

## Lab 1

# Optimal Reentry of a Spacecraft

**Lab Objective:** *We consider the problem of minimizing the heating experienced by a spacecraft during reentry. The boundary value problem (BVP) associated with the reentry of a spacecraft is inherently challenging: the craft must descend quickly enough to enter the atmosphere, but pull out soon enough to prevent overheating or crashing. Problems involving variational calculus and optimal control often include the numerical solution of a challenging BVP.*

A fundamental topic considered in aerospace engineering is the process of landing a spacecraft. Landing a spacecraft requires a massive reduction in the kinetic energy of the craft. That reduction can be accomplished either through the use of massive quantities of fuel (very expensive), or by transforming kinetic energy into heat. That heat must then be absorbed by the atmosphere and the spacecraft. The question then is how to choose the optimal path for reentry into the atmosphere, where the total heating experienced by the craft is minimized.

We begin by giving a control system<sup>1</sup> describing the path of a spacecraft through the atmosphere (we assume the spacecraft is similar to the Apollo craft). Our dependent variables to consider are the velocity  $v$  of the spacecraft, the angle  $\gamma$  of the flight path, and the normalized altitude  $\xi$  above the Earth's surface. ( $\xi = h/R$ , where  $R$  is the radius of the Earth and  $h$  the altitude of the spacecraft above the Earth.) A control variable  $u$  will be used to represent the angle of attack of the spacecraft. The flight path can then be described by

$$\begin{aligned}\dot{v} &= -s\rho v^2 C_D(u) - \frac{g \sin(\gamma)}{(1 + \xi)^2}, \\ \dot{\gamma} &= s\rho v C_L(u) + \frac{v \cos(\gamma)}{R(1 + \xi)} - \frac{g \cos \gamma}{v(1 + \xi)^2}, \\ \dot{\xi} &= \frac{v \sin \gamma}{R}.\end{aligned}\tag{1.1}$$

$C_D$  and  $C_L$  represent drag and lift coefficients, and are functions of  $u$ :  $C_D(u) = 1.174 - .9 \cos u$ ,  $C_L(u) = 0.6 \sin u$ . The atmospheric density  $\rho$  is a function of height,

---

<sup>1</sup>This control problem and its numerical solution are thoroughly described in 'Introduction to Numerical Analysis' by J. Stoer, R. Bulirsch (pg 524). We will mirror their presentation throughout this lab.



Figure 1.1: Apollo 8 during launch

which we represent by  $\rho(\xi) = \rho_0 e^{-R\beta\xi}$ , where  $\rho_0$  is the atmospheric density at the surface of the earth. Other parameters include the force of gravity  $g$ , and  $s = \frac{1}{2}S/m$ , where  $S$  is the frontal area of the craft and  $m$  is its mass. The numerical values we will use are coded below, along with the drag and lift functions.

```
from __future__ import division
from math import pi, sqrt, sin, cos, exp
from numpy import linspace, array, tanh, cosh, ones, arctan
import numpy as np
from scipy.special import erf
from scipy.optimize import root

from bvp6c import bvp6c, bvpinit, deval
from structure_variable import struct

R = 209
beta = 4.26
rho0 = 2.704e-3
g = 3.2172e-4
s = 26600

def C_d(u):
    return 1.174 - 0.9*cos(u)

def C_l(u):
    return 0.6*sin(u)
```

We will require that the trajectory of the spacecraft satisfy the boundary con-

ditions

$$\begin{aligned}
v(0) &= 0.36 \quad (36000 \text{ ft/sec}) & v(T) &= 0.27 \\
\gamma(0) &= -8.1^\circ \frac{\pi}{180^\circ} & \gamma(T) &= 0 \\
\xi(0) &= \frac{4}{R} \quad (h = 400000 \text{ ft}) & \xi(T) &= \frac{2.5}{R}
\end{aligned} \tag{1.2}$$

where  $T$  represents the time at the end of the (first) reentry maneuver. These boundary conditions are similar to those encountered at the end of each Apollo mission to the moon.

To simplify notation, we will also write (1.1) in the form  $y' = G(y)$ , where  $y = [y_0, y_1, y_2]^T = [v, \gamma, \xi]^T$  and  $G$  has component functions  $G = [G_0, G_1, G_2]^T$ . The total heating is

$$J[u] = \int_0^T 10y_0^3 \sqrt{\rho}.$$

The Hamiltonian corresponding to this control system is

$$H = 10y_0^3 \sqrt{\rho} + \lambda_0 G_0 + \lambda_1 G_1 + \lambda_2 G_2, \tag{1.3}$$

where  $\lambda = [\lambda_0, \lambda_1, \lambda_2]^T$  is the adjoint variable. The optimal state and adjoint equations are thus given by

$$\begin{aligned}
\dot{y} &= H_\lambda, \\
\dot{\lambda} &= -H_y
\end{aligned} \tag{1.4}$$

To our boundary conditions we add the terminal condition that  $H = 0$  at  $t = T$ . Finally, from the condition  $\frac{\partial H}{\partial u} = 0$  we find that the optimal control satisfies

$$\tan u = \frac{6\lambda_1}{9y_0\lambda_0}. \tag{1.5}$$

Most BVP solvers require an equal number of differential equations and boundary conditions. Currently we have a free boundary value problem; there are 6 ODEs and 7 boundary conditions, and the length of the reentry maneuver,  $T$ , is still unknown. By making the transformation  $x = t/T$ , and treating  $T$  as a dependent variable, the BVP is now defined on the interval  $(0, 1)$  and is augmented with an additional ODE:

$$\begin{aligned}
y' &= TH_\lambda, \quad ' = \frac{d}{dx}, \\
\lambda' &= -TH_y, \\
T' &= 0.
\end{aligned} \tag{1.6}$$

This BVP has 7 ODEs, and with the 7 boundary conditions introduced earlier it has the required form.



