praticata, il parametro ε è positivo e quindi si ha un numero di prede maggiore e un numero di predatori minore rispetto al caso senza pesca.

4 Risoluzione del modello di Lotka-Volterra

Il sistema di Lotka-Volterra (3) di equazioni differenziali, sebbene di due sole equazioni in due incognite e quindi apparentemente molto semplice, non può in realtà essere risolto con i metodi analitici che di solito vengono insegnati nei corsi di analisi dei primi anni di università.

Volterra ideò un procedimento analitico-grafico [12] che gli permetteva di risolvere le equazioni in un modo piuttosto laborioso, seppur di indubbia valenza didattica (si veda pag. 355-357 di [5] per una descrizione dettagliata del metodo). Si trattava ovviamente di un metodo che necessitava solo di calcoli eseguibili manualmente, da momento che a quel tempo non esistevano ancora macchine calcolatrici automatiche.

Grazie allo sviluppo dei calcolatori, è oggi possibile risolvere il problema per via numerica, cioè grazie ad algoritmi di calcolo implementati con linguaggi di programmazione. Si può procedere, ad esempio, ad una implementazione alle differenze finite o mediante il metodo di Runge-Kutta (implicito) o ancora con metodi espliciti che approssimino il valore della derivata mediante lo sviluppo in serie. Esistono a questo riguardo diversi strumenti a disposizione sia dei ricercatori che degli studenti a seconda che si affronti il problema per ragioni di studio scientifico o a scopo didattico, per esempio:

- **il foglio di calcolo**[4]: Uno dei punti di forza del foglio di calcolo è che, una volta impostato, è sufficiente cambiare i dati in ingresso per ottenere in maniera istantanea i risultati. Le formule contenute in ogni cella aggiornano, infatti, in tempo reale i valori ottenuti. Tali risultati possono poi essere già organizzati in una pagina facilmente stampabile in modo che, nel caso di analisi di dati ripetute sempre nel medesimo modo, si debba semplicemente inserire i dati di ingresso per avere un rapporto di misura.

[4] I fogli di calcolo sono ampiamente utilizzati in tutti i settori del calcolo scientifico in quanto risultano di facile utilizzo. L'analisi dei dati e la loro visualizzazione grafica sono, infatti, operazioni immediate. Esistono sul mercato molti fogli di calcolo, tutti molto simili per quanto riguarda il loro utilizzo e le loro potenzialità. Segnaliamo qui il foglio di calcolo Calc incluso nella distribuzione del pacchetto OpenOffice. Il sito ufficiale del pacchetto OpenOffice in lingua italiana è il seguente: `http://it.openoffice.org/`. Dal sito è possibile scaricare il pacchetto per i principali sistemi operativi e nella lingua desiderata. Si tratta di un foglio di calcolo con caratteristiche simili a foglio di calcolo Excel del pacchetto Office di Microsoft. A differenza di Office, OpenOffice è però distribuito sotto i termini della licenza GNU Lesser General Public License (GNU LGPL). È quindi possibile scaricare ed utilizzare gratuitamente tale programma.

Questa caratteristica positiva può diventare però anche uno svantaggio qualora sia necessaria una maggiore duttilità per analizzare dei dati per la prima volta. Anche in questo caso è spesso più comodo e veloce utilizzare Octave per il quale è anche possibile trovare librerie di funzioni molto specifiche o scriversene ad hoc.;

- **Matlab o Octave**[5]: in questo caso si può utilizzare direttamente la funzione ODE45 basata sul metodo di Runge-Kutta oppure creare un m-file con un programma basato su cicli FOR che approssimi il valore della derivata mediante lo sviluppo in serie;

[5]Il sito ufficale del software Octave è il seguente: www.octave.org. Qui è possibile scaricare il programma, il manuale, accedere ai vari pacchetti di funzioni e al forum di discussione.

