Appendice D
Tecniche di calcolo per esperimenti numerici

D.1 Il calcolo numerico nella risoluzione delle equazioni differenziali

Una delle più importanti e interessanti possibilità nel campo dell'analisi numerica è certamente quella di risolvere equazioni differenziali: queste, come è noto, costituiscono il modello matematico descrittivo di una grandissima varietà di fenomeni, ma non sempre presentano soluzioni di tipo analitico.

Per questo motivo si ravvisa l'opportunità di esporre brevemente i metodi risolutivi usati in alcuni dei programmi descritti in questo libro, con particolare riferimento alla risoluzione numerica della classica equazione del moto

$$\frac{d^2x}{dt^2} = f(t,x,v)$$

che costituisce un'equazione differenziale del secondo ordine, riconducibile ad un sistema di due equazioni differenziali del primo ordine del tipo:

$$dx/dt = v$$

$$dv/dt = f(t,x,v) \qquad (1)$$

con le condizioni iniziali:

$x(t_0) = x_0$ (posizione iniziale)
$v(t_0) = v_0$ (velocità iniziale)

È opportuno osservare che, in tutti i tipi di moto studiati, il tempo t non appare esplicitamente nell'espressione dell'accelerazione, data dalla funzione $f(t, x, v)$, cosicché la funzione $f(t,x, v)$ stessa risulta dipendente dalle sole variabili x e v; nel seguito verrà quindi indicata semplicemente con $f(x, v)$.

Vediamo ora i vari metodi di calcolo usati per studiare, istante per istante, l'evoluzione di un moto qualsiasi, caratterizzato dal sistema di equazioni (1): vedremo che i metodi più semplici possono trovare soddisfacente applicazione nei casi in cui gli intervalli di integrazione siano ragionevolmente non troppo grandi.

Essenzialmente il procedimento di calcolo si fonda sulle seguenti considerazioni: se a un dato istante t_0 sono noti i valori delle grandezze $x(t_0) = x_0$, $v(t_0) = v_0$ e quindi, nota la funzione $f(x,v)$, anche il valore dell'accelerazione $a_0 = f(x_0,v_0)$, allora è possibile determinare con una certa approssimazione i valori che le stesse grandezze assumono in un istante t_1 successivo, così vicino a t_0 che nell'intervallo di tempo t_1–t_0 l'accelerazione possa ritenersi costante; i nuovi valori x_1, v_1, a loro volta, consentono di determinare quelli corrispondenti a un istante t_2, successivo a t_1, e così via, in modo da descrivere completamente il moto considerato in un intervallo di tempo qualsiasi.

D.2 Il metodo di Eulero

Il metodo di Eulero è fra quelli più semplici e consente di ottenere due successioni $\{x_k\}$ e $\{v_k\}$ dei valori x_k e v_k, calcolati negli istanti t_k, mediante le equazioni ricorsive:

$$x_{k+1} = x_k + v_k Dt$$
$$v_{k+1} = v_k + a_k Dt = v_k + f(x_k, v_k)\, Dt$$

che costituiscono un'ovvia, ma non troppo accurata approssimazione delle equazioni

$$dx/dt = v$$

$$dv/dt = a = f(x, v)$$

D.3 Il metodo di Eulero modificato e la tecnica 'predictor-corrector'

Vediamo come si può migliorare il metodo di Eulero nel caso più frequente in cui l'accelerazione sia una funzione $a = f(x)$ della sola coordinata x :

all'istante t_0 risulta $\quad a_0 = f\ (x_0)$

e all'istante $t_0 + Dt/2$ $\quad v_{m0} = v_0 + a_0 Dt/2$

che è la velocità media nell'intervallo Dt, la quale consente di calcolare la posizione x_1 raggiunta all'istante t_1

$$x_1 = x_0 + v_{m0} Dt \approx x(t_1)$$

I valori successivi saranno quindi:

$$\begin{aligned} a_1 &= f(x_1) \\ v_{m1} &= v_{m0} + a_1 Dt \\ x_2 &= x_1 + v_{m1} Dt \end{aligned}$$

e così di seguito.

t_0		t_1		t_2
x_0		x_1		x_2
a_0		a_1		a_2
v_0	v_{m0}		v_{m1}	

Più semplicemente, dopo aver posto

$$\begin{aligned} a &= f(x_0) \\ v &= v_0 + a \cdot Dt/2 \\ x &= x_0 + v \cdot Dt \end{aligned}$$

basterà reiterare le istruzioni

$$\begin{aligned} a &= f(x) \\ v &= v + a \cdot Dt \\ x &= x + v \cdot Dt \end{aligned}$$

Questo metodo fornisce risultati accettabili quando l'osservazione richiede tempi abbastanza brevi; in caso contrario le approssimazioni previste nei singoli passi, sia pure in misura molto piccola, sommandosi possono dar luogo ad errori anche vistosi.