Figure 1.2: The Apollo 8 mission was the first to orbit the moon and return to earth. After a flight of three days from earth, they orbited the moon ten times in 20 hours before making the return trip. This photograph shows a portion of the far side of the moon, as seen by the Apollo 8.

**Problem 1.** Write a function `ode` that implements the ODEs for the original control system along with the adjoint equations (1.6). Notice that the adjoint variables and the final time are coordinates of  $y : y_3 = p_0, y_4 = p_1, y_5 = p_2$ , and  $y_6 = T$ .

```
def ode(x,y):
    # Parameters:
    # x: independent variable (unused in our ODEs)
    # y: vector-valued dependent variable; it is an ndarray
    #     with shape (7,)

    # Returns:
    # ndarray of length (7,) that evaluates the RHS of the ODES
```

## Constructing an Initial Guess

We will use the BVP solver `bvp6c`. Like any solver capable of handling nonlinear problems, `bvp6c` requires an initial guess to jump-start its Newton-like iteration process. Our nonlinear BVP is very sensitive, and requires an initial guess that is quite close to the solution. This sensitivity is physically meaningful. The spacecraft is traveling at a speed far greater than a typical aircraft. If the control is not aggressive, the spacecraft will fall/‘bounce’ back into space as it encounters the atmosphere at a high velocity. However, if the control lasts too long, the craft will

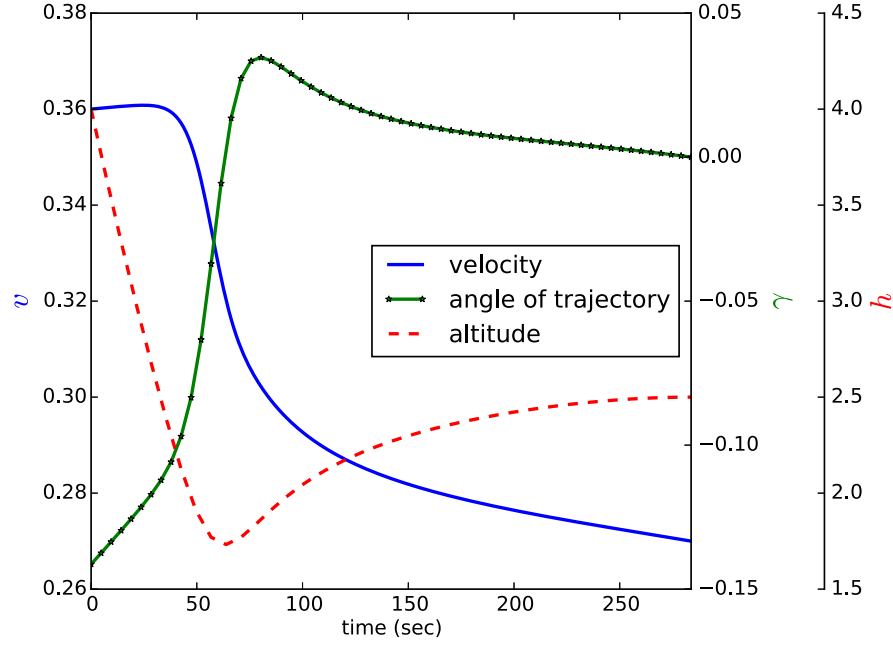


Figure 1.3: The optimal path for the reentry maneuver of a spacecraft. This path minimizes the heating of the spacecraft, and satisfies (1.6),(1.2), and the terminal condition  $H(T) = 0$ .

overheat or crash.

Since this is a sensitive problem, we will use a heuristic method to construct a good initial guess. From aerospace engineers we know that the control  $u$  should empirically look like Figure 1.4; we can create a smooth approximation of the form  $u = p_0 \operatorname{erf}(p_1(p_2 - t/T))$ , where  $p_0, p_1$ , and  $p_2$  are unknown constants. To help us determine these constants, and to find good initial guesses for  $v, \gamma$ , and  $\xi$ , we define an auxiliary BVP to help us come up with a good initial guess for the original BVP:

$$\begin{aligned}
 \dot{y}_0 &= -s\rho y_0^2 C_D(u) - \frac{g \sin(y_1)}{(1 + y_2)^2}, \\
 \dot{y}_1 &= s\rho y_0 C_L(u) + \frac{y_0 \cos(y_1)}{R(1 + y_2)} - \frac{g \cos y_1}{y_0(1 + y_2)^2}, \\
 \dot{y}_2 &= \frac{y_0 \sin y_1}{R}, \\
 \dot{p}_0 &= 0, \\
 \dot{p}_1 &= 0, \\
 \dot{p}_2 &= 0.
 \end{aligned} \tag{1.7}$$