Si tratta in realtà di un vero e proprio ambiente di calcolo, simile per struttura e modalità di funzionamento al più noto programma commerciale MATLAB, tanto che alcuni lo considerano un suo clone. A differenza di MATLAB, però, Octave è distribuito sotto i termini della licenza GNU General Public License (GNU GPL). È quindi possibile utilizzare gratuitamente tale programma, cosa che favorisce molto un investimento di tempo per imparare il suo utilizzo in quanto non comporta l'acquisto di alcuna licenza: è sufficiente scaricare il programma ed installarlo su ogni macchina. Esistono versioni per i principali sistemi operativi (Linux, Mac OS X, Windows e OS/2). Nel seguito della trattazione ci riferiremo quindi ad Octave, ma per quanto detto è possibile riportare tutto anche in ambiente MATLAB. Non ci soffermeremo qui sulle enormi potenzialità che Octave offre sia per il calcolo che per la visualizzazione grafica. Per questo rimandiamo sia al manuale che viene installato insieme al programma, sia alla vastissima documentazione che è possibile trovare in Internet. Data la forte somiglianza con MATLAB, è molto spesso possibile consultare manuali e utilizzare risorse scritte originariamente per MATLAB (e viceversa). Richiamiamo qui solo il fatto che Octave può funzionare sia fornendogli una riga di codice alla volta nella finestra di comando, sia scrivendo un vero e proprio programma in un file con estensione .m (ad esempio nomefile.m) che potrà poi essere richiamata dal programma semplicemente scrivendo nomefile nella finestra di comando. L'editore distribuito con la versione Windows di Octave (attivabile scrivendo edit sulla riga di comando) facilita la scrittura del programma allineando opportunamente i comandi e utilizzando colori diversi a seconda che si stia digitando un comando, una variabile, un testo etc. Il linguaggio di programmazione è estremamente semplice da apprendere per chi abbia già programmato con un qualunque altro linguaggio ed è agevolato dal fatto che le variabili non devono essere dichiarate poiché Octave è in grado di riconoscerne il tipo al loro primo uso ed utilizza una allocazione dinamica della memoria. I risultati dei calcoli sono in prima battuta restituiti sullo schermo, ma è possibile averli anche su file in formato testo. È così poi possibile importarli in un foglio di calcolo per una ulteriore elaborazione. È possibile anche scrivere sottoprogrammi che vengano poi richiamati dal programma principale, sia in forma di file con estensione .m che vengono eseguiti come da riga di comando, sia sotto forma di funzioni. Queste ultime hanno il vantaggio di utilizzare solo le variabili fornite in ingresso e di restituire solo quelle richieste in uscita mentre le variabili utilizzate all'interno della funzione vengono poi dimenticate. Esistono una molteplicità di "package", cioè di librerie di funzioni, scritte da vari utenti su un argomento specifico e che possono essere scaricati gratuitamente dal sito web ufficiale o da quelli personali dei vari utenti.

- **R**[6]: non lo utilizzeremo in questo articolo, ma un ottimo esempio del suo impiego nella risoluzione delle equazioni di Lotka-Volterra utilizzando i dati sulle popolazioni di linci e di lepri in Canada si può trovare in [7].

Per poter implementare il modello descritto dal sistema (3) è necessario, come passo preliminare, discretizzare le equazioni in gioco, cioè passare dal dominio continuo del tempo ad un dominio discreto [1].

È anche necessario discretizzare le equazioni in modo che risultino di tipo esplicito, cioè il valore della funzione ad un certo istante deve essere funzione dei soli valori ai passi precedenti.

Come discretizzazione di prima approssimazione delle equazioni del sistema (3), si consideri l'approssimazione del primo ordine della derivata data da:

$$x(t+\Delta t) = x(t) + \frac{dx(t)}{dt}\Delta t \tag{6}$$

che, ricordando che $x(t)$ può essere indicato semplicemente con x e $\frac{dx(t)}{dt}$ indicato con $\dot{x}$, la (6) può essere riscritta:

$$x(t+\Delta t) = x + \dot{x}\Delta t \tag{7}$$

Analogamente per la popolazione dei predatori:

$$y(t+\Delta t) = y + \dot{y}\Delta t \tag{8}$$

Andando a sostituire $\dot{x}$ e $\dot{y}$ dalle (3) nelle (7) e (8) si ottiene:

$$\begin{cases} x(t+\Delta t) = (a-by)\,x\,\Delta t + x & \text{(9a)} \\ y(t+\Delta t) = (-c+dx)\,y\,\Delta t + y & \text{(9b)} \end{cases}$$

Tale sistema si risolve numericamente mediante un algoritmo che calcoli il valore delle funzioni x e y all'istante $(t+\Delta t)$ utilizzando i loro valori all'istante precedente (t). Indicando quindi con $\dot{y}_n$ e $\ddot{y}_n$ i valori delle variabili al passo n, il sistema (9) può essere riscritto come:

$$\begin{cases} x_{n+1} = (a-by_n)\,x_n\,\Delta t + x_n & \text{(10a)} \\ y_{n+1} = (-c+dx_n)\,y_n\,\Delta t + y_n & \text{(10b)} \end{cases}$$

[6]Si tratta di un programma sviluppato appositamente per trattare grandi quantità di dati statistici. Anch'esso, come Octave è distribuito sotto i termini della licenza GNU General Public License (GNU GPL) e disponibile per tutti i tipi di piattaforme. Il sito ufficiale del software R è il seguente: www.r-project.org. Il sito è completamente in inglese, ma esistono ottimi manuali in italiano reperibili sia su Internet che in libreria.

