Gravity Turn Maneuver

Jeff Anding
jganding@outlook.com

Introduction

This is a model of a single stage rocket gravity turn maneuver. It is integrated in two parts a) one from launch to point of start of the turn and b) the point from when gravity turn angle is set to a non-vertical (ie., <90 deg) value. Though I don't use Maple units in this model, I believe there is unit consistency throughout the model.

Some things to note about this model:

1. There is a slight discontinuity at the point where the two integration models connect. However, the final conditions in the first integration are used as the initial conditions in the 2nd integration.
2. There are probably much easier ways to manipulate the arrays (using built-in Maple functions perhaps) than I have used here. That said, what I have used gets the job done.

Initialization

```
> restart
> with(plots) : with(plottools) :
```

Section 1

> Initialize key parameters of the model. Use this section to try different values that drive the model. The initial conditions of the differential equations are defined and can be adjusted below in sections 3 and 4 of the worksheet.

$A := \left(\frac{\mathrm{Pi}\cdot 2.5^2}{4}\right);$ *# frontal area of vehicle with diameter of 2.5 m*

$C_D := 0.03;$ *# coefficient of drag*

$\rho_0 := 1.225;$ *# density of air at sea level and standard temperature*

$H := 8.4;$

 # the height scale of the exponential fall for air density calculations in km (converted to m in

the formula)

$T_0 := 350000;$ *# vehicle thrust in N*

$m_0 := 11300;$ *# initial mass of vehicle (propellant mass plus payload) in kg*

$m_p := 10000;$ *# initial mass of propellant in kg*

$m_f := m_0 - m_p;$ *# final mass of vehicle one propellant burns off. mass of payload in kg*

$r_{earth} := 6.378e6;$ *# radius of earth in meters*

$I_{sp} := 300;$ *# engine specific impulse*

$G := 6.6743e{-}11;$ *# gravitational constant*

$m_{earth} := 5.97219e24;$ *# mass of earth in kg*

$t_{g\_turn} := 15;$ *# time at initiation of gravity turn in seconds*

$$A := 4.908738522$$

$$C_D := 0.03$$

$$\rho_0 := 1.225$$

$$H := 8.4$$

$$T_0 := 350000$$

$$m_0 := 11300$$

$$m_p := 10000$$

$$m_f := 1300$$

$$r_{earth} := 6.378\ 10^{6}$$

$$I_{sp} := 300$$

$$G := 6.6743\ 10^{-11}$$

$$m_{earth} := 5.97219\ 10^{24}$$

$$t_{g\_turn} := 15 \tag{3.1}$$

> $\mu := G \cdot m_{earth}$

$$\mu := 3.986018772\ 10^{14} \tag{3.2}$$

> $g_0 := G \cdot \frac{m_{earth}}{r_{earth}^{\ 2}}$ *# gravitational pull of earth at it's surface*

$$g_0 := 9.798741706 \tag{3.3}$$

> *# Key values for information* ==

$t_f := \text{floor}\left(\frac{m_p \cdot g_0 \cdot I_{sp}}{T_0}\right);$ *# t final. Time at propellant burnout.*

$thrust\_to\_mass\_ratio := evalf\left(\frac{T_0}{m_0}\right);$

$propellant\_to\_mass\_ratio := evalf\left(\frac{m_p}{m_0}\right);$

$m_{burn\_rate} := evalf\left(\frac{m_p}{t_f}\right);$

$$t_f := 83$$

$$thrust\_to\_mass\_ratio := 30.97345133$$

$$propellant\_to\_mass\_ratio := 0.8849557522$$

$$m_{burn\_rate} := 120.4819277 \quad (3.4)$$

> # Run the sim for longer than the fuel lasts ==============================

$duration := (\text{floor}(2.5 \cdot t_f))$
 # run the sim 2.5 times as long as the fuel lasts - change this if desired

$$duration := 207 \quad (3.5)$$

Section 2

> # Change of gravitational acceleration with altitude =======================

$eq_g := g(t) = g_0 \cdot \left(\frac{r_{earth}}{r(t)}\right)^2 : eq_g$

$$g(t) = \frac{3.986018772\ 10^{14}}{r(t)^2} \quad (4.1)$$

> # Thrust over time ==

$eq_{thrust} := T = piecewise(t \le t_f, T_0, 0) : eq_{thrust}$

$$T = \begin{cases} 350000 & t \le 83 \\ 0 & otherwise \end{cases} \quad (4.2)$$

> # Vehicle mass over time ==

$eq_{mass} := m(t) = piecewise\left(t \le t_f, m_0 - \frac{T}{g(t) \cdot I_{sp}} \cdot t, t > t_f, m_f\right) : eq_{mass}$

$$m(t) = \begin{cases} 11300 - \frac{T\, t}{300\, g(t)} & t \le 83 \\ 1300 & 83 < t \end{cases} \quad (4.3)$$

> $eq_{mass} := subs(eq_{thrust}, eq_{mass}) : eq_{mass}$

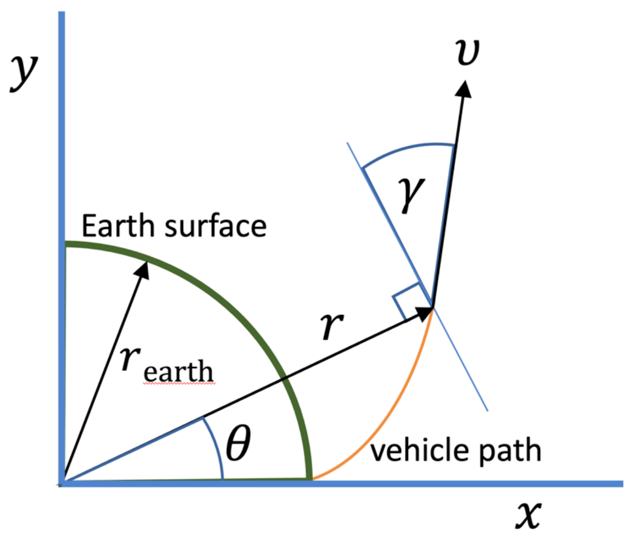
$$m(t) = \begin{cases} 11300 - \dfrac{\left(\begin{cases} 350000 & t \le 83 \\ 0 & \text{otherwise} \end{cases}\right) t}{300\, g(t)} & t \le 83 \\ 1300 & 83 < t \end{cases} \tag{4.4}$$

> # *Air density as a function of altitude at time t* ============================

$eq_{density} := \rho(t) = \rho_0 \cdot \exp\left(-\dfrac{r(t) - r_{earth}}{H \cdot 1000}\right) : eq_{density}$

$$\rho(t) = 1.225\, e^{-0.0001190476190\, r(t) + 759.2857140} \tag{4.5}$$

> # *Equations of motion for r (distance (radius) from center of earth), γ (angle of the flight path), υ (velocity), and θ (angle of declination)*

> 

> $deq_{radial} := diff(r(t), t) = \upsilon(t) \cdot \sin(\gamma(t)) : deq_{radial}$

$$\frac{d}{dt} r(t) = \upsilon(t) \sin(\gamma(t)) \tag{4.6}$$

> $deq_{declination} := diff(\theta(t), t) = \dfrac{\upsilon(t) \cdot \cos(\gamma(t))}{r(t)} : deq_{declination}$

$$\frac{d}{dt} \theta(t) = \frac{\upsilon(t) \cos(\gamma(t))}{r(t)} \tag{4.7}$$

> $deq_{velocity} := diff(\upsilon(t), t) = \dfrac{T}{m(t)} - \dfrac{1}{(2 \cdot m(t))} \cdot A \cdot C_D \cdot \rho(t) \cdot \upsilon(t)^2 - \dfrac{\mu}{r(t)^2} \cdot \sin(\gamma(t)) : deq_{velocity}$

$$\frac{d}{dt} \upsilon(t) = \frac{T}{m(t)} - \frac{0.07363107785\, \rho(t)\, \upsilon(t)^2}{m(t)} - \frac{3.986018772\, 10^{14} \sin(\gamma(t))}{r(t)^2} \tag{4.8}$$

> $deq_{velocity} := subs(eq_{density}, deq_{velocity}) : deq_{velocity}$

$$\frac{d}{dt} \upsilon(t) = \frac{T}{m(t)} - \frac{0.09019807037\, e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{m(t)} \tag{4.9}$$

$$- \frac{3.986018772\ 10^{14} \sin(\gamma(t))}{r(t)^2}$$

> $deq_{velocity} := subs(eq_{thrust}, deq_{velocity}) : deq_{velocity}$

$$\frac{d}{dt} \upsilon(t) = \frac{\begin{cases} 350000 & t \leq 83 \\ 0 & otherwise \end{cases}}{m(t)} - \frac{0.09019807037\ e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{m(t)} - \frac{3.986018772\ 10^{14} \sin(\gamma(t))}{r(t)^2} \quad \textbf{(4.10)}$$

> $deq_{velocity} := subs(eq_{mass}, deq_{velocity}) : deq_{velocity}$

$$\frac{d}{dt} \upsilon(t) = \frac{\begin{cases} 350000 & t \leq 83 \\ 0 & otherwise \end{cases}}{\begin{cases} 11300 - \frac{\left(\begin{cases} 350000 & t \leq 83 \\ 0 & otherwise \end{cases}\right) t}{300\, g(t)} & t \leq 83 \\ 1300 & 83 < t \end{cases}} - \frac{0.09019807037\ e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{\begin{cases} 11300 - \frac{\left(\begin{cases} 350000 & t \leq 83 \\ 0 & otherwise \end{cases}\right) t}{300\, g(t)} & t \leq 83 \\ 1300 & 83 < t \end{cases}} - \frac{3.986018772\ 10^{14} \sin(\gamma(t))}{r(t)^2} \quad \textbf{(4.11)}$$

