

FISICA E INFORMATICA: UN AMBIENTE PER LO STUDIO DEL MOTO. 1

Sandro Ronca

La fisica procede nello studio dei fenomeni naturali ideando e sviluppando modelli mediante i quali cerca di individuare le leggi che governano l'universo. I modelli nascono da osservazioni e ipotesi che si fanno sulla natura e la loro importanza è dovuta al fatto che, attraverso di essi le descrizioni matematiche degli eventi naturali acquistano significato e coerenza all'interno di una rappresentazione il più possibile unitaria della realtà fisica.

Costruendo un modello, operiamo una serie di astrazioni attraverso le quali tentiamo di cogliere il significato profondo del succedersi degli eventi, cerchiamo cioè di stabilire leggi di valore universale che siano in grado di descrivere il maggior numero possibile di fatti, anzi, come gli sforzi della Fisica moderna testimoniano, la nostra maggiore ambizione è di possedere una teoria di grande generalità che possa ricondurre a pochi principi fondamentali tutto quanto avviene nell'universo. Se fossimo costretti a ragionare in termini rigorosamente fenomenici, basandoci cioè solamente sulla nostra esperienza sensoriale, l'impresa, di fronte all'enorme varietà dei fenomeni, apparirebbe del tutto disperata.

Fisica, modelli, simulazioni

Ogni modello comporta, in più o meno ampia misura, un'interpretazione della realtà. La prima legge della dinamica comporta l'ipotesi che sia possibile ridurre a zero le forze esterne agenti sul corpo, si fonda cioè su un'estrapolazione ricavata da dati empirici di cui non è possibile una verifica diretta. D'altra parte dovremo in qualche maniera definire il concetto di forza. La seconda legge $F = ma$ ma può essere considerata empirica, ma allora dovremo dare un significato al concetto di massa inerziale. Il moto planetario trova una descrizione all'interno del sistema concettuale fornito dalla gravitazione newtoniana, ma un insieme diverso di ipotesi potrebbe dare risultati altrettanto validi: ne è esempio la teoria della relatività che vede lo stesso moto come effetto della curvatura dello

spazio-tempo. Ciò che importa è che il modello produca delle asserzioni che siano confrontabili con i dati sperimentali, in modo che sia possibile un raccordo tra il suo sistema concettuale e il mondo fisico e quindi una valutazione della sua attendibilità. È grazie a questo modo di procedere che la Fisica costruisce un sistema coerente di idee e di concetti, aventi per oggetto la realtà, con vaste implicazioni filosofiche e culturali. Mettere in evidenza questi aspetti e sfruttare il loro elevato potenziale educativo è certamente un compito importante che l'insegnamento di questa materia deve affrontare. Dal punto di vista strettamente didattico si tratta di coniugare la pura conoscenza dei fatti con l'identificazione, la comprensione e l'uso del paradigma interpretativo che la scienza propone. In altre parole si tratta di far comprendere ed assimilare il metodo scientifico, passando indubbiamente attraverso lo sviluppo di creatività, senso critico, capacità di analisi e sintesi, uso appropriato del linguaggio matematico. Una sfida interessante, visto che il pericolo di una presentazione della Fisica come serie di esercitazioni matematiche, o, peggio, come insieme di nozioni propedeutiche allo studio di materie tecniche, è sempre in agguato. Vi deve essere ovviamente un corso di strumenti, supporti, attività didattiche ben correlate, per raggiungere gli obiettivi enunciati. Alcuni di questi strumenti possono far riferimento alle tecnologie informatiche, che sono oggetto di queste riflessioni.

L'elaboratore ha l'affascinante potere di permettere la sperimentazione con le idee e con i concetti, verificandone in tempo quasi reale gli effetti. In questo senso un laboratorio di informatica può essere considerato un vero e proprio «labora-

torio delle idee», le cui potenzialità didattiche sono enormi e sicuramente non completamente esplorate.

Esistono molte e diverse modalità di utilizzo dell'Informatica in ambito didattico, ma qui vogliamo prendere in considerazione la simulazione ed, in particolare, la simulazione costruita dagli stessi studenti.