Nal caso in cui si voglia utilizzare un Δt più grande (e quindi diminuire i tempi di calcolo) o aumentare la precisione, è necessario utilizzare una approssimazione del secondo ordine della derivata. Si ha dunque per la popolazione delle prede e per quella dei predatori:

$$\begin{cases} x(t+\Delta t) = x(t) + \dfrac{dx(t)}{dt}\Delta t + \dfrac{1}{2}\dfrac{d^2x(t)}{dt^2}\Delta t^2 & \text{(11a)} \\ \\ y(t+\Delta t) = y(t) + \dfrac{dy(t)}{dt}\Delta t + \dfrac{1}{2}\dfrac{d^2y(t)}{dt^2}\Delta t^2 & \text{(11b)} \end{cases}$$

che, per comodità, riscriveremo:

$$\begin{cases} x(t+\Delta t) = \dot{x}\Delta t + \dfrac{1}{2}\ddot{x}\Delta t^2 & \text{(12a)} \\ \\ y(t+\Delta t) = \dot{y}\Delta t + \dfrac{1}{2}\ddot{y}\Delta t^2 & \text{(12b)} \end{cases}$$

Indicando quindi con $\dot{x}_n$, $\ddot{x}_n$ i valori delle derivate seconde rispetto al tempo al passo n, il sistema (11) diventa:

$$\begin{cases} x_{n+1} = x_n + \dot{x}_n\Delta t + \dfrac{1}{2}\ddot{x}_n\Delta t^2 & \text{(13a)} \\ \\ y_{n+1} = y_n + \dot{y}_n\Delta t + \dfrac{1}{2}\ddot{y}_n\Delta t^2 & \text{(13b)} \end{cases}$$

dove x_n e y_n si ricavano direttamente dalle (3a) e (3b):

$$\begin{cases} \dot{x}_n = ax_n - bx_n y_n & \text{(14a)} \\ \dot{y}_n = -cy_n + dx_n y_n & \text{(14b)} \end{cases}$$

e da cui, derivando, si ottengono:

$$\begin{cases} \ddot{x}_n = a\dot{x}_n - b\left(\dot{x}_n y_n + x_n \dot{y}_n\right) & \text{(15a)} \\ \\ \ddot{y}_n = -c\dot{y}_n + d\left(\dot{x}_n y_n + x_n \dot{y}_n\right) & \text{(15b)} \end{cases}$$

Per ottenere il sistema in una formula risolubile direttamente, le (14) andrebbero sostituite sia nelle (13) che nelle (15). Le (15) così ottenute andrebbero poi sostituite nelle (13). Dal punto di vista operativo, però, abbiamo preferito tenerle

separate ed eseguire i calcoli parziali di volta in volta piuttosto che ottenere una formula che a colpo d'occhio sia quasi illeggibile.

4.1 Effetto della pesca

Nel caso in cui il sistema predatore-preda sia soggetto a pesca, il sistema (5), analogamente a quanto fatto per il (10), si può riscrivere come:

$$\begin{cases} x_{n+1} = (a - \varepsilon - by_n)\, x_n \Delta t + x_n & \text{(16a)} \\ y_{n+1} = (-c - \varepsilon + dx_n)\, y_n \Delta t + y_n & \text{(16b)} \end{cases}$$

Per quanto riguarda lo svilippo in serie del secondo ordine, si utilizzano ancora le (13), dove però le (14) e le (15) vanno sostituite, rispettivamente, dalle seguenti (17) e (18):

$$\begin{cases} \dot{x}_n = (a - \varepsilon) x_n - b x_n y_n & \text{(17a)} \\ \dot{y}_n = (-c - \varepsilon) y_n + d x_n y_n & \text{(17b)} \end{cases}$$

da cui, derivando:

$$\begin{cases} \ddot{x}_n = (a - \varepsilon)\dot{x}_n - b\left(\dot{x}_n y_n + x_n \dot{y}_n\right) & \text{(18a)} \\ \ddot{y}_n = -(c + \varepsilon)\dot{y}_n + d\left(\dot{x}_n y_n + x_n \dot{y}_n\right) & \text{(18b)} \end{cases}$$

4.2 Limitazione nella crescita

Introducendo una limitazione alla crescita della popolazione delle prede descritta, per esempio, dalla curva logistica con valore limite K, il sistema (3) diventa:

$$\begin{cases} \dot{x} = ax\left(1 - \dfrac{x}{K}\right) - bxy & \text{(19a)} \\ \dot{y} = -cy + dxy & \text{(19b)} \end{cases}$$

che, analogamente alla (10), si può riscrivere in modo da essere implementato in un algoritmo di integrazione basato sull'approssimazione del primo ordine della derivata:

$$\begin{cases} x_{n+1} = (a - by_n)\, x_n \Delta t - \dfrac{ax_n^2}{K} + x_n & \text{(20a)} \\ y_{n+1} = (-c + dx_n)\, y_n \Delta t + y_n & \text{(20b)} \end{cases}$$

Per una approssimazione del secondo ordine si utilizzano sempre le (13) con le (21) e (22) riportate qui di seguito:

$$\begin{cases} \dot{x}_n = ax_n \left(1 - \dfrac{x_n}{K}\right) - bx_n y_n & \text{(21a)} \\ \dot{y}_n = -cy_n + dx_n y_n & \text{(21b)} \end{cases}$$

$$\begin{cases} \ddot{x}_n = a\dot{x}_n \left(1 - \dfrac{2x_n}{K}\right) - b\left(\dot{x}_n y_n + x_n \dot{y}_n\right) & \text{(22a)} \\ \ddot{y}_n = -c\dot{y}_n + d\left(\dot{x}_n y_n + x_n \dot{y}_n\right) & \text{(22b)} \end{cases}$$

4.3 Esempi di implementazione

4.3.1 Con il foglio di calcolo

Gli algoritmi sopra descritti possono essere agevolmente implementati con il foglio di calcolo. In figura 3 è mostrata una parte del foglio di calcolo che risolve il modello di Lotka-Volterra logistico. I principali calcoli da impostare sono mostrati nella tabella 3. In modo del tutto analogo si possono preparare i fogli di calcolo per il modello con prede e predatori soggetti a pesca etc.

4.3.2 Con un m-file di Octave

L'equazione (10) sopra descritta è state implementata con Octave nel codice 1, mentre la (13) è stata tradotta in linguaggio Octave per il caso senza pesca, con accrescimento logistico della popolazione delle prede e in presenza di pesca rispettivamente nei codici 2, 3 e 4.

Le figure 4, 5 mostrano alcune figure ottenute con una approssimazione del primo ordine, mentre le figure 6 e 7 si riferiscono ad una approssimazione della derivata del secondo ordine. La figura 8 mostra come il sistema evolva verso un

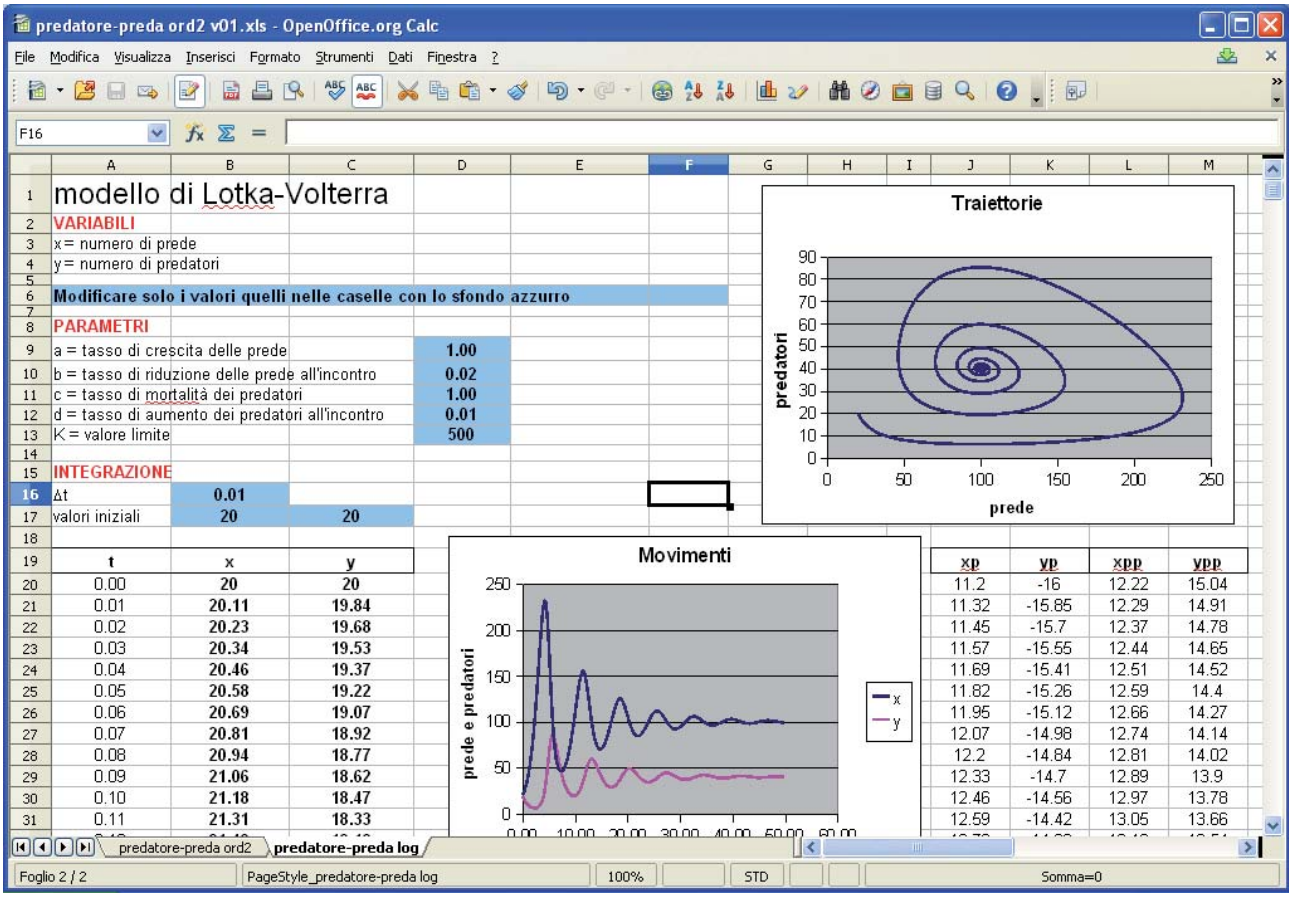

Figura 3: Foglio di calcolo per il modello predatore-preda logistico

punto di equilibrio stabile nel caso di equazione logistica per la popolazione delle prede e la figura 9 evidenzia il fatto che in presenza di pesca si ha un aumento del numero di prede. Nel caso di derivata approssimata al secondo ordine il passo di integrazione può essere sensibilmente ridotto.

