

Solutions Lab 2

1a) Test Case: Using $V_0 = 65$ m/s, $\dot{m}_{fuel} = 0$ kg/s and $\rho = 0$ kg/m³.

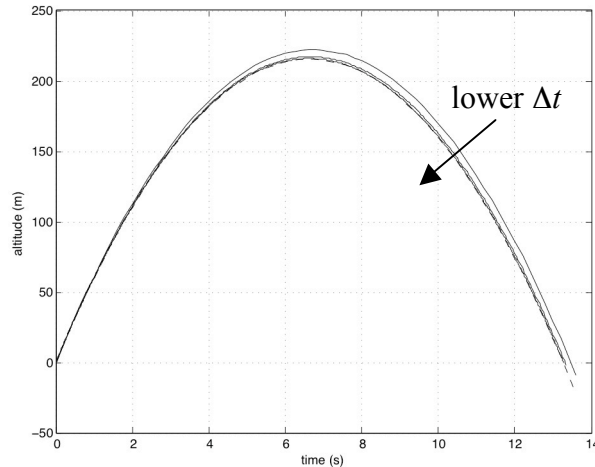
Under these conditions, the equation of motion simplifies to:

$$m \frac{dV}{dt} = -g,$$

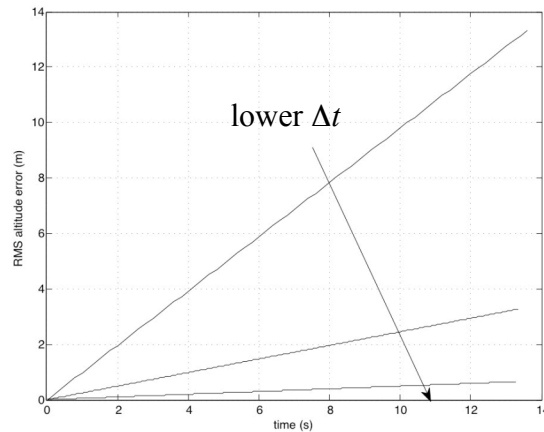
which can be easily solved after integrating twice for V and h with $t_0 = 0$ s:

$$V = V_0 - gt \quad \text{and} \quad h = h_0 + V_0 t - \frac{1}{2}gt^2$$

1b) The following plot shows results for $\Delta t = 0.2$ s, 0.05s and 0.01s, the exact solution is also plotted.



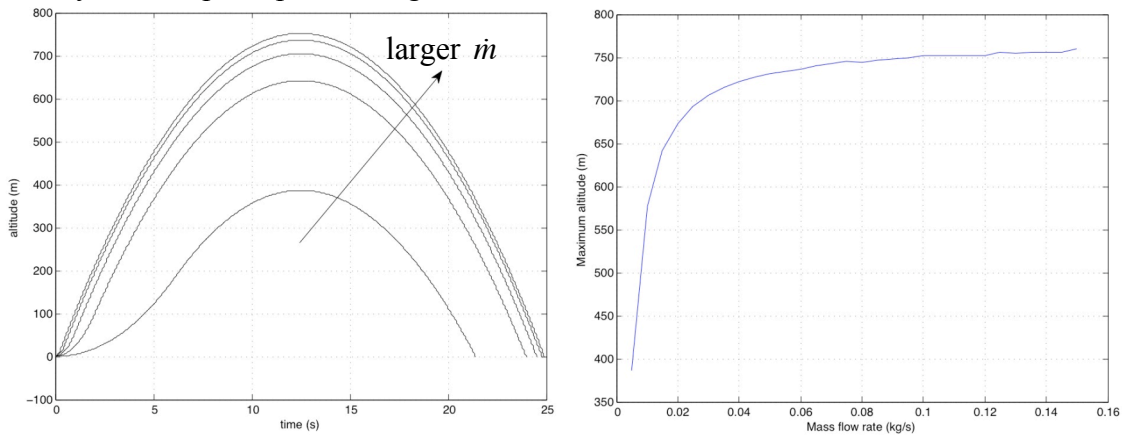
The dashed line shows the exact solution. Solid lines approach the dashed as Δt becomes smaller. It is clear that small Δt improves accuracy, at additional computational cost. A better way to illustrate the improvement would be taking the RMS error $\sqrt{(h - h_{exact})^2}$, which is shown in the plot below.



It can be clearly seen that the error propagates, or increases, with time. But is also clear that such errors can be reduced by using small time steps. Another way to increase accuracy is to make use of a 2nd order approximation for a derivative, which is out of the scope of this problem set.

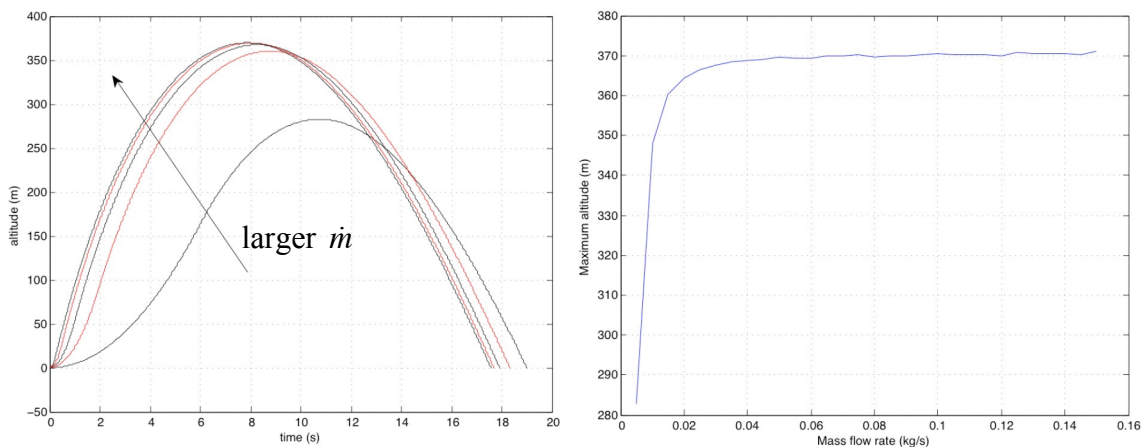
1c) The algorithm is consistent, as the error tends to zero with smaller time steps, although care should be taken when calculating for long times, as error propagation may decrease the accuracy, as exemplified above.

2a) The burn times for the cases shown vary from 6 to 25 seconds ($t_{burn} = m_{fuel} / \dot{m}_{fuel}$). If we fix our time step, we should select it according to the smallest time scales. If we select $\Delta t = 0.01$ sec, we will obtain enough points for very good accuracy. In this problem, such selection appears trivial, as the code will run quite fast in practically any computer language, but as the nature of the problem becomes more complicated, the criteria for selecting time steps becomes more involved and could be the difference between minutes and days of computer power usage.



It can be seen that as the flow rate increases, the altitude curves become more and more similar. To the right we can see this more clearly; we have a plot of the maximum altitude as a function of mass flow rate. A higher burn rate means more thrust, and therefore a higher rate of momentum increase, thus higher altitude. In the limit of very small thrust (just enough to compensate for gravity) no momentum is imparted to the rocket, this is the *hovering* mode.

2b) Lower burn rates are required in this case since drag is a function of velocity squared. Too much initial force would give a higher velocity and higher drag. As can be seen below, the altitude becomes insensitive to \dot{m} sooner than in 2a).



Loss mechanisms if fuel is burned... i) too slowly, gravity losses dominate, while ii) too quickly, drag losses become the most important.

Example of Matlab code:

```
% initial conditions
v(1) = 0;
h(1) = 0;
m(1) = .15;
t(1) = 0;

% Fuel mass and flow rate
mf = 0.03;
mdot = .12;

% operating conditions
rho = 1.22;
Cd = 0.35;
g = 9.8;
ue = 550;
A=0.001;

% time step
dt = 0.01;

% initial sign of the velocity vector
vect = 1;

j = 1;
while h(j)>=0,
    j = j+1;
    v(j) = v(j-1) + dt*(vect*mdot*ue/m(j-1) - 0.5*Cd*A*rho*v(j-1)*abs(v(j-1))/m(j-1) -
g);
    h(j) = h(j-1) + v(j-1)*dt;
    if h(j)<h(j-1)
        vect = -1;
    end
    m(j) = m(j-1) - mdot*dt;
    t(j) = t(j-1)+dt;
    if m(j) <= (m(1)-mf)
        mdot = 0;
    end
end
end
```