SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Laurea in Fisica
Laboratorio di Calcolo - **Esercitazione n. 3**

Un *rover* per l'esplorazione di Marte è fatto cadere liberamente da una quota h, partendo da fermo. All'istante $t_0 = 0$ la posizione del *rover* è $x_0 = h$ e la sua velocità $v_0 = 0$. Trascorso un tempo Δt il dispositivo si troverà a una quota $x(t) = h - \frac{1}{2}gt^2$ e avrà acquistato una velocità pari a $v(t) = gt$, dove $g = 3.711$ ms^{-1} è l'accelerazione di gravità su Marte.

Il tempo di volo complessivo si prevede essere

$$t_M = \sqrt{\frac{2h}{g}}\,.$$

Il rover è progettato per resistere a un impatto che avviene a una velocità pari al massimo a 150 km/h.

Scrivi, in una directory di nome `roverFalling`, un programma che calcoli la quota e le velocità raggiunta dal rover a diversi istanti di tempo t compresi tra il momento del lancio e quello in cui tocca il suolo marziano a intervalli di Δt. Il programma deve fare i passi seguenti.

(1) Chiede all'utente di inserire la quota di lancio h in m. La quota dev'essere positiva. Qualora non lo fosse il programma deve segnalare l'errore all'utente e permettergli di reinserirla.
(2) Chiede all'utente di inserire il tempo Δt in s. Anche questo numero dev'essere positivo e l'utente deve poterlo reinserire qualora non lo fosse. Il valore di Δt è richiesto anche essere piccolo in confronto al tempo di volo t_M. Qualora cosí non fosse l'utente deve poter reinserire il dato.
(3) Assegna la quota e la velocità del *rover* a due variabili che, attraverso un ciclo iterativo, cambiano secondo la regola data sopra. L'iterazione termina quando il rover tocca il suolo. La condizione di arresto del ciclo non può essere determinata dal tempo di volo, perché in una successiva versione del programma tale tempo potrebbe fluttuare a causa della presenza di perturbazioni del moto del rover. La condizione di uscita dal ciclo deve dunque essere determinata dalla quota raggiunta.
(4) A ogni passo dell'iterazione stampa i valori del tempo trascorso dallo sgancio, della quota e della velocità separati da uno spazio.
(5) Al termine della simulazione il programma informa l'utente sulla sorte del rover. Se questo è sopravvissuto all'impatto si congratula con l'utente, altrimenti ne decreta il licenziamento.

Modifica il programma in maniera da ridurre il tempo Δt man mano che la quota raggiunta dal *rover* diminuisce. Quando la quota h si è ridotta di almeno un fattore 2 rispetto a quella che aveva all'inizio della simulazione diminuisci il tempo Δt del 50 %. Successivamente, ogni volta che la quota diminuisce di un fattore 2 rispetto a quella che il veicolo aveva all'istante della riduzione, esegui la stessa operazione. Esegui la simulazione con diversi valori di Δt e osserva quel che accade al diminuire di Δt.

Modifica il programma, cambiando la forma della struttura d'iterazione (se hai scelto un `while` usa un `for` o viceversa, scegliendo opportunamente la variabile di controllo). Quale tra le due forme ti sembra migliore?

Una volta che il programma sembra funzionare eseguilo con il comando

```
./nome_del_programma > rover.dat
```

dove `nome_del_programma` è il nome che hai scelto per il tuo programma. In questo modo quello che di norma viene presentato sullo schermo viene scritto sul file `rover.dat`. Nota che in questo modo non potrai vedere eventuali istruzioni fornite dal programma all'utente e il terminale resterà in attesa dei tuoi input. Poi usa il comando `gnuplot` e, al prompt di `gnuplot`, dai il comando `plot 'rover.dat' using 1:2`. Quindi, sempre in `gnuplot`, esegui il comando `plot 'rover.dat' using 1:3` (`quit` è il comando per uscire da `gnuplot`). Osserva il risultato e cerca di capire come funziona `gnuplot`.