> $deq_{velocity} := subs(eq_{g}, deq_{velocity}) : deq_{velocity}$

$$\frac{d}{dt} \upsilon(t) = \frac{\begin{cases} 350000 & t \leq 83 \\ 0 & otherwise \end{cases}}{\begin{cases} 11300 - 8.362563060\ 10^{-18} \left(\begin{cases} 350000 & t \leq 83 \\ 0 & otherwise \end{cases}\right) r(t)^2\, t & t \leq 83 \\ 1300 & 83 < t \end{cases}} \quad \textbf{(4.12)}$$

$$-\frac{0.09019807037\, e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{\begin{cases} 11300 - 8.362563060\, 10^{-18} \left(\begin{cases} 350000 & t \le 83 \\ 0 & \mathit{otherwise} \end{cases}\right) r(t)^2\, t & t \le 83 \\ 1300 & 83 < t \end{cases}}$$

$$-\frac{3.986018772\, 10^{14} \sin(\gamma(t))}{r(t)^2}$$

> $deq_{fltpath} := diff(\gamma(t), t) = -\frac{\mu}{\upsilon(t) \cdot r(t)^2} \cdot \cos(\gamma(t)) + \frac{\upsilon(t) \cdot \cos(\gamma(t))}{r(t)} : deq_{fltpath}$

$$\frac{d}{dt}\gamma(t) = -\frac{3.986018772\, 10^{14} \cos(\gamma(t))}{\upsilon(t)\, r(t)^2} + \frac{\upsilon(t) \cos(\gamma(t))}{r(t)} \tag{4.13}$$

> *# Now to see the final DE's together* ================================

deq_{radial}; $deq_{declination}$; $deq_{velocity}$; $deq_{fltpath}$;

$$\frac{d}{dt} r(t) = \upsilon(t) \sin(\gamma(t))$$

$$\frac{d}{dt}\theta(t) = \frac{\upsilon(t) \cos(\gamma(t))}{r(t)}$$

$$\frac{d}{dt}\upsilon(t) = \frac{\begin{cases} 350000 & t \le 83 \\ 0 & \mathit{otherwise} \end{cases}}{\begin{cases} 11300 - 8.362563060\, 10^{-18} \left(\begin{cases} 350000 & t \le 83 \\ 0 & \mathit{otherwise} \end{cases}\right) r(t)^2\, t & t \le 83 \\ 1300 & 83 < t \end{cases}}$$

$$-\frac{0.09019807037\, e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{\begin{cases} 11300 - 8.362563060\, 10^{-18} \left(\begin{cases} 350000 & t \le 83 \\ 0 & \mathit{otherwise} \end{cases}\right) r(t)^2\, t & t \le 83 \\ 1300 & 83 < t \end{cases}}$$

$$-\frac{3.986018772\, 10^{14} \sin(\gamma(t))}{r(t)^2}$$

$$\frac{d}{dt}\gamma(t) = -\frac{3.986018772\, 10^{14} \cos(\gamma(t))}{\upsilon(t)\, r(t)^2} + \frac{\upsilon(t) \cos(\gamma(t))}{r(t)} \tag{4.14}$$

Section 3

> # =================== Solution Part 1 ===========================

> $\theta_0 := 0;\ \gamma_0 := 90 \cdot \frac{\mathrm{Pi}}{180};$

put the initial conditions into variables so we can use them later elsewhere if needed

$$\theta_0 := 0$$

$$\gamma_0 := \frac{\pi}{2} \tag{5.1}$$

> $initcond1 := r(0) = r_{earth}, \theta(0) = \theta_0, \upsilon(0) = 1\mathrm{e}{-10}, \gamma(0) = \gamma_0;$

$$initcond1 := r(0) = 6.378\ 10^6, \theta(0) = 0, \upsilon(0) = 1.\ 10^{-10}, \gamma(0) = \frac{\pi}{2} \tag{5.2}$$

> $soln1 := dsolve(\{deq_{radial}, deq_{declination}, deq_{velocity}, deq_{fltpath}, initcond1\}, numeric, \{r(t), \theta(t), \upsilon(t), \gamma(t)\}, output = Array([seq(i, i = 1..t_{g\_turn})]))$

$$soln1 := \begin{bmatrix} [\ t \quad \gamma(t) \quad r(t) \quad \theta(t) \quad \upsilon(t)\ ] \\ \begin{bmatrix} \text{15 x 5 Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \end{bmatrix} \tag{5.3}$$

> $solTable1 := soln1[2][1]$

$$solTable1 := \begin{bmatrix} \text{15 x 5 Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{5.4}$$

> $upperbound(solTable1)$

$$15, 5 \tag{5.5}$$

> $ub1 := upperbound(solTable1)[1]$

$$ub1 := 15 \tag{5.6}$$

Section 4

> # =================== Solution Part 2 ==========================

> # Solving the DEs after the start of the gravity turn at t_{g_turn} we start at a new "zero" time but
need to adjust for a new t_f... Must redefine all equations that depent on t_f because t_{g_turn}
seconds have already been used up so t_f is now closer upon us.

>

> $t_{f2} := t_f - t_{g\_turn};$

$$t_{f2} := 68 \quad (6.1)$$

> # Thrust over time ==

$eq_{thrust} := T = piecewise(t \le t_{f2}, T_0, 0)$

$$eq_{thrust} := T = \begin{cases} 350000 & t \le 68 \\ 0 & otherwise \end{cases} \quad (6.2)$$

>

> # Vehicle mass over time ======================================

$eq_{mass} := m(t) = piecewise\left(t \le t_{f2}, m_0 - \frac{T}{g_0 \cdot I_{sp}} \cdot t, t > t_{f2}, m_f\right) : eq_{mass}$

$$m(t) = \begin{cases} 11300 - 0.0003401797327\, T\, t & t \le 68 \\ 1300 & 68 < t \end{cases} \quad (6.3)$$

> $eq_{mass} := subs(eq_{thrust}, eq_{mass}) : eq_{mass}$

$$m(t) = \begin{cases} 11300 - 0.0003401797327 \left(\begin{cases} 350000 & t \le 68 \\ 0 & otherwise \end{cases}\right) t & t \le 68 \\ 1300 & 68 < t \end{cases} \quad (6.4)$$

> $\mathbf{deq_{velocity} := diff(\upsilon(t), t) = \frac{T}{m(t)} - \frac{1}{(2 \cdot m(t))} \cdot A \cdot C_D \cdot \rho(t) \cdot \upsilon(t)^2 - \frac{\mu}{r(t)^2} \cdot \sin(\gamma(t)) : deq_{velocity}}$

$$\frac{d}{dt}\, \upsilon(t) = \frac{T}{m(t)} - \frac{0.07363107785\, \rho(t)\, \upsilon(t)^2}{m(t)} - \frac{3.986018772\ 10^{14} \sin(\gamma(t))}{r(t)^2} \quad (6.5)$$

> $deq_{velocity} := subs(eq_{density}, deq_{velocity}) : deq_{velocity}$

$$\frac{d}{dt}\, \upsilon(t) = \frac{T}{m(t)} - \frac{0.09019807037\, e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{m(t)} - \frac{3.986018772\ 10^{14} \sin(\gamma(t))}{r(t)^2} \quad (6.6)$$

> $deq_{velocity} := subs(eq_{thrust}, deq_{velocity}) : deq_{velocity}$

$$\frac{d}{dt}\, \upsilon(t) = \frac{\begin{cases} 350000 & t \le 68 \\ 0 & otherwise \end{cases}}{m(t)} - \frac{0.09019807037\, e^{-0.0001190476190\, r(t) + 759.2857140}\, \upsilon(t)^2}{m(t)} - \frac{3.986018772\ 10^{14} \sin(\gamma(t))}{r(t)^2} \quad (6.7)$$

> $deq_{velocity} := subs\left(eq_{mass}, deq_{velocity}\right) : deq_{velocity}$

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\upsilon(t) = \frac{\begin{cases} 350000 & t \le 68 \\ 0 & otherwise \end{cases}}{\begin{cases} 11300 - 0.0003401797327\left(\begin{cases} 350000 & t \le 68 \\ 0 & otherwise \end{cases}\right)t & t \le 68 \\ 1300 & 68 < t \end{cases}} - \frac{0.09019807037\,\mathrm{e}^{-0.0001190476190\,r(t) + 759.2857140}\,\upsilon(t)^2}{\begin{cases} 11300 - 0.0003401797327\left(\begin{cases} 350000 & t \le 68 \\ 0 & otherwise \end{cases}\right)t & t \le 68 \\ 1300 & 68 < t \end{cases}} - \frac{3.986018772\;10^{14}\,\sin(\gamma(t))}{r(t)^2} \tag{6.8}$$

>

> $\boldsymbol{\gamma_1} := 50\cdot\frac{\mathbf{Pi}}{\mathbf{180}}$ *# set the flight path angle to the pitch-over setting at* t_{g_turn}

$$\gamma_1 := \frac{5\,\pi}{18} \tag{6.9}$$

> $initcond2 := r(0) = solTable1\left[t_{g\_turn}, 3\right], \theta(0) = solTable1\left[t_{g\_turn}, 4\right], \upsilon(0) = solTable1\left[t_{g\_turn}, 5\right], \gamma(0) = \gamma_1;$

$$initcond2 := r(0) = 6.38056505238347\;10^{6}, \theta(0) = -1.40047292509927\;10^{-9}, \upsilon(0) = 354.287251054473, \gamma(0) = \frac{5\,\pi}{18} \tag{6.10}$$

> $soln2 := dsolve\left(\left\{deq_{radial}, deq_{declination}, deq_{velocity}, deq_{fltpath}, initcond2\right\}, numeric, \left\{r(t), \theta(t), \upsilon(t), \gamma(t)\right\}, output = Array\left(\left[seq\left(i, i = 1 ..duration - t_{g\_turn}\right)\right]\right)\right)$

$$soln2 := \left[\begin{array}{c} \left[\; t \quad \gamma(t) \quad r(t) \quad \theta(t) \quad \upsilon(t)\;\right] \\ \left[\begin{array}{c} \mathit{192\ x\ 5\ Matrix} \\ \mathit{Data\ Type{:}\ anything} \\ \mathit{Storage{:}\ rectangular} \\ \mathit{Order{:}\ Fortran\_order} \end{array}\right] \end{array}\right] \tag{6.11}$$

> $solTable2 := soln2[2][1]$

(6.12)

$$solTable2 := \begin{bmatrix} \text{192 x 5 Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{6.12}$$

> upperbound(solTable2)

$$192, 5 \tag{6.13}$$

> ub2 := upperbound(solTable2)[1]

$$ub2 := 192 \tag{6.14}$$

Section 5

> # ============== Aggregate the solution arrays into one array =============

> ub1; ub2; ub1 + ub2

$$15$$
$$192$$
$$207 \tag{7.1}$$

> solTable := Array(1 ..ub1 + ub2, 1 ..5)

$$solTable := \begin{bmatrix} \text{1..207 x 1..5 Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{7.2}$$

```
> for j from 1 to ub1 do
    solTable[j, 1] := j;
    solTable[j, 2] := solTable1[j, 2];
    solTable[j, 3] := solTable1[j, 3];
    solTable[j, 4] := solTable1[j, 4];
    solTable[j, 5] := solTable1[j, 5];
  end do:
  for j from ub1 + 1 to ub1 + ub2 do
    solTable[j, 1] := j;
    solTable[j, 2] := solTable2[j − ub1, 2];
    solTable[j, 3] := solTable2[j − ub1, 3];
    solTable[j, 4] := solTable2[j − ub1, 4];
    solTable[j, 5] := solTable2[j − ub1, 5];
   end do:
> solTable
```

$$\begin{bmatrix} \text{1..207 x 1..5 Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{7.3}$$

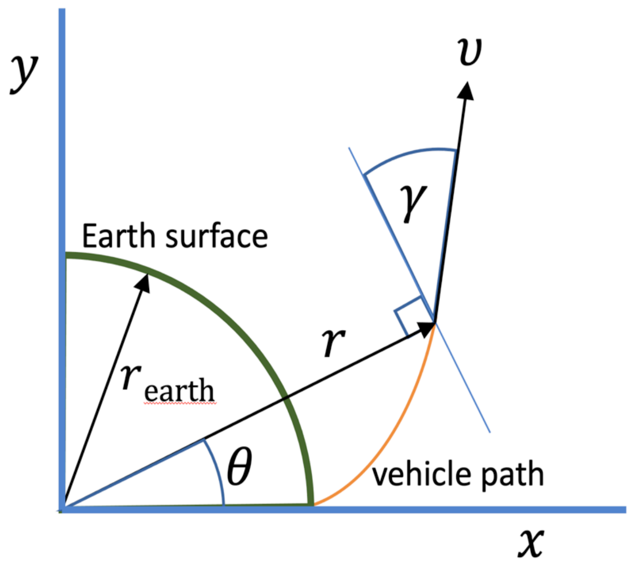
> $ub := upperbound(solTable)[1]$

$$ub := 207 \quad (7.4)$$

Section 6

Now use the solution to calculate profiles of various flight variables:

- solTable column 1 is t (time)
- solTable column 2 is γ (angle of flight path)
- solTable column 3 is r (radius)
- solTable column 4 is θ (angle of declination)
- solTable column 5 is υ (velocity)

> 

>

> *# Acceleration (as numerical derivative of velocity) ======================*

```
accel_t := Array(1..ub, 1..1) :
for j from 2 to ub do
   accel_t[j, 1] := solTable[j, 5] − solTable[j − 1, 5];
end do:
accel_t[1, 1] := accel_t[2, 1] :
```

> $accel_t$

1..207 x 1..1 Array
Data Type: anything
Storage: rectangular
Order: Fortran_order

(8.1)

>

> *# Altitude above sea level ===*

```
s_t := Array(1..ub, 1..1) :
```

```
for j from 1 to ub do
    s_t[j, 1] := solTable[j, 3] - r_earth;
end do:
> s_t
```

$$\begin{bmatrix} \text{1..207 x 1..1 Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad \textbf{(8.2)}$$

```
> # Vehicle mass vs time ==========================================
```

$mass_t := Array(1..ub, 1..1):$

for j **from** 1 **to** ub **do**

$$mass_t[j, 1] := piecewise\left(solTable[j, 1] \le t_f, m_0 - \frac{piecewise(solTable[j, 1] \le t_f, T_0, 0)}{g_0 \cdot I_{sp}} \cdot solTable[j, 1], solTable[j, 1] > t_f, m_f\right)$$

end do:

> $mass_t$

$$\begin{bmatrix} \text{1..207 x 1..1 Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad \textbf{(8.3)}$$

>

```
> # Dynamic Pressure ==========================================
```

$dp_t := Array(1..ub, 1..1):$

for j **from** 1 **to** ub **do**

$$dp_t[j, 1] := \frac{1}{(2 \cdot mass_t[j, 1])} \cdot A \cdot C_D \cdot \rho_0 \cdot \exp\left(-\frac{s_t[j, 1]}{H \cdot 1000}\right) \cdot solTable[j, 5]^2;$$

end do:

> dp_t

$$\begin{bmatrix} \text{1..207 x 1..1 Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \quad \textbf{(8.4)}$$

>

> # *Air density at vehicle as a function of time* ==============================

$dens_t := Array(1..ub, 1..1):$
for j **from** 1 **to** ub **do**
$$dens_t[j,1] := \rho_0 \cdot \exp\left(-\frac{s_t[j,1]}{H \cdot 1000}\right);$$
end do:

> $dens_t$

$$\left[\begin{array}{c} 1..207 \text{ x } 1..1 \text{ Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{array}\right] \quad (8.5)$$

>

> # *Air density at vehicle as a function of altitude* ===========================

$dens_{alt} := Array(1..ub, 1..2):$
for j **from** 1 **to** ub **do**
$dens_{alt}[j,1] := s_t[j,1];$
$dens_{alt}[j,2] := dens_t[j,1];$
end do:

> $dens_{alt}$

$$\left[\begin{array}{c} 1..207 \text{ x } 1..2 \text{ Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{array}\right] \quad (8.6)$$

>

> # *Convert polar coordinates to rectangular for plotting ascent profile* =========

$ascent := Array(1..ub, 1..2);$
$ground := Array(1..ub, 1..2);$
for j **from** 1 **to** ub **do**
$ascent[j,1] := solTable[j,3] \cdot \sin(solTable[j,4]);$
$ascent[j,2] := (solTable[j,3] \cdot \cos(solTable[j,4])) - r_{earth};$
$ground[j,1] := r_{earth} \cdot \sin(solTable[j,4]);$
$ground[j,2] := (r_{earth} \cdot \cos(solTable[j,4])) - r_{earth};$
end do:

$$ascent := \begin{bmatrix} \textit{1..207 x 1..2 Array} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{bmatrix}$$

$$ground := \begin{bmatrix} \textit{1..207 x 1..2 Array} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{bmatrix} \quad (8.7)$$

>

> # Thrust as a function of time ====================================

$thrust_t := Array(1..ub, 1..1);$

for j **from** 1 **to** ub **do**

$thrust_t[j, 1] := piecewise\left(solTable[j, 1] \leq t_f, T_0, 0\right);$

end do:

$$thrust_t := \begin{bmatrix} \textit{1..207 x 1..1 Array} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{bmatrix} \quad (8.8)$$

>

> # find max Q ==

$Q\_max := 0:$

for j **from** 1 **to** ub **do**

if $dp_t[j, 1] > Q\_max$ **then**

$Q\_max := dp_t[j, 1]:$

$best\_j := j:$

end if;

end do:

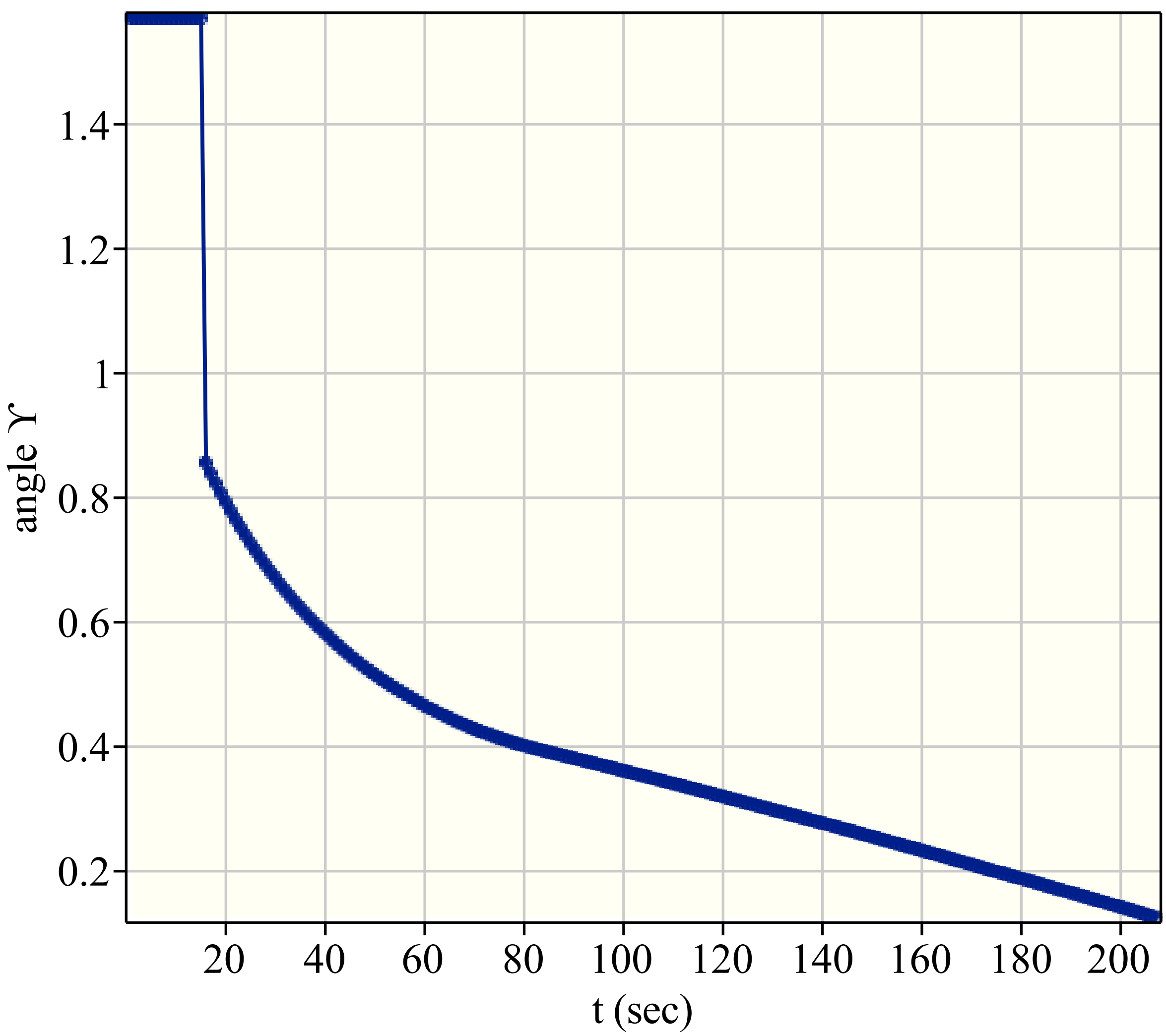
>

>

Section 7

> # Solution plots ==

> $Plot_{\gamma} := dataplot(solTable[1..ub, 2], labels = [\text{"t (sec)"}, \text{"angle Υ"}], labeldirections = [\text{"horizontal"}, \text{"vertical"}], size = [600, 400], symbolsize = 8, background = \text{"Ivory"}, filled = [color = \text{"Cyan"}, transparency = 0.9], axes = box, axis = [gridlines = [10, color = \text{"gray"}]])$



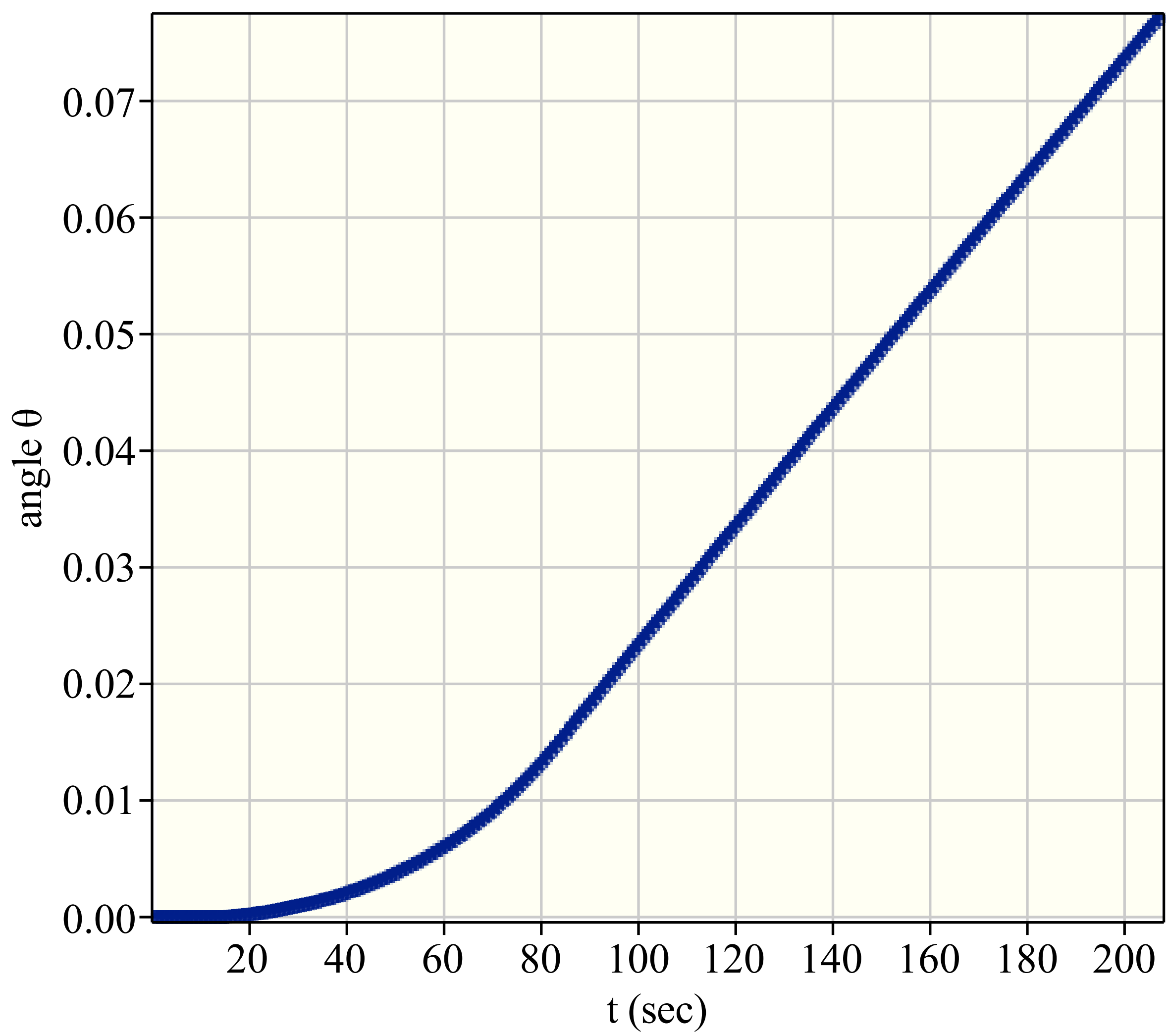
> $total\_change\_of\_\Upsilon\_in\_radians := solTable2[ub2, 2] - \gamma_1; total\_change\_of\_\Upsilon\_in\_degrees := \frac{total\_change\_of\_\Upsilon\_in\_radians \cdot 180}{\text{Pi}}$

$$total\_change\_of\_\Upsilon\_in\_radians := -0.746748106446743$$

$$total\_change\_of\_\Upsilon\_in\_degrees := -42.7855148531977 \quad \textbf{(9.1)}$$

>

> $Plot_{declination} := dataplot(solTable[1..ub, 4], labels = ["t (sec)", "angle θ"], labeldirections = ["horizontal", "vertical"], size = [600, 400], symbolsize = 8, background = "Ivory", filled = [color = "Cyan", transparency = 0.9], axes = box, axis = [gridlines = [10, color = "gray"]])$



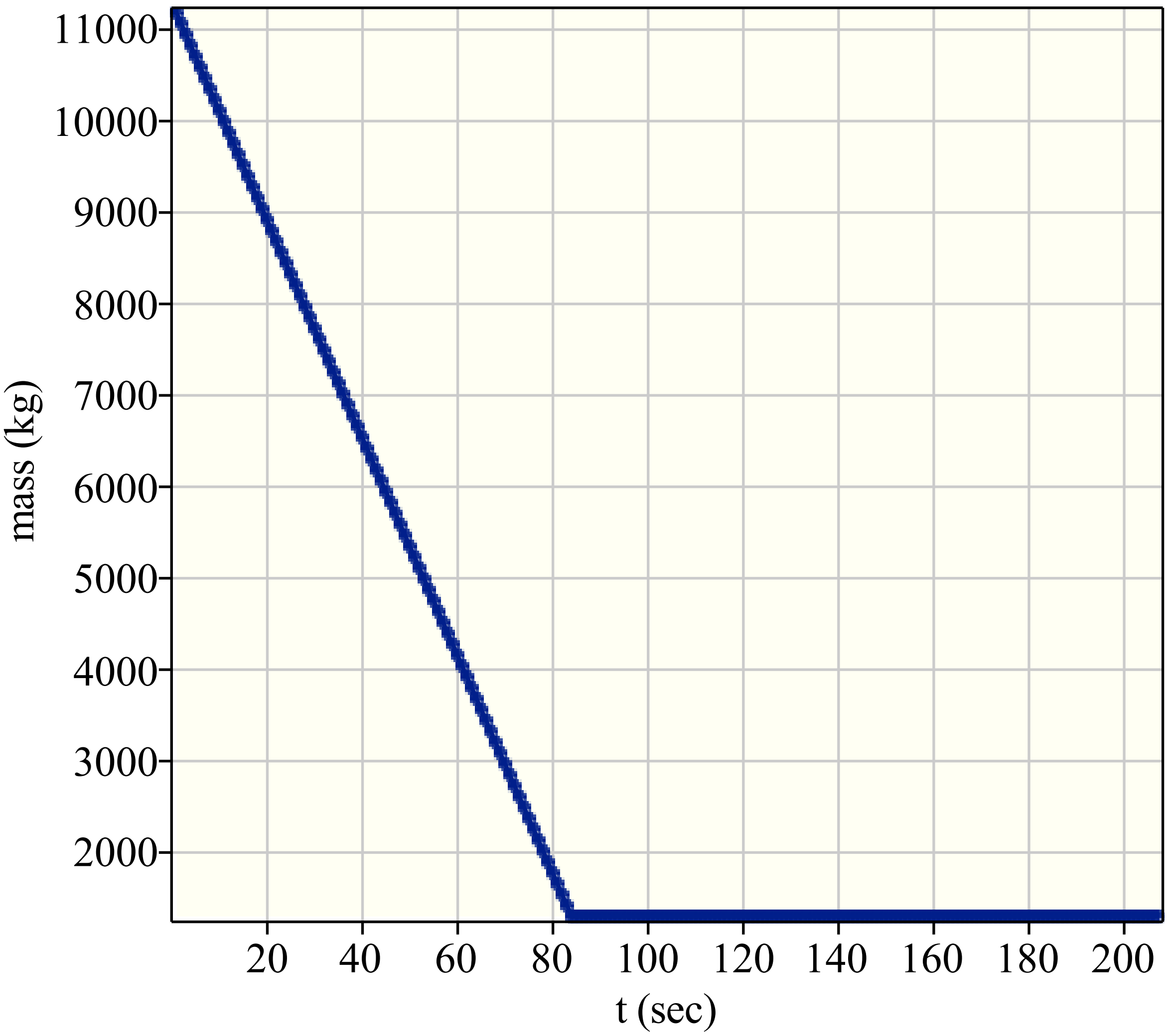
> $total\_change\_of\_\theta\_in\_radians := solTable[ub, 4] - \theta_0; total\_change\_of\_\theta\_in\_degrees := \frac{total\_change\_of\_\theta\_in\_radians \cdot 180}{Pi}$

$$total\_change\_of\_\theta\_in\_radians := 0.0770611205031900$$

$$total\_change\_of\_\theta\_in\_degrees := 4.41527696880533 \quad \textbf{(9.2)}$$

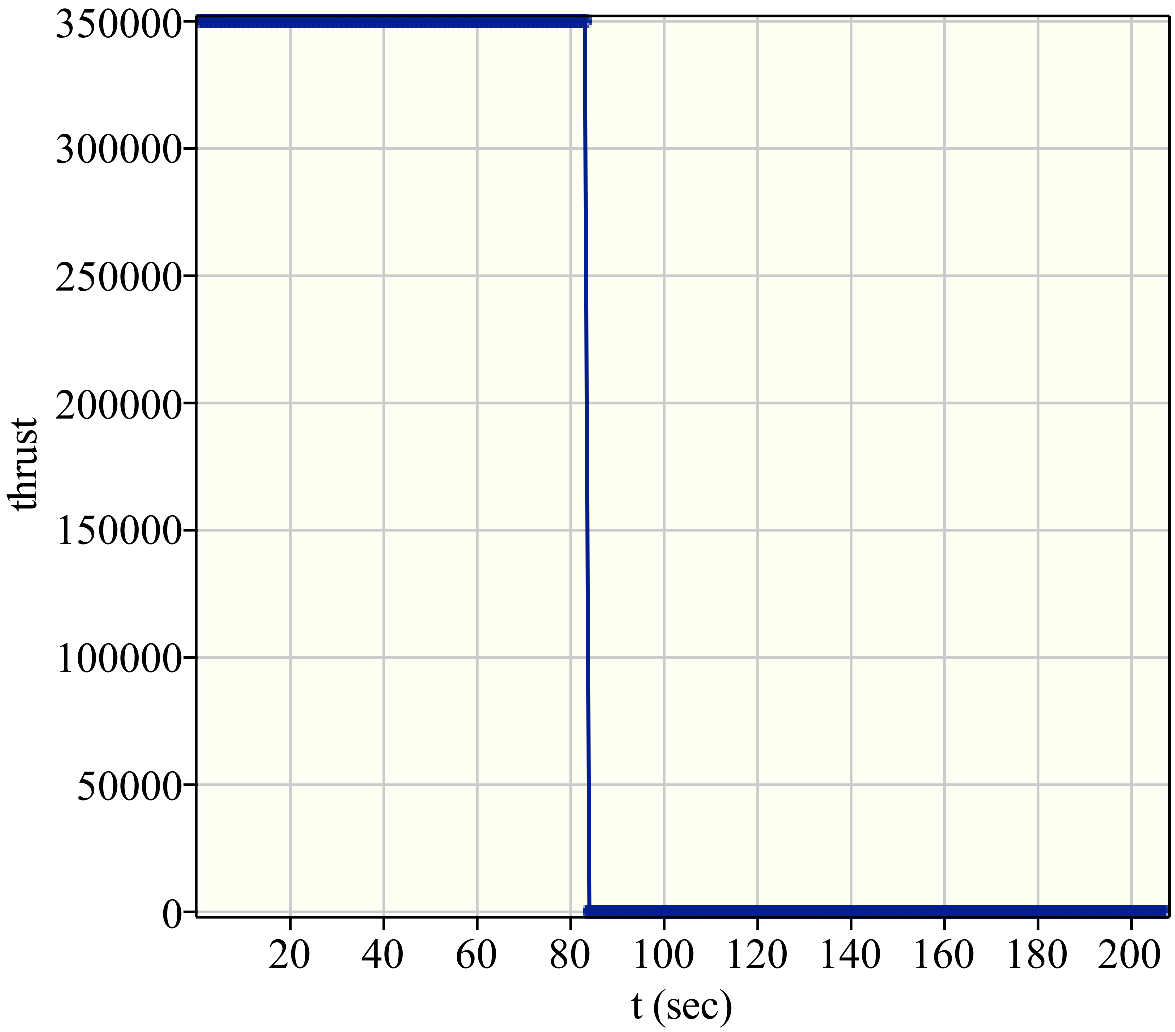
>

> $Plot_{mass} := dataplot\left(mass_t[1..ub, 1], labels = [\text{"t (sec)"}, \text{"mass (kg)"}], labeldirections = [\text{"horizontal"}, \text{"vertical"}], size = [600, 400], symbolsize = 8, background = \text{"Ivory"}, filled = [color = \text{"Cyan"}, transparency = 0.9], axes = box, axis = [gridlines = [10, color = \text{"gray"}]]\right)$



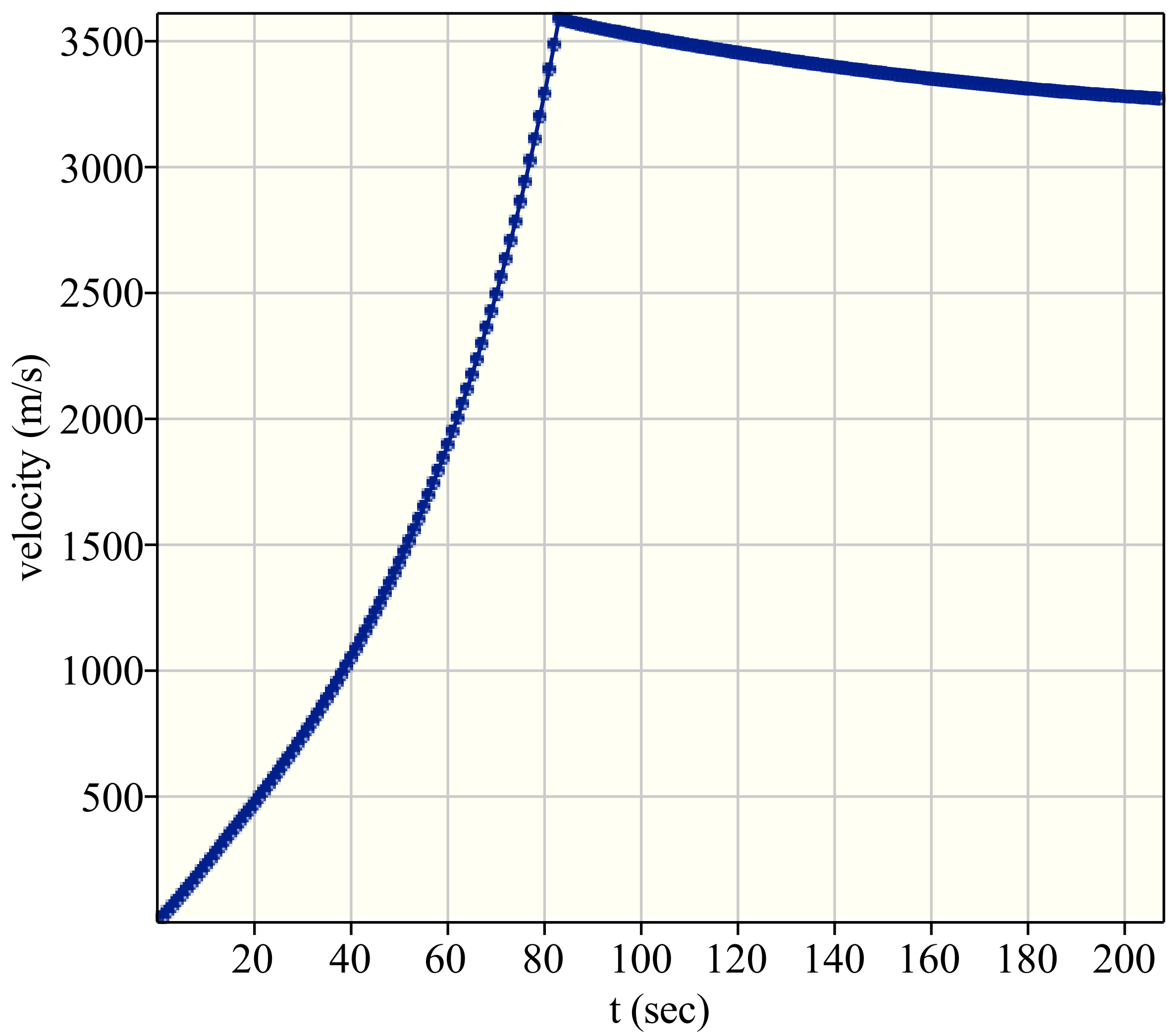
>

> $Plot_{thrust} := dataplot\left(thrust_t[1..ub, 1], labels = [\text{"t (sec)"}, \text{"thrust"}], labeldirections = [\text{"horizontal"}, \text{"vertical"}], size = [600, 400], symbolsize = 8, background = \text{"Ivory"}, filled = [color = \text{"Cyan"}, transparency = 0.9], axes = box, axis = [gridlines = [10, color = \text{"gray"}]]\right)$



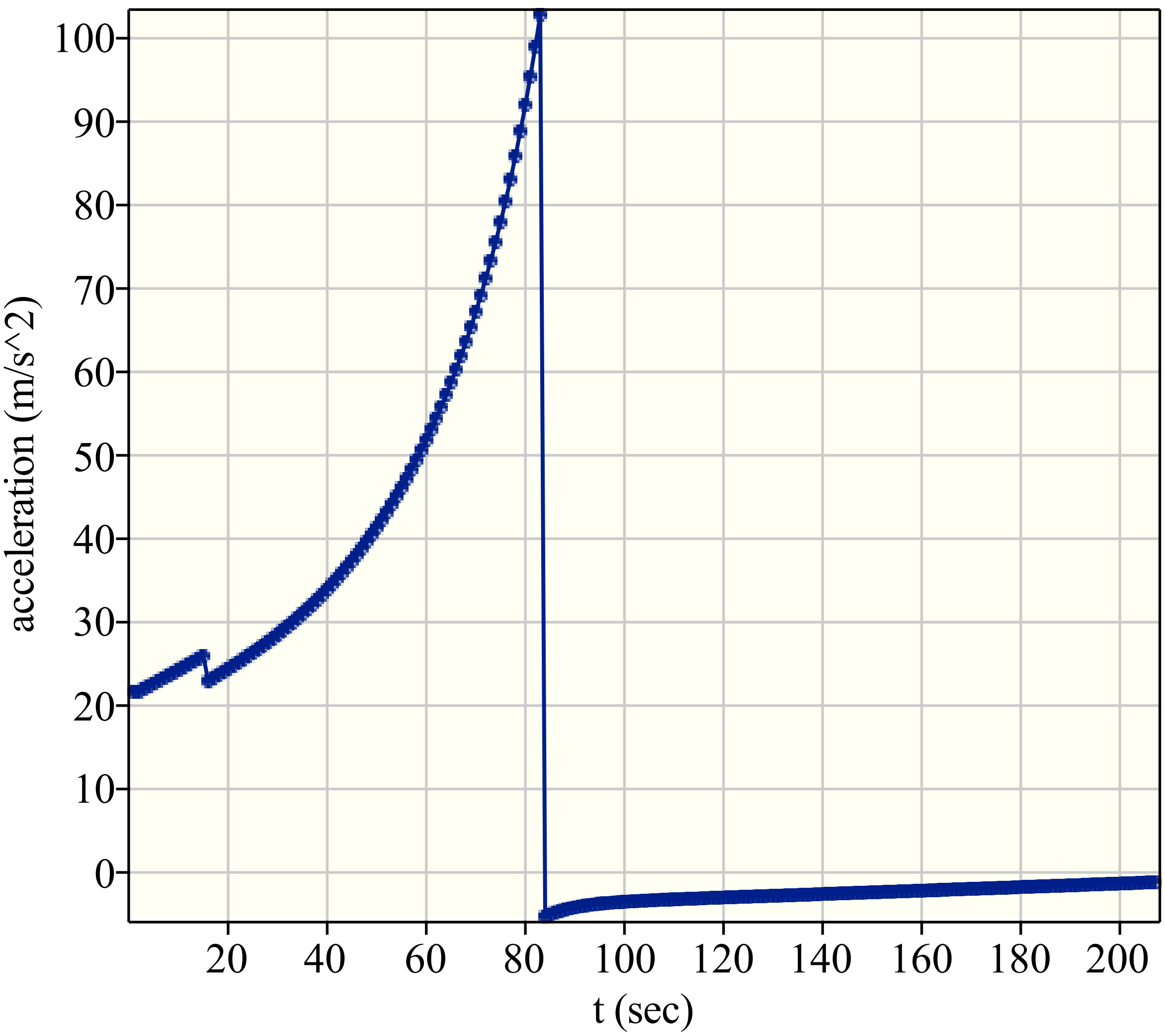
>

> $Plot_{vel} := dataplot(solTable[1..ub, 5], labels = ["t (sec)", "velocity (m/s)"], labeldirections = ["horizontal", "vertical"], size = [600, 400], symbolsize = 8, background = "Ivory", filled = [color = "Cyan", transparency = 0.9], axes = box, axis = [gridlines = [10, color = "gray"]])$



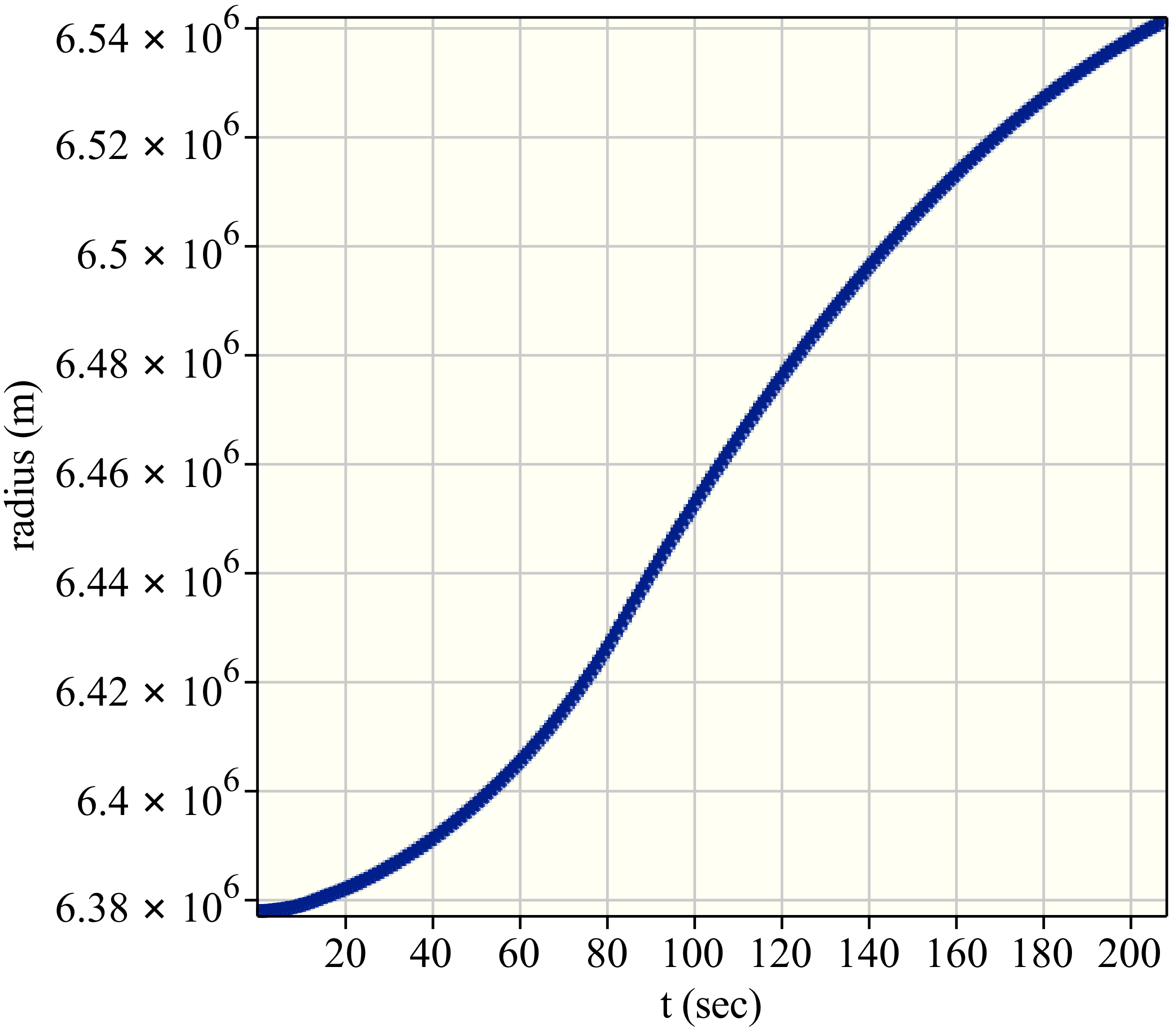
>

> $Plot_{accel} := dataplot\left(accel_t[1..ub, 1..1], labels = [\text{"t (sec)"}, \text{"acceleration (m/s^2)"}], labeldirections = [\text{"horizontal"}, \text{"vertical"}], size = [600, 400], symbolsize = 8, background = \text{"Ivory"}, filled = [color = \text{"Cyan"}, transparency = 0.9], axes = box, axis = [gridlines = [10, color = \text{"gray"}]]\right)$



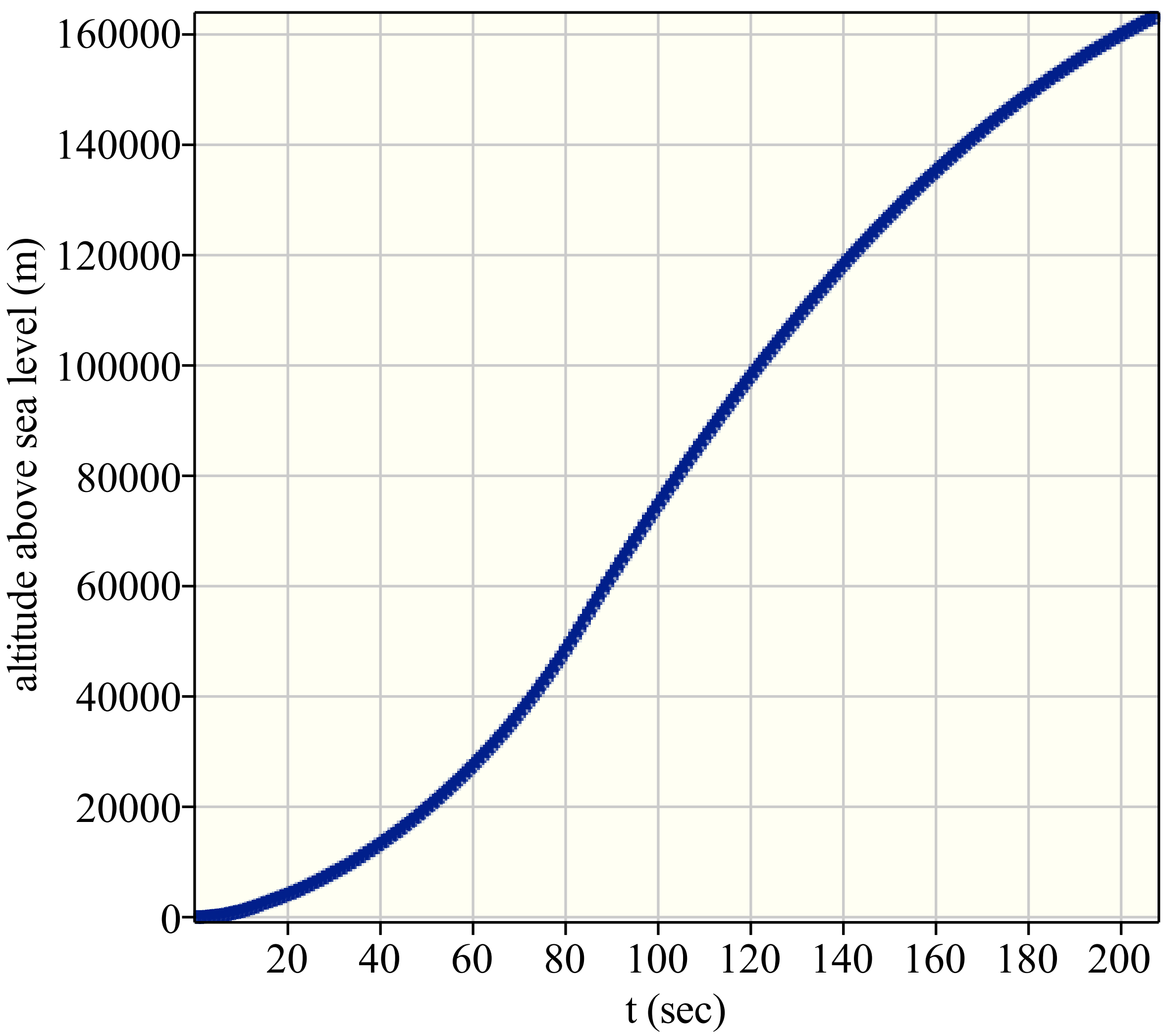
>

> $Plot_{radius} := dataplot(solTable[1..ub, 3], labels = ["t (sec)", "radius (m)"], labeldirections = ["horizontal", "vertical"], size = [600, 400], symbolsize = 8, background = "Ivory", filled = [color = "Cyan", transparency = 0.9], axes = box, axis = [gridlines = [10, color = "gray"]])$



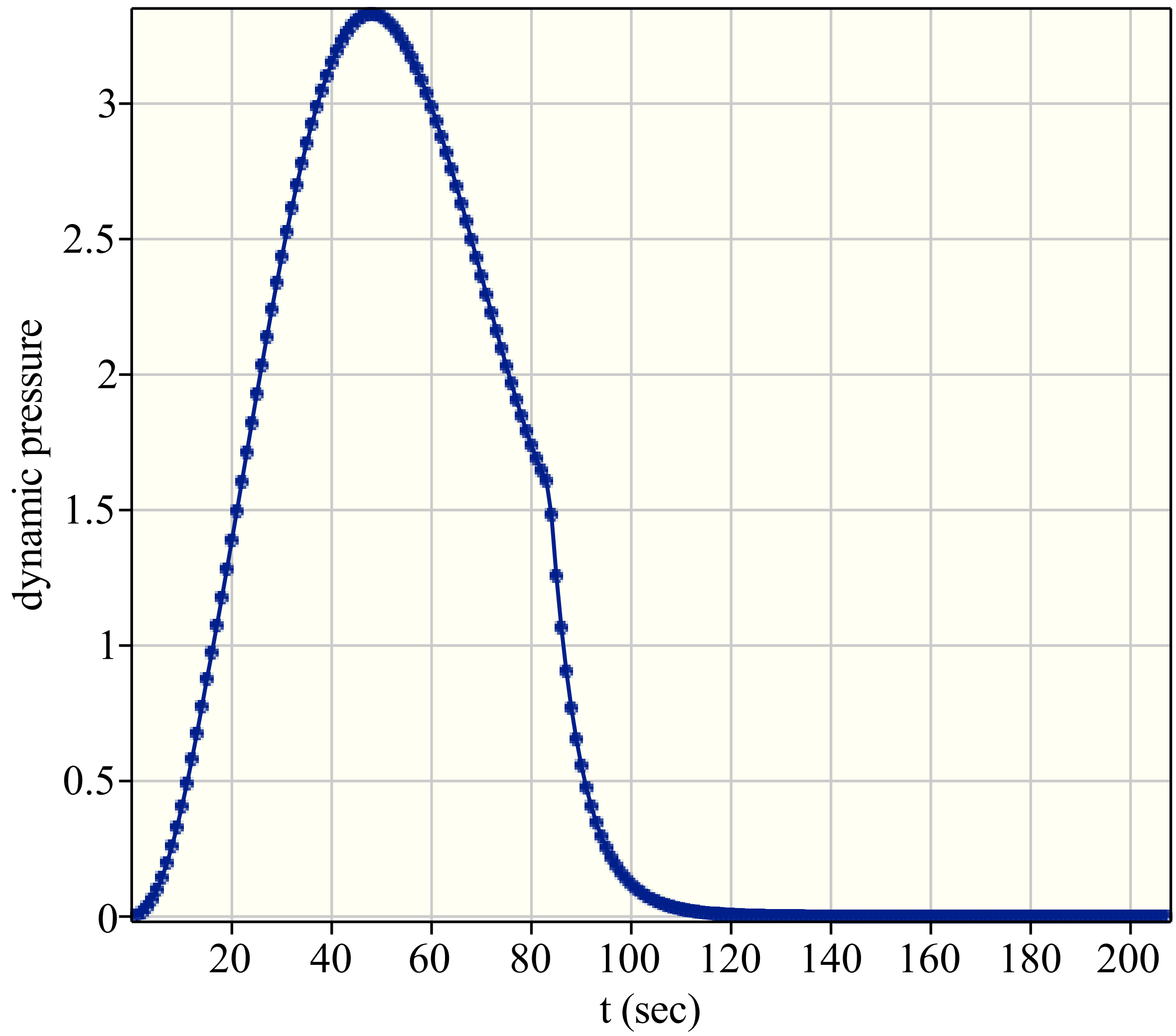
>

> $Plot_{alt} := dataplot\left(s_t[1..ub, 1..1], labels = ["t (sec)", "altitude above sea level (m)"], labeldirections = ["horizontal", "vertical"], size = [600, 400], symbolsize = 8, background = "Ivory", filled = [color = "Cyan", transparency = 0.9], axes = box, axis = [gridlines = [10, color = "gray"]]\right)$



>

> $Plot_{dyn\_press} := dataplot\left(dp_t[1..ub, 1..1], labels = ["t (sec)", "dynamic pressure"], labeldirections = ["horizontal", "vertical"], size = [600, 400], symbolsize = 8, background = "Ivory", filled = [color = "Cyan", transparency = 0.9], axes = box, axis = [gridlines = [10, color = "gray"]]\right)$



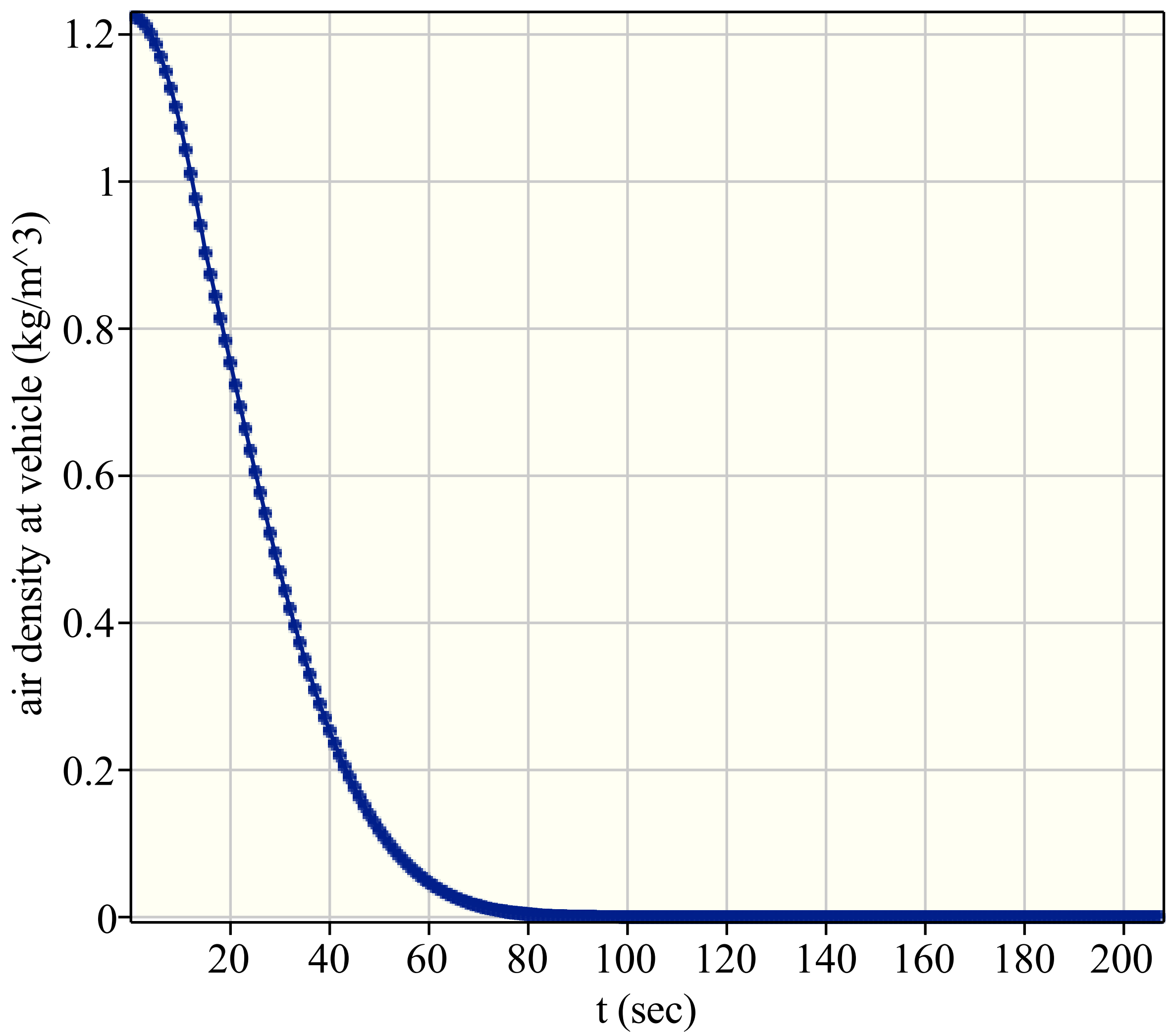
> $Q_{max} := Q\_max;\ Time\_at\_Q_{max} := best\_j;$

$$Q_{max} := 3.33147797217689$$

$$Time\_at\_Q_{max} := 48 \quad (9.3)$$

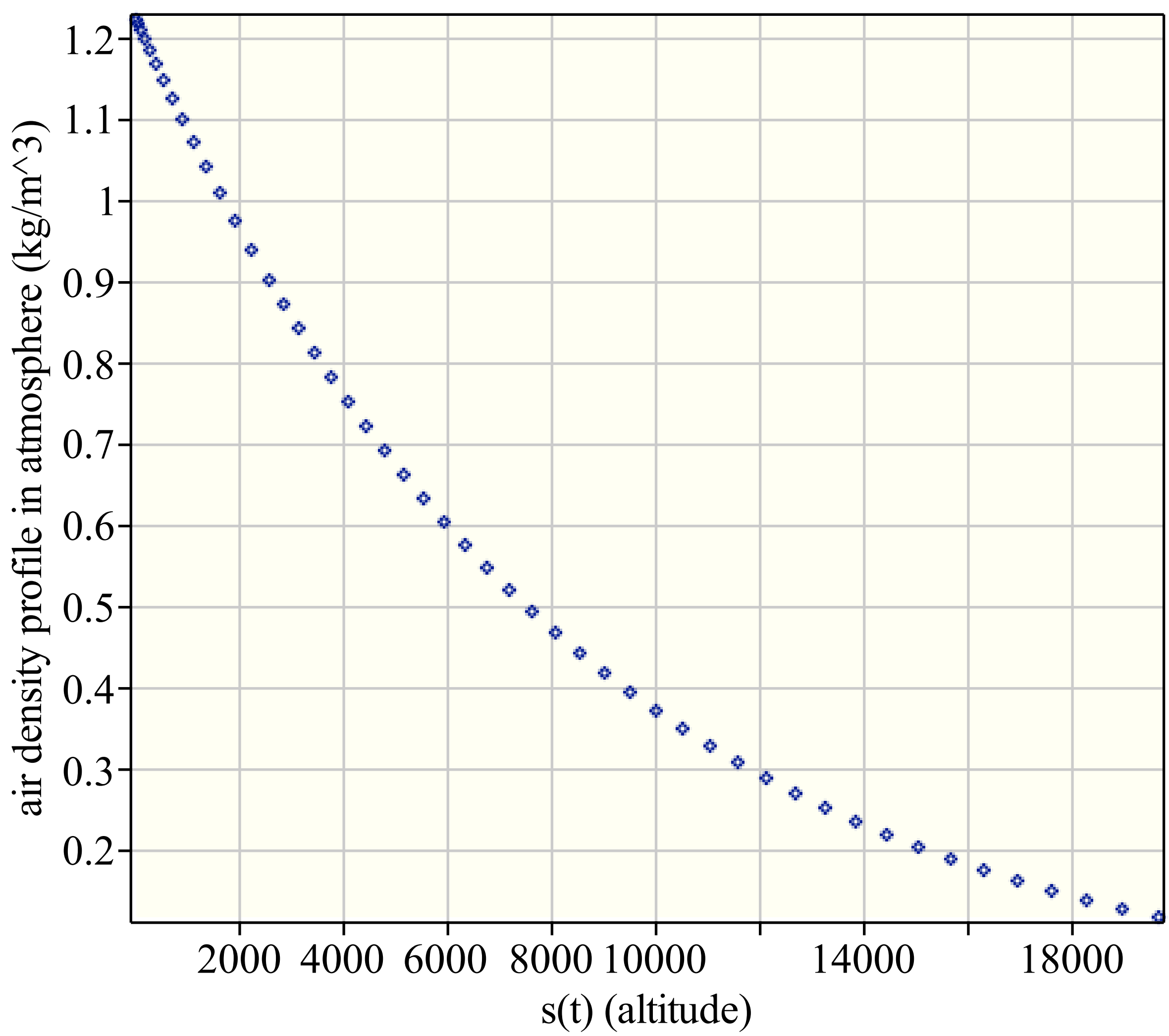
>

> $Plot_{air\_density\_v\_time} := dataplot\left(dens_t[1..ub, 1..1], labels = ["t (sec)", "air density at vehicle (kg/m^3)"], labeldirections = ["horizontal", "vertical"], size = [600, 400], symbolsize = 8, background = "Ivory", filled = [color = "Cyan", transparency = 0.9], axes = box, axis = [gridlines = [10, color = "gray"]]\right)$



>

> $Plot_{air\_density\_v\_alt} := pointplot\big(dens_{alt}[1..50, 1..2]$, *color* = "Navy" , *labels* = ["s(t) (altitude)", "air density profile in atmosphere (kg/m^3)"], *labeldirections* = ["horizontal", "vertical"], *size* = [600, 400], *symbolsize* = 8, *background* = "Ivory", *filled* = [*color* = "Cyan", *transparency* = 0.9], *axes* = *box*, *axis* = [*gridlines* = [10, *color* = "gray"]])

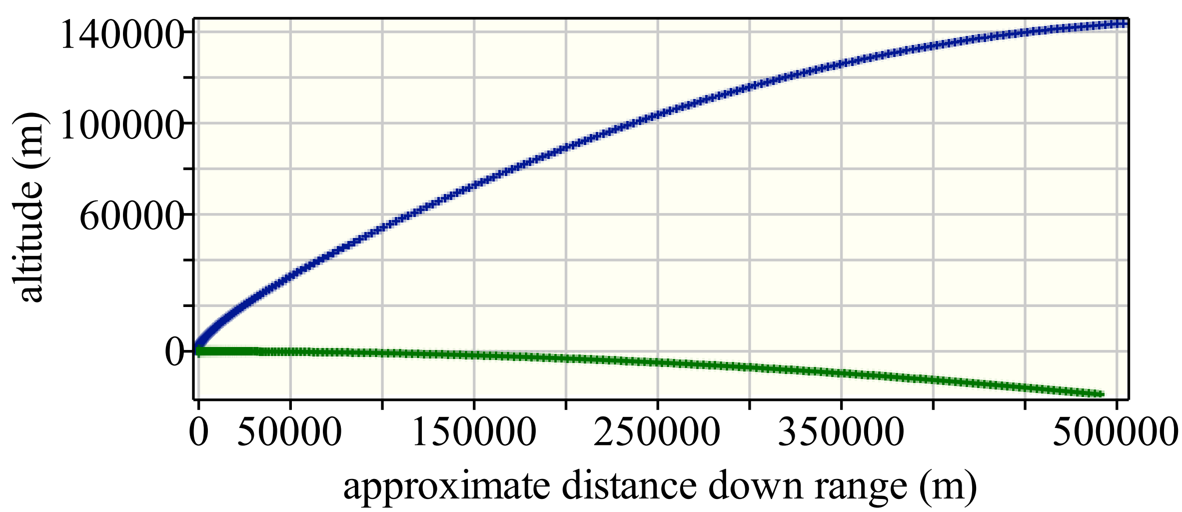


>

> $Plot_{ascent} := pointplot(ascent[1..ub, 1..2],\ color = \text{"Navy"},\ labels = [\text{"approximate distance down range (m)"}, \text{"altitude (m)"}],\ labeldirections = [\text{"horizontal"}, \text{"vertical"}],\ size = [950, 200],\ symbolsize = 8,\ background = \text{"Ivory"},\ filled = [color = \text{"Cyan"},\ transparency = 0.9],\ axes = box,\ axis = [gridlines = [10,\ color = \text{"gray"}]]):$

> $Plot_{ground} := pointplot(ground[1..ub, 1..2],\ color = \text{"DarkGreen"},\ labels = [\text{"approximate distance down range (m)"}, \text{"altitude (m)"}],\ labeldirections = [\text{"horizontal"}, \text{"vertical"}],\ size = [950, 200],\ style = pointline,\ symbolsize = 2,\ background = \text{"Ivory"},\ filled = [color = \text{"Cyan"},\ transparency = 0.9],\ axes = box,\ axis = [gridlines = [10,\ color = \text{"gray"}]]):$

> $display(Plot_{ascent}, Plot_{ground})$



> $grav\_turn\_initiated\_at := s_t[t_{g\_turn}, 1]\mathrm{m};\ grav\_turn\_initiated\_at := t_{g\_turn}\mathrm{s};$

$$grav\_turn\_initiated\_at := 2565.05238347314\ \mathrm{m}$$

$$grav\_turn\_initiated\_at := 15\ \mathrm{s} \qquad \textbf{(9.4)}$$

>

Conclusions

Modeling a rocket's gravity turn maneuver in Maple is a good exercise in demonstrating the capabilities of Maple while providing a good case to learn about equations of the gravity turn maneuver.

References

Credits and references:

1. Launch Dynamics and Gravity Turn Maneuvers by David Yaylali found at this link:
   https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiDvMTewLH_AhX0lWoFHQ6iCPMQFnoECA4QAQ&url=https%3A%2F%2Fwww.asthecroworbits.com%2FFiles%2FLaunchDynamics.pdf&usg=AOvVaw1mWq34xdq8-yzsMhAtEtO0
   Also, the diagrams in this model depicting how key variables relate to the model are heavily inspired by (though not copied directly from) the diagram in the paper by Mr. Yaylali.
2. the information at:
   https://mintoc.de/index.php/Gravity_Turn_Maneuver
   I attribute credit to the author(s) of this material found at "mintOC.de"

The information at these references belongs to the authors and not me - I used the information to help build the math models above but all the credit in the original sources of information goes to them, not me.

See Also

For more information on creating effective Maple Documents, see this Tips & Techniques.