This auxiliary BVP is defined on the interval  $[0, T]$ , where  $T$  is unknown. We guess at  $T$ : the maneuver will occur quickly, so how about 230 seconds? Below we create functions for the ODE and for the boundary conditions.

```

T0 = 230

def ode_auxiliary(x,y):
    u = y[3]*erf( y[4]*(y[5]-(1.*x)/T0) )
    rho = rho0*exp(-beta*R*y[2])
    out = array([-s*rho*y[0]**2*C_d(u) - g*sin(y[1])/(1+y[2])**2,
                  ( s*rho*y[0]*C_l(u) + y[0]*cos(y[1])/(R*(1 + y[2])) -
                    g*cos(y[1])/(y[0]*(1+y[2])**2) ),
                  y[0]*sin(y[1])/R,
                  0,
                  0,
                  0      ])

    return out

def bcs_auxiliary(ya,yb):
    out1 = array([ ya[0]-.36,
                   ya[1]+8.1*pi/180,
                   ya[2]-4/R
                   ])
    out2 = array([ yb[0]-.27,
                   yb[1],
                   yb[2]-2.5/R
                   ])

    return out1, out2

```

The two main functions used by `bvp6c` are `bvpinit` and `deval`. `solve` requires an initial guess, which you will create in Problem 2.

```

options = struct()
# options include abstol, reltol, singularterm, stats, vectorized, maxnewpts, ←
    slopeout,xint
options.abstol, options.reltol = 1e-8, 1e-7
options.fjacobian = ode_auxiliary_jacobian
options.bcjacobian = bcs_auxiliary_jacobian
options.nmax = 2000

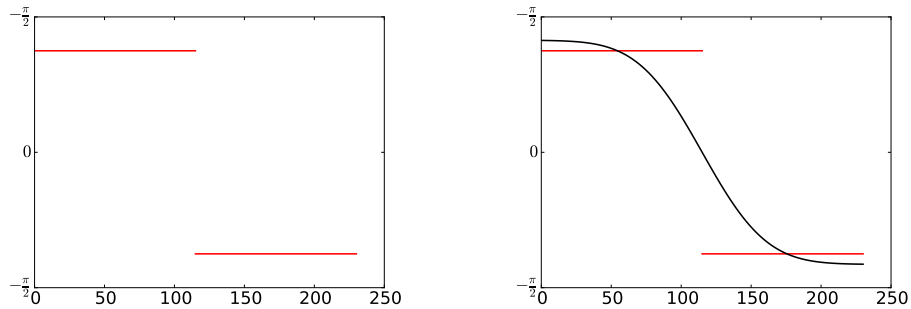
solinit = bvpinit(np.linspace(0,1,100),initial_guess)
sol = bvp6c(ode,bcs,solinit,options)

N = 240
xint = linspace(0,T0,N+1)
num_sol_auxiliary, _ = deval(sol,xint)

```

**Problem 2.** Write the function `guess_auxiliary` referenced in the code above. This function provides an initial guess to `bvp_solver` for the auxiliary BVP described by (1.7) and (1.2). Use the heuristic data provided in Figure 1.4 to find good estimates of  $p_0, p_1$ , and  $p_2$ . Use Figure 1.3 to estimate the trajectories of  $y_0, y_1$ , and  $y_2$ . Then run the code above to check that your initial guess is adequate.

At this point we have constructed good initial guesses for the dependent variables  $y_0, y_1, y_2$ , and  $y_6$  (representing the total time of the maneuver) in the original BVP



Heuristic for the control  $u$ , provided by engineers.

A smooth initial approximation of the control.

Figure 1.4: We construct a smooth estimate for the control  $u$ , by supposing the control has the form  $u = p_0 \operatorname{erf}(p_1(p_2 - t/T))$  and estimating parameters  $p_0, p_1, p_2$ .

(1.6). We now need to construct initial guesses for the adjoint variables  $y_3, y_4$ , and  $y_5$ .

By reexamining the condition  $H_u = 0$ , we find that the optimal control  $u$  satisfies

$$\sin u = \frac{-0.6y_4}{\alpha} \quad \cos u = \frac{-0.9y_0y_3}{\alpha}$$

where  $\alpha = \sqrt{(0.6y_4)^2 + (0.9y_0y_3)^2}$ . From this we know that  $y_3 < 0$ , since  $\cos u > 0$ . A simple guess would be  $y_3 = -1$ . (Recall that the adjoint variables are unique up to some scaling.) We can then approximate  $y_4$  from the relationship

$$\tan u = \frac{6y_4}{9y_0y_3}.$$

To approximate  $y_5$ , we use the identity  $H = 0$ .

**Problem 3.** Adapt your previous code to solve the original, dimension seven BVP. Use the solution of the auxiliary BVP to construct a good initial guess. Plot the control  $u$ . How long does the reentry maneuver take?