

In questa esercitazione vogliamo migliorare l'accuratezza della simulazione della discesa su Marte di un *rover*. Fai dunque una copia del programma che hai realizzato per l'esercitazione 3 da modificare in questa occasione. Metti la copia in una *directory* di nome `windRover` e assegna il nome `<cognome_1>_<cognome_2>.c` al *file* con il codice sorgente, dove, evidentemente `<cognome_i>` rappresenta il cognome di uno degli autori.

Nella versione odierna il rover parte sempre da fermo da una quota fissata pari a  $h = 200$  m. Il passo di simulazione ha una durata costante  $dt = 5$  ms. Ricordiamo che la quota  $y(t)$  raggiunta al tempo  $t > 0$  e la corrispondente velocità  $v(t)$  sono

$$y(t) = h - \frac{1}{2}gt^2, \quad v(t) = gt,$$

dove  $g = 3.711 \text{ ms}^{-1}$  è l'accelerazione di gravità su Marte. Il programma deve fare i passi seguenti.

- (1) Partendo da  $t = 0$ , calcola la posizione e la velocità del rover a tutti gli istanti di tempo successivi, intervallati di  $dt$  costante, memorizzando i valori delle variabili che rappresentano queste grandezze in altrettanti *array* di taglia opportuna. Il calcolo degli elementi degli *array* si arresta quando il rover tocca il suolo.
- (2) Una volta che il rover abbia toccato il suolo il programma stampa il messaggio `# rover atterrato. Velocita' all'impatto:` , seguito dall'indicazione della velocità raggiunta al momento dell'impatto espressa in km/h.
- (3) La presenza di vento nell'atmosfera di Marte è simulata come segue: ogni volta che il programma esegue un passo di simulazione, una raffica di vento sposta il rover a destra o a sinistra, in modo casuale, di una quantità  $\delta$ . Il valore di  $\delta$  dev'essere inserito a cura dell'utente del programma e dev'essere  $\delta \leq 1$  m.
- (4) Il programma, per ciascuna delle posizioni  $y(t)$  calcolate in precedenza, determina il valore della coordinata orizzontale  $x(t)$  del rover durante la caduta, prendendo come origine la posizione di partenza. Si osservi che la posizione raggiunta al tempo  $t$  dipende da quella raggiunta al tempo  $t - 1$ .
- (5) Terminata la simulazione il programma stampa sullo schermo i valori di  $t$ ,  $x(t)$ ,  $y(t)$  e  $v(t)$ , partendo dall'ultimo punto simulato e andando indietro nel tempo.

Una volta che il programma sembra funzionare esegui con il comando

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

dove `nome_del_programma` è il nome che hai scelto per il tuo eseguibile. In questo modo quello che di norma viene presentato sullo schermo è 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 che, grazie ai cancellati (`#`) inseriti all'inizio di ogni stringa, non hai bisogno di modificare il *file* `rover.dat`. `gnuplot` ignora le righe che iniziano con questo segno.

Usa il comando `plot 'rover.dat' using 2:3 with lines`. Cosa stai osservando? A quali coordinate atterra il rover?

- (6) Modifica il programma inserendo una variante. In questa variante l'effetto del vento è proporzionale, con una costante  $\delta \leq 1$ , alla differenza di quota tra l'istante in cui se ne simula l'effetto e l'istante precedente.
- (7) Fai in modo che l'utente possa scegliere, nel corso dell'esecuzione del programma, tra un tipo di simulazione e l'altro.

Usando il programma modificato esegui due simulazioni in condizioni diverse e registra i dati in due *file* diversi (ad esempio `roverA.dat` e `roverB.dat`). Puoi confrontare direttamente i risultati usando `gnuplot` facendo un grafico dei valori presenti nei due *file* usando un comando del tipo `plot 'roverA.dat' using 2:3, 'roverB.dat' using 2:3`.

Esegui le stesse simulazioni più volte e confronta i risultati di simulazioni diverse fatte nelle stesse condizioni. Cosa se ne può dedurre sull'effettiva casualità dei numeri estratti con `rand48()`?