press, 2006.

20. J. Awerjcewicz, *Modeling, Simulation and Control of Nonlinear Engineering Dynamical Systems, State-of-the-Art, Presperctives and Applications*, Heidelberg, Germany: Springer, 2008.

21. L.D. Duncan, *Basic considerations in the development of an unguided rocket trajectory simulation model*, Technical report N^0 5076, Atmospheric Sciences Laboratory, United States Army Electronics Command, 1966.

22. Chinchuluun A, Enkhbat R, Pardalos P.M. *A novel approach for nonconvex optimal control problems*, Optimization: A Journal of Mathematical Programming and Operations Research, vol. 58, pp.781–789, 2009.

23. R. Gabasov, F.M. Kirillova, and S.V. Prischepova, *Optimal feedback control*, London, Springer-Verlag, 1995.

24. R.R. Garrett, *Numerical Methodes For Solving Optimal Control Problems*, Master Thesis, University of Tennessee, Knoxville, 2015.

25. P. Howlett, *The Optimal Control of a Train*, Annals of Operations Research, vol. 98, pp. 65–87, 2000.