

# Martian Lander

**Due Time:** 23.59, Sun 02 December 2018

**Earnings:** 9% of your final grade

**NOTE:** Plan to finish a few days early to avoid last minute hardware/software or other unexpected holdups, for which no allowance is given.

**NOTE:** The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.

**Purpose:** Solve Ordinary differential Equations (ODEs) in real-time for a planetary lander flight simulator.

## Algorithm

While the user wishes to continue, the application will ask the user to select whether to run the simulation one more time. The simulation uses Heun's adaptation of Euler's method, as explained in lecture notes, to solve the following differential equations that describe the descent of a landing module on Mars under gravity with atmospheric drag and an additional booster term B:

$$\begin{aligned} dv/dt &= g - c*(v + a*(v/v_{max})^3) - B \\ v &= dx/dt \end{aligned}$$

The values of the constants are  $g = 3.7$ ,  $c = 0.13$ ,  $a = 8.3$ ,  $v_{max} = 46$

Solving these equations means calculating the velocity  $v$  and position  $x$  of the module as time  $t$  (seconds) passes after it starts its descent from an initial location 1000 meters above the Martian surface at the instant of being released from the mother ship. It then falls to the surface under the above equations.

The meanings of the terms are:

1.  $dv/dt$  is the acceleration of the lander module as it falls to the surface where  $v$  (in metres/second) is positive in a downward direction. At the instant of release  $v$  is zero and  $x$  is 1000 metres and time  $t$  (in seconds) is zero. Time is measured from the computer clock.
2.  $g$  is the downward acceleration caused by gravity (which on Mars is only 38% of Earth's gravity).
3.  $c*(v + a*(v/v_{max})^3)$  is the deceleration caused by atmospheric drag. It is small when the velocity is small and gets large as a power law as the velocity increases. At some high velocity (called the terminal velocity) it balances gravity and acceleration becomes zero..
4.  $B$  is the deceleration caused by a burn from the braking rocket on the landing module. This always acts in an upward direction to counter gravity. This rocket can be turned on and off and adjusted from the keyboard using the keys W (increases  $B$ ) and E (decreases  $B$ ). However, braking uses up fuel at the rate of  $B$  units per second and so depletes the fixed reserve of 100 units.

When the simulation starts, the lander is 1000 meters above the Martian surface at the instant of being released from the mother ship. It then falls to the surface under the above equations. The purpose of the simulation is to get a soft landing which is taken as a speed of  $< 1$  metre/sec at a distance above the surface of  $< 1$  metre. If the module hits the surface faster than this it gets bounced back up again ( $v$  becomes  $-v$ ) until gravity pulls it down again. The best landing is a straight descent without a single bounce. The number of bounces is recorded. Try to land without a single bounce.

There is a "bare-bones" console executable *lander.exe* in a zip file on BS that runs the simulation as a single shot – it does not offer a repeat and does not save the results to file. Your version should work the same and display the same information in the same way. In addition add the following:

1. offer the user the option of repeating the simulation
2. save the results of all simulations to file and display them on-screen at the end in order of best-to-worst landings together with the name of the user. The best is shortest time with fewest bounces.

Set up a Win32 console project in Visual Studio 2015 with the name *ass3*. In a file named *ass3.cpp* using the C (or/C++) programming language write the code to implement the application as described above. Remember to implement the Heun improvement – not just the Euler method.

See the Marking Sheet for how you can lose marks, but you definitely lose 60% or more if:

- your application won't build in Visual Studio 2015
- your application crashes in normal operation
- I can't build it because you submitted the wrong files or the files are missing, even if it's an honest mistake – 100% deduction.

**What to Submit :** Use Brightspace to submit this assignment as a zip file (**not** RAR, not 9zip, not 7 zip) containing only the source code (C or C++) file ass3.cpp. The name of the zipped folder **must** contain your name as a prefix so that I can identify it, for example using my name the file would be tyleraAss3CST8233.zip. It is also vital that you include the Cover Information (as specified in the Submission Standard) as a file header (you don't need to include an additional separate file header) in your source file so the file can be identified as yours. Use comment lines in the file to include the information.

Don't send me the file as an email attachment – it will get 0.

**NOTE: due to exams, this assignment cannot be late.**