Esercizio

- Scrivere una funzione di Matlab per approssimare la soluzione del problema di Cauchy

$$\begin{cases} y'(t) = f(t, y(t)) & t \in [t_0, t_0 + T] \\ y(t_0) = y_0 \end{cases}$$

usando il metodo di Eulero esplicito.

$$\begin{cases} u_{i+1} = u_i + hf(t_i, u_i) & i = 0, 1, \dots, n-1 \\ u_0 = y_0 \end{cases}$$

dove $h = T/n$, $t_i = t_0 + ih$ per $i = 0, \dots, n$ e $u_i \approx y(t_i)$.

Euler

```
function [t,u]=euler(f,t0,y0,T,n)
h=T/n;
t=[t0:h:t0+T];
u(1)=y0;
for i=1:n
    K=feval(f,t(i),u(i));
    u(i+1)=u(i)+h*K;
end
```

Verifica dell'ordine di convergenza

Sia

$$e(h) := \max_{i=0,\dots,n} |y(t_i) - u_i| \, .$$

Se $e(h) = O(h^p)$ allora

$$e(h/2) \approx \frac{e(h)}{2^p}$$

quindi

$$2^p \approx \frac{e(h)}{e(h/2)}$$

e

$$p \approx \frac{\log[e(h)] - log[e(h/2)]}{\log 2} \, .$$

Esercizio

Scrivere uno script di Matlab che

- approssimi la soluzione del problema di Cauchy

$$\begin{cases} y' = -t^2 y & t \in [0,1] \\ y(0) = 1 \end{cases}$$

  usando il metodo di Taylor di secondo ordine;
- verifichi esperimentalmente l'ordine di convergenza del metodo;
- faccia il grafico della soluzione approssimata e della soluzione esatta $y(t) = \exp(-t^3/3)$.

Soluzione

```
clear
n=input('Numero sottointervalli: ');
h=1/n;
t=[0:h:1];
u(1)=1;
for i=1:n
    u(i+1)=u(i)-h*t(i)^2*u(i)+h^2/2*u(i)*(t(i)^4-2*t(i));
end
err1=max(abs(u-exp(-t.^3/3)));
h=h/2;
t=[0:h:1];
u(1)=1;
for i=1:2*n
    u(i+1)=u(i)-h*t(i)^2*u(i)+h^2/2*u(i)*(t(i)^4-2*t(i));
end
err2=max(abs(u-exp(-t.^3/3)));
papp=(log(err1)-log(err2))/log(2)
tt=linspace(0,1);
y=exp(-tt.^3/3);
plot(t,u,tt,y)
legend('App','Esatta')
```

Esercizio

- Scrivere una funzione di Matlab per approssimare la soluzione del problema di Cauchy

$$\begin{cases} y' = -t^2 y \quad t \in [0,1] \\ y(0) = 1 \end{cases}$$

usando il metodo di Crank-Nicolson.

- Scrivere una script per verificare esperimentalmente l'ordine di convergenza del metodo e fare il grafico della soluzione approssimata e della soluzione esatta.

Soluzione

```
function [t,u]=eserCN(n)
t0=0;
y0=1;
T=1;
h=T/n;
t=[t0:h:t0+T];
u(1)=y0;
for i=1:n
    u(i+1)=(1-h/2*t(i)^2)/(1+h/2*t(i+1)^2)*u(i);
end
```

```
n=input('Numero di sottointervalli: ');
[t,u]=eserCN(n);
y=exp(-t.^3/3);
err1=max(abs(y-u));
[t,u]=eserCN(2*n);
y=exp(-t.^3/3);
err2=max(abs(y-u));
papp=(log(err1)-log(err2))/log(2)
tt=linspace(0,1);
y=exp(-tt.^3/3);
plot(t,u,tt,y)
legend('App','Esatta')
```

Esercizio

- Scrivere una funzione di Matlab per approssimare la soluzione del problema di Cauchy

$$\begin{cases} y'(t) = f(t, y(t)) & t \in [t_0, t_0 + T] \\ y(t_0) = y_0 \end{cases}$$

usando il metodo di Heun:

$$\begin{cases} K_1 = f(t_i, u_i)\,, \quad K_2 = f(t_i + h, u_i + hK_1) \\ u_{i+1} = u_i + \dfrac{h}{2}(K_1 + K_2) \quad i = 0, 1, \ldots, n-1 \\ u_0 = y_0 \end{cases}$$

dove $h = T/n$ e $t_i = t_0 + ih$ per $i = 0, \ldots, n$.

Heun

```
function [t,u]=heun(f,t0,y0,T,n)
h=T/n;
t=[t0:h:t0+T];
u(1)=y0;
for i=1:n
    K1=feval(f,t(i),u(i));
    K2=feval(f,t(i)+h,u(i)+h*K1);
    u(i+1)=u(i)+h/2*(K1+K2);
end
```

Esercizio

- Scrivere una funzione di Matlab per approssimare la soluzione del problema di Cauchy

$$\begin{cases} y'(t) = f(t, y(t)) & t \in [t_0, t_0 + T] \\ y(t_0) = y_0 \end{cases}$$

usando il metodo di Adams-Bashforth a quattro passi:

$$\begin{cases} u_{i+1} = u_i + \dfrac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) & i = 3, \dots, n-1 \\ u_0 = y_0 \end{cases}$$

dove $h = T/n$ e $t_i = t_0 + ih$ per $i = 0, \dots, n$.

- Per calcolare u_1, u_2 e u_3 usare
  - il metodo di Eulero;
  - il metodo di Heun;
  - il metodo di Runge-Kutta 4.
- Stimare per i tre casi l'ordine di convergenza del metodo di Adams-Bashforth a quattro passi.

Adams-Bashforth 4 passi

```
function [t,u]=AB4(fun,t0,y0,T,n)
h=T/n;
t=[t0:h:t0+T];
u(1)=y0;
for i=1:3
    f(i)=feval(fun,t(i),u(i));
    K1=f(i);
    K2=feval(fun,t(i)+h/2,u(i)+h/2*K1);
    K3=feval(fun,t(i)+h/2,u(i)+h/2*K2);
    K4=feval(fun,t(i)+h,u(i)+h*K3);
    u(i+1)=u(i)+h/6*(K1+2*K2+2*K3+K4);
end
for i=4:n
    f(i)=feval(fun,t(i),u(i));
    u(i+1)=u(i)+h/24*(55*f(i)-59*f(i-1)+37*f(i-2)-9*f(i-3));
end
```