Utilizzando incrementi di tempo *Dt* sempre più piccoli si allungano i tempi di lavoro, ma, entro certi limiti, si riducono gli errori.

Un ulteriore affinamento del calcolo si può ottenere valutando le grandezze in gioco come viene di seguito indicato.

Accelerazione iniziale: $a_0 = f(x_0)$

velocità presunta dopo il tempo *Dt*

considerando costante l'accelerazione a_0
(formula di Eulero): $v = v_0 + a_0 Dt$

suo valor medio nell'intervallo *Dt*: $v_m = (v + v_0)/2$

posizione raggiunta dopo il tempo *Dt*: $x = x_0 + v_m Dt$

accelerazione presunta dopo il tempo *Dt*: $a = f(x)$

suo valor medio nell'intervallo *Dt*: $a_m = (a_0 + a)/2$

velocità dopo il tempo *Dt*
(formula di Eulero modificata): $v = v_0 + a_m Dt$

I valori finali di x e di v dovranno essere assunti come valori iniziali nel successivo intervallo di tempo *Dt*; basterà allora porre $x_0 = x$, $v_0 = v$ e ripetere il processo.

Questo metodo fa parte di una categoria di procedimenti noti come *predictor-corrector*: essi consistono nell'uso di una formula (*predictor*) che fornisce una prima previsione del valore che la grandezza in esame assume alla fine di un dato intervallo, seguito dall'applicazione di una formula più corretta (*corrector*), che la approssima con maggiore precisione.

Nel caso appena esaminato, ad esempio, la formula di Eulero e la formula di Eulero modificata rappresentano una coppia *predictor-corrector* relativa alla velocità.

D.4 Reiterazione della tecnica predictor-corrector

L'ultimo metodo descritto fornisce risultati che, in particolari condizioni, si possono ritenere soddisfacenti; volendo tuttavia migliorarli ulteriormente, basterà reiterare la formula *corrector*, sia per x che per v, all'interno del medesimo intervallo, calcolando più volte il valor medio fra lo stesso valore iniziale e quello finale ultimo ottenuto:

```
a0 = f(x0)
  per ogni valore di k fra 0 e n
    inizio
      vk+1 = v0 + (a0 + ak)Dt/2
      xk+1 = x0 + (v0 + vk+1)Dt/2
      ak+1 = f(xk+1)
    fine
```

Al crescere di n aumentano i tempi di lavoro, ma le successioni dei valori di v, x, a, convergono; cioè al crescere di n la differenza fra due termini consecutivi tende a zero.

In pratica, entro i limiti determinati dalla precisione del calcolatore e utilizzando intervalli Dt sufficientemente piccoli, si nota che tale convergenza viene raggiunta dopo poche iterazioni.

All'uscita dal ciclo basterà indicare con x_0 e v_0 gli ultimi valori ottenuti di x e di v: essi saranno assunti come dati iniziali nel successivo intervallo di tempo.

D.5 Il metodo di Runge-Kutta del quarto ordine

La semplicità e la buona precisione, anche per intervalli di integrazione abbastanza grandi, rendono il metodo di Runge-Kutta, nelle sue numerose varianti, uno dei più usati.

In particolare, il metodo di Runge-Kutta del quarto ordine, assai comune, applicato al solito sistema di equazioni

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = f(x, v)$$

fornisce, ad ogni passo costante Dt, le soluzioni:

$$\begin{aligned} x &= x_0 + [h_1 + 2(h_2 + h_3) + h_4]/6 \\ v &= v_0 + [k_1 + 2(k_2 + k_3) + k_4]/6 \end{aligned} \qquad (2)$$

dove x_0 e v_0 sono i valori della posizione e della velocità all'istante t_0 e x e v i valori delle stesse grandezze nell'istante successivo $t_0 + Dt$, e inoltre:

$$\begin{aligned}
h_1 &= v_0 Dt & k_1 &= Dt \cdot f(x_0, v_0) \\
h_2 &= \left(v_0 + \frac{1}{2} k_1\right) Dt & k_2 &= Dt \cdot f\left(x_0 + \frac{1}{2} h_1, v_0 + \frac{1}{2} k_1\right) \\
h_3 &= \left(v_0 + \frac{1}{2} k_2\right) Dt & k_3 &= Dt \cdot f\left(x_0 + \frac{1}{2} h_2, v_0 + \frac{1}{2} k_2\right) \\
h_4 &= (v_0 + k_3) Dt & k_4 &= Dt \cdot f(x_0 + h_3, v_0 + k_3)
\end{aligned} \tag{3}$$

