```
> restart; unprotect(γ, Γ)
```

The Reentry-Problem (optimal braking maneuver for Apollo 11 type space capsule) as a Boundary Value Problem

Translated Literature (In English): Stoer/Bulirsch, "Introduction to Numerical Analysis", Springer-Verlag, Berlin, Heidelberg. New York 2002.
Original Literature (In German): Stoer/Bulirsch, "Einführung in die Numerische Mathematik II", Springer-Verlag, Berlin,
Heidelberg. New York 1973

Autor: Folkmar Bornemann, 23.2.2000;
Revision für Maple 8 am 2.2.2003;
Revision für Maple 10 am 1.2.2006
Revision für Maple 2015 am 12.10.2015

§1. The problem statement

A. The state variables

Velocity
> $v := x_1$:
Flight-path angle
> $\gamma := x_2$:
Normalized height (as a multiple of Earth's radius)
> $\xi := x_3$:

B. The control variable

Control parameter used for adjusting the motion of the space capsule (Angle)
> u :

C. Auxiliary variables

Density of air (R = Radius of Earth= 20 900 000 ft = 6370 km)
> $\rho := \rho 0\, e^{(-\beta R \xi)} : \beta = 4.26, R = 209$:
Aerodynamic Drag coefficient
> $C_D := c_1 - c_2 \cos(u) : c_1 = 1.174,\ c_2 = 0.9$:
Aerodynamic Lift coefficient
> $C_L := c_3 \sin(u) : c_3 = 0.6$:

D. The constituent Differential Equations

The acceleration, which is given by the time derivative $V = \frac{d}{dt} v(t)$ of the velocity, is

> $V := -\frac{Sm \rho v^2 C_D}{2} - \frac{g \sin(\gamma)}{(1+\xi)^2}$:

Here *Sm* denotes the ratio of the Front surface to the mass of the capsule, *g* is the gravitational acceleration (in Imperial units)

> $Sm = 53200,\ g = 0.32172e-3$:

The time derivative $\Gamma = \frac{d}{dt} \gamma(t)$ of the flight-path angle is

> $\Gamma := \frac{Sm \rho v C_L}{2} + \frac{v \cos(\gamma)}{R(1+\xi)} - \frac{g \cos(\gamma)}{v(1+\xi)^2}$:

The time derivative $\Xi = \frac{d}{dt} \xi(t)$ of the normalized height is

> $\Xi := \frac{v \sin(\gamma)}{R}$:

E. The boundary conditions

(a) The initial position of the capsule as it enters the Earth's atmosphere at $t = 0$:

Velocity = 36 000 ft/sec = 11 km/s = Mach 33

> $v(0) = 0.36$:

Initial flight-path angle (in °)

> $\gamma(0) = -8.1$:

Initial height= 400 000 ft = 122 km

> $\xi(0) = \frac{4}{R}$:

(b) Required conditions over the Pacific by the end of the braking maneuver at time $t = T$:

Velocity = 27 000 ft/sec = 8.2 km/s = Mach 25

> $v(T) = 0.27$:

The flight-path angle as the capsule should be parallel to the surface of the Earth

> $\gamma(T) = 0$:

Height = 250 000 ft = 76 km

> $\xi(T) = \frac{2.5}{R}$:

F. The objective function

The total stagnation point convective heating per unit area is given by the following integral.:

$$J = \int_0^T \Phi \, dt$$

The range of integration is from t=0, the time when the vehicle hits the 400,000 ft atmospheric level, until a time instant T. The vehicle is to be manueuvered into an initial position favorable for the final splashdown in the pacific. Through the freely disposable parameter u, the maneuver is to be executed in such a way that the heating J becomes minimal.

> $\Phi := 10\, v^3 \sqrt{\rho}$

$$\Phi := 10\, x_1^3 \sqrt{\rho 0\, e^{-\beta R x_3}} \tag{1.1}$$

§2. Formulation as a Boundary Value Problem

The theory of optimal control shows that the problem can be regarded as a ***variational problem with differential equation constraints***.

This leads to the following formulation: Construct the ***Hamilton-function*** H, consisting of the linear combination of the function (Φ) inside the integral (J) which is to be minimized and the ***adjoint variables (Lagrange multipliers)***. This forms the right hand side of the constituent differential equations:

$$H := \Phi + \lambda_1 V + \lambda_2 \Gamma + \lambda_3 \Xi :$$

Thus, we now obtain, for each state variable, the right hand side of the differential equation by differentiating H with respect to the corresponding adjoint variable; these are of course just the constituent differential equations as discussed in (D)...

> $f_1 := \frac{\partial}{\partial \lambda_1} H$

$$f_1 := -\frac{1}{2}\, Sm \rho 0\, e^{-\beta R x_3} x_1^2 \left(c_1 - c_2 \cos(u)\right) - \frac{g \sin(x_2)}{(1 + x_3)^2} \tag{2.1}$$

> $f_2 := \frac{\partial}{\partial \lambda_2} H$

$$f_2 := \frac{1}{2}\, Sm \rho 0\, e^{-\beta R x_3} x_1\, c_3 \sin(u) + \frac{x_1 \cos(x_2)}{R\,(1 + x_3)} - \frac{g \cos(x_2)}{x_1\,(1 + x_3)^2} \tag{2.2}$$

> $f_3 := \frac{\partial}{\partial \lambda_3} H$

(2.3)

$$f_3 := \frac{x_1 \sin(x_2)}{R} \tag{2.3}$$

For each of the adjoint variables we obtain the right hand side by the negative derivative of *H* with respect to the corresponding state variables:

> $f_4 := -\frac{\partial}{\partial x_1} H$

$$f_4 := -\left(\frac{1}{2} Sm\,\rho 0\, e^{-\beta R x_3} c_3 \sin(u) + \frac{\cos(x_2)}{R(1+x_3)} + \frac{g\cos(x_2)}{x_1^2 (1+x_3)^2}\right)\lambda_2 + Sm\,\rho 0\, e^{-\beta R x_3} x_1 (c_1 - c_2\cos(u))\lambda_1 - \frac{\sin(x_2)\lambda_3}{R} - 30\, x_1^2 \sqrt{\rho 0\, e^{-\beta R x_3}} \tag{2.4}$$

> $f_5 := -\frac{\partial}{\partial x_2} H$

$$f_5 := -\left(-\frac{x_1 \sin(x_2)}{R(1+x_3)} + \frac{g\sin(x_2)}{x_1 (1+x_3)^2}\right)\lambda_2 + \frac{g\cos(x_2)\lambda_1}{(1+x_3)^2} - \frac{x_1\cos(x_2)\lambda_3}{R} \tag{2.5}$$

> $f_6 := -\frac{\partial}{\partial x_3} H$

$$f_6 := -\left(-\frac{1}{2} Sm\,\rho 0\,\beta\, R\, e^{-\beta R x_3} x_1 c_3 \sin(u) - \frac{x_1\cos(x_2)}{R(1+x_3)^2} + \frac{2\,g\cos(x_2)}{x_1 (1+x_3)^3}\right)\lambda_2 - \left(\frac{1}{2} Sm\,\rho 0\,\beta\, R\, e^{-\beta R x_3} x_1^2 (c_1 - c_2\cos(u)) + \frac{2\,g\sin(x_2)}{(1+x_3)^3}\right)\lambda_1 + \frac{5\, x_1^3 \rho 0\,\beta\, R\, e^{-\beta R x_3}}{\sqrt{\rho 0\, e^{-\beta R x_3}}} \tag{2.6}$$

The optimal control is then given by ***(Pontryagin's Minimum principle)***

> $\frac{\partial}{\partial u} H = 0$:

in our case, as

> $U_{solve} := solve\left(\frac{\partial}{\partial u} H = 0,\ u\right)$

$$U_{solve} := \arctan\left(\frac{c_3 \lambda_2}{c_2 \lambda_1 x_1}\right) \tag{2.7}$$

(Caution: the innocent looking main branch of the function arctan limits the control-values to the

interval $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. *In the model the ocntrol function* $u(t)$ *can be an arbitrary real number; because the value of* $\mathrm{u(t)}$ *only enters in the form of* $\sin(u)$ *and* $\cos(u)$. *The values of* $\mathrm{u(t)}$ *in the interval* $[-\pi, \pi]$ *produces the same result. For flight-path angle* $\gamma(0) = -8.1$ *this does not play any role, in contrast to the shallower flight-path angle of the Apollo 11 with* $\gamma(0) = -6.5$.
What does one have to do in order to overcome this difficulty and to be able to compute for $\gamma(0) = -6.5$ *as well? The CAS-Systems like Maple should be used with mathematical expertise.)*

Thus we have 6 equations with 6 conditions, which looks good but the time *T* is still unknown. Thus we add another state variable using a trivial equation *T(t) = 0*, wherein the time *T* has been transformed to the interval [0,1]. This new state variable is then included as a factor in front of the remaining differential equations.

The variational formulation provides another associated boundary condition, namely the ***Transversality condition*** $H(t = T) = 0$.

§3. Programming the right hand side of the system of differential equations

After the following calculations, we will obtain the 7 differential equations of our boundary value problem:

```
> fct := [u = U_solve, seq(dx_i = T f_i, i = 1..6), dx_7 = 0]
```

$$fct := \left[u = \arctan\left(\frac{c_3\,\lambda_2}{c_2\,\lambda_1\,x_1}\right),\ dx_1 = T\left(-\frac{1}{2}\,Sm\,\rho 0\,e^{-\beta R x_3}\,x_1^2\,(c_1 - c_2\cos(u))\right)\right. \tag{3.1}$$

$$\left. -\frac{g\sin(x_2)}{(1+x_3)^2}\right),\ dx_2 = T\left(\frac{1}{2}\,Sm\,\rho 0\,e^{-\beta R x_3}\,x_1\,c_3\sin(u) + \frac{x_1\cos(x_2)}{R\,(1+x_3)}\right.$$

$$\left. -\frac{g\cos(x_2)}{x_1\,(1+x_3)^2}\right),\ dx_3 = \frac{T x_1\sin(x_2)}{R},\ dx_4 = T\left(-\left(\frac{1}{2}\,Sm\,\rho 0\,e^{-\beta R x_3}\,c_3\sin(u)\right.\right.$$

$$\left.\left. + \frac{\cos(x_2)}{R\,(1+x_3)} + \frac{g\cos(x_2)}{x_1^2\,(1+x_3)^2}\right)\lambda_2 + Sm\,\rho 0\,e^{-\beta R x_3}\,x_1\,(c_1 - c_2\cos(u))\,\lambda_1\right.$$

$$\left. -\frac{\sin(x_2)\,\lambda_3}{R} - 30\,x_1^2\sqrt{\rho 0\,e^{-\beta R x_3}}\right),\ dx_5 = T\left(-\left(-\frac{x_1\sin(x_2)}{R\,(1+x_3)}\right.\right.$$

$$+\frac{g\sin(x_2)}{x_1\,(1+x_3)^2}\Bigg)\lambda_2+\frac{g\cos(x_2)\,\lambda_1}{(1+x_3)^2}-\frac{x_1\cos(x_2)\,\lambda_3}{R}\Bigg),\ dx_6=T\Bigg(-\Bigg($$

$$-\frac{1}{2}\,Sm\,\rho 0\,\beta\,R\,e^{-\beta R x_3}\,x_1\,c_3\sin(u)-\frac{x_1\cos(x_2)}{R\,(1+x_3)^2}+\frac{2\,g\cos(x_2)}{x_1\,(1+x_3)^3}\Bigg)\lambda_2$$

$$-\left(\frac{1}{2}\,Sm\,\rho 0\,\beta\,R\,e^{-\beta R x_3}\,x_1^2\,(c_1-c_2\cos(u))+\frac{2\,g\sin(x_2)}{(1+x_3)^3}\right)\lambda_1$$

$$+\frac{5\,x_1^3\,\rho 0\,\beta\,R\,e^{-\beta R x_3}}{\sqrt{\rho 0\,e^{-\beta R x_3}}}\Bigg),\ dx_7=0\Bigg]$$

Let's look at the cost of evaluating the right hand side if we simple program it directly:

```
with(CodeGeneration):
with(codegen,optimize,makeproc,cost):
cost(fct[]);
```

$$34\ functions+113\ multiplications+79\ subscripts+25\ divisions+8\ assignments+39\ additions \qquad \textbf{(3.2)}$$

Considering an average of 5 flops per function call this leads to 347 flops!

Hence one does not program ir right away. We try to save the intermediate values and reuse them, such as cos(x2) etc. In this way we can save almost a factor of 3 in effort:

```
> fctopt:=[optimize(fct,'tryhard')];cost(fctopt[]);
```

$$fctopt:=\Bigg[t16=x_3,\ t10=1+t16,\ t4=\frac{1}{t10},\ t5=\frac{1}{t10^2},\ t39=2\,t4\,t5,\ t38=g\,t5,$$

$$t17=x_2,\ t7=\sin(t17),\ t37=g\,t7,\ t18=x_1,\ t15=\frac{1}{t18},\ t36=g\,t15,\ t35$$

$$=\beta\,R,\ t21=\frac{1}{R},\ t9=\cos(t17),\ t34=t21\,t9,\ t33=\rho 0\,e^{-t16\,t35},\ t32=t18\,t21,$$

$$t31=t9\,t36,\ t30=Sm\,t33,\ t20=c_2,\ t12=\lambda_2,\ t13=\lambda_1,\ t19=c_3,\ u$$

$$= \arctan\left(\frac{t19\,t12\,t15}{t20\,t13}\right),\ t2 = c_1 - t20\cos(u),\ t29 = t2\,t30,\ t28$$

$$= t19\sin(u)\ t30,\ t14 = t18^2,\ t27 = t14\,t33\,t35,\ t26 = \frac{1}{2}\ t28,\ t22 = pow\left(t33,\right.$$

$$\left.\frac{1}{2}\right),\ t11 = \lambda_3,\ dx_1 = T\left(-\frac{1}{2}\ t14\,t29 - t5\,t37\right),\ dx_2 = T\,(-t5\,t31 + (t34\,t4$$

$$+ t26)\ t18),\ dx_3 = T\,t7\,t32,\ dx_4 = T\left(-\left(t26 + \left(t21\,t4 + \frac{t38}{t18^2}\right)\ t9\right)\ t12\right.$$

$$\left. + t18\,t13\,t29 - t7\,t21\,t11 - 30\,t14\,t22\right),\ dx_5 = T\,((-t11\,t32 + t13\,t38)\ t9 - ($$

$$-t32\,t4 + t36\,t5)\ t12\,t7),\ dx_6 = T\left(-\left(t31\,t39 + \left(-\frac{1}{2}\ t28\,t35\right.\right.\right.$$

$$\left.\left.\left. - t5\,t34\right)\ t18\right)\ t12 - \left(\frac{1}{2}\ Sm\,t2\,t27 + t37\,t39\right)\ t13 + \frac{5\ t18\,t27}{t22}\right),\ dx_7 = 0\Bigg]$$

$$16\ subscripts + 40\ assignments + 19\ additions + 8\ divisions + 66\ multiplications + 7\ functions \quad \textbf{(3.3)}$$

Thus we only need 128 flops.

We need this function evaluation only in a standard programming language. Julia codes are similar to C and MATLAB. Maple has a built-in code optimization for both but we use MATLAB since it is easier to convert this code to Julia (from Maple 2016 onwards one has the option to convert directly to Julia):

```
Matlab(fct,optimize,output="reentry_f.m");
```

§4. Programming the boundary conditions

The 6 boundary conditions on both the boundaries can be written easily. We must program the Transversality condition $H(t = T) = 0$:

> $rfct := subs(seq(x_i = xb_i,\ i = 1..6),\ [u = U_{solve},\ r_7 = H])$:

We do not program this function in this form because the cost would be 117 flops

```
> cost(rfct[]);
```

$$12\ functions + 35\ multiplications + 28\ subscripts + 9\ divisions + 2\ assignments + 13\ additions \quad \textbf{(4.1)}$$

Instead, we program with 82 flops:

```
> ropt:=[optimize(rfct,'tryhard')]; cost(ropt[]);
```

$$ropt := \left[t48 = xb_3,\ t44 = 1 + t48,\ t59 = \frac{g}{t44^2},\ t50 = xb_1,\ t58 = \frac{t50}{R},\ t57 = \rho 0\, e^{-\beta R\, t48},\ t56 = Sm\, t57,\ t54 = t50^2,\ t52 = c_2,\ t51 = c_3,\ t49 = xb_2,\ t47 = \lambda_1,\ t46 = \lambda_2,\ t45 = \frac{1}{t50},\ t42 = \sin(t49),\ u = \arctan\left(\frac{t51\, t46\, t45}{t52\, t47}\right),\ r_7 = \left(\frac{1}{2}\, t50\, t51 \sin(u)\ t56 + \left(\frac{t58}{t44} - t45\, t59\right)\cos(t49)\right) t46 + \left(-\frac{1}{2}\, t54\, (c_1 - t52\cos(u))\ t56 - t42\, t59\right) t47 + t42\, \lambda_3\ t58 + 10\ t50\, t54\, \sqrt{t57}\right]$$

$$10\ subscripts + 17\ assignments + 9\ additions + 6\ divisions + 27\ multiplications + 8\ functions \qquad \textbf{(4.2)}$$

Hence we program using the optimized function:

```
> Matlab(rfct,optimize,output="reentry_bc.m");
```

§5. Verification of the minimum principle

From the Pontryagin's Minimum principle we obtain the necessary condition for the existence of a minimal solution:

```
> 0 ≤ ∂^2/∂u^2 H:
```

Let's we what we obtain:

```
> 0 ≤ simplify(∂^2/∂u^2 H)
```

$$0 \le -\frac{1}{2}\, Sm\, \rho 0\, e^{-\beta R\, x_3}\, x_1 \left(x_1\, c_2 \cos(u)\, \lambda_1 + c_3 \sin(u)\, \lambda_2\right) \qquad \textbf{(5.1)}$$

The condition is as follows *(the control u is constrained to the interval* $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, *which is equivalent to* $0 \le \cos(u)$*)*

```
> λ1 x1 c2 + λ2 c3 tan(u) ≤ 0:
```

and after subtituting the control from (2.7) we obtain

```
> subs(u = U_solve, %) : simplify(%)
```

$$\frac{c_2^2\, \lambda_1^2\, x_1^2 + c_3^2\, \lambda_2^2}{x_1\, c_2\, \lambda_1} \le 0 \qquad \textbf{(5.2)}$$

finally $\lambda_1 \le 0$. From the formula for the control $u = U_{solve}$

$$u = \arctan\left(\frac{c_3\, \lambda_2}{c_2\, \lambda_1\, x_1}\right) \qquad \textbf{(5.3)}$$

it follows that $\lambda_1 \neq 0$. Which leads to $\lambda_1 < 0$ verifying the minimum principle; this condition is an *a posteriori* check.

§6. Calculating a starting trajectory for the multiple shooting method

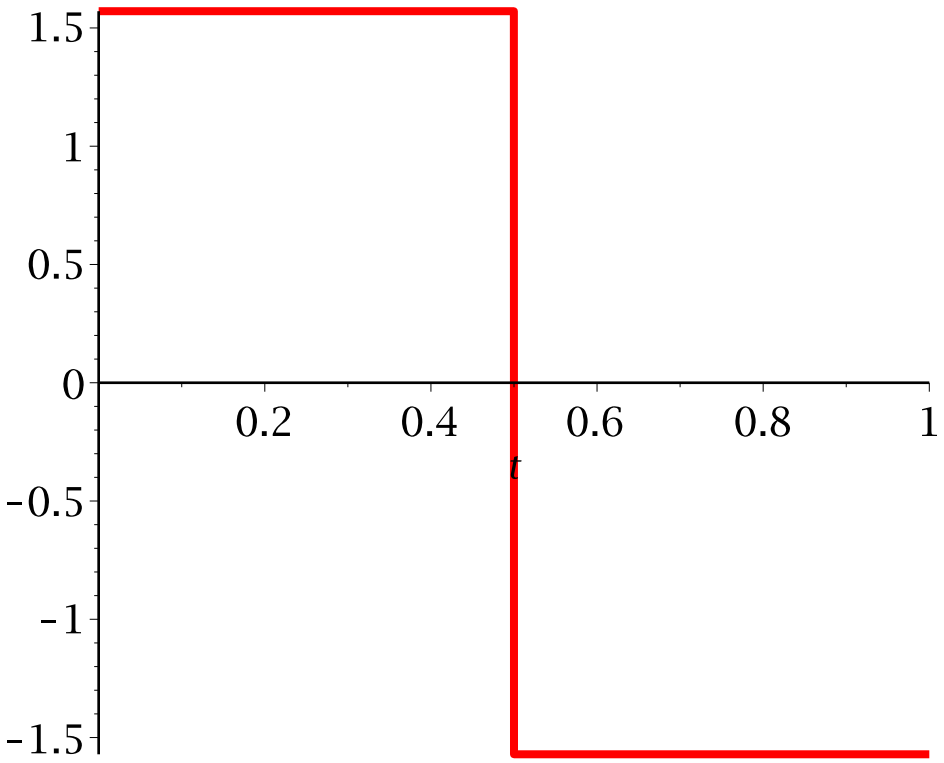
Newton's multiple shooting method requires initial values for the state variables and the adjoint variables. As long as the nodes of the multiple shooting method are not unknown, we need sensible starting values,that is starting trajectories, over the entire time interval. We can then determine the nodes of the multiple shooting method using these starting trajectories.

The system of differential equations has a highly sensitive dependence on the inital values and also exhibits the phenomenon of moving singularities. This is the mathematical formulation of the danger involved in the reentry maneuver. This sensitivity is a consequence of the effect of atmospheric forces, and the physical interpretation of the singularity is a "crash" of the capsule or a "hurling back" into space. These phenomena are independent of the problem of "failure due to heating" during re-entry, which we want to prevent and hence optimize the heating to a minimum.

The idea to get sensible starting trajectory is the following: find a "half-decent" control *u* to solve the boundary value problem (BVP), i.e. get a valid (but far from optimal) trajectory. Then Newton's method needs to only find a better *u* such that the trajectory stays valid and the heat is optimized.

How to get a "half-decent" control-function *u*? In the linear case this would be the so called "bang-bang"-control, i.e. the control function *u* only takes one of the two values -π/2 and +π/2. Here +π/2 means maximal (up-)lift and -π/2 means minimal (up-)lift. At the beginning one would start with maximal (up-)lift in order to slowdown the fall, but then at some time one would use minimal (up-)lift, in order to assure not to stay in outer space (in order to reach the surface of the earth). Using this "half-decent" idea one has only one unknown, the time when to switch from +π/2 to -π/2. The control-function *u* has then the form:

$$> \ plot\left(-\frac{\pi\,\mathrm{signum}(t-0.5)}{2},\ t=0..1,\ color=[red],\ thickness=3\right)$$



Problem:

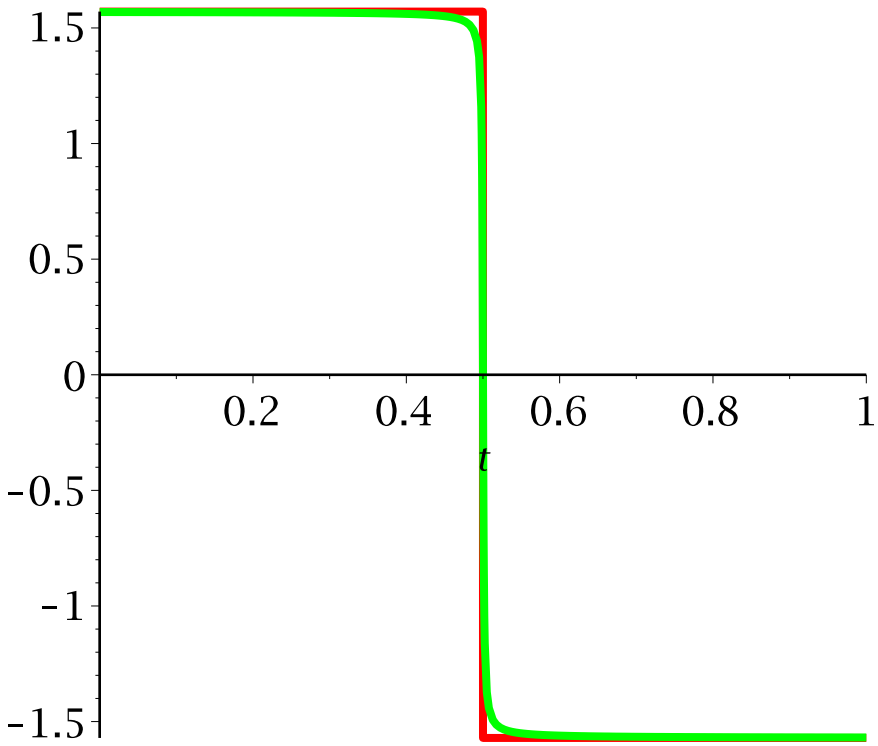
We thus obtained 3 differential equations, 6 constraints, but only two parameters (Duration of the maneuver and the switchover time). It lacks another parameter, here we take the amplitude of the control, not to be set to the full maximum of -π/2. Moreover, the continuity of control is affected adversely.

Thus we try a 2-parameter family of step-like function:

> $U_{test} := p_1 \arctan(1000\ (p_2 - t))$:

For $p_1 = 1$ and $p_2 = \frac{1}{2}$ (which will also be our starting values) we obtain

> $plot\left(\left[-\frac{\pi\ \mathrm{signum}(t-0.5)}{2}, subs\left(p_1 = 1, p_2 = \frac{1}{2}, U_{test}\right)\right], t = 0..1, color = [red, green], thickness = 3\right)$



Now we need an initial guess for *T*. For this, we make a rough calculation. The differential equation for the normalized height is,

$$\frac{\mathrm{d}}{\mathrm{d}t}\, h(t) = v \sin(\gamma)\ ,$$

we know that the velocity v should reduce from 27000 ft/sec to 36000ft/sec and the flight-path angle γ should increase from -8.1° to 0°. This results in the loss of 150000ft in the height. Setting all these constants in the average values we obtain:

150000 ft / *T* = 30000 ft/sec sin(4°),

which gives

T = (150000 / 30000) (180/4/3) sec = 75 sec,

in the order of 1 minute.

Otherwise, we can only hope to solve the auxiliary boundary value problem with the forward or backward shooting. In fact, we suffer with the forward shooting method which results in a shipwreck (or the crashing of the capsule). However, the reverse shooting method provides a viable solution.

Thus we have a start trajectory! No, not quite. We have the start trajectory for the state variables. How to arrive at a meaningful starting trajectory for the adjoint variables is discussed later in §8.

§7. Programming the auxiliary boundary value problem

Briefly

> $fct := [u = U_{test}, seq(dx_i = Tf_i, i = 1..3), seq(dx_i = 0, i = 4..6)]:$

> fct

$$\left[u = -p_1 \arctan(-1000\,p_2 + 1000\,t),\ dx_1 = T\left(-\frac{1}{2}\,Sm\rho 0\,e^{-\beta R x_3}\,x_1^2\,(c_1 - c_2\cos(u)) - \frac{g\sin(x_2)}{(1+x_3)^2}\right),\ dx_2 = T\left(\frac{1}{2}\,Sm\rho 0\,e^{-\beta R x_3}\,x_1\,c_3\sin(u) + \frac{x_1\cos(x_2)}{R\,(1+x_3)} - \frac{g\cos(x_2)}{x_1\,(1+x_3)^2}\right),\ dx_3 = \frac{T x_1\sin(x_2)}{R},\ dx_4 = 0,\ dx_5 = 0,\ dx_6 = 0\right] \tag{7.1}$$

> $Matlab(fct, optimize, output = \text{"aux\_f.m"})$

§8. Initial guess for the adjoint variables

From the Pontryagin's Minimum principle we obtain the necessary condition $\lambda_1 < 0$. Thus we keep the starting value of λ_1 simply:

> $\lambda_{start,\,1} := -1:$

setting this value in the optimal control we obtain an initial guess for λ_2 as:

> $\lambda_{start,\,2} := solve(subs(\lambda_1 = \lambda_{start,\,1}, u = U_{solve}), \lambda_2)$

$$\lambda_{start,\,2} := -\frac{\tan(u)\,c_2\,x_1}{c_3} \tag{8.1}$$

Finally using $H = 0$ (which is valid for the optimal solution) we obtain an initial guess for λ_3

> $\lambda_{start,\,3} := simplify(solve(subs(\lambda_1 = \lambda_{start,\,1}, \lambda_2 = \lambda_{start,\,2}, H = 0), \lambda_3))$

$$\lambda_{start,\,3} := -\frac{1}{2}\,\frac{1}{x_1\,c_3\,(x_3^2 + 2\,x_3 + 1)\cos(u)\sin(x_2)}\left(e^{-\beta R x_3}\,R\,Sm\rho 0\,c_1\,c_3\,x_1^2\,x_3^2\cos(u) + 2\,e^{-\beta R x_3}\,R\,Sm\rho 0\,c_1\,c_3\,x_1^2\,x_3\cos(u) - e^{-\beta R x_3}\,R\,Sm\rho 0\,c_2\,c_3\,x_1^2\,x_3^2 + e^{-\beta R x_3}\,R\,Sm\rho 0\,c_1\,c_3\,x_1^2\cos(u) - 2\,e^{-\beta R x_3}\,R\,Sm\rho 0\,c_2\,c_3\,x_1^2\,x_3 + 20\sqrt{\rho 0\,e^{-\beta R x_3}}\,R\,c_3\,x_1^3\,x_3^2\cos(u) - e^{-\beta R x_3}\,R\,Sm\rho 0\,c_2\,c_3\,x_1^2 + 40\sqrt{\rho 0\,e^{-\beta R x_3}}\,R\,c_3\,x_1^3\,x_3\cos(u) + 20\,x_1^3\sqrt{\rho 0\,e^{-\beta R x_3}}\,R\,c_3\cos(u)\right) \tag{8.2}$$

$$-2\sin(u)\cos(x_2)\,c_2\,x_1^2\,x_3+2\sin(u)\cos(x_2)\,R\,g\,c_2-2\sin(u)\cos(x_2)\,c_2\,x_1^2$$

$$+2\,g\sin(x_2)\,R\,c_3\cos(u)\Big)$$

We program these expressions which are to be evaluated as a function.

> $fct := \left[u = U_{test},\ seq\left(\lambda_i = \lambda_{start,\,i},\ i = 1..3\right)\right]:$

> $Matlab(fct,\ optimize,\ output = \text{"multiplier\_start.m"})$

We will use the boundary value problem solver (bvpsol) from the ODEInterface package in Julia (https://github.com/luchr/ODEInterface.jl)

§9. Create another function to help with the graphics of the solution

> $fct := \left[u = U_{solve},\ h = H,\ \phi = \Phi\right]:$

> $Matlab(fct,\ optimize,\ output = \text{"utility.m"})$