Spring 2019

# Stanford
# AA 203: Introduction to Optimal Control and Dynamic Optimization
# Problem set 6, due on May 15

**Problem 1:** Read the article by Betts "Survey of numerical methods for trajectory optimization," AIAA J. of Guidance, Control and Dynamics, 21:193-207, 1998, and write a 1/2 page summary of his suggestions/conclusions. (The article is available on Canvas.)

**Problem 2:** Consider the optimal control problem

$$(\mathbf{OCP})_1 \quad \begin{cases} \min \ h(y(1)) = -y(1) \\ \dot{x}(t) = -x(t)\,u(t) + y(t)\,u^2(t), \ \ \dot{y}(t) = x(t)\,u(t) - 3y(t)\,u^2(t) \\ x(0) = 1, \ y(0) = 0 \\ 0 \leq u(t) \leq 1, \ t \in [0, 1] \end{cases}$$

where $0 \leq x(t) \leq 1$ and $0 \leq y(t) \leq 1$ represent concentrations of chemical substances that react according to the above differential equations, under a temperature control action represented by $0 \leq u(t) \leq 1$. The final time is fixed, namely: $t_f = 1$. The objective consists of maximizing the concentration of the second substance $y$ starting from a maximal concentration of the first substance $x$.

The effectiveness of direct methods for optimal control problems heavily depends on the rule used to numerically integrate the differential equations (and the cost). Herein, we describe a simple rule for integration known as *trapezoidal rule*, which resembles the classical forward Euler scheme, but is much more efficient. Specifically, consider a dynamical system $\dot{x} = f(x(t), u(t))$ and select two points $a < b$ in $[0, t_f]$. By the fundamental theorem of calculus, one has

$$x(b) = x(a) + \int_a^b f(x(t), u(t)) \ dt.$$

When $a$ and $b$ are "close enough," the previous integral can be approximated by the area of the trapezoid with vertices $a$, $f(a)$, $f(b)$, and $b$ (see figure 1). One then obtains the approximation

$$x(b) \simeq x(a) + \frac{f(a) + f(b)}{2}(b - a).$$

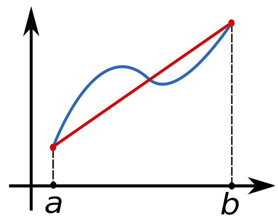


Figure 1: Trapezoidal approximation.

Using such an approximation (referred to as trapezoidal approximation), and given a time discretization $0 = t_0 < t_1 < \cdots < t_N = t_f$, the differential constraints can then be transcribed into the following set of constraints:

$$x(t_{i+1}) - x(t_i) - (t_{i+1} - t_i)\,\frac{f(t_i) + f(t_{i+1})}{2} = 0, \quad i = 0, \ldots, N-1\ . \tag{1}$$

(a) Numerically solve problem $(\mathbf{OCP})_1$ by implementing a direct method leveraging the trapezoidal rule (1). Specifically, fill in the `Matlab` scripts `fDyn.m`, `cost.m`, and `constraint.m` available in the folder `Trapezoidal/ChemicalReaction`, and then run the script `collocation.m` to obtain a solution. Provide plots for the time evolutions of $x$, $y$ and $u$.

(b) What is the optimal quantity of the second substance $y$ at the final time $t_f = 1$?

**Problem 3:** In this problem we will revisit the problem of computing optimal trajectories for a rocket, by considering a more sophisticated version of the Goddard model, namely:

$$(\mathbf{OCP})_2 \begin{cases} \ \min\ h(y(t_f), v(t_f), m(t_f)) = -y(t_f) \\ \dot{y}(t) = v(t),\ \dot{v}(t) = \dfrac{u(t)}{m(t)} - g(y(t)) - \dfrac{D}{m(t)}\rho(y(t))\, v^2(t),\ \dot{m}(t) = -bu(t) \\ y(0) = y_0,\ v(0) = v_0,\ m(0) = m_0 \\ y(t_f) \in \mathbb{R}, v(t_f) \in \mathbb{R},\ m(t_f) = m_f \\ 0 \leq u(t) \leq u_{\max},\ t \in [0, t_f] \end{cases}$$

where $y(t)$ represents the height reached by the rocket at time $t$, $v(t)$ is the velocity of the rocket at time $t$, $m(t)$ is the mass of the rocket at time $t$, the gravitational acceleration is modeled by

$$g(y) = \frac{\mu}{(y(t) + r_E)^2}$$

(where $\mu > 0$ is a constant, $r_E$ is the radius of Earth, and $b > 0$ is the fuel consumption ratio –assumed constant– ), and the drag force is modeled by $\dfrac{D}{m(t)}\rho(y(t))v^2(t)$ (where $D > 0$ is a constant, and $\rho(y(t))$ captures the variation of air density with respect to height). The initial conditions are given by $(y_0, v_0, m_0)$, the final time $t_f$ is fixed and given, the final height $y(t_f)$ and final velocity $v(t_f)$ are free, and the final desired mass $m_f$ accounts for the requirement to have some residual amount of propellant left for further maneuvers.

(a) Numerically solve $(\mathbf{OCP})_2$ by implementing a direct method leveraging the trapezoidal rule (1). Specifically, fill in the scripts `Matlab` scripts `fDyn.m`, `cost.m`, and `constraint.m` available in the folder `Trapezoidal/Goddard`, and then run the script `collocation.m` to obtain a solution. (Numerical values for all constants are provided in the scripts.) Provide plots for the time evolutions of $y$, $v$ $m$, and $u$.

(b) What is the minimum number of discretization points `N` required by the above method to converge (i.e., `Matlab` output: `convergence = 1`)?

(c) Numerically solve $(\mathbf{OCP})_2$ by implementing a direct method leveraging the Hermite-Simpson rule presented in class. Specifically, fill in the scripts `fDyn.m`, `cost.m`, and `constraint.m` available in the folder `HermiteSimpson/Goddard`, and then run the script `collocation.m` to obtain a solution. Provide plots for the time evolutions of $y$, $v$ $m$, and $u$.

(d) What is the minimum number of discretization points `N` required by the above method to converge? Compare and discuss this result with the one obtained in (b).

Learning goals for this problem set:

**Problem 1:** To gain general knowledge about *numerical* methods for trajectory optimization.

**Problems 2 & 3:** To learn how to implement direct methods with different integration schemes, and learn the relative benefits.