

Esercizio 1

Dobbiamo innanzitutto capire a quale equazione obbedisce il numero di batteri della colonia. Dal fatto che in tempi uguali la soluzione aumenti dello stesso fattore in tempi uguali deduciamo che la legge che regola la crescita dei batteri è esponenziale, cioè detta y la quantità di batteri dipendente dal tempo t espresso in ore vale

$$y(t) = K \cdot a^t$$

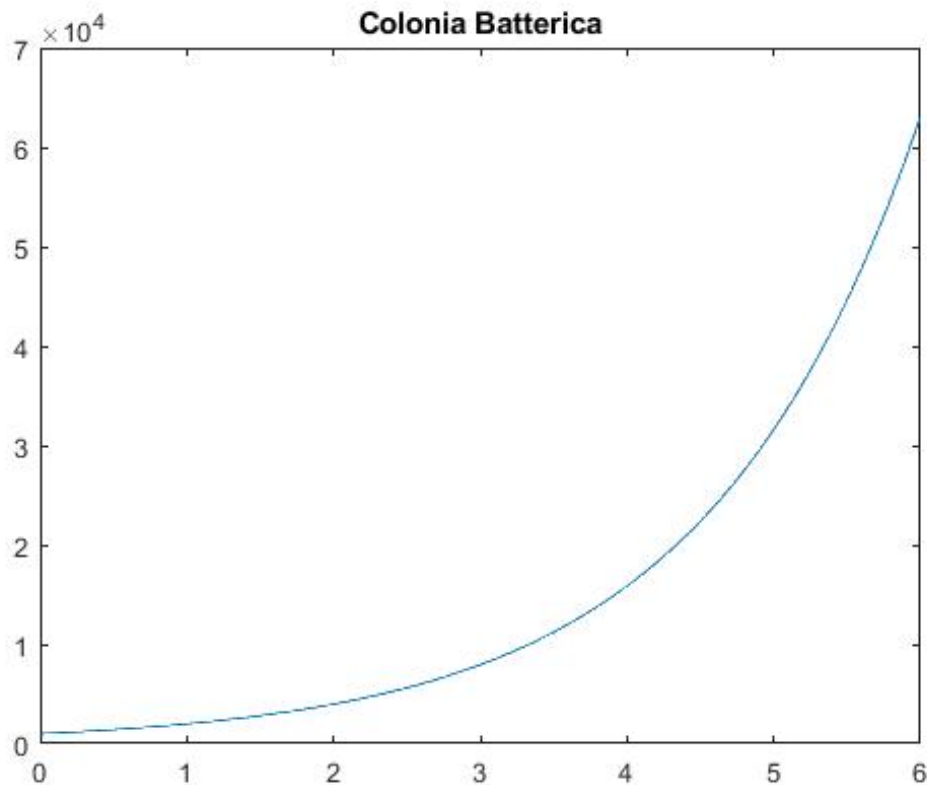
dal fatto che ogni ora il tempo raddoppi abbiamo che $a = 2$, e dal fatto che all'inizio i batteri siano 1000 abbiamo $K = 1000$ (anche se quest'ultima informazione non cambia l'equazione differenziale del problema). Ne segue che $y(t) = 1000 \cdot 2^t$, che risolve l'equazione differenziale

$$y'(t) = \log 2 \cdot y$$

per cui la function che risolve il problema col metodo di Eulero in funzione di un generico valore iniziale (nel nostro caso 1000) e del passo h (che settiamo a 0.01) sarà:

```
1 function []=coloniabatteri(y0,h)
2 %y0=valore iniziale, h passo di integrazione
3 f=@(x,y) log(2)*y;
4 [x,u]=eulero(f,[0,6],y0,h);
5 plot(x,u)
6 title('Colonia Batterica');
7 end
```

Il grafico di tale soluzione numerica nelle prime sei ore è il seguente:

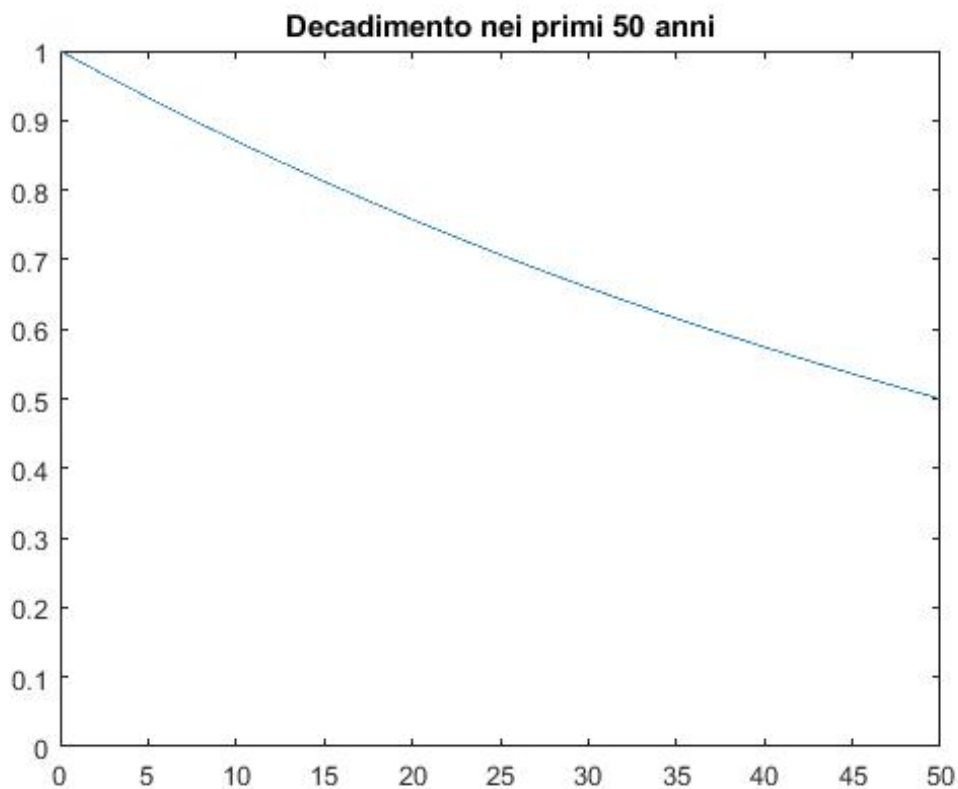


Esercizio 2

La funzione `plutonio` seguente restituisce il tempo necessario ad avere, come richiesto dal testo, una riduzione di $\frac{1}{10}$ della quantità di plutonio iniziale (per farla abbiamo usato la funzione `find`) in funzione del passo h del metodo di Runge-Kutta (che setteremo a 0.01). Il grafico è quello, sempre come da testo, nel decadimento nei primi 50 anni.

```
1 function k=plutonio(h)
2 %h e' il passo di integrazione
3 fun=@(x,y) -1/50*log(2)*y;
4 tspan=[0,200];
5 y0=1;
6 [t,u]=RK4(fun,tspan,y0,h);
7 l=find(u<0.1,1);
8 k=t(l);
9 m=find(t>50,1);
10 plot(t(1:m),u(1:m))
11 title('Decadimento nei primi 50 anni')
12 axis([0 50 0 1])
13 end
```

Il tempo richiesto alla divisione per 10 è 166.1 anni. Il grafico nel decadimento nei primi 50 anni è il seguente:

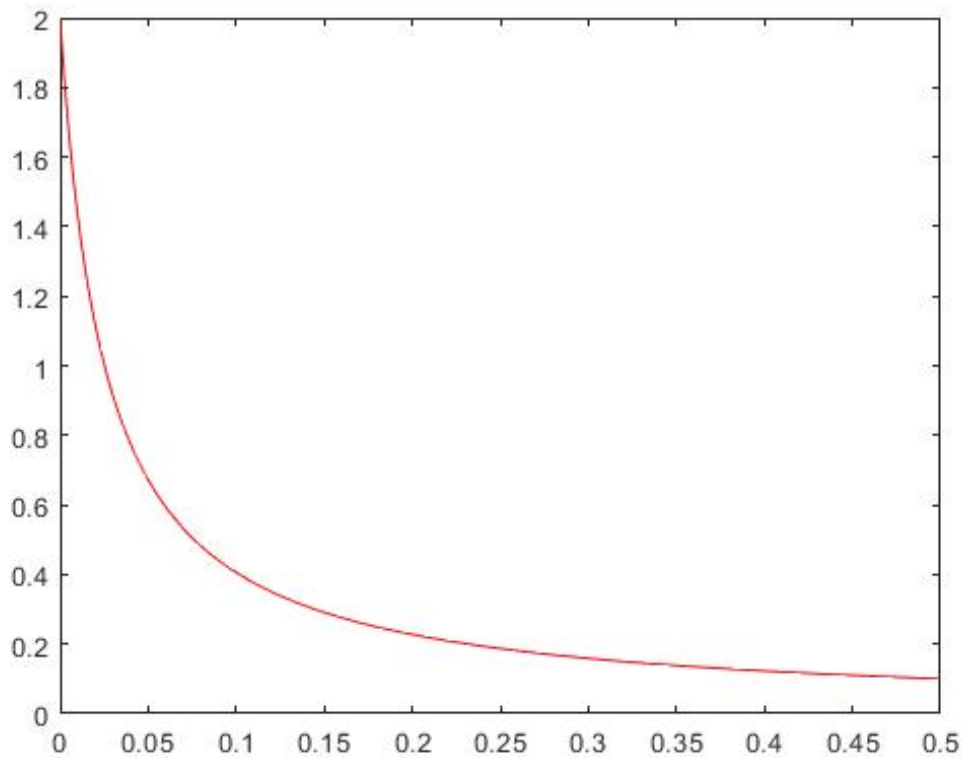


Esercizio 3

Lo script seguente applica la routine `ode45` all'equazione logistica con i parametri assegnati.

```
1 %la popolazione diventa un decimo di quella iniziale a t~0.23
2 a=0.2;
3 K=0.01;
4 b=a/K;
5 y0=2;
6 f=@(x,y) y*(a-b*y);
7 [t,u] = ode45(f,[0 0.5],y0);
8 plot(t,u,'r');
```

Dal disegno ricaviamo che intorno a $t = 0.23$ la popolazione sarà diventata un decimo di quella iniziale. Il grafico richiesto è il seguente:



Esercizio 4

Innanzitutto scriviamo una funzione che implementi l'equazione logistica con il parametro mathusiano α dipendente dal tempo, e che la risolva mediante la routine `ode45`:

```
1 function [x,y]=eqlogistica2(y0)
2 f=@(t,y) y*(1/2+cos(2*pi*t))*(1-y/100);
3 [x,y] = ode45(f,[0 20],y0);
4 end
```

Fatto ciò, scriviamo uno script che applichi tale funzione con il parametro iniziale settato a 1, 10, 50, 200 e che faccia i quattro grafici richiesti:

```
1 [x1,y1]=eqlogistica2(1);
2 [x2,y2]=eqlogistica2(10);
3 [x3,y3]=eqlogistica2(50);
4 [x4,y4]=eqlogistica2(200);
5 subplot(2,2,1)
6 plot(x1,y1)
7 title('y0=1')
8
9 subplot(2,2,2)
10 plot(x2,y2)
11 title('y0=10')
12
13 subplot(2,2,3)
14 plot(x3,y3)
15 title('y0=50')
16
17 subplot(2,2,4)
18 plot(x4,y4)
19 title('y0=200')
```

In tutti e quattro i casi la soluzione tende alla retta $y = 100$, ovviamente in tre casi crescendo e in uno decrescendo.

