



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

In generale un'equazione differenziale del primo ordine è del tipo:

$$y' = f(t, y)$$

- dove l'operatore $(\cdot)' \equiv d/dt$ indica la derivazione rispetto alla t
- f una funzione delle variabili scalari t, y .
- l'incognita vettoriale $y = y(t)$, funzione della variabile indipendente t



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

Una qualsiasi equazione differenziale ordinaria, di qualsiasi ordine, può sempre essere ridotta ad un sistema di equazioni differenziali del primo ordine. Ad esempio l'equazione

$$\frac{d^2 g}{dt^2} + q(t) \frac{dg}{dt} = r(t)$$

nell'incognita scalare $g(t)$, può essere riscritta sotto forma di sistema:

$$\begin{cases} \frac{dy_1}{dt} = y_2(t) \\ \frac{dy_2}{dt} = r(t) - q(t)y_2(t) \end{cases}$$

nella funzione vettoriale incognita $y(t) = \{y_1(t), y_2(t)\}^T$, dove l'incognita originale compare nella posizione $y_1(t) = g(t)$ mentre $y_2(t)$ è una nuova variabile dipendente, spesso chiamata "variabile ausiliaria".



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

Il linguaggio MATLAB possiede una funzione predefinita, *ode45* (*ode* sta per Ordinary Differential Equations), che permette un'agevole soluzione di questo tipo di problemi evitando all'utente la necessità di un'implementazione a basso livello dell'algoritmo di integrazione

Il problema di valori iniziali è il seguente:

$$\frac{d^2 y}{dt^2} + 3 \cos^2 t - 2 = 0$$
$$y(0) = \frac{dy(0)}{dt} = 0$$



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

Si può facilmente verificare che la soluzione analitica del problema è costituita dalla funzione:

$$y(t) = \frac{1}{4}t^2 + \frac{3}{8}\cos(2t) - \frac{3}{8}$$

Per risolvere il problema assegnato è necessario implementare una funzione MATLAB, che chiameremo *myfunction2x1* per ricordare che l'output è un vettore colonna 2×1 , e salvarla in un M-file avente lo stesso nome nella cartella di lavoro scelta dall'utente.



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

Un esempio di file `myfunction_2x1.m` è il seguente:

```
function dydt = myfunction_2x1(t,y)
% MYFUNCTION_2X1 : funzione di 3 variabili, (t,y1,y2), a due componenti
% da valutare nella risoluzione dei problemi di valori iniziali associati
% all'equazione
% (1)          y'' + 3 cos^2(t) - 2 = 0
%
% La (1) si riscrive in forma di sistema di equazioni del primo ordine
% (2a)         y1' = y2
% (2b)         y2' = -3cos^2(t) + 2
%
% Alloca ed inizializza un vettore colonna 2x1
dydt = zeros(2,1);

% Definisce il vettore dydt, variabile di output
dydt(1) = y(2);      % implementa la dipendenza (2a)
dydt(2) = ...        % implementa la dipendenza (2b)
    - 3*(cos(t))^2 + 2;
```



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

La sequenza di comandi che adotteremo sarà del tipo:

```
>> NSTEP = 40; tFin = 2*pi;
>> t = linspace(0,tFin,NSTEP)
>> y0 = [0 0];
>> [Tt,Y] = ode45(@myfunction_2x1,t,y0);
>> Yexact = 0.25*Tt.^2+(3./8.)*cos(2.*Tt) -(3./8.);
>> plot(Tt,Y(:,1),'*',Tt,Yexact,'-')
>> xlabel('t'); ylabel('y'); legend('Runge-Kutta','Soluzione esatta');
```

- Viene fissato inizialmente un numero di passi di integrazione *NSTEP*
- Un valore finale della variabile indipendente *tFin*
- Successivamente viene assegnato un vettore riga *t* con il comando *linspace*, che assegna ai suoi *NSTEP* elementi dei valori equispaziati
- La funzione *ode45* viene poi invocata dopo aver fissato il vettore delle condizioni iniziali *y0*
- Nella chiamata alla funzione *ode45* viene passato quello che nel gergo di MATLAB è un **function handle**, cioè il **puntatore a funzione** *@myfunction2x1* che si ottiene antepoendo il simbolo speciale *@* al nome della funzione che definisce il problema differenziale studiato



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

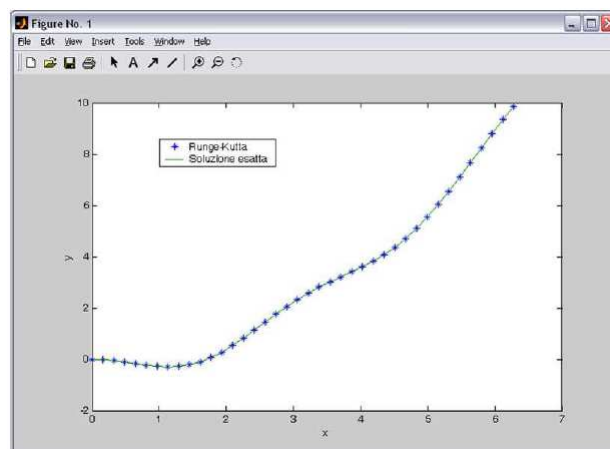
Considerazioni

La funzione `ode45` restituisce un vettore Tt , in questo caso identico a t , ed una matrice Y di $NSTEP$ righe e 2 colonne (la prima è la soluzione, mentre la seconda è la derivata). Ciascuna delle righe rappresenta la soluzione numerica ottenuta in corrispondenza di ciascuno dei valori discreti della variabile indipendente contenuti in Tt . Il motivo per cui viene restituito un vettore come Tt nasce dalla possibilità di invocare funzioni come `ode45` abilitando l'opzione di passo di integrazione variabile in caso di forte variabilità locale della funzione che implementa la derivata



Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

Il grafico della soluzione esatta e della soluzione numerica del problema proposto





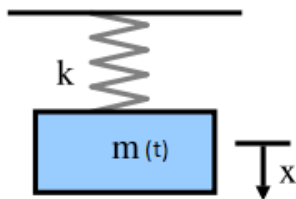
Funzioni MATLAB per la risoluzione di equazioni differenziali ordinarie

Il grafico della soluzione esatta e della soluzione numerica del problema proposto



Massa (variabile)-molla

Si prenda in considerazione un semplice sistema massa molla, la cui particolarità sia quella di avere una massa variabile.



Processando la risposta di tale sistema con l'uso della wavelet, si identificheranno anomalie (spikes) durante il moto oscillatorio.

$$m(t) \frac{d^2 x}{dt^2} = -kx$$

$$x(0) = x_0 \quad \dot{x}(0) = \dot{x}_0$$

La legge di variazione della massa è:

$$m(t) = 1 - a \left[t - b \left(|t - 3.5|^{0.2} + 3.5^{0.2} \right) \right]$$

Quando $0 < a < 1$ e $b = 0$, $m(t)$ risulta una legge lineare decrescente.



Massa (variabile)-molla

Per risolvere la seguente equazione differenziale

$$m(t) \frac{d^2 x}{dt^2} = -kx$$

Si pone

$$\begin{cases} \dot{x}(t) = z(t) = v(t) \\ \dot{z}(t) = -\frac{k}{m(t)} x(t) \end{cases}$$

velocità

Si ha quindi che

$$\dot{x}(t) = \frac{d}{dt} x \approx \frac{x(t+\Delta t) - x(t)}{\Delta t} = v(t)$$



Massa (variabile)-molla

$$x(t+\Delta t) - x(t) = v(t)\Delta t$$

discretizzandola risulta

$$x(k+1) = x(k) + v(k)\Delta t$$

Stesso vale per

$$\dot{z}(t) = -\frac{k}{m(t)} x(t) \quad \text{dove } \dot{z}(t) \text{ è una velocità, per cui}$$

$$\dot{z}(t) = \frac{d}{dt} z \approx \frac{z(t+\Delta t) - z(t)}{\Delta t} = -\frac{k}{m(t)} x(t)$$



Massa (variabile)-molla

da cui si ottiene

$$z(t + \Delta t) - z(t) = -\frac{k}{m(t)} x(t) \Delta t$$

discretizzandola

$$z(k+1) - z(k) = -\frac{k}{m(k)} x(k) \Delta t$$

$$z(k+1) = z(k) - \frac{k}{m(k)} x(k) \Delta t$$



In Matlab

`%Massa variabile`

`close all`

`clear`

`clc`

`passi=100;`

`h=passi;`

`y(1)=0;`

`v(1)=1;`

`for k=1:1000`

`y(k+1)=y(k)+(1/h)*v(k);`

`v(k+1)=v(k)-(1/h)*y(k)/(1-0.1*((k/h)-abs((k/h)-3.5).^0.2+3.5^0.2));`

`contr(k)=(1-0.1*((k/h)-(abs((k/h)-3.5).^0.2)+3.5^0.2));`

`end`



In Matlab

```
figure(1)
plot(contr)
title('Legge di variazione della massa')
```

```
figure(2)
plot(y)
title('Legge del moto')
```

```
u=y;
```

```
ur=real(u);
```



In Matlab

```
w='db13';
[c,l]=wavedec(ur,6,w);
```

```
for i=1:6
    D(i,:)=wrcoef('d',c,l,w,i);
    A(i,:)=wrcoef('a',c,l,w,i);
end
```

```
tt=1+20:length(ur)-20;
```



In Matlab

```
figure(6)
subplot(7,1,1);plot(tt,ur(tt),'r');
title('Coefficienti di dettaglio')

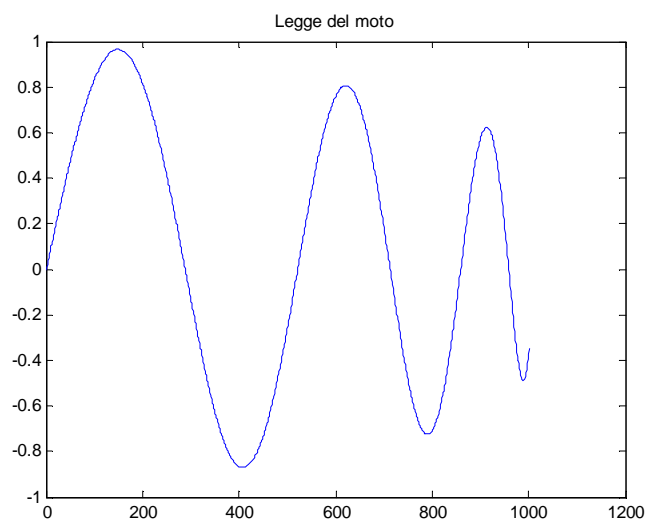
for i=1:6
    subplot(7,1,i+1);
    plot(tt,D(6-i+1,tt),'g');
end

figure(7)
subplot(7,1,1);plot(tt,ur(tt),'r');
title('Coefficienti di approssimazione')

for i=1:6
    subplot(7,1,i+1);
    plot(tt,A(6-i+1,tt),'b');
end
```

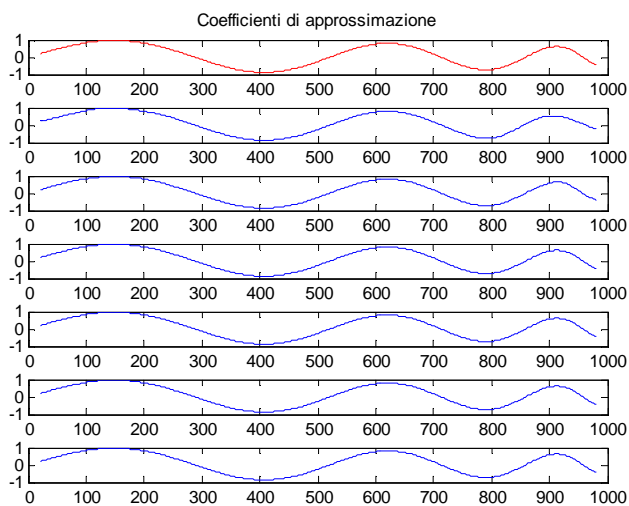


Risultati

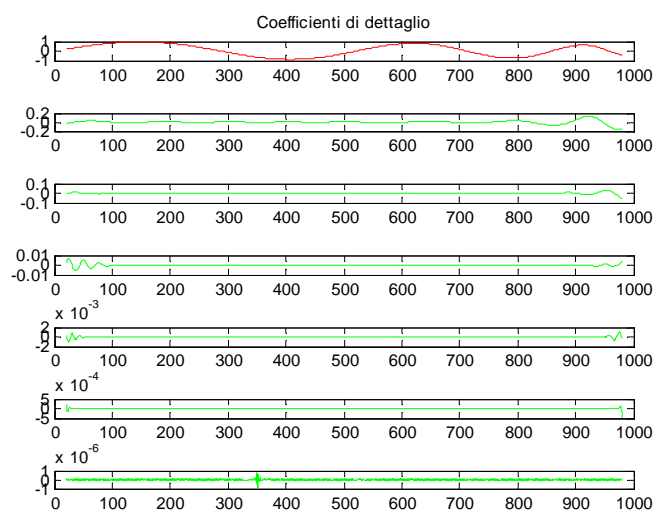




Risultati



Risultati





Risultati