Dalle ultime relazioni si ricava che:

$$h_1 + 2h_2 + 2h_3 + h_4 = [6v_0 + (k_1 + k_2 + k_3)]Dt$$

e quindi le (2) diventano:

$$\begin{aligned}
x &= x_0 + [v_0 + (k_1 + k_2 + k_3)/6]Dt \\
v &= v_0 + [k_1 + 2(k_2 + k_3) + k_4]/6
\end{aligned} \tag{4}$$

Si può notare che queste equazioni, rispetto alle (2), consentono un calcolo molto più rapido; esse infatti esprimono i valori sia di x che di v in funzione dei medesimi coefficienti k_i ricavabili dalle seguenti relazioni:

$$k_1 = Dt \cdot f(x_0, v_0)$$

$$k_2 = Dt \cdot f\left(x_0 + \frac{1}{2} v_0 Dt, v_0 + \frac{1}{2} k_1\right)$$

$$k_3 = Dt \cdot f\left(x_0 + \frac{1}{2} v_0 Dt + \frac{1}{4} k_1 Dt, v_0 + \frac{1}{2} k_2\right)$$

$$k_4 = Dt \cdot f\left(x_0 + v_0 Dt + \frac{1}{2} k_2 Dt, v_0 + k_3\right)$$

ottenute dalle (3) con successive sostituzioni.

D.6 Confronto dei metodi di calcolo

Il programma che segue pone a confronto i metodi descritti. Esso studia il moto di un oscillatore armonico, visualizzando il grafico della funzione $x = x(t)$ in funzione del tempo.

Due sono i test per la valutazione della bontà del metodo usato: il grafico di $x = x(t)$ è costruito insieme a quello teorico (di diverso colore) di $x = x_0 \cdot \cos(\Omega t)$ e in ogni istante viene controllata la conservazione dell'energia mediante la visualizzazione del rapporto E/E_0 fra l'energia totale E e quella iniziale E_0.

È opportuno osservare che in questo caso l'equazione del moto diventa

$$\frac{d^2x}{dt^2} = -\frac{k}{m}\,x = f(x)$$

e quindi la funzione $a = f(x,v)$ dipende dalla sola coordinata x; i coefficienti k_i usati nel metodo di Runge-Kutta risultano allora così semplificati:

$$k_1 = Dt \cdot f(x_0)$$

$$k_2 = Dt \cdot f\left(x_0 + \frac{1}{2}\,v_0 Dt\right)$$

$$k_3 = Dt \cdot f\left(x_0 + \frac{1}{2}\,v_0 Dt + \frac{1}{4}\,k_1 Dt\right)$$

$$k_4 = Dt \cdot f\left(x_0 + v_0 Dt + \frac{1}{2}\,k_2 Dt\right)$$

Nel capitolo 5 il metodo di Runge-Kutta viene applicato al caso del moto di un oscillatore armonico smorzato, caratterizzato dall'equazione differenziale:

$$\frac{d^2x}{dt^2} = -\frac{k}{m}\,x - \frac{c_s}{m}\,v = f(x, v)$$

con k costante elastica e c_s coefficiente di smorzamento.

D.6.1 Il programma in Pascal

```
Program provarm;

const
      tavolozza=2;

      verde=1;rosso=2;giallo=3;
      sfondo=9;
      dt=0.08;k=2.0;m=1;

type  vettore=array[0..10] of real;

var   ot,e,e0,er,tempo,pixel:real;
      x0,v0,a0,x1,v1,a1,w:real;
      h,k1,k2,k3,k4:real;
      x,v,a:vettore;
      i,iteraz,t:integer;
      gr,ri:integer;

      xo,yo,ux,uy:real;
      y,f:array[1..2] of real;
      grafico:array[1..2] of string[10];
      ch:char;
      err:integer;

Procedure ParametriFinestra(xmin,ymin,xmax,ymax:real);
begin (* ParametriFinestra *)
  ux:=320/(xmax-xmin);
  uy:=200/(ymax-ymin);
  xo:=-xmin*ux;
  yo:=200+ymin*uy
end; (* ParametriFinestra *)

procedure intestazione(i:integer);
begin
  graphcolormode;
  graphbackground(sfondo);
  palette(tavolozza);
  draw(round(xo),round(yo),round(xo+319*ux),round(yo),
    giallo);
  gotoxy(1,1);writeln('dt=',dt:3:2,' k=',k:3:1);
  gotoxy(1,22);writeln('grafico di ',grafico[i]);
  gotoxy(1,24);write('E/Eo=');
  gotoxy(25,22);write('passi:')
end;

Procedure accendi(var asc,ord:real;col:integer);
var xt,yt:integer;
begin
   xt:=round(xo+asc*ux);
```

```
    if (yo-ord*uy>=0) and (yo-ord*uy<=199) then
     begin
       yt:=round(yo-ord*uy);
       plot(xt,yt,col);
     end
end;(* accendi *)

begin (* armonico *)
  ot:=dt*sqrt(k/m);
  ClrScr;
  grafico[1]:='x = x(t)';
  grafico[2]:='E = E(t)';

  gotoxy(1,10);
  writeln('Studio del moto di un corpo soggetto a una forza
           F=-kx');
  writeln;
  writeln('opzione grafici:');writeln;
  writeln('1- grafico di x=x(t): spostamento');
  writeln('2- grafico di E=E(t): energia totale');
  ch:=' ';
  repeat
     read (Kbd,Ch)
  until Upcase(Ch) in ['1','2'];
  val(ch,gr,err);

(* valori iniziali *)
    x0:=0.1;
    v0:=0;
    a0:=-k*x0;

  (* calcolo parametri per la finestra grafica *)
    e:=0.5*(v0*v0+k*x0*x0);
    f[1]:=2*x0;f[2]:=2*e;

  (* scelta del metodo di calcolo *)
    clrscr;
    gotoxy(8,14);
    writeln('Metodo di calcolo:');writeln;
    writeln('1- Eulero');
    writeln('2- Reiterazione predictor-corrector');
    writeln('3- Runge-Kutta');
    ch:=' ';
    repeat
     read (Kbd,Ch)
    until Upcase(Ch) in ['1','2','3'];
    val(ch,ri,err);
    if ri=1 then v0:=v0+a0*dt/2;
    if ri=2 then
              begin
                write('quante volte? ');readln(iteraz);
```

```
                x[0]:=x0;v[0]:=v0;a[0]:=a0
              end;

  (* predisposizione dello schermo grafico *)
   ParametriFinestra(0,-f[gr],319,f[gr]);
   intestazione(gr);
   pixel:=0;tempo:=0;er:=0;

   repeat
      if pixel=319 then  (* fine schermata *)
                   begin
                    pixel:=0;
                    intestazione(gr);
                   end;
      pixel:=pixel+1;
      tempo:=tempo+1;
      gotoxy(6,24);write(er:5:3);

      gotoxy(31,22);write(tempo:6:0);

      case ri of
                                          (* Eulero *)
      1:       begin
                x1:=x0+v0*dt;
                a1:=-k*x1;
                v1:=v0+a1*dt;
                x0:=x1;v0:=v1;
               end;

      2:       begin              (* Predictor-corrector
          iterato *)
                for i:=0 to iteraz do
                   begin
                     v[i+1]:=v[0]+(a[0]+a[i])*dt/2;
                     x[i+1]:=x[0]+(v[0]+v[i+1])*dt/2;
                     a[i+1]:=-k*x[i+1]
                   end;
                t:=iteraz+1;
                x[0]:=x[t];v[0]:=v[t];a[0]:=a[t];
                x1:=x[t];v1:=v[t]
               end;

      3:       begin                      (* Runge-Kutta *)
                w:=-dt*k;
                h:=w*dt/2;
                k1:=w*x0;
                k2:=k1+h*v0;
                k3:=k2+h*k1/2;
                k4:=k1+h*(2*v0+k2);
                x1:=x0+dt*(v0+(k1+k2+k3)/6);
                v1:=v0+(k1+2*k2+2*k3+k4)/6;
```

```
                x0:=x1;v0:=v1
              end
      end;

      y[1]:=x1;

  (* energia  [e0=energia iniziale] *)
      e:=0.5*(v1*v1+k*x1*x1);
      y[2]:=e;
      if tempo=1 then e0:=e;
      er:=e/e0;

  (* costruzione del grafico *)
      accendi(pixel,y[gr],giallo);
      if gr=1 then          (* per confronto con y=a cos(jt) *)
           begin
           y[gr]:=0.1*cos(tempo*ot);
           accendi(pixel,y[gr],rosso)
           end
   until KeyPressed
end.
```

D.6.2 Il programma in BASIC

```
10 'Provarm
20 '(oscillatore armonico)
30 KEY OFF:CLS:CLEAR
35 :
40 REM ---- costanti ----
50 SCREEN 0:WIDTH 80
60 TAVOLOZZA=2:SFONDO=9
70 VERDE=1:ROSSO=2:GIALLO=3
80 DT=.1:K=1:M=1:OT=DT*SQR(K/M)
90 PRINT
95 :
100 REM ---- valori iniziali ----
110 X0=.1:V0=0
120 A0=-(K/M)*X0
130 DEF FNA(X)=-(K/M)*X
140 E0=.5*(M*V0*V0+K*X0*X0)
145 :
150 REM ---- scelta del metodo di calcolo ----
160 PRINT "Moto di un corpo soggetto a una forza F=-kx"
170 LOCATE 5,1:PRINT "Scelta del metodo di calcolo:":PRINT
180 PRINT "    1 - Eulero"
190 PRINT "    2 - Eulero modificato iterato "
200 PRINT "    3 - Runge-Kutta (4°ordine)"
210 GOSUB 760:IF RI<1 OR RI>3 THEN 210
220 IF RI=1 THEN V0=V0+A0*DT/2
230 PRINT
```

```
240 IF RI=2 THEN INPUT "quante volte? ",ITERAZ:
    X(0)=X0:V(0)=V0:A(0)=A0
250 SCREEN 1:COLOR SFONDO,TAVOLOZZA
260 WINDOW (0,-2*X0)-(320,2*X0)
270 GOSUB 430
280 PIXEL=0:TEMPO=0
285 :
290 WHILE PIXEL <320
300     PIXEL=PIXEL+1:TEMPO=TEMPO+1
310     ON RI GOSUB 510,570,660
320     REM ---- calcolo dell'energia ----
330     E=.5*M*V*V+.5*K*X*X
340     REM ---- visualizzazione ----
350     PSET(PIXEL,.1*COS(TEMPO*OT)),ROSSO

360     PSET(PIXEL,X),GIALLO
370     LOCATE 1,24:PRINT "E/Eo=";USING"#.###";E/E0
380 LOCATE 22,31:PRINT TEMPO:LOCATE 2,1
390 K$=INKEY$:IF K$<>"" THEN IF K$=CHR$(27) THEN END
400 WEND
410 PIXEL=0:GOSUB 430:GOTO 290
415 :
420 REM ---- intestazione ----
430 CLS:LINE (0,0)-(319,0)
440 LOCATE 1,2:PRINT "Dt=";DT;"  k=";K
450 LOCATE 22,1:PRINT "x = x(t)"
460 POKE &H4E,2:PRINT "x = Xo cos(Ωt)"
470 POKE &H4E,3:LOCATE 22,25:PRINT "passi:";
480 LOCATE 24,25:PRINT "ESC per finire";
490 RETURN
495 :
500 REM ---- Eulero ----
510 X=X0+V0*DT
520 A=-(K/M)*X
530 V=V0+A*DT
540 X0=X:V0=V
550 RETURN
555 :
560 REM ---- Reiterazione predictor-corrector ----
570 FOR I=0 TO ITERAZ
580 V(I+1)=V(0)+(A(0)+A(I))*DT/2
590 X(I+1)=X(0)+(V(0)+V(I+1))*DT/2
600 A(I+1)=-(K/M)*X(I+1)
610 NEXT I
620 X(0)=X(I):V(0)=V(I):A(0)=A(I)
630 X=X(I):V=V(I)
640 RETURN
650 REM ---- Runge-Kutta (4°ordine) ----
660 K1=FNA(X0)*DT
670 K2=FNA(X0+V0*DT/2)*DT
680 K3=FNA(X0+(V0+K1/2)*DT/2)*DT
```

```
690 K4=FNA(X0+(V0+K2/2)*DT)*DT
700 X=X0+(V0+(K1+K2+K3)/6)*DT
710 V=V0+(K1+2*(K2+K3)+K4)/6
720 X0=X:V0=V
730 RETURN
750 REM —— attesa di risposta ——
760 R$=INKEY$:IF R$="" THEN 760
770 RI=VAL(R$)
780 RETURN
```

D.6.3 Risultati

I grafici che si ottengono con i due metodi di calcolo, quello di Eulero e quello di Runge-Kutta, mostrano significative differenze: dopo oltre 20000 passi, il primo, evidenzia un accentuato sfasamento fra i grafici di $x=x(t)$ e di $x=x_0 \cdot \cos(\Omega t)$ a confronto e quindi un notevole accumulo di errori. Il secondo, con la sovrapposizione dei due grafici e la conservazione dell'energia dopo ben 25000 passi, prova una apprezzabile precisione di calcolo.

