

Tutoraggio #7

Federico Pichi

5 giugno 2019

Esercizio 1

Implementare in Matlab i metodi di risoluzione multistep per risolvere il problema di Cauchy $y' = f(t, y)$ con dati iniziali u_0 , per t in (t_0, t_f) , passo h e per coefficienti i vettori a e b nei casi di Eulero esplicito, Eulero implicito, Crank-Nicolson, Punto medio e Simpson tale che restituisca in output i valori della soluzione u calcolati nei vari istanti temporali t .

Esercizio 2

Sono assegnati i seguenti problemi di Cauchy, dei quali è riportata la soluzione esatta:

- 1) $y'(t) = \sin(t)(1 + \cos(t) - y(t))$, $y(0) = 3$ SOL: $y(t) = 2 + \cos(t)$
- 2) $y'(t) = (2t + y(t))^2$, $y(0) = 0$ SOL: $y(t) = \sqrt{2} \tan(\sqrt{2}t) - 2t$
- 3) $y'(t) = y(t)(1 - y(t))$, $y(0) = 0.5$ SOL: $y(t) = y(t) = e^t / (1 + e^t)$

Scrivere uno script che utilizzi il codice multistep per risolvere numericamente nell'intervallo temporale $[t_0, t_N] = [0, 1]$, con passo $h = 0.01$ i problemi di Cauchy assegnati con i metodi di Eulero esplicito, Eulero implicito, Crank-Nicolson, Punto medio e Simpson. Per i metodi impliciti fissare tolleranza $1.e - 8$ e numero massimo di iterate 20. Per i metodi a più passi utilizzare il metodo di Heun per calcolare i valori di innesco.

Esercizio 3

Studiare l'help delle functions `ode45` e `ode15s` ed utilizzarli per la soluzione numerica dei problemi di Cauchy elencati al punto precedente, effettuando il plot di soluzione ed errore corrispondente.

Soluzione 1

Codice 1: Soluzione 1

```
1 function [ t,u ] = multistep( a,b,tf,t0,u0,h,fun,dfun,tol,itmax )
2 %Risolva il problema di Cauchy  $y'=f(t,y)$  e dati iniziali  $u_0$ , per  $t$  in
3  $(t_0,tf)$  con passo  $h$  e con il metodo multistep avente per coefficienti  $a$  e  $b$ 
4 %In output restituisce i valori della soluzione calcolati nei vari istanti
5 %temporali

7 f=fun(t0,u0);
8 p=length(a)-1;
9 u=u0;
10 nt=fix((tf-t0(1))/h);

12 for k=1:nt-p
13     lu=length(u);
14     G=a'*u(lu:-1:lu-p)+h*b(2:p+2)'*f(lu:-1:lu-p);
15     t0=[t0;t0(lu)+h];
16     %if b(1)==0
17     % disp('Il metodo e' esplicito');
18     % unew=G; fnew=fun(t0(end),unew);
19     %else
20     % disp('Il metodo e' implicito');
21     unew=u(lu);
22     t=t0(lu+1);
23     it=0;
24     err=tol+1;
25     while err>tol && it<itmax
26         y=unew;
27         den=1-h*b(1)*dfun(t,y);
28         fnew=fun(t,y);
29         if den==0
30             it=itmax+1;
31         else
32             it=it+1;
33             unew=unew-(unew-h*b(1)*fnew-G)/den;
34             err=abs(y-unew);
35         end
36     %end
37 end
38 u=[u;unew];
39 f=[f;fnew];
40 end
41 t=t0;
42 end
```

Soluzione 2

Codice 2: Soluzione 2

```

1 s=menu('Scegli ode da risolvere','D(y(t))=sin(t).*(1+cos(t)-y(t))',...
2       'D(y(t))=(2t+y(t)).^2','D(y(t))=y(t).*(1-y(t))');

4 t0=0;
5 tf=1;
6 h=0.01;
7 tol=1.e-8;
8 itmax=20;

10 switch s
11     case 1
12         fun=inline('sin(t).*(1+cos(t)-y)','t','y');
13         dfun=inline('-sin(t)','t','y');
14         u0=3;

16         %EULERO
17         a=1;
18         b=[0;1];
19         [te,ue]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
20         subplot(1,5,1)
21         plot(te,ue,'r')
22         hold on
23         plot(te,2+cos(te),'b')
24         hold off
25         title('Eulero')
26         ee=norm(ue-2-cos(te),inf);

28         %EULERO IMPLICITO
29         a=1;
30         b=[1;0];
31         [tei,uei]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
32         subplot(1,5,2)
33         plot(tei,uei,'r')
34         hold on
35         plot(tei,2+cos(tei),'b')
36         hold off
37         title('Eulero implicito')
38         eei=norm(uei-2-cos(tei),inf);

40         %CRANK-NICOLSON
41         a=1;
42         b=[0.5;0.5];
43         [tcn,ucn]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
44         subplot(1,5,3)
45         plot(tcn,ucn,'r')
46         hold on
47         plot(tcn,2+cos(tcn),'b')
48         hold off
49         title('Crank-Nicolson')
50         ecn=norm(ucn-2-cos(tcn),inf);

52         %PUNTO MEDIO
53         f0=fun(t0,u0);
54         u1=u0+(h/2)*(f0+fun(h,u0+h*f0));
55         a=[0;1];
56         b=[0;2;0];
57         [tpm,upm]=multistep(a,b,tf,[t0;t0+h],[u0;u1],h,fun,dfun,tol,itmax);
58         subplot(1,5,4)
59         plot(tpm,upm,'r')
60         hold on
61         plot(tpm,2+cos(tpm),'b')
62         hold off
63         title('Punto medio')
64         epm=norm(upm-2-cos(tpm),inf);

```

```

66     %SIMPSON
67     f0=fun(t0,u0);
68     u1=u0+(h/2)*(f0+fun(h,u0+h*f0));    %Dovrei utilizzare RK4
69     a=[0;1];
70     b=[1/3;4/3;1/3];
71     [ts,us]=multistep(a,b,tf,[t0;t0+h],[u0;u1],h,fun,dfun,tol,itmax);
72     subplot(1,5,5)
73     plot(ts,us,'r')
74     hold on
75     plot(ts,2+cos(ts),'b')
76     hold off
77     title('Simpson')
78     es=norm(us-2-cos(ts),inf);

80     figure
81     plot(te,abs(ue-2-cos(te)),tei,abs(uei-2-cos(tei)))
82     title('Grafico del valore assoluto dell''errore di approssimazione')
83     legend('Eulero esplicito','Eulero implicito','Location','NorthEastOutside')

85     figure
86     plot(tcn,abs(ucn-2-cos(tcn)),tpm,abs(upm-2-cos(tpm)),ts,abs(us-2-cos(ts)))
87     title('Grafico del valore assoluto dell''errore di approssimazione')
88     legend('Crank-Nicolson','Punto medio','Simpson','Location','NorthEastOutside')

91     case 2
92     fun=inline('(2*t+y).^2','t','y');
93     dfun=inline('2*(2*t+y)','t','y');
94     u0=0;

96     %EULERO
97     a=1;
98     b=[0;1];
99     [te,ue]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
100    subplot(1,5,1)
101    plot(te,ue,'r')
102    hold on
103    plot(te,sqrt(2)*tan(sqrt(2)*te)-2*te,'b')
104    hold off
105    title('Eulero')
106    ee=norm(ue-(sqrt(2)*tan(sqrt(2)*te)-2*te),inf);

108    %EULERO IMPLICITO
109    a=1;
110    b=[1;0];
111    [tei,uei]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
112    subplot(1,5,2)
113    plot(tei,uei,'r')
114    hold on
115    plot(tei,sqrt(2)*tan(sqrt(2)*tei)-2*tei,'b')
116    hold off
117    title('Eulero implicito')
118    eei=norm(uei-(sqrt(2)*tan(sqrt(2)*tei)-2*tei),inf);

120    %CRANK-NICOLSON
121    a=1;
122    b=[0.5;0.5];
123    [tcn,ucn]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
124    subplot(1,5,3)
125    plot(tcn,ucn,'r')
126    hold on
127    plot(tcn,sqrt(2)*tan(sqrt(2)*tcn)-2*tcn,'b')
128    hold off
129    title('Crank-Nicolson')
130    ecn=norm(ucn-(sqrt(2)*tan(sqrt(2)*tcn)-2*tcn),inf);

132    %PUNTO MEDIO

```

```

133     f0=fun(t0,u0);
134     u1=u0+(h/2)*(f0+fun(h,u0+h*f0));
135     a=[0;1];
136     b=[0;2;0];
137     [tpm,upm]=multistep(a,b,tf,[t0;t0+h],[u0;u1],h,fun,dfun,tol,itmax);
138     subplot(1,5,4)
139     plot(tpm,upm,'r')
140     hold on
141     plot(tpm,sqrt(2)*tan(sqrt(2)*tpm)-2*tpm,'b')
142     hold off
143     title('Punto medio')
144     epm=norm(upm-(sqrt(2)*tan(sqrt(2)*tpm)-2*tpm),inf);

146     %SIMPSON
147     f0=fun(t0,u0);
148     u1=u0+(h/2)*(f0+fun(h,u0+h*f0));
149     a=[0;1];
150     b=[1/3;4/3;1/3];
151     [ts,us]=multistep(a,b,tf,[t0;t0+h],[u0;u1],h,fun,dfun,tol,itmax);
152     subplot(1,5,5)
153     plot(ts,us,'r')
154     hold on
155     plot(ts,sqrt(2)*tan(sqrt(2)*ts)-2*ts,'b')
156     hold off
157     title('Simpson')
158     es=norm(us-(sqrt(2)*tan(sqrt(2)*ts)-2*ts),inf);

160     figure
161     plot(te,abs(ue-2-cos(te)),tei,abs(uei-2-cos(tei)))
162     title('Grafico del valore assoluto dell''errore di approssimazione')
163     legend('Eulero esplicito','Eulero implicito','Location','NorthEastOutside')

165     figure
166     plot(tcn,abs(ucn-2-cos(tcn)),tpm,abs(upm-2-cos(tpm)),ts,abs(us-2-cos(ts)))
167     title('Grafico del valore assoluto dell''errore di approssimazione')
168     legend('Crank-Nicolson','Punto medio','Simpson','Location','NorthEastOutside')

171 case 3
172     fun=inline('y.*(1-y)','t','y');
173     dfun=inline('1-2*y','t','y');
174     u0=0.5;

176     %EULERO
177     a=1;
178     b=[0;1];
179     [te,ue]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
180     subplot(1,5,1)
181     plot(te,ue,'r')
182     hold on
183     plot(te,(exp(te)./(1+exp(te))),'b')
184     hold off
185     title('Eulero')
186     ee=norm(ue-(exp(te)./(1+exp(te))),inf);

188     %EULERO IMPLICITO
189     a=1;
190     b=[1;0];
191     [tei,uei]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
192     subplot(1,5,2)
193     plot(tei,uei,'r')
194     hold on
195     plot(tei,(exp(tei)./(1+exp(tei))),'b')
196     hold off
197     title('Eulero implicito')
198     eei=norm(uei-(exp(tei)./(1+exp(tei))),inf);

200     %CRANK-NICOLSON

```

```

201     a=1;
202     b=[0.5;0.5];
203     [tcn,ucn]=multistep(a,b,tf,t0,u0,h,fun,dfun,tol,itmax);
204     subplot(1,5,3)
205     plot(tcn,ucn,'r')
206     hold on
207     plot(tcn,(exp(tcn)./(1+exp(tcn))), 'b')
208     hold off
209     title('Crank-Nicolson')
210     ecn=norm(ucn-(exp(tcn)./(1+exp(tcn))),inf);

212     %PUNTO MEDIO
213     f0=fun(t0,u0);
214     u1=u0+(h/2)*(f0+fun(h,u0+h*f0));
215     a=[0;1];
216     b=[0;2;0];
217     [tpm,upm]=multistep(a,b,tf,[t0;t0+h],[u0;u1],h,fun,dfun,tol,itmax);
218     subplot(1,5,4)
219     plot(tpm,upm,'r')
220     hold on
221     plot(tpm,(exp(tpm)./(1+exp(tpm))), 'b')
222     hold off
223     title('Punto medio')
224     epm=norm(upm-(exp(tpm)./(1+exp(tpm))),inf);

226     %SIMPSON
227     f0=fun(t0,u0);
228     u1=u0+(h/2)*(f0+fun(h,u0+h*f0));
229     a=[0;1];
230     b=[1/3;4/3;1/3];
231     [ts,us]=multistep(a,b,tf,[t0;t0+h],[u0;u1],h,fun,dfun,tol,itmax);
232     subplot(1,5,5)
233     plot(ts,us,'r')
234     hold on
235     plot(ts,(exp(ts)./(1+exp(ts))), 'b')
236     hold off
237     title('Simpson')
238     es=norm(us-(exp(ts)./(1+exp(ts))),inf);

240     figure
241     plot(te,abs(ue-2-cos(te)),tei,abs(uei-2-cos(tei)))
242     title('Grafico del valore assoluto dell'errore di approssimazione')
243     legend('Eulero esplicito','Eulero implicito','Location','NorthEastOutside')

245     figure
246     plot(tcn,abs(ucn-2-cos(tcn)),tpm,abs(upm-2-cos(tpm)),ts,abs(us-2-cos(ts)))
247     title('Grafico del valore assoluto dell'errore di approssimazione')
248     legend('Crank-Nicolson','Punto medio','Simpson','Location','NorthEastOutside')

250 end

```

Soluzione 3

Codice 3: Soluzione 3

```

1 s=menu('Scegli ode da risolvere','D(y(t))=sin(t).*(1+cos(t)-y(t))',...
2       'D(y(t))=(2t+y(t)).^2','D(y(t))=y(t).*(1-y(t))');
3 t0=0;
4 tf=1;
5 tOf=[t0 tf];
6 h=0.01;
7 tol=1.e-8;
8 itmax=20;

10 switch s
11     case 1
12         fun=inline('sin(t).*(1+cos(t)-y)','t','y');
13         u0=3;
14         [t,y]=ode15s(fun,tOf,u0);
15         [t1,y1]=ode45(fun,tOf,u0);
16         e15=norm(y-(2+cos(t)),inf);
17         e45=norm(y1-(2+cos(t1)),inf);

19         plot(t,y,'r')
20         hold on
21         plot(t,2+cos(t),'b')
22         hold off
23         title('ode15s')

25         figure
26         plot(t1,y1,'r')
27         hold on
28         plot(t1,2+cos(t1),'b')
29         hold off
30         title('ode45')

32         figure
33         plot(t,abs(y-(2+cos(t))), 'm')
34         title('Errore ode15s')

36         figure
37         plot(t1,abs(y1-(2+cos(t1))), 'g')
38         title('Errore ode45')

40     case 2
41         fun=inline('(2*t+y).^2','t','y');
42         u0=0;
43         [t,y]=ode15s(fun,tOf,u0);
44         [t1,y1]=ode45(fun,tOf,u0);
45         e15=norm(y-(sqrt(2)*tan(sqrt(2)*t)-2*t),inf);
46         e45=norm(y1-(sqrt(2)*tan(sqrt(2)*t1)-2*t1),inf);

48         plot(t,y,'r')
49         hold on
50         plot(t,sqrt(2)*tan(sqrt(2)*t)-2*t,'b')
51         hold off
52         title('ode15s')

54         figure
55         plot(t1,y1,'r')
56         hold on
57         plot(t1,sqrt(2)*tan(sqrt(2)*t1)-2*t1,'b')
58         hold off
59         title('ode45')

61         figure
62         plot(t,abs(y-(sqrt(2)*tan(sqrt(2)*t)-2*t)), 'm')
63         title('Errore ode15s')

```

```
65     figure
66     plot(t1,abs(y1-(sqrt(2)*tan(sqrt(2)*t1)-2*t1)), 'g')
67     title('Errore ode45')

69     case 3
70     fun=inline('y.*(1-y)', 't', 'y');
71     u0=0.5;
72     [t,y]=ode15s(fun,tOf,u0);
73     [t1,y1]=ode45(fun,tOf,u0);
74     e15=norm(y-(exp(t)./(1+exp(t))),inf);
75     e45=norm(y1-(exp(t1)./(1+exp(t1))),inf);

77     plot(t,y, 'r')
78     hold on
79     plot(t,(exp(t)./(1+exp(t))), 'b')
80     hold off
81     title('ode15s')

83     figure
84     plot(t1,y1, 'r')
85     hold on
86     plot(t1,(exp(t1)./(1+exp(t1))), 'b')
87     hold off
88     title('ode45')

90     figure
91     plot(t,abs(y-(exp(t)./(1+exp(t)))), 'm')
92     title('Errore ode15s')
93     figure
94     plot(t1,abs(y1-(exp(t1)./(1+exp(t1)))), 'g')
95     title('Errore ode45')

97 end
```