La realizzazione di una simulazione comporta l'attuazione di quei passaggi che, al di là della fase osservativa, tipicamente caratterizzano la costruzione di un modello in Fisica: 1) bisogna mettere a fuoco le idee che stanno alla base del modello, le quali 2) devono essere tradotte in un insieme di relazioni formali e 3) fornire un output attraverso cui sia possibile verificarne la correttezza e la consistenza. Inoltre 4) si deve mettere, se possibile, in rilievo il grado di unificazione che il modello offre, spiegando un numero sufficientemente ampio di fatti, e 5) considerare l'eventualità di implicazioni teoriche o empiriche non direttamente evidenti. Vedremo come praticamente tutti questi punti saranno toccati durante la realizzazione del progetto.

Il progetto

Ci proponiamo allora di fornire agli allievi gli strumenti per la costruzione di un ambiente di simulazione adatto allo studio del moto. I prerequisiti informatici consistono nella conoscenza elementare del linguaggio Pascal con particolare riferimento all'uso delle procedure e delle funzioni. Questa attività è indirizzata a studenti del triennio, ma ciò non toglie che, con una guida opportuna, interessanti risultati si possano ottenere anche nel biennio. Il problema che assegneremo agli allievi sarà di costruire un programma in grado di visualizzare e muovere oggetti grafici in accordo con le leggi della dinamica e di verificare la possibilità di descrivere con esso i principali tipi di moto.

Il seguito di questo e di un articolo successivo riguarderà la descrizione di un programma che vuole solamente costituire un modello di riferimento in rapporto alle considerazioni di tipo metodologico che verranno via via proposte e viene

quindi dato nella forma più essenziale possibile, con il non celato scopo di dimostrare che è possibile raggiungere importanti risultati educativi con un dispendio veramente limitato di risorse.

Cominciamo la descrizione del programma riferendoci alla codifica in Turbo Pascal (5.0). L'inserimento di nuovi valori delle variabili avviene agendo sul codice sorgente. Tutte le inizializzazioni di variabili interessanti si trovano all'inizio del

programma principale, in modo da consentirne un'agevole modifica. La procedura *Inizia_Grafica* serve ad inizializzare l'hardware grafico e a rilevare automaticamente il tipo di scheda. L'istruzione *GetAspectRatio* permette di mantenere i rapporti dimensionali per ogni tipo di scheda grafica (CGA, EGA, VGA). *Crea_Oggetto* visualizza e memorizza l'immagine di un cerchio di raggio dato: è questo l'oggetto che dovrà essere animato. In queste prime due procedure non

vi è nulla di didatticamente rilevante per cui possono essere fornite dal docente senza ulteriori commenti. Le funzioni *Xsc* e *Ysc* operano una trasformazione delle coordinate da reali ad intere attraverso un fattore di scala (il margine serve solo a spostare il reticolo rispetto ai limiti dello schermo). L'area di visualizzazione del monitor rappresenta il mondo in cui i nostri oggetti dinamici devono muoversi. Se vogliamo che sia possibile una verifica quantitativa del model-

```

program Moto;
uses
  Crt, Graph;
Const
  Margine = 10;
  raggio = 5;
  g = 9.81;
var
  P : pointer;
  Size,Xasp,Yasp : word;
  ScalaX,ScalaY,LatoR,PassoR : real;
  XO,YO,X,Y,Vx,Vy,Ax,Ay,Deltat,t,Tempo,Fx,Fy,Massa : real;

procedure Inizia_Grafica;
var Gd,Gm : integer;
begin
  Gd := Detect;
  Initgraph(Gd,Gm,'');
  GetAspectRatio(Xasp,Yasp);
end;

procedure Crea_Oggetto(r:integer);
var X0,Y0,X1,X2,Y1,Y2:integer;
begin
  X0 := 20; Y0 := 20;
  X1 := X0 - r; X2 := X0 + r;
  Y1 := Y0 - r; Y2 := Y0 + r;
  setcolor(lightred);
  setfillstyle(1,red);
  circle(X0,Y0,r);
  floodfill(X0,Y0,lightred);
  Size := ImageSize(X1,Y1,X2,Y2);
  GetMem(P,Size);
  GetImage(X1,Y1,X2,Y2,P');
  ClearDevice;
end;

function Xsc(X:real):integer;
begin
  Xsc := round(X*scalaX) + margine;
end;

function Ysc(Y:real):integer;
begin
  Ysc := round(Y*scalaY) + margine;
end;

procedure reticolo(passo:real);
var i:real;
begin
  setcolor(darkgray);
  i := 0;
  While i <= LatoR do
  begin
    line(Xsc(i),Ysc(0),Xsc(i),Ysc(LatoR));
    line(Xsc(0),Ysc(i),Xsc(LatoR),Ysc(i));
    i := i + passo;
  end;

procedure Poni_Oggetto(X,Y:real);
begin
  Putimage(Xsc(X)-raggio,Ysc(Y)-raggio,P',XorPut);
end;

procedure Incrementa(Xi,Yi,Vix,Viy,Ax,Ay,dt:real;var Xf,Yf,Vfx,Vfy:real);
begin
  Vfx := Vix + Ax*dt;
  Vfy := Viy + Ay*dt;
  Xf := Xi + Vfx*dt;
  Yf := Yi + Vfy*dt;
end;[Incrementa]

procedure Forza_costante(Fx,Fy,Massa:real;var Accx, Accy:real);
begin
  Accx := Fx/Massa;
  Accy := Fy/Massa;
end;

procedure muovi_oggetto(tmax:real);
begin
  t := 0; X := X0; Y := Y0;
  Poni_Oggetto(X,Y);
  readln;
  Forza_Costante(Fx,Fy,Massa,Ax,Ay);
  while (t <= tmax) and (X <= LatoR) and (Y <= LatoR) do
  begin
    Poni_Oggetto(X,Y); [Cancella Immagine precedente]
    ["Inserire qui procedure per Forze non costanti"]
    PutPixel(Xsc(X),Ysc(Y),lightgray);
    Incrementa(X,Y,Vx,Vy,Ax,Ay,Deltat,X,Y,Vx,Vy);
    ["Condizioni su X e/o Y per moti vincolati"]
    Poni_Oggetto(X,Y);
    t := t + Deltat;
  end;
end;

begin
  {-----Variabili su cui intervenire-----}
  LatoR := 6;      PassoR := 1;
  X0 := 1;  Y0 := 1;  Vx := 2;  Vy := 0;
  Massa := 200;   Fx := 0;   Fy := Massa*g;
  Deltat := 0.005;  Tempo := 1;
  {-----}
  Inizia_Grafica;
  ScalaX := (GetMaxY-20)/LatoR*Yasp/Xasp;
  ScalaY := (GetMaxY-20)/LatoR;
  Crea_oggetto(raggio);
  Reticolo(PassoR);
  Muovi_Oggetto(Tempo);
  readln;
  CloseGraph;
end.

```

lo, quest'area deve costituire una rappresentazione in scala dello spazio bidimensionale. In altre parole dovremo stabilire che N punti schermo (pixel) equivalgono ad un metro. I fattori di scala sono definiti all'inizio del programma principale. La procedura Reticolo, evidenzia appunto un reticolo con passo variabile (PassoR). La variabile LatoR consente di modificare l'ampiezza dell'area quadrata in esame (effetto zoom). L'origine (0,0) si trova nel vertice in alto a sinistra.

Poni_Oggetto permette di visualizzare la pallina nel punto schermo (Xsc, Ysc) corrispondente alle coordinate reali (X, Y). Notare che deve essere sottratta la misura del raggio per avere coincidenza con il centro della circonferenza. L'opzione XORPUT significa che viene esercitata un'operazione di or esclusivo sui pixel, con che viene conservata l'immagine di sfondo, mentre due PutImage riferentisi a coordinate identiche ed alla stessa immagine, applicati in sequenza, ne provo-

cano la cancellazione. La procedura Incrementa riceve in ingresso i valori di posizione, velocità ed accelerazione, secondo le loro componenti cartesiane ed il valore dell'intervallo temporale dt e fornisce in uscita (variabili precedute dalla parola chiave VAR) le variabili posizione e velocità opportunamente incrementate. Data la sua importanza, torneremo ancora su questa procedura. *(continua)*

Sandro Ronca