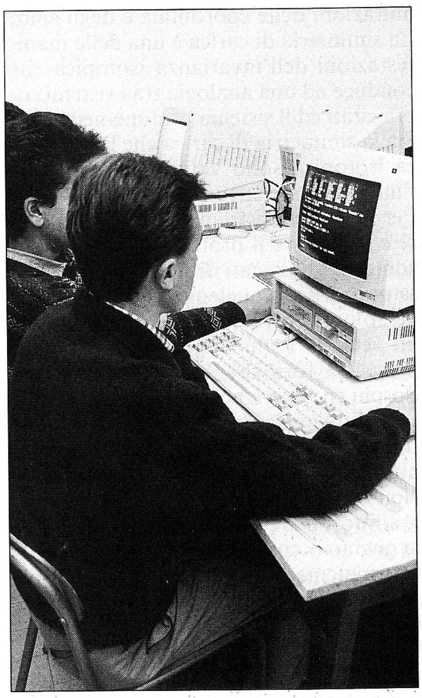
FISICA E INFORMATICA: UN AMBIENTE PER LO STUDIO DEL MOTO. 2

Sandro Ronca

Sul precedente numero della rivista è stato presentato un programma, con relativo listato, che consente la costruzione di un ambiente di simulazione adatto allo studio del moto. L'esercizio che qui viene proposto consiste nella modifica di alcune parti del programma di base per analizzare altri tipi di moto. Gli studenti verranno stimolati nello sviluppo di strumenti sia teorici che informatici.

Continuiamo la descrizione del programma iniziata con l'articolo precedente (*cfr. Nuova Secondaria*, 5, 15-1-93, p. 78).

La procedura Forza-Costante serve per calcolare le componenti dell'accelerazione, note le componenti della forza. Anche se, più semplicemente, le componenti dell'accelerazione potrebbero essere fornite direttamente, abbiamo scelto questa via per evitare la separazione degli aspetti cinematici da quelli dinamici. Inoltre vi è un'esigenza di uniformità poichè, per ogni tipo di moto analizzato, avremo una procedura che svolgerà gli stessi compiti. La procedura Muovi_Oggetto costituisce assieme a Incrementa, il cuore vero e proprio del programma. Essa è costituita da un ciclo WHILE controllato dalla variabile t che simula il flusso temporale. Il ciclo si arresta quando si raggiunge il valore Tmax, che viene passato alla procedura come parametro, oppure quando si esce dai limiti di visualizzazione. Una classica tecnica di animazione consiste nel cancellare un oggetto prima di farlo apparire nella nuova posizione: questo è esattamente ciò che fanno le due chiamate a Poni_Oggetto (quella fuori dal ciclo serve per la prima visualizzazione); con Readln il programma attende la pressione del tasto Invio. L'istruzione



PutPixel permette di visualizzare la traiettoria dell'oggetto. Sono previsti degli spazi in cui porremo le chiamate di procedura per moti soggetti a forze variabili o a particolari vincoli.

Il programma principale non dovrebbe presentare difficoltà di comprensione. Notiamo solo che la variabile GetMaxY utilizzata nel calcolo dei fattori di scala non deve essere dichiarata perchè preesistente: viene automaticamente inizializzata e contiene il valore dell'ampiezza verticale (in pixel) dello schermo, dipendente dalla scheda grafica.

Considerazioni metodologiche

In questa versione base il programma consente di studiare moti bidimensionali rettilinei uniformi e uniformemente accelerati. Come si vede osservando la codifica, esso è molto semplice e si presta bene ad essere, con minimo sforzo, ampliato e modificato per comprendere un insieme più vasto di moti. Di fatto gli studenti verranno coinvolti nello sviluppo degli strumenti teorici (analisi del problema) ed informatici (scrittura delle procedure) necessari sia alla realizzazione del programma di base che degli ampliamenti successivi, con i quali verranno simulati il moto circolare, il moto armonico, l'oscillatore armonico, il pendolo ed il moto planetario.

La simulazione

Il problema di realizzare l'ambiente di simulazione può essere convenientemente scomposto in tre sotto-problemi: *a)* creare e visualizzare l'oggetto; *b)* realizzare un modello in scala dello spazio bidimensionale; *c)* simulare il moto secondo le leggi della dinamica. Nessuna difficoltà dovrebbe presentarsi per i primi due punti. È consigliabile, dopo aver scritto le necessarie procedure (fino a Poni_Oggetto compresa), invitare gli studenti a stendere un brevissimo programma principale con cui, variando i parametri X e Y di Poni_Oggetto (X, Y), si possa verificare la correttezza del posizionamento sul reticolo (la sequenza è: Inizia_Grafica; ScalaX: = ...; ScalaY: = ...; Crea_Oggetto (raggio); Reticolo (PassoR); Poni_Oggetto (X, Y); Readln; Closegraph;), dando anche diversi valori ai parametri LatoR e PassoR (PassoR: = 1 darà un passo del reticolo corrispondente ad 1 metro). Successivamente si può sostituire la chiamata a Poni_Oggetto, con un semplice ciclo come ad esempio:

```
LatoR: = 10; PassoR: = 1;
X: = 1; Y: = 5;
Poni_Oggetto (X, Y);
While X< = LatoR do
begin
  Poni_Oggetto (X, Y);
  X: = X + 0.25;
  Poni_Oggetto (X, Y);
end;
```

con il quale si provoca lo spostamento della pallina orizzontalmente con un incremento della X corrispondente, in questo caso, a 25 cm. È bene sperimentare modificando le variabili finchè il funzionamento di questa prima parte non sia perfettamente chiaro.

È probabile che, già in questa fase, gli allievi si pongano dei problemi del tipo: «come posso muovere la pallina secondo un angolo di 45° rispetto all'orizzontale?», «perché non riesco ad ottenere una traiettoria curva?», ecc. Se ciò non accade spontaneamente è opportuno che l'insegnante intervenga, ponendo egli stesso quesiti come i precedenti. Ciò ha lo scopo di preparare il terreno per i futuri sviluppi. Tali curiosità, naturali o indotte, possono anche essere sfruttate per raggiungere obiettivi che trascendono gli scopi per cui questa attività viene impostata (ad esempio si può approfittarne per chiarire il significato di dipendenza funzionale: «cosa succede se $Y = X^2$?»).

Il punto *c)* deve naturalmente essere analizzato in maggior dettaglio, costituendo la parte essenziale e didatticamente più significativa del programma. Anche qui possiamo evidenziare alcuni punti essenziali.

a) A partire da valori iniziali noti, dobbiamo calcolare velocità e posizioni successive utilizzando solamente le difinizioni di velocità $v = \delta x/\delta t$ e accelerazione $a = \delta v/\delta t$ e la seconda legge della dinamica $F = ma$, poichè secondo il nostro modello esse rappresentano il livello più elementare e fondamentale di descrizione del movimento (*v.* precedente articolo p. 78, punto 1).

b) Dovremo simulare il flusso del tempo con una variabile (t) che verrà opportunamente incrementata (Deltat) all'interno di un ciclo. Qualche problema potrà porsi sul significato da attribuire alla variabile tempo. Il nostro modello implica un fattore di scala temporale che non è totalmente controllabile in quanto dipende dall'hardware (la durata del ciclo di istruzione) e dalla risoluzione (l'incremento Deltat) che si vuole ottenere. L'unità di tempo del modello subisce una dilatazione per piccoli incrementi temporali, in relazione al maggior numero di operazioni che sono richieste, e ciò comporta una diminuizione della velocità del moto.

c) La procedura di calcolo fa riferimento al cosiddetto metodo passo a passo (noto anche come metodo di Eulero) per la risoluzione numerica delle equazioni differenziali. Esso afferma che: data un'equazione differenziale $dy/dx = f(x, y)$ con la condizione iniziale $y(x_0) = y_0$ se $y(x_i) = y_i$ e $dy/dx_{(x=xi)} = f(x_i,y_i)$ detto $\delta x = x_{i+1} - x_i$, allora è possibile approssimare l'equazione differenziale con l'equazione alle differenze finite: $y_{i+1} = y_i + \delta x f(x_i, y_i)$. Esistono alcune ottime ragioni per introdurre, in forma semplificata, questo metodo: è un metodo di uso generale, particolarmente potente quando si voglia analizzare un fenomeno senza risolvere equazioni differenziali; si presta molto bene ad essere applicato al calcolo automatico; fa riferimento diretto alla «fisica del problema»; mette in evidenza la causalità dei fenomeni. Vale la pena di soffermarci su quest'ultimo punto perchè può servirci per mettere a fuoco l'idea fondamentale della Fisica classica: se spiegheremo la procedura di calcolo scrivendo ad esempio che $x(t + \delta t) = x(t) + \delta x$ dovremo far capire che con questa for-

mula stiamo implicitamente ammettendo la nostra fede nella causalità, crediamo cioè che lo stato di un sistema ad un certo istante di tempo sia perfettamente determinato dal suo stato all'istante immediatamente precedente. Sappiamo che proprio la crisi del determinismo ha contrassegnato il passaggio dalla Fisica classica alla Fisica quantistica (che non viene quasi mai affrontata nelle nostre scuole) ed è evidente che si potrebbe aprire qui un'interessante ed utile discussione.
Il metodo passo a passo può essere descritto così:
1) Attribuiamo ad x il valore iniziale $x = x_i$;
2) Calcoliamo l'incremento δx;
3) Calcoliamo il valore finale $x_f = x + \delta x$;
4) Attribuiamo il nuovo valore alla variabile $x = x_f$;
5) ripetiamo dal passo 2).
La traduzione in Turbo Pascal è data dalle procedure Incrementa e Muovi_Oggetto la cui struttura mi sembra sufficientemente chiara (*v.* precedente articolo, p. 78, punto 2: la formalizzazione).

Come utilizzare il programma base

Chiederemo inizialmente agli studenti di porre le condizioni per simulare moti rettilinei uniformi secondo varie direzioni nel piano e velocità (Fx = 0, Fy = 0, valori opportuni per vx e vy). Varrà quindi la pena di far notare che il principio di inerzia viene automaticamente rispettato essendo implicitamente presente nelle istruzioni che formano la stessa procedura Incrementa. La consistenza del modello potrà essere verificata rilevando le distanze percorse in un intervallo di tempo stabilito (ottenuto fissando un valore per il parametro di Muovi_Oggetto Tmax). Successivamente la sperimentazione riguarderà i moti uniformemente accelerati e i moti composti ovviamente proponendo sempre le opportune verifiche. Un suggerimento in questo senso potrebbe essere quello di riprodurre con la simulazione, le traiettorie ricavate da fotografie stroboscopiche di cui molti testi di Fisica sono ricchi. Come passo successivo si possono proporre problemi che se affrontati con una simulazione di questo tipo riescono ad assumere un aspetto ludico creando motivazioni molto più forti e risultati molto più significativi di quelli che si possono ottenere usando carta e penna. Facciamo qualche esempio: data la posizione iniziale della pallina, stabilito che essa si muoverà in un campo gravitazionale, determinare i parametri iniziali perchè colpisca un determinato punto del piano; determinare l'istante di lancio di un oggetto da un velivolo in moto rettilineo uniforme perchè colpisca un punto dato.

Sandro Ronca