Una volta implementato l'algoritmo, è molto interessante andare a vedere cosa succede al variare dei parametri in gioco. Ad esempio, è interessante vedere come si degradi la soluzione all'aumentare di Δt sia nel caso di approssimazione della derivata del primo ordine che del secondo, oppure modificare il parametro ε nel modello con pesca verificando che con $\varepsilon = 0$ si ottengono gli stessi risultati ottenuti nel modello in cui la pesca non sia praticata.

4.3.3 Con la funzione ODE45 di Octave

Il sistema (3) può essere risolto anche in modo più diretto utilizzando le funzioni `ODE23` e `ODE45` di Octave. In tal caso è sufficiente impostare tale sistema in una

cella	calcolo
B21	`=B20+J20*$B$16+0.5*L20*$B$16^2`
C21	`=C20+K20*$B$16+0.5*M20*$B$16^2`
J21	`=$D$9*B21*(1-B21/$D$13)-$D$10*B21*C21`
K21	`=-$D$11*C21+$D$12*B21*C21`
L21	`=$D$9*J21*(1-2*B21/$D$13)-$D$10*(J21*C21+B21*K21)`
M21	`=-$D$11*K21+$D$12*(J21*C21+B21*K21)`

Tabella 3: Calcoli da inserire nel foglio di calcolo

funzione che verrà poi richiamata da `ODE45` ad ogni passo di integrazione. Il listato 5 mostra l'm-file in cui si impostano i parametri da passare alla `ODE45` e la chiamata alla stessa. Tale funzione viene chiamata due volte, una prima volta per integrare le equazioni di Lotke-Volterra classiche 6, la seconda volta per integrare quelle in cui si è posta una curva logistica per le prede 7.

Codice 1: Algoritmo di integrazione delle equazioni di Lotka-Volterra con approssimazione del primo ordine della derivata.

```
clear
close all

% scelta dei parametri
x0=20;          % condizione iniziale prede
y0=20;          % condizione iniziale predatori
t0=0;           % tempo iniziale
tf=30;          % tempo finale
dt=0.001;       % Δ t
a=1;            % parametro a del modello
b=0.02;         % parametro b del modello
c=1;            % parametro c del modello
d=0.01;         % parametro d del modello

% inizializzazione
t=t0:dt:tf;
Npunti=size(t,2);
x=zeros(1,Npunti);
y=zeros(1,Npunti);
x(1)=x0; y(1)=y0;

% calcolo
for n=1:Npunti-1
  x(n+1)=x(n)+(a-b*y(n))*x(n)*dt;
  y(n+1)=y(n)+(-c+d*x(n))*y(n)*dt;
end

% grafico dei risultati
figure,plot(t,x,'b',t,y,'g')
legend('preda','predatore')
title('Evoluzione delle popolazioni')
xlabel('tempo')
ylabel('numero di individui')

figure,plot(x,y)
title('Piano delle fasi')
xlabel('preda')
ylabel('predatore')
```

Codice 2: Algoritmo di integrazione delle equazioni di Lotka-Volterra con approssimazione del secondo ordine della derivata.

```
clear
close all

% scelta dei parametri
x0=20;          % condizione iniziale prede
y0=20;          % condizione iniziale predatori
t0=0;           % tempo iniziale
tf=30;          % tempo finale
dt=0.001;       % Δ t
a=1;            % parametro a del modello
b=0.02;         % parametro b del modello
c=1;            % parametro c del modello
d=0.01;         % parametro d del modello

% inizializzazione
t=t0:dt:tf;
Npunti=size(t,2);
x=zeros(1,Npunti);
y=zeros(1,Npunti);
xp=zeros(1,Npunti-1);
yp=zeros(1,Npunti-1);
xpp=zeros(1,Npunti-1);
ypp=zeros(1,Npunti-1);
x(1)=x0; y(1)=y0;

% calcolo
for n=1:Npunti-1
  xp(n)=(a-b*y(n))*x(n);
  yp(n)=(d*x(n)-c)*y(n);
  xpp(n)=a*xp(n)-b*(xp(n)*y(n)+x(n)*yp(n));
  ypp(n)=-c*yp(n)+d*(xp(n)*y(n)+x(n)*yp(n));
  x(n+1)=x(n)+xp(n)*dt+1/2*xpp(n)*dt^2;
  y(n+1)=y(n)+yp(n)*dt+1/2*ypp(n)*dt^2;
end

% grafico dei risultati
figure,plot(t,x,'b',t,y,'g')
legend('preda','predatore')
title('Evoluzione delle popolazioni')
xlabel('tempo')
ylabel('numero di individui')

figure,plot(x,y)
title('Piano delle fasi')
xlabel('preda')
ylabel('predatore')
```

Codice 3: Algoritmo di integrazione delle equazioni di Lotka-Volterra con il modello logistico per le prede (approx. derivata ordine 2).

```
clear; close all

% scelta dei parametri
x0=20;          % condizione iniziale prede
y0=20;          % condizione iniziale predatori
t0=0;           % tempo iniziale
tf=60;          % tempo finale
dt=0.001;       % Δ t
a=1;            % parametro a del modello
b=0.02;         % parametro b del modello
c=1;            % parametro c del modello
d=0.01;         % parametro d del modello
K=500;          % valore limite per le prede

% inizializzazione
t=t0:dt:tf;
Npunti=size(t,2);
x=zeros(1,Npunti);
y=zeros(1,Npunti);
xp=zeros(1,Npunti-1);
yp=zeros(1,Npunti-1);
xpp=zeros(1,Npunti-1);
ypp=zeros(1,Npunti-1);
x(1)=x0; y(1)=y0;

% calcolo
for n=1:Npunti-1
  xp(n)=(a*(1-x(n)/K)-b*y(n))*x(n);
  yp(n)=(d*x(n)-c)*y(n);
  xpp(n)=a*xp(n)*(1-2*x(n)/K)-b*(xp(n)*y(n)+x(n)*yp(n));
  ypp(n)=-c*yp(n)+d*(xp(n)*y(n)+x(n)*yp(n));
  x(n+1)=x(n)+xp(n)*dt+1/2*xpp(n)*dt^2;
  y(n+1)=y(n)+yp(n)*dt+1/2*ypp(n)*dt^2;
end

% grafico dei risultati
figure,plot(t,x,'b',t,y,'g')
legend('preda','predatore')
title('Evoluzione delle popolazioni')
xlabel('tempo')
ylabel('numero di individui')

figure,plot(x,y)
title('Piano delle fasi')
xlabel('preda')
ylabel('predatore')
```

Codice 4: Algoritmo di integrazione delle equazioni di Lotka-Volterra con in presenza di pesca (approx. derivata ordine 2).

```
clear; close all

% scelta dei parametri
x0=20;          % condizione iniziale prede
y0=20;          % condizione iniziale predatori
t0=0;           % tempo iniziale
tf=30;          % tempo finale
dt=0.001;       % Δ t
a=1;            % parametro a del modello
b=0.02;         % parametro b del modello
c=1;            % parametro c del modello
d=0.01;         % parametro d del modello
e=0.5;          % parametro di intensità della pesca

% inizializzazione
t=t0:dt:tf;
Npunti=size(t,2);
x=zeros(1,Npunti);
y=zeros(1,Npunti);
xp=zeros(1,Npunti-1);
yp=zeros(1,Npunti-1);
xpp=zeros(1,Npunti-1);
ypp=zeros(1,Npunti-1);
x(1)=x0; y(1)=y0;

% calcolo
for n=1:Npunti-1
  xp(n)=((a-e)-b*y(n))*x(n);
  yp(n)=(d*x(n)-(c+e))*y(n);
  xpp(n)=(a-e)*xp(n)-b*(xp(n)*y(n)+x(n)*yp(n));
  ypp(n)=-(c+e)*yp(n)+d*(xp(n)*y(n)+x(n)*yp(n));
  x(n+1)=x(n)+xp(n)*dt+1/2*xpp(n)*dt^2;
  y(n+1)=y(n)+yp(n)*dt+1/2*ypp(n)*dt^2;
end

% grafico dei risultati
figure,plot(t,x,'b',t,y,'g')
legend('preda','predatore')
title('Evoluzione delle popolazioni')
xlabel('tempo')
ylabel('numero di individui')

figure,plot(x,y)
title('Piano delle fasi')
xlabel('preda')
ylabel('predatore')
```

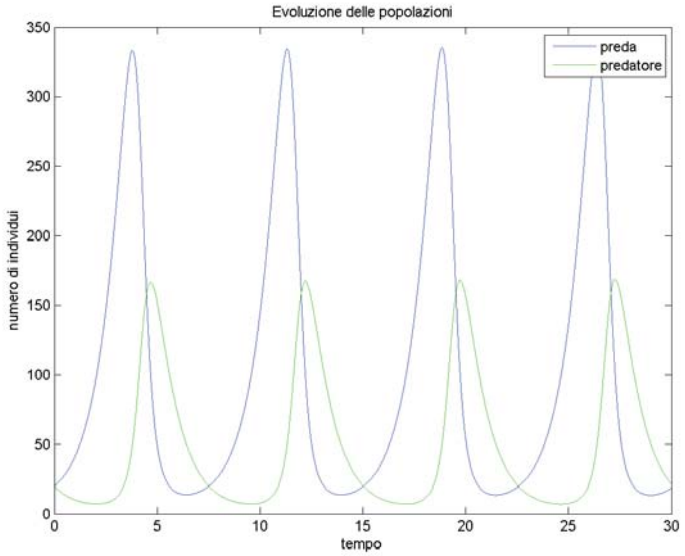

Figura 4: Andamento delle popolazioni di prede e predatori (approx. della derivata del primo ordine - codice 1).

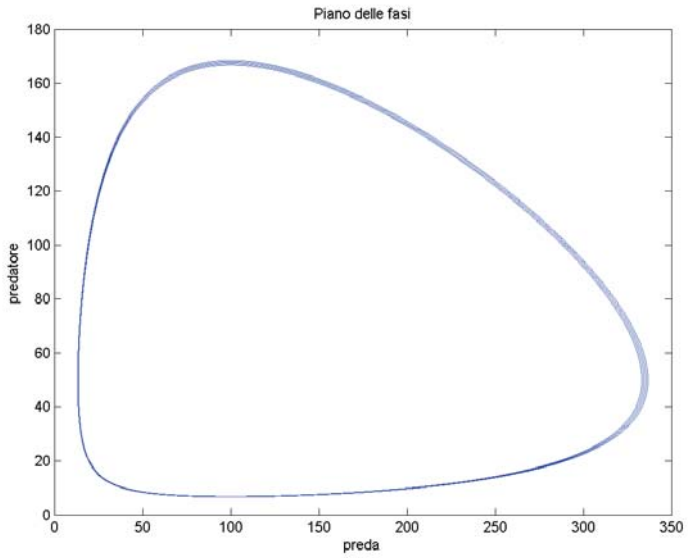

Figura 5: Piano delle fasi prede e predatori (approx. della derivata del primo ordine - codice 1).

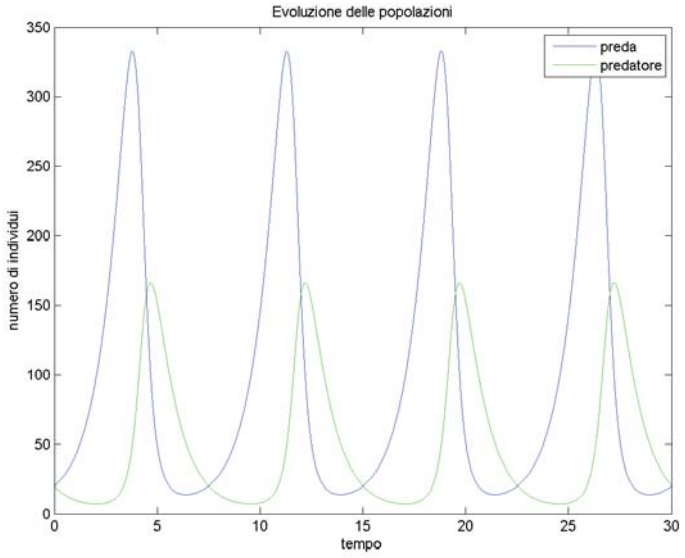

Figura 6: Andamento delle popolazioni di prede e predatori (approx. della derivata del secondo ordine - codice 2).

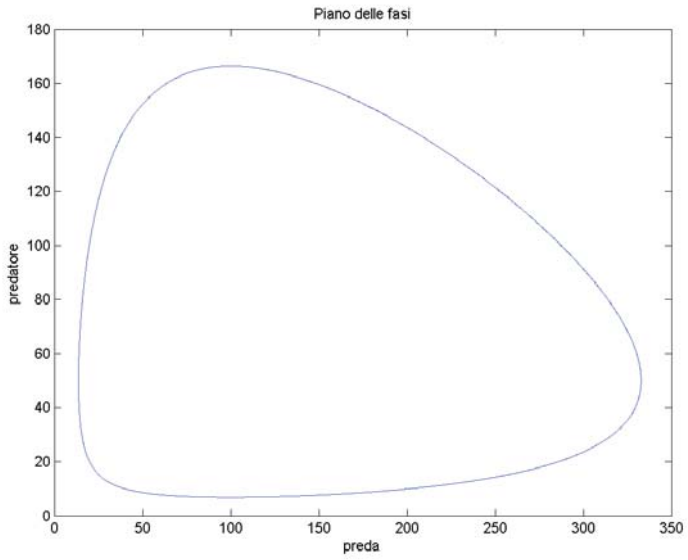

Figura 7: Piano delle fasi prede e predatori (approx. della derivata del secondo ordine - codice 2).

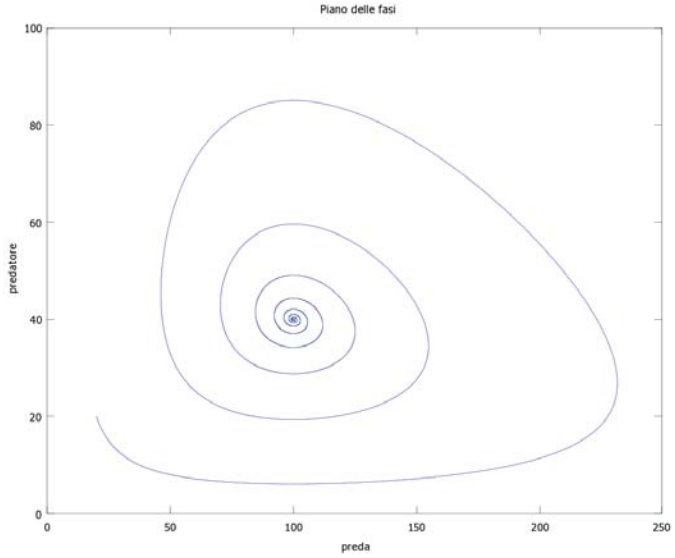

Figura 8: Piano delle fasi prede e predatori (approx. della derivata del secondo ordine e modello logistico - codice 3)

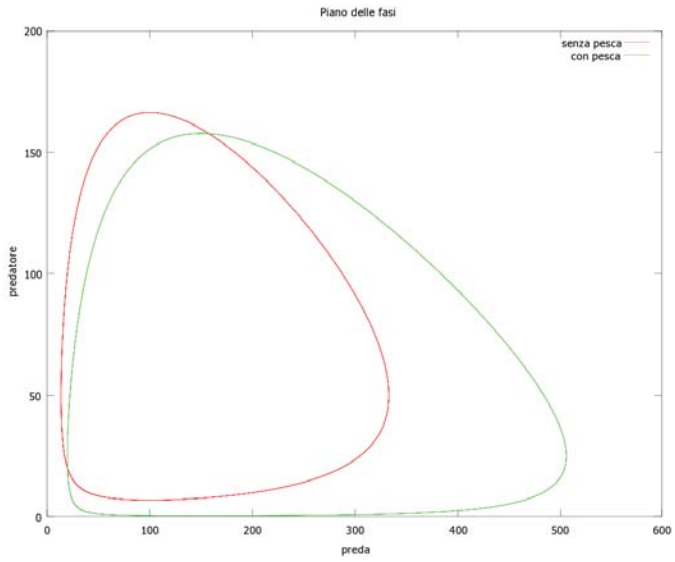

Figura 9: Piano delle fasi prede e predatori (con e senza pesca - codice 4)

Codice 5: File di impostazione dei parametri del modello di Lotka-Volterra integrato con la funzione `ODE45` di Octave.

```
clear; close all

% scelta dei parametri
x0=20;          % condizione iniziale prede
y0=20;          % condizione iniziale predatori
t0=0;           % tempo iniziale
tf=30;          % tempo finale
dt=0.001;       % Δ t

% integrazione delle equazioni differenziali
%%% i parametri a, b, c, d sono contenuti nella funzione
%%% che viene chiamata
tfinal=tf*(1+eps);
options=odeset('MaxStep',dt*100);   % max passo di integraz.
%%% con Octave
[t,f]=ode45(@lotkavolterra,[t0 tfinal],[x0 y0],options);
[tl,fl]=ode45(@lotkavolterralog,[t0 tfinal],[x0 y0],options);
%%% con Matlab
%[tl,fl]=ode45('lotkavolterra',[t0 tfinal],[x0 y0],options);
%[tl,fl]=ode45('lotkavolterralog',[t0 tfinal],[x0 y0],options);

x=f(:,1); y=f(:,2);
xl=fl(:,1); yl=fl(:,2);

% grafico dei risultati
figure,plot(t,x,'b',t,y,'g')
legend('preda','predatore')
title('Evoluzione delle popolazioni')
xlabel('tempo')
ylabel('numero di individui')

figure,plot(x,y)
title('Piano delle fasi')
xlabel('preda')
ylabel('predatore')

figure,plot(tl,xl,'b',tl,yl,'g')
legend('preda','predatore')
title('Evoluzione delle popolazioni con modello logistico')
xlabel('tempo')
ylabel('numero di individui')

figure,plot(xl,yl)
title('Piano delle fasi con modello logistico')
xlabel('preda')
ylabel('predatore')
```

Codice 6: Funzione contenente il modello di Lotka-Volterra.

```
function fp=lotkavolterra(t,f)
%LOTKAVOLTERRA  modello predatore-preda di Lotka-Volterra

a=1;            % parametro a del modello
b=0.02;         % parametro b del modello
c=1;            % parametro c del modello
d=0.01;         % parametro d del modello

fp=diag([a-b*f(2) -c+d*f(1)])*f;
```

Codice 7: Funzione contenente il modello di Lotka-Volterra con modello logistico per la popolazione delle prede.

```
function fp=lotkavolterralog(t,f)
%LOTKAVOLTERRALOG  modello predatore-preda logistico

a=1;            % parametro a del modello
b=0.02;         % parametro b del modello
c=1;            % parametro c del modello
d=0.01;         % parametro d del modello
K=500;          % