Interoperating direct and indirect optimal control solvers

Olivier Cots,

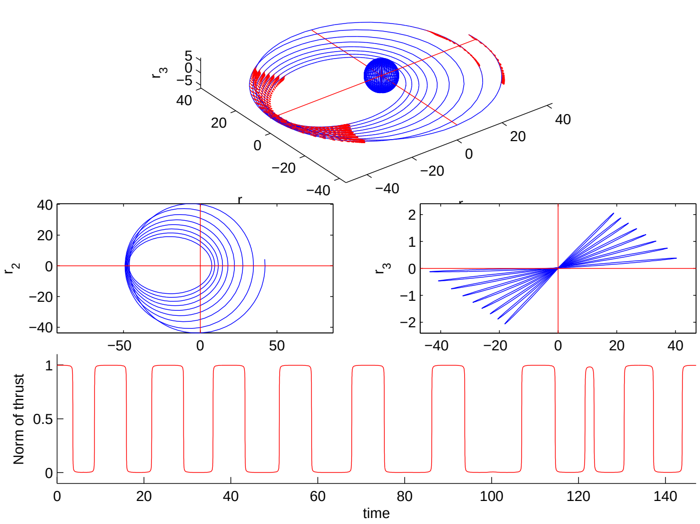
Joint work with J.-B. Caillau, P. Martinon
and the invaluable help of **Inria Sophia SED**

9th International Congress on Industrial and Applied Mathematics (ICIAM 2019),

July 15-19, Valencia, Spain.

Contents

Scientific context - Optimal control



Minimum time control of the Kepler equation (CNES / TAS / Inria / CNRS collaboration):

$$\min \int_0^{t_f} \|u\| \, \mathrm{d}t \equiv \max m(t_f)$$

$$\ddot{q} = -\frac{\mu}{\|q\|^3} + \frac{u}{m}, \quad \dot{m} = -\beta \|u\|, \quad \|u\| \leq T_{\max}, \quad t \in [0, t_f].$$

Fixed initial and final Keplerian orbits, free final time t_f.

Scientific context - Optimal Control Problem

$$
\text{(OCP)} \quad \begin{cases} \min \quad g(x(t_0), x(t_f)) + \int_{t_0}^{t_f} f^0(t, x(t), u(t))\, \mathrm{d}t \\[2ex] \dot{x}(t) = f(t, x(t), u(t)), \quad u(t) \in U, \quad t \in [t_0\,, t_f] \text{ a.e.,} \\ h(t, x(t), u(t)) \leq 0, \quad t \in [t_0\,, t_f] \\ c(x(t_0), x(t_f)) = 0, \end{cases}
$$

- g : boundary cost
- f^0 : running cost
- f : dynamics
- U : control domain
- h : control/state constraints
- c : boundary constraints

Numerical methods:

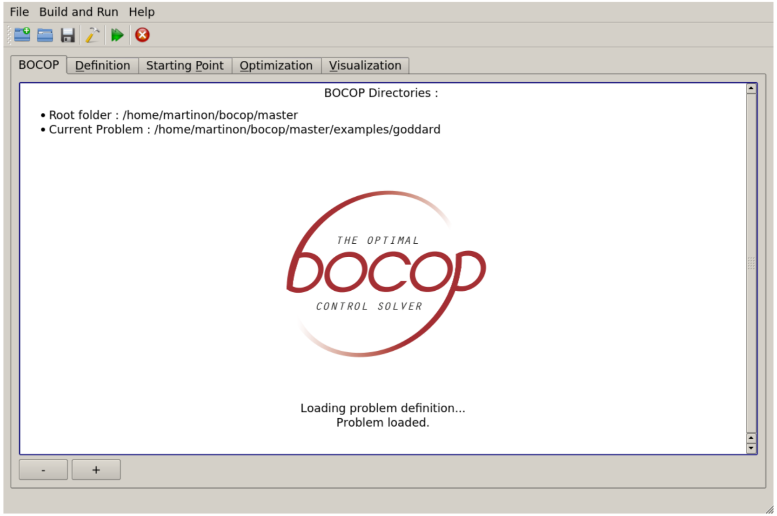
- Direct and Indirect: simple shooting, multiple shooting, collocation.
  - indirect: OCP $\longrightarrow$ PMP $\longrightarrow$ NLE: Newton solvers.
  - direct: OCP $\longrightarrow$ NLP $\longrightarrow$ KKT: Interior points solvers, SQP solvers.
- HJB (value function, cf Dyn. Prog.) and LMI (to compute a lower bound).
- Homotopy techniques to solve a one-parameter family of ocp's.
- Conjugate points computation.

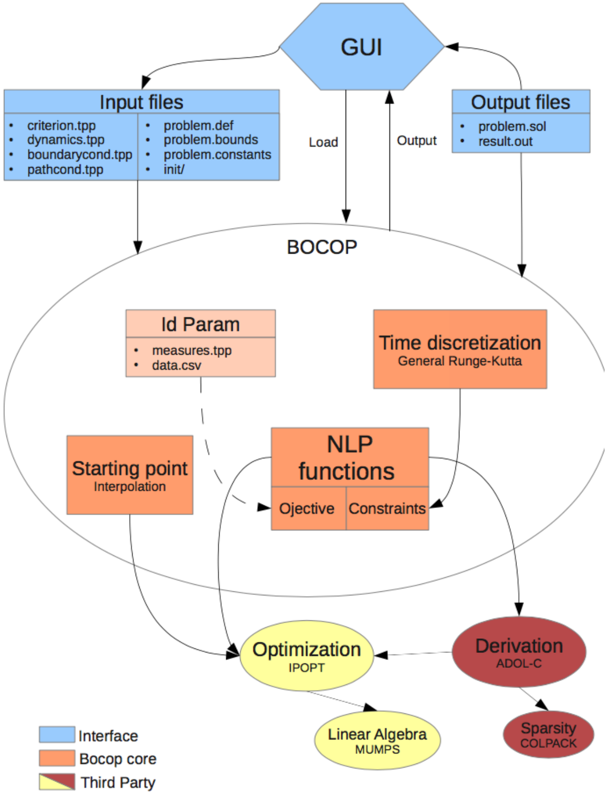
Softwares:

- `Bocop`: direct method (collocation) and "HJB",
- `HamPath`: indirect method (simple and multiple shooting), homotopy and conjugate points,
- ...

See appendix A of the following ref for a survey on numerical tools: S. M. Rogers, *Optimal Control of Nonholonomic Mechanical Systems*, PHD Thesis, 2017.

The Bocop project (bocop.org) began in 2010 in the framework of the Inria-Saclay initiative for an open source optimal control toolbox, and is supported by team COMMANDS.



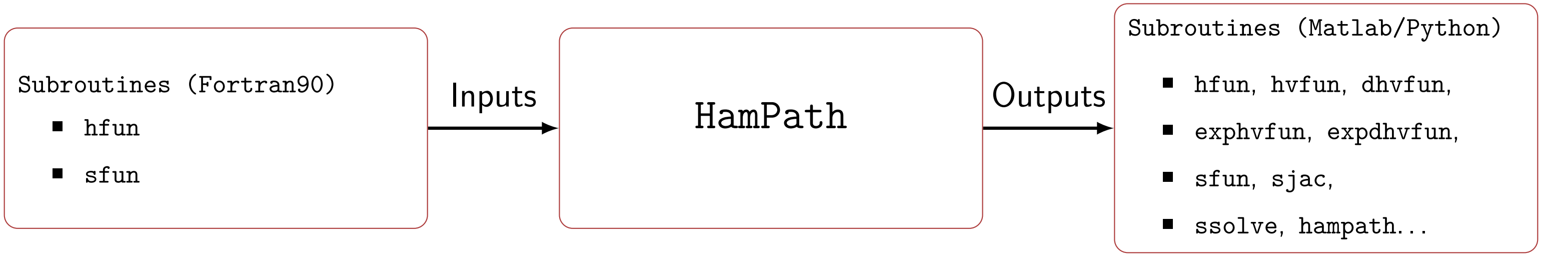


- Direct method: OCP $\longrightarrow$ NLP $\longrightarrow$ KKT
- interior point solver IPOPT
- automatic differentiation Adol-C/CppAD

Technological context - HamPath software

The `HamPath` package (hampath.org) is a an open-source software developed since 2009, jointly by CNRS (Toulouse, Dijon) and Inria (Sophia).

- Indirect method: OCP $\longrightarrow$ PMP $\longrightarrow$ NLE
- Newton solver: `hybrj` from `MINPACK`
- automatic differentiation: `TAPENADE`
- integration: Runge-Kutta schemes with variable step-size as `dopri5`, `dop853`, `radau5` (9, 13)



Possibilities: indirect simple and multiple shooting, homotopy and conjugate points computation

Context - Objectives

Inria project (started June 1st 2019):

ct (Control Toolbox) - Towards a collaborative set of reference tools to solve ocp's

Objectives.

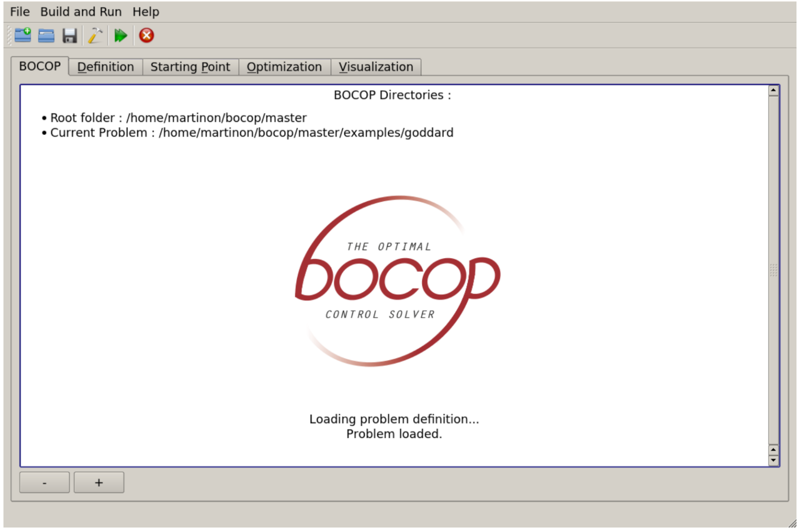
- Interoperate `Bocop` and `HamPath` (complementary approaches),
- Provide a high-end interface.

People involved.

- McTAO (Inria Sophia): J.-B. Caillau, J.-B. Pomet
- APO (CNRS): O. Cots, J. Gergaud
- CAGE (Inria Paris): P. Martinon
- COMMANDS (Inria Saclay): F. Bonnans
- Sophia SED

Interoperating of `Bocop` and `HamPath`: GUI demo, Python notebook demo.

Direct approach - Bocop software



- C / C++, linux/win/mac, GUI, EPL License
- OCP, model identification, delay systems
- General Runge-Kutta schemes, IPOPT solver, automatic differentiation (AdolC/CppAD)

Direct approach - Time discretization, Runge Kutta schemes

$$
(\mathrm{OCP}) \qquad \begin{cases} \min \quad g(x(t_0), x(t_f)) + \int_{t_0}^{t_f} f^0(t, x(t), u(t))\,\mathrm{d}t \\ \dot{x}(t) = f(t, x(t), u(t)), \quad u(t) \in U, \quad t \in [t_0\,, t_f] \text{ a.e.}, \\ h(t, x(t), u(t)) \leq 0, \quad t \in [t_0\,, t_f], \\ c(x(t_0), x(t_f)) = 0, \end{cases}
$$

Time discretization. $(t_i)_{i=0,\ldots,N}$, usually uniform with step h.

Decision variable. new state and control variables: $X := (x_i, u_i)$.

NLP Transcription. The OCP is reformulated in terms of the unknown X.

- objective: boundary cost $g(x_0, x_N)$, running cost as sum of terms $f^0(t_i, x_i, u_i)$.
- boundary conditions: $c(x_0, x_N)$.
- path constraints: $h(t_i, x_i, u_i)$, $i = 0, \cdots, N$.
- dynamics: general Runge Kutta formulas

General RK formula. s stages, coefficients a_{ij}, b_i, c_i such that $\sum_{i=1}^{s} b_i = 1$ and $c_i = \sum_{j=1}^{i-1} a_{ij}$.

Butcher form:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

The formula for one step is:

$$k_i = f\big(t_0 + c_i h, x_0 + h \sum_{j=1}^{s} a_{ij} k_j\big), \quad \text{for } i \in [\![1\,,s]\!],$$

$$x_1 = x_0 + h \sum_{i=1}^{s} b_i k_i,$$

Setting the unknown $X := (x_i, u_i, k_i)$, we finally obtain the discretized problem (NLP)

$$\min F(X), \quad C_{\text{LB}} \leq C(X) \leq C_{\text{UB}}.$$

$\rightarrow$ A direct method solves (NLP) as an approximation of (OCP).

Remark. KKT for NLP tends to PMP for OCP when $h \rightarrow 0$.

1D rocket ascent with maximal final mass:

- State: $q = (h, v, m)$
- Objective: $\max\ m(t_f)$
- Dynamics:

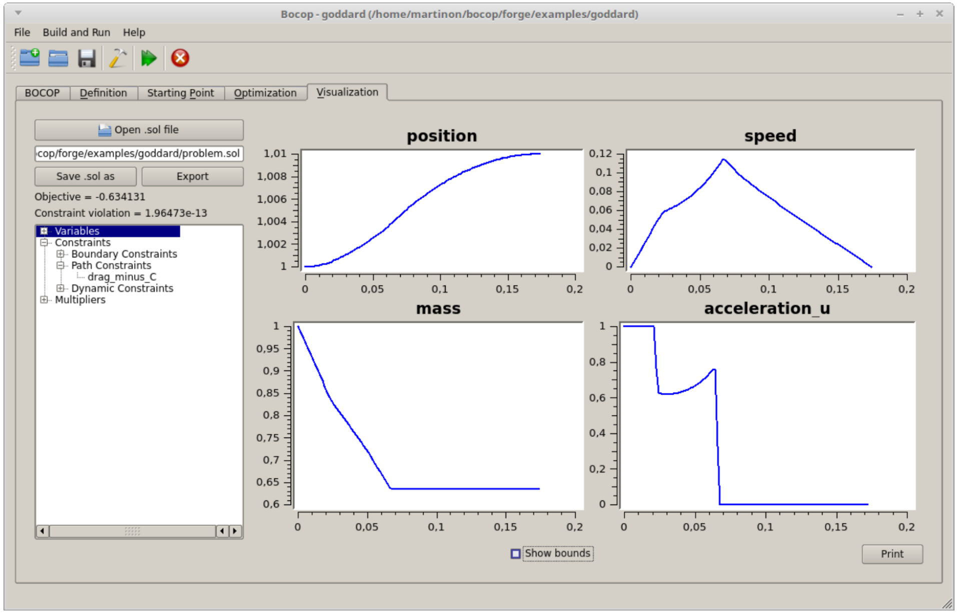
$$\begin{cases} \dot{h} &= v \\ \dot{v} &= \dfrac{1}{m}(c\,u - D(h,v)) - g(h) \\ \dot{m} &= -b\,u \end{cases}$$

- Constraints:

$$\begin{cases} t_f \text{ free} \\ u \in [0,1] \\ h(0) = 1,\ \ v(0) = 0,\ \ m(0) = 1 \\ h(t_f) = 1.01 \end{cases}$$

Direct approach - Bocop - Goddard problem

1D rocket ascent with maximal final mass: solution of the form B^+SB^0.

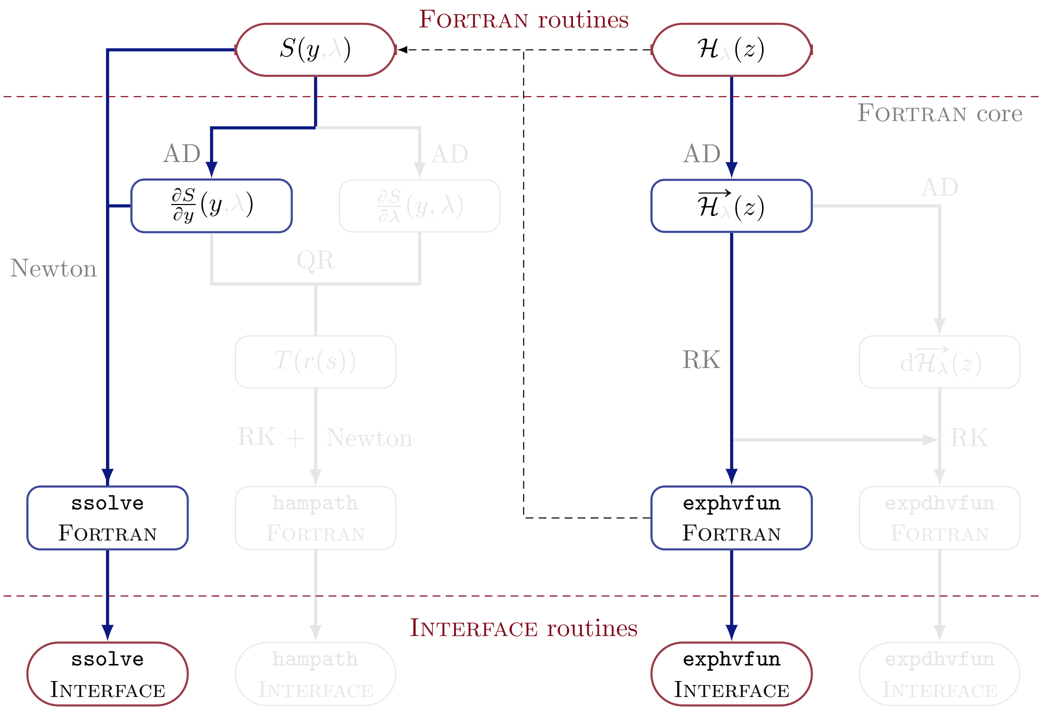


++ no a priori on the structure, robust to the initial guess,

– accuracy can be limited by discretization/structure.

New Bocop: micro-swimmer example.

Indirect approach - HamPath software



Let consider:

$$(\mathrm{P}_1) \qquad \begin{cases} J(u(\cdot)) := \dfrac{1}{2}\displaystyle\int_0^{t_f} u(t)^2 \, \mathrm{d}t \longrightarrow \min \\ \dot{x}(t) = -x(t) + u(t), \quad u(t) \in \mathbb{R}, \quad t \in [0\,,\,t_f] \text{ p.p.}, \quad x(0) = x_0, \\ x(t_f) = x_f, \end{cases}$$

with $t_f := 1$, $x_0 := -1$, $x_f := 0$ and $\forall\, t \in [0\,,\,t_f]$, $x(t) \in \mathbb{R}$.

Let consider:

$$
(\mathrm{P}_1) \qquad \begin{cases} J(u(\cdot)) := \dfrac{1}{2}\displaystyle\int_0^{t_f} u(t)^2 \,\mathrm{d}t \longrightarrow \min \\ \dot{x}(t) = -x(t) + u(t), \quad u(t) \in \mathbb{R}, \quad t \in [0\,,\,t_f] \text{ p.p.}, \quad x(0) = x_0, \\ x(t_f) = x_f, \end{cases}
$$

with $t_f := 1$, $x_0 := -1$, $x_f := 0$ and $\forall\, t \in [0\,,\,t_f]$, $x(t) \in \mathbb{R}$.

- Pseudo-Hamiltonian: $H(x,p,p^0,u) := p\,(-x+u) + p^0\,\frac{1}{2}u^2, \quad p^0 = -1$ (cas normal)
- Maximization condition: $u_s(x,p) := -p/p^0 = p, \quad \left(\frac{\partial^2 H}{\partial u^2} = p^0 = -1 < 0\right)$,
- We get the boundary value problem (BVP) :

$$
(\mathrm{BVP}_1) \qquad \begin{cases} \dot{x}(t) &= \quad \partial_p H[t] = -x(t) + u_s(x(t),p(t)) = -x(t) + p(t), \\ \dot{p}(t) &= -\partial_x H[t] = \quad p(t), \\ x(0) &= x_0, \quad x(t_f) = x_f, \end{cases}
$$

where $[t] := (x(t), p(t), p^0, u_s(x(t), p(t)))$.

We want to solve (BVP$_1$):

$$(\mathrm{BVP}_1) \qquad \begin{cases} \dot{x}(t) &= \quad \partial_p H[t] = -x(t) + p(t), \\ \dot{p}(t) &= -\partial_x H[t] = \quad p(t), \\ x(0) &= x_0, \quad x(t_f) = x_f. \end{cases}$$

- We introduce :

$$\overrightarrow{H}(z,u) := \left(\frac{\partial H}{\partial p}(z, p^0, u), -\frac{\partial H}{\partial x}(z, p^0, u) \right), \quad z := (x, p).$$

- We denote by $z(\cdot, x_0, p_0)$ the solution of $\dot{z}(t) = \overrightarrow{H}(z(t), u_s(z(t)))$, $z(0) = (x_0, p_0)$.

We want to solve (BVP$_1$):

$$(\mathrm{BVP}_1) \qquad \begin{cases} \dot{x}(t) &= \quad \partial_p H[t] = -x(t) + p(t), \\ \dot{p}(t) &= -\partial_x H[t] = \quad p(t), \\ x(0) &= x_0, \quad x(t_f) = x_f. \end{cases}$$

- We introduce :

$$\vec{H}(z,u) := \left(\frac{\partial H}{\partial p}(z, p^0, u), -\frac{\partial H}{\partial x}(z, p^0, u) \right), \quad z := (x,p).$$

- We denote by $z(\cdot, x_0, p_0)$ the solution of $\dot{z}(t) = \vec{H}(z(t), u_s(z(t)))$, $z(0) = (x_0, p_0)$.
- We define the shooting function by:

$$\begin{array}{rccl} S\colon & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & p_0 & \longmapsto & S(p_0) := \Pi_x(z(t_f, x_0, p_0)) - x_f, \end{array}$$

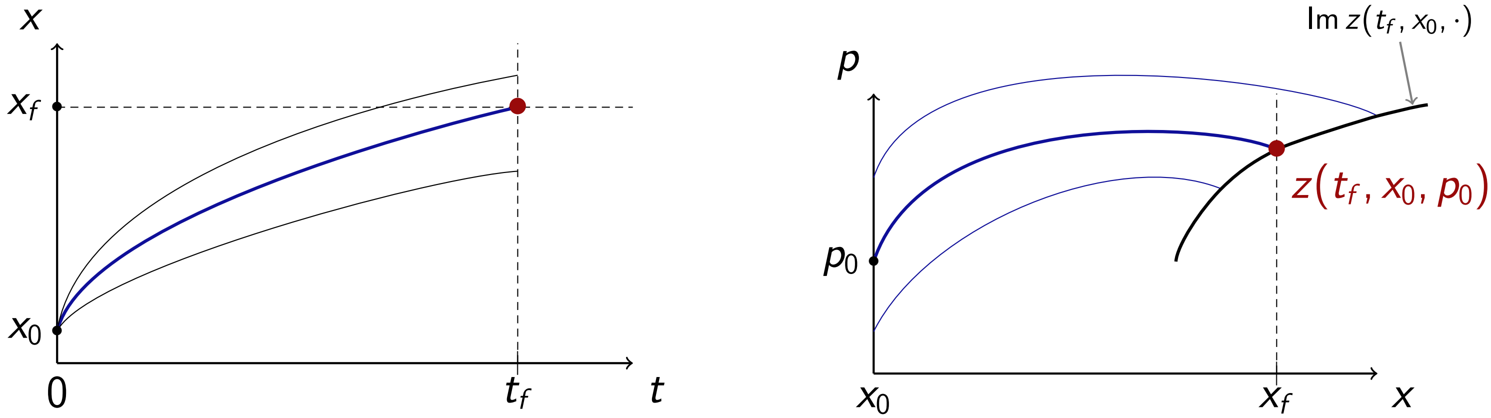
où $\Pi_x(x,p) = x$.

- Solve (BVP$_1$) amount to solve $S(p_0) = 0$. This is the simple shooting method.

Indirect approach - Simple shooting

Shooting function: $S(p_0) = \Pi_x(z(t_f, x_0, p_0)) - x_f = x(t_f, x_0, p_0) - x_f$.

Shooting method: solve $S(p_0) = 0$.



Algorithms:

- $z(\cdot, x_0, p_0)$ is computed with Runge-Kutta schemes.
- $S(p_0) = 0$ is solved by a Newton method (sensitive to initialization).

Let consider: $H(z,u) = H_0(z) + u\, H_1(z)$, H_0, H_1 smooth, $z = (x,p)$, $u \in [u_{\min}\,, u_{\max}]$.

The maximization condition gives:

$$\begin{cases} u_{\max} & \text{si} \quad H_1(z) > 0, \\ u_s(z) & \text{si} \quad H_1(z) = 0, \\ u_{\min} & \text{si} \quad H_1(z) < 0, \end{cases}$$

with $u_s(z) \in (u_{\min}\,, u_{\max})$ the *singular control*.

We define

$$\begin{aligned} \overrightarrow{H_+} &= \overrightarrow{H_0} + u_{\max}\, \overrightarrow{H_1}, \\ \overrightarrow{H_s} &= \overrightarrow{H_0} + u_s\, \overrightarrow{H_1}, \\ \overrightarrow{H_-} &= \overrightarrow{H_0} + u_{\min}\, \overrightarrow{H_1}, \end{aligned}$$

1D rocket ascent with maximal final altitude:

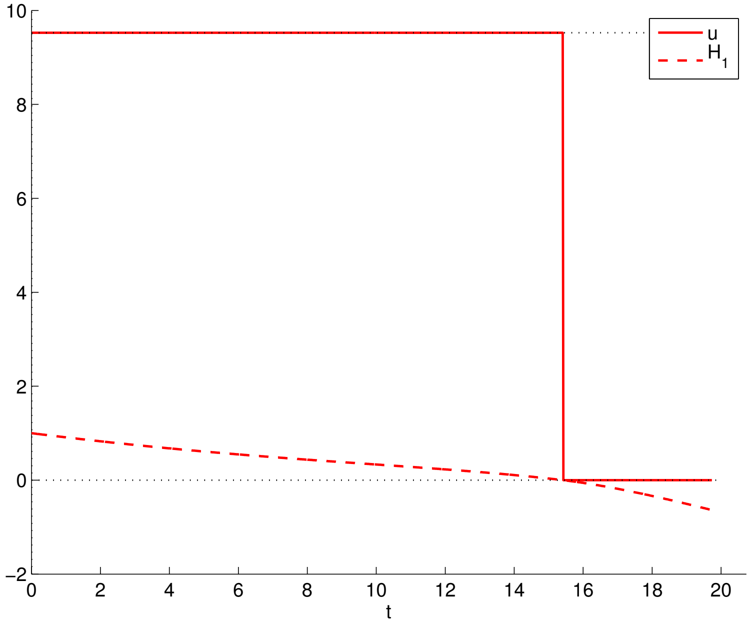
- State: $q = (h, v, m)$
- Objective: $\max h(t_f)$
- Dynamics:

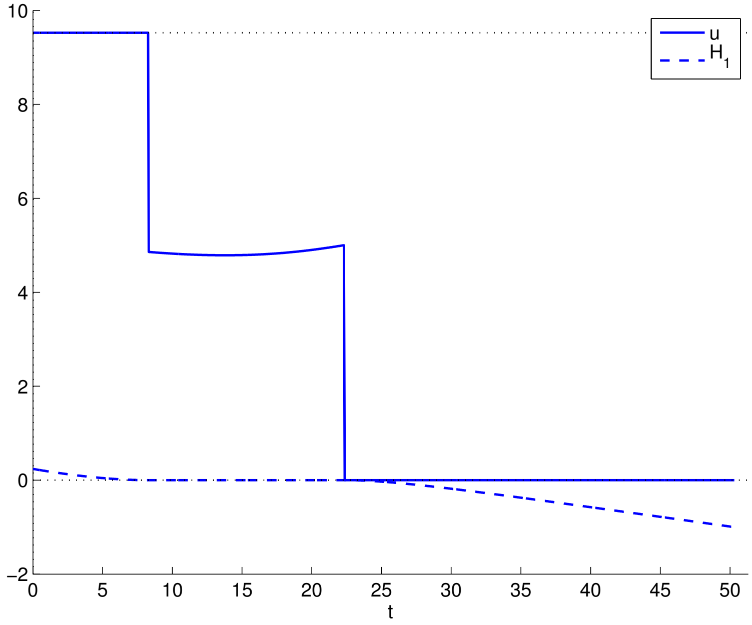
$$\begin{cases} \dot{h} &= v \\ \dot{v} &= \dfrac{1}{m}(u - D(h,v)) - g(h) \\ \dot{m} &= -u \end{cases}$$

- Constraints:

$$\begin{cases} t_f \text{ fixed} \\ u \in [0, u_{\max}] \\ h(0) = v(0) = 0, \;\; m(0) = 214.839 \\ \hfill m(t_f) = 67.9833 \end{cases}$$

Optimal control u and *swtiching function* H_1 (recall $H(z,u) = H_0(z) + u\,H_1(z)$).





$t_f = 20$. The solution is Bang-Bang.

$t_f = 50$. The solution is Bang-Singular-Bang.

Indirect approach - Conclusion

- Indirect method: OCP $\longrightarrow$ PMP $\longrightarrow$ NLE.

+ fast to converge,
+ accurate,

– – work to be done by hand: a priori on the structure,

– – sensitive to initialization.

- New `HamPath`: Kepler demo.

Interoperate Bocop and HamPath

Inria project (started June 1st 2019):

ct (Control Toolbox) - Towards a collaborative set of reference tools to solve ocp's

Objectives.

- Interoperate `Bocop` and `HamPath` (complementary approaches),
- Provide a high-end interface.

People involved.

- McTAO (Inria Sophia): J.-B. Caillau, J.-B. Pomet
- APO (CNRS): O. Cots, J. Gergaud
- CAGE (Inria Paris): P. Martinon
- COMMANDS (Inria Saclay): F. Bonnans
- Sophia SED

Interoperate Bocop and HamPath

Starting point. The following scheme commutes if the discretization of the state-costate equation is a partitioned Runge-Kutta and symplectic scheme.

$$\begin{tikzcd}
\text{OCP} \arrow[r, "\text{PMP}"] \arrow[d, "\text{Disc.}"'] & \text{BVP} \arrow[d, "\text{Disc.}"] \\
\text{NLP} \arrow[r, "\text{KKT}"] & \text{NLE}
\end{tikzcd}$$

See: E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration. Structure–Preserving Algorithms for Ordinary Differential Equations*, vol 31 of *Springer Series in Computational Mathematics*, 2006.

Key idea. Use `Bocop` to initialize `HamPath` and determine the optimal control structure and active state constraints if any.

Remark. The determination of the structure in an automatic way is an open question, but some examples have been treated manually, see for instance:

- Bonnard, B.; Claeys, M.; Cots, O.; Martinon, P.; *Geometric and numerical methods in the contrast imaging problem in nuclear magnetic resonance*. Acta Appl. Math. **135** (2015), no. 1, 5–45.
- Cots, O.; Gergaud, J.; Goubinat, D.; *Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft*. Optim. Control Appl. Meth. **39** (2018), no. 1, 281–301.

Conclusions

- About interoperating Bocop and HamPath (complementary approaches):

Interoperating bocop and hampath: simple optimal control problem

The idea of this notebook is to see how to use hampath on a simple optimal control problem whose solution is smooth:

$$\int_0^1 |u|^2 \, \mathrm{d}t \to \min,$$
$$\dot{x} = v,$$
$$\dot{v} = -\lambda v^2 + u,$$

with x and v fixed at $t = 0$ and $t = 1$. The Hamiltonian of the problem is:

$$H = p_x v + p_v(-\lambda v^2 + u) - u^2/2$$

Entrée [1]:
```
# Imports
import matplotlib
%matplotlib notebook
from matplotlib import pyplot as plt
```

Step 1: bocop

Entrée [3]:
```
# Bocop call (uses a definition file)
import bocop
if(bocop.launchProblem("controlLisse", "../test/problem_bocop", 100) == 0):
    print("Bocop OK")
```

Entrée [ ]:
```
# Reads solution file
objective, time_steps, time_stages, state, control, parameters, costate, boundary_mult, status = bocop.readSolFile('pr
initial_costate = boundary_mult[0:2]
print("bocop solution (costate approximation):", initial_costate)
```

Step 2: hampath

Entrée [ ]:
```
# Imports
import hampath
import numpy as np
```

Entrée [ ]:
```
# Initializations
t0      = 0.0                                              # Initial time
tf      = 1.0                                              # Final time
q0      = np.array([-1.0,0.0])                             # Initial state
n       = len(q0)                                          # State dimension
par     = np.array([t0, tf, q0[0], q0[1], 0.0, 0.0, 0.0])  # t0, tf, x_0, v_0, x_f, v_f, lambda_0
options = hampath.Options(TolX=1e-10, TolOdeAbs=1e-10)
```

Entrée [ ]:
```
# Imports pre-compiled plugin
import use_case_3
!pygmentize ./use_case_3/hfun.f90
```

Entrée [ ]:
```
# hampath call (single shooting)
yGuess = initial_costate # initial guess from bocop
[p0sol,ssol,nfev,njev,flag] = hampath.ssolve(use_case_3,yGuess,options,par)
```

- About providing a high-end interface: the work is in progress but do not hesitate to try the current versions of Bocop and HamPath.

References

- Bocop: bocop.org,
- HamPath: hampath.org.
- Bonnans, F.; Giorgi, D.; Grelard, V.; Maindrault, S.; Martinon, P.; *Bocop — A collection of examples*. Tech. report, 2015.
- Bonnard, B.; Claeys, M.; Cots, O.; Martinon, P.; *Geometric and numerical methods in the contrast imaging problem in nuclear magnetic resonance*. Acta Appl. Math. **135** (2015), no. 1, 5–45.
- Caillau, J.-B.; Cots, O.; Gergaud, J.; *Differential pathfollowing for regular optimal control problems*. Optim. Methods Softw. **27** (2012), no. 2, 177–196.
- Cots, O.; Gergaud, J.; Goubinat, D.; *Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft*. Optim. Control Appl. Meth. **39** (2018), no. 1, 281–301.