Development of a Multi-phase Optimal Control Software for Aerospace applications (MPOPT)

EPFL

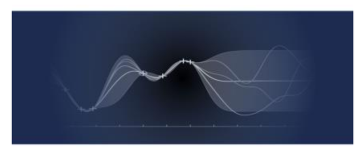
Master thesis
Spring 2020

Automatic control Laboratory, EPFL

Master thesis in Mechanical Engineering

mpopt 0.1.3

`pip install mpopt`



pypi v0.1.3 | build passing | coverage 94% | docs passing

https://github.com/mpopt

Overview

An open-source, extensible, pseudo-spectral collocation based multi-phase nonlinear optimal control problem (OCP) solver package named MPOPT is developed in Python programming language as part of the thesis.

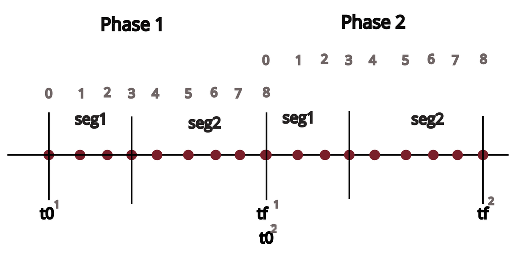
- **Main features of the solver are :**
  - Customizable collocation approximation (Legendre / Chebyshev polynomial based/User defined)
  - Novel adaptive grid refinement schemes
  - NLP solution using algorithmic differentiation
  - Sophisticated post-processing module

Method and usage

Multi-phase Optimal Control Problem (OCP)

↓

Pseudo-spectral Collocation (Global or Multi-segment)



↓

Nonlinear Programming Problem (NLP)

↓

Solve and retrieve solution

Van der Pol Oscillator (2D) : OCP

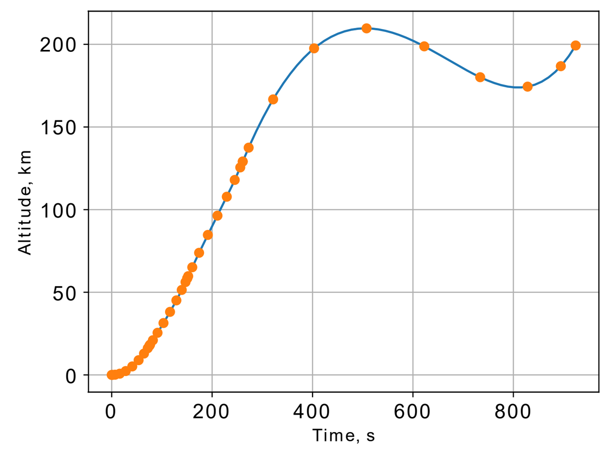
$$
\begin{aligned}
\min_{x,u} \quad & J = 0 + \int_{t_0}^{t_f} \left(x_0^2 + x_1^2 + u^2\right) dt \\
\text{subject to} \quad & \dot{x_0} = (1 - x_1^2)x_0 - x_1 + u \\
& \dot{x_1} = x_0 \\
& x_1 \geq -0.25 \\
& -1 \leq u \leq 1 \\
& x_0(t_0) = 0;\ x_1(t_0) = 1; \\
& t_0 = 0.0;\ t_f = 10
\end{aligned}
$$

Solution under 10 lines of Python code

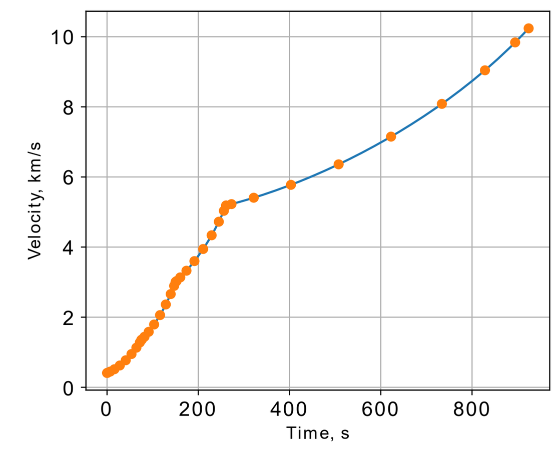
```python
from mpopt import mp
ocp = mp.OCP(n_states=2, n_controls=1, n_phases=1)
ocp.dynamics[0]=lambda x,u,t:[(1-x[1]*x[1])*x[0]-x[1]+u,x[0]]
ocp.running_costs[0]=lambda x,u,t: x[0]*x[0]+x[1]*x[1]+u*u
ocp.lbx[0][1] = -0.25
ocp.lbu[0], ocp.ubu[0] = -1, 1
ocp.x00[0] = [0, 1]
ocp.lbtf[0], ocp.ubtf[0] = 10, 10
mpo, post = mp.solve(ocp, n_segments=1, poly_orders=15,"LGR")
```

Examples of software usage : Results

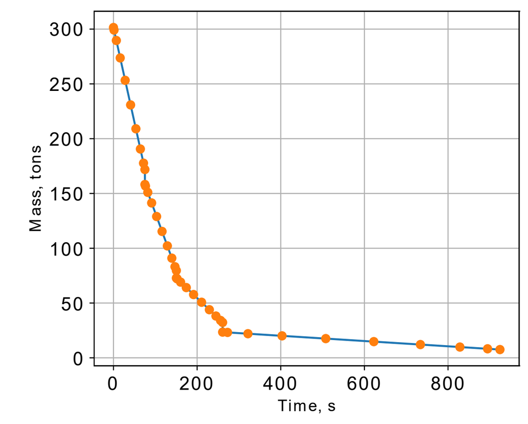
Multi-stage rocket ascent trajectory optimization : Delta-III Rocket



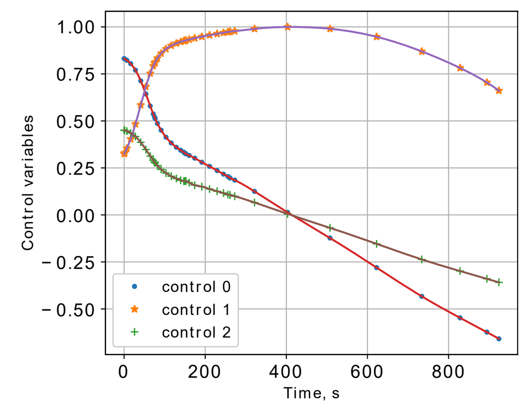
Optimal Altitude profile



Launch vehicle velocity



Mass decay profile



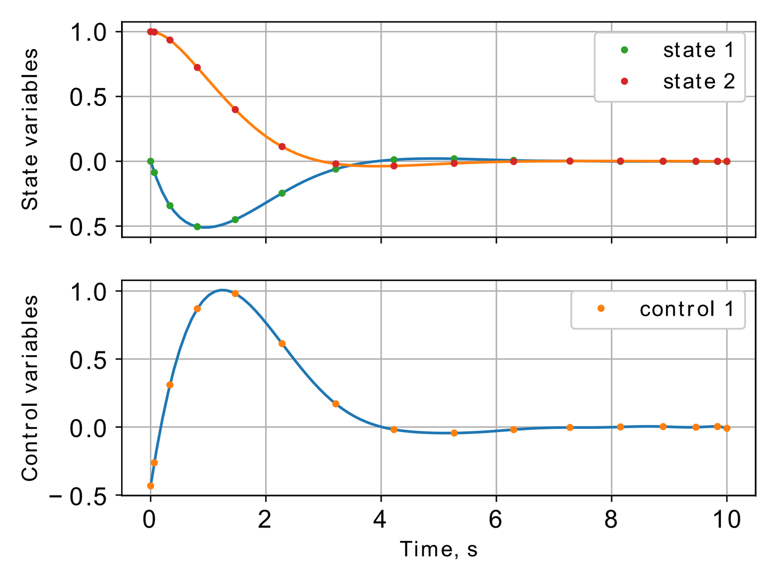
Thrust vector direction

Solver	Payload, kg	Tf, s	solve time, s
MPOPT	7529.7129	924.1393	3.20
PSOPT *	7529.6610	924.1413	3.35
GPOPS-II *	7529.7123	-	-
SOCS *	7529.7125	-	-

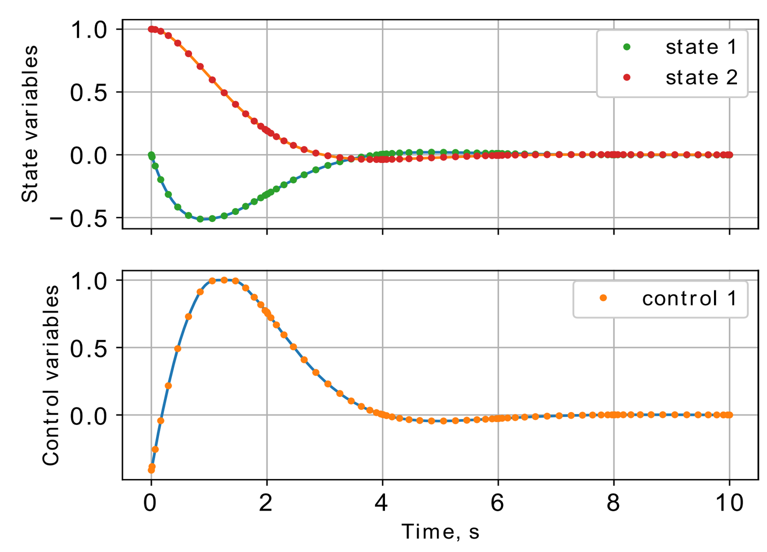
*Popular optimal control software

Table: Comparison with similar software

Global, multi-segment collocation e.g. : van der Pol Oscillator (2D)



Global collocation : Chebyshev-Gauss-Lobatto (CGL)



Multi-segment collocation : Legendre-Gauss-Radau (LGR)

Case	Optimal cost	NLP var.	NLP cons.	Time total	Time NLP sol.
LGR, 15 nodes	2.87373	50	47	85.2 ms ± .842 ms	≈ 9.5ms
CGL, 15 nodes	2.87397	50	47	84.2 ms ± 1.14 ms	≈ 8.7ms
LGR, 5 seg, 75 nodes	2.87332	230	227	326 ms ± .872 ms	≈ 30ms

Adaptive grid refinement schemes

Accurate transcription of continuous time OCP into an NLP is a challenging exercise. Grid refinement overcomes this problem by adapting the location of collocation nodes so that the solution is robust.

Three promising adaptive schemes are developed

Iterative adaptive schemes

Initialize segments with equal width

↓

Solve OCP

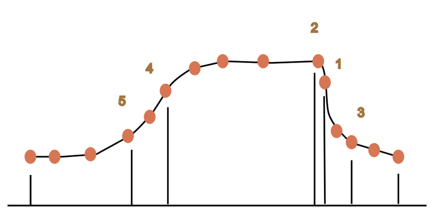
↓

Re-compute segment widths based on present solution

↓

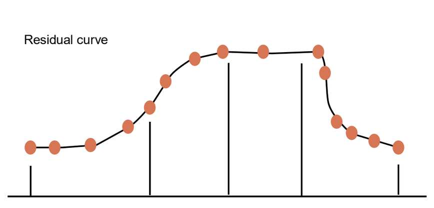
Until convergence

1. Heuristic method



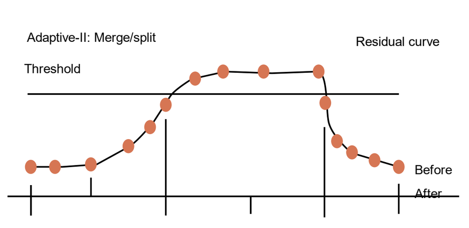
Control slope defines Pivots (1, 2, 3,4, 5) -> pivots decide refined segments

2. Equal residual segments



Residual in dynamics is maintained equal in all segments -> New segment widths

3. Merge/split method

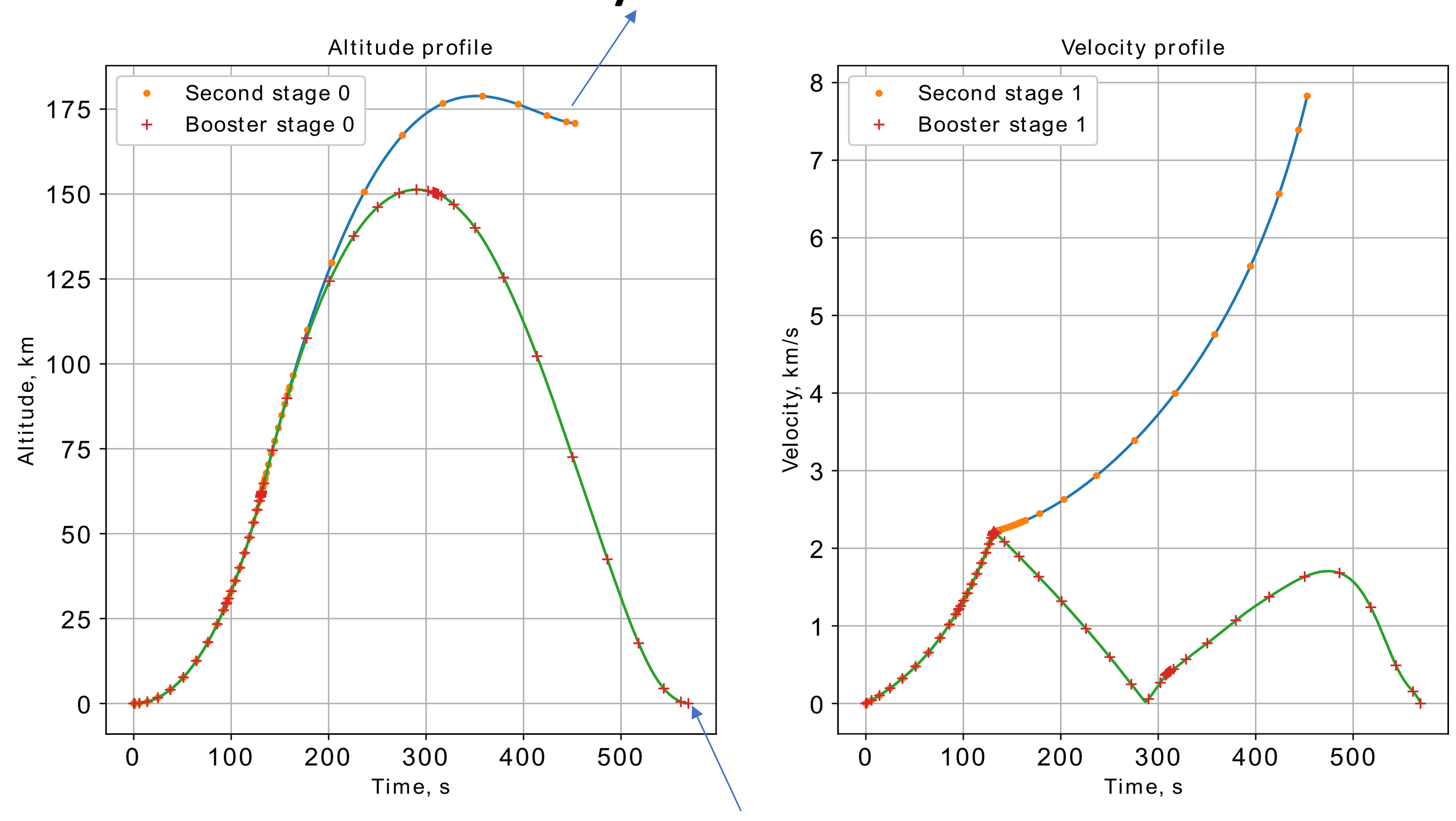


Consecutive segments with acceptable residual are merged and other segments are split

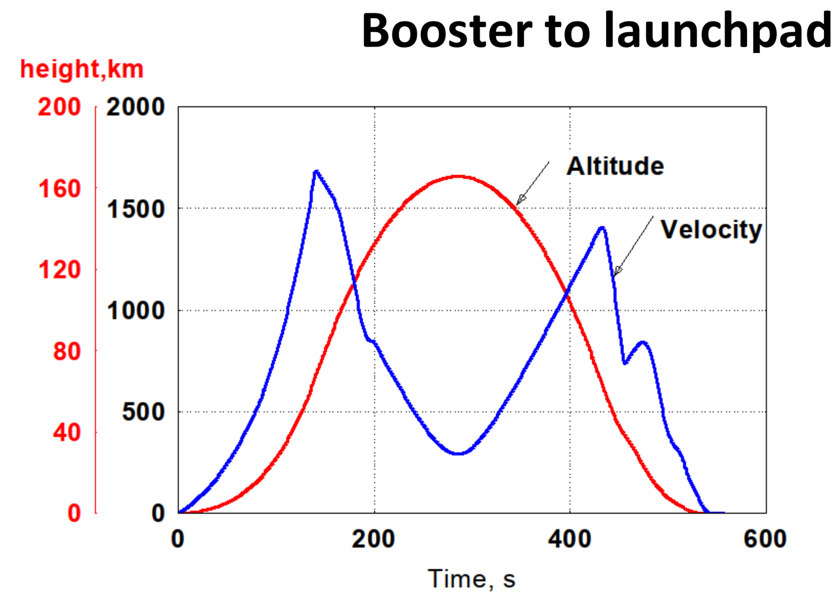
Non- Iterative adaptive scheme → **Add segment width as optmization variables** → **Solve for the optimal solution, grid at one go**

Case study : Reusable rocket trajectory optimization

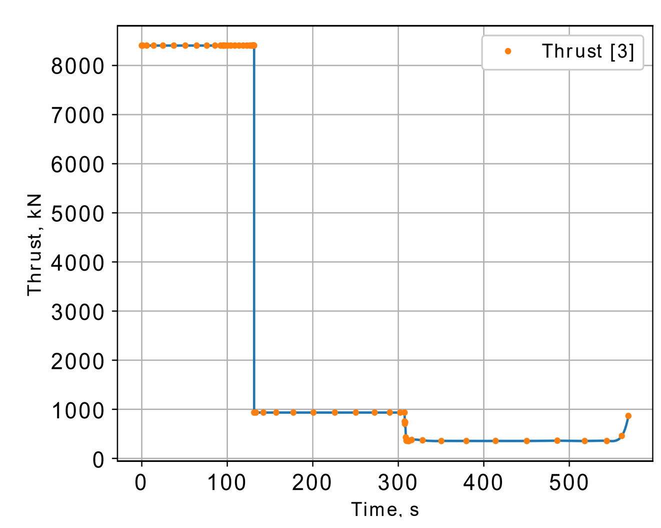
Payload to orbit



Booster to launchpad



SpaceX Falcon9 Telemetry: NROL-76



Optimal booster thrust profile: MPOPT

Parameter	Reference	non-adaptive	Adaptive:Heuristic	Adaptive-III
Payload, kg	17310	17139.5	17267.4	17451.4
mass at landing, kg	22100	22100	22100	22100
MECO, s	131.4	131.4	131.4	131.4
SECO, s	453.4	453.98	453.52	452.86
Landing, s	569.7	579.7	581.3	569.1
Solve time, s	1.9	2.3	2.86	17.3

Author
Devakumar THAMMISETTY, GM

Supervisor
Prof. Colin Jones, LA, EPFL

External examiner
Dr. UP Rajeev, VSSC, ISRO

Acknowledgements
Petr Listov, Doctoral Assistant, EPFL