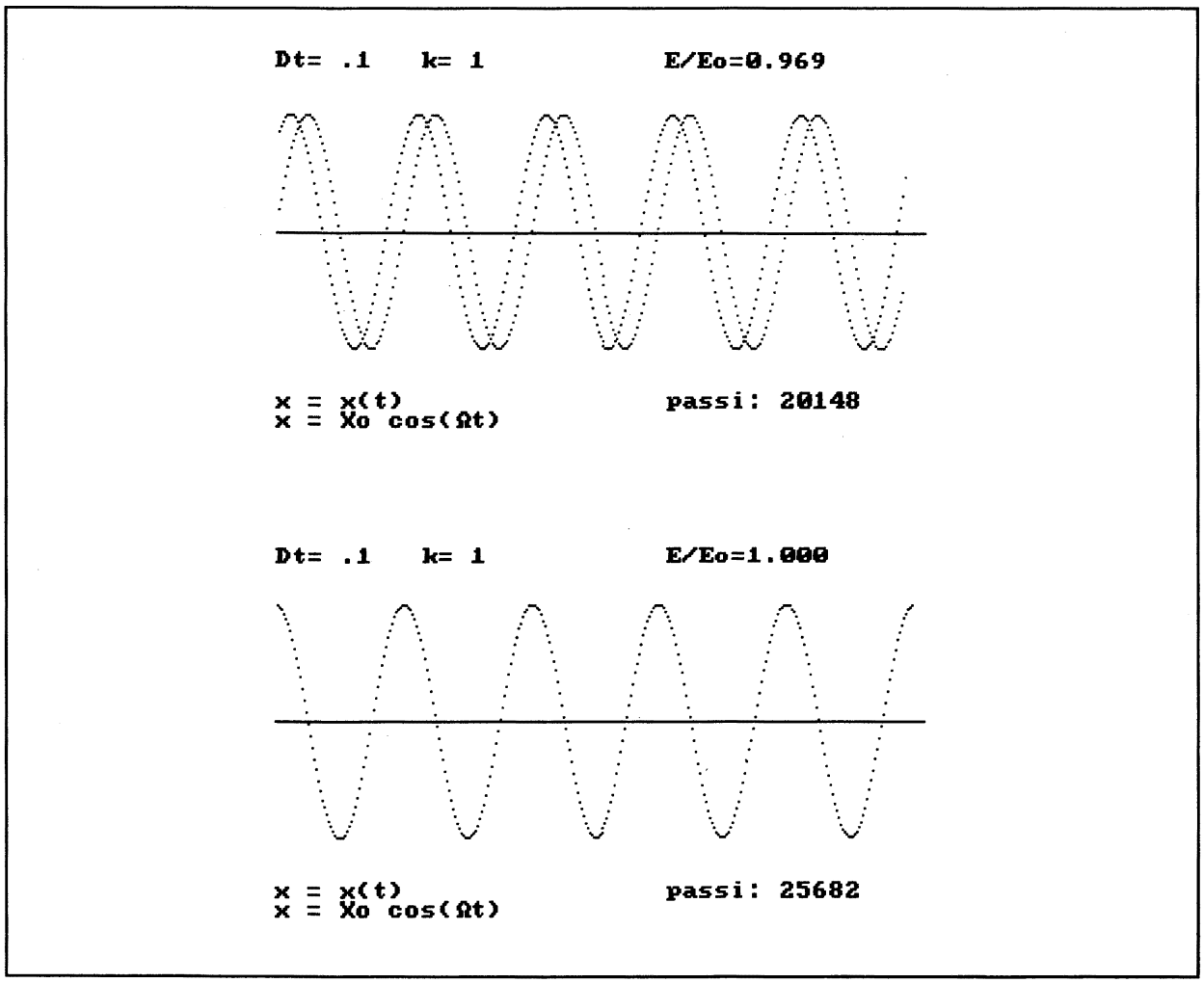

Figura 44

Note: