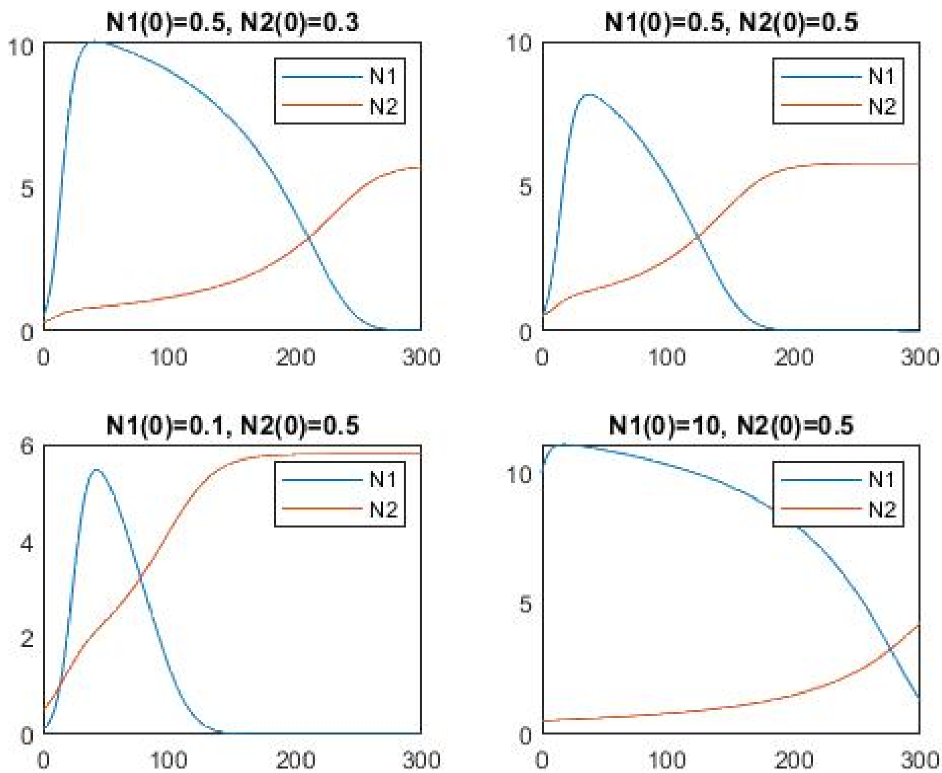
Esercizio 1

Scriviamo uno script che risolva con `ode45` e disegni il grafico dell'equazione del testo in quattro casi diversi a seconda dei parametri utilizzati. Utilizziamo ovviamente i parametri del testo anche per i parametri dell'equazione e per l'intervallo di integrazione.

```
f=@(t,n) [0.21827*(1-n(1)/13-n(2)/3.71429)*n(1); 0.06069*(1-n(2)/5.8-n(1)/13.2118)*n(2)];
[t1,sol1]=ode45(f,[0 300], [0.5; 0.3]);
[t2,sol2]=ode45(f,[0 300], [0.5; 0.5]);
[t3,sol3]=ode45(f,[0 300], [0.1; 0.5]);
[t4,sol4]=ode45(f,[0 300], [10; 0.5]);
subplot(2,2,1)
plot(t1,sol1(:,1),t1,sol1(:,2))
legend('N1','N2')
title('N1(0)=0.5, N2(0)=0.3')
subplot(2,2,2)
plot(t2,sol2(:,1),t2,sol2(:,2))
legend('N1','N2')
title('N1(0)=0.5, N2(0)=0.5')
subplot(2,2,3)
plot(t3,sol3(:,1),t3,sol3(:,2))
legend('N1','N2')
title('N1(0)=0.1, N2(0)=0.5')
subplot(2,2,4)
plot(t4,sol4(:,1),t4,sol4(:,2))
legend('N1','N2')
title('N1(0)=10, N2(0)=0.5')
```



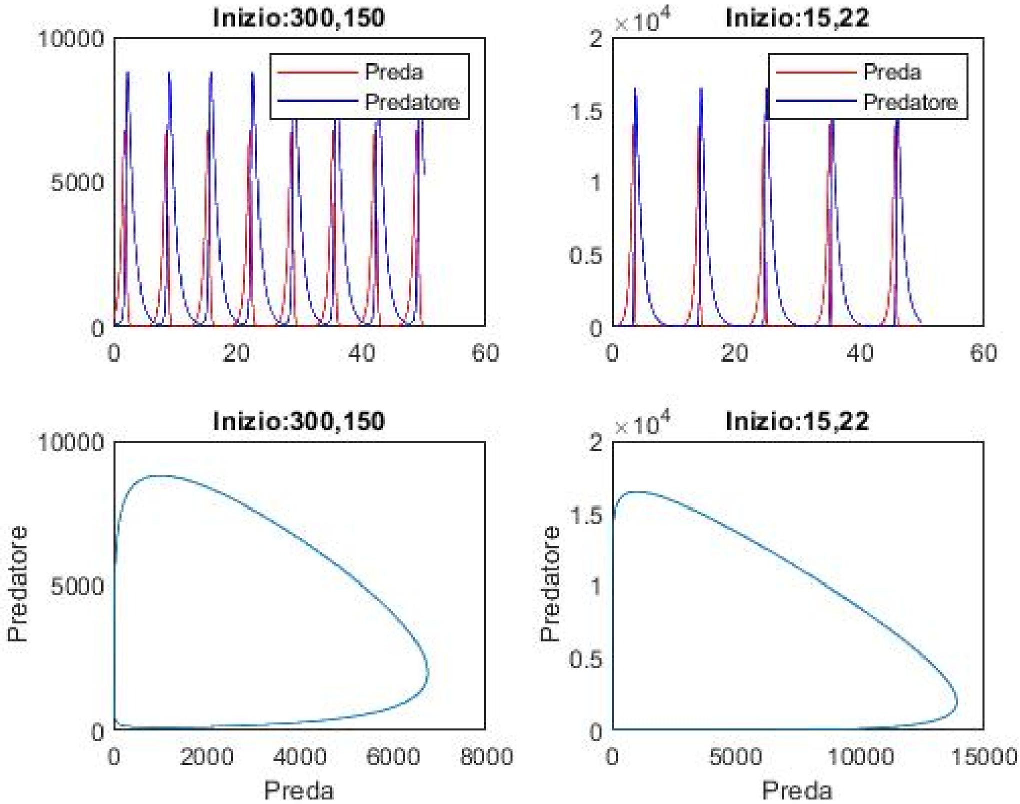
Notiamo che $N2$ tende a 5.8, e che $N1$ tende a zero in tutti e quattro i casi.

Esercizio 2

La seguente function implementa il modello di Lotka-Volterra in funzione di due parametri t_{max} e del passo h del metodo di Runge-Kutta usato nella risoluzione dello stesso.

```
function []=lotkavolterra(tmax,h)
alfa=2;
gamma=0.001;
alfa1=1;
beta1=0.001;
fun=@(t,x) [x(1)*(alfa-gamma*x(2)); x(2)*(-alfa1+beta1*x(1))];
[t1,u1]=RK4(fun,[0:tmax],[300;150],h);
[t2,u2]=RK4(fun,[0:tmax],[15;22],h);

subplot(2,2,1)
plot(t1,u1(1,:),'r',t1,u1(2,:),'b');
legend('Preda','Predatore');
title('Inizio:300,150');

subplot(2,2,2)
plot(t2,u2(1,:),'r',t2,u2(2,:),'b');
legend('Preda','Predatore');
title('Inizio:15,22');

subplot(2,2,3)
plot(u1(1,:),u1(2,:))
xlabel('Preda')
ylabel('Predatore')
title('Inizio:300,150');

subplot(2,2,4)
plot(u2(1,:),u2(2,:))
xlabel('Preda')
ylabel('Predatore')
title('Inizio:15,22');
```

Abbiamo settato $t_{max} = 50$ espresso in ore e, come ormai standard, $h = 0.01$. Osserviamo il simpatico fenomeno per cui i predatori consumano interamente la loro scorta di cibo per poi estinguersi, far ri-aumentare il numero di prede e riconsumarli tutti nuovamente. Questo comportamento è evidente anche dai due grafici inferiori che descrivono le prede in funzione dei predatori, che sono chiaramente ciclici.

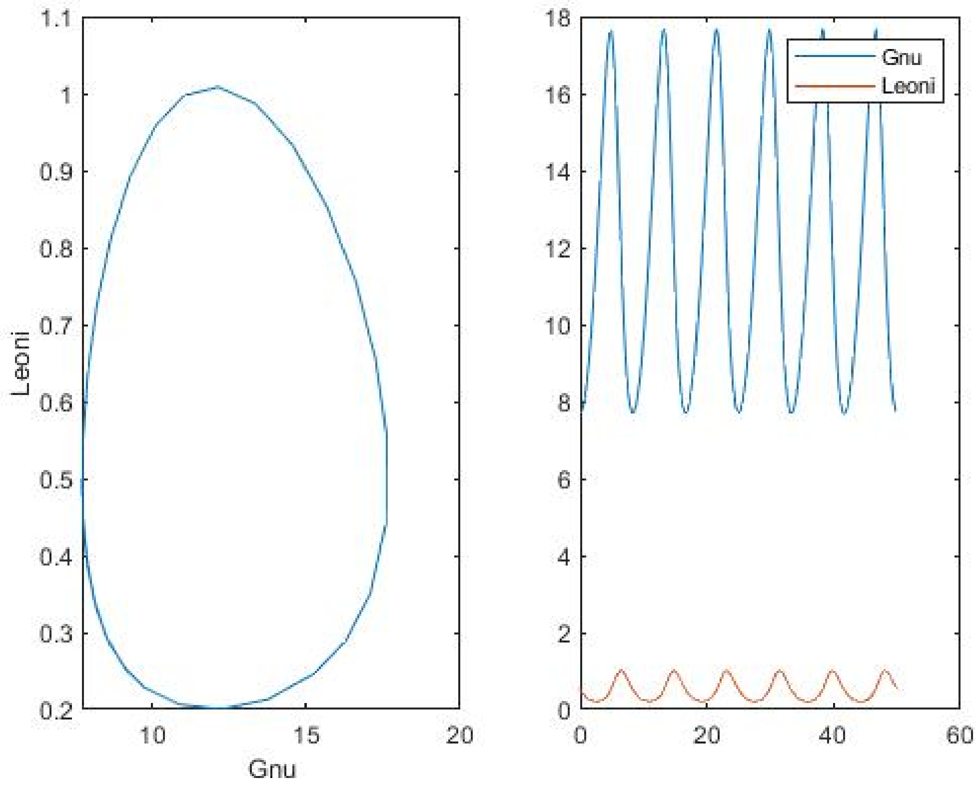

Inizio:300,150
Preda
Predatore
Inizio:15,22
$\times 10^4$
Preda
Predatore
Inizio:300,150
Predatore
Preda
Inizio:15,22
$\times 10^4$
Predatore
Preda

Esercizio 3

Esattamente come nella funzione precedente, anche qui implementiamo il modello di Lotka-Volterra, ma stavolta lo risolviamo con `ode45`. Abbiamo usato il comando `find` per risolvere un solo problema differenziale e limitarci a ridurre l'intervallo di integrazione a manina, invece di far partire un'altra routine di `ode45`.

```
f=@(t,x) [0.405*x(1)-0.81*x(1)*x(2); -1.5*x(2)+0.125*x(1)*x(2)];
[t,sol]=ode45(f,[0 25], [7.7; 0.5]);
T=find(t>10,1);
subplot(1,2,1)
plot(sol(1:T,1),sol(1:T,2));
xlabel('Gnu')
ylabel('Leoni')
subplot(1,2,2)
plot(t,sol(:,1),t,sol(:,2));
legend('Gnu','Leoni')
```

Le considerazioni qualitative sono molto simili a quelle dell'esercizio precedente, non fosse che la popolazione delle prede non si azzera mai, ma tende nel momento di massima proliferazione della popolazione di predatori al valore 8. Per il resto, il comportamento è ancora oscillatorio.



Esercizio 4

L'equazione differenziale che descrive il fenomeno è $x(t)' = rx(t)y(t), y(t)' = -rx(t)y(t)$, dove $x(t)$ sono gli infetti e $y(t)$ sono i malati. La malattia non e' mortale, ma tende a cronicizzarsi: detto r il tasso di infezione, la function seguente restituisce il tempo necessario ad avere il numero di infetti pari ai 4/5 della popolazione in funzione di N numero di individui e i_0 numero di infetti iniziali.

```
function y=malattia(i0,r,N,T)
f=@(t,x) [r*x(1)*x(2);-r*x(1)*x(2)];
[t,sol]=ode45(f,[0 T], [i0; N-i0]);
plot(t,sol(:,1),'r',t,sol(:,2),'g');
legend('Infetti','Malati');
l=find(sol(:,1)>0.8*N,1);
y=t(l);
end
```

Abbiamo posto, come da testo, $i_0 = 1$, $N = 1000$ e $r = 0.0001$. Inoltre abbiamo integrato con $T = 200$. Il risultato ottenuto è 83.3536 ore, e il grafico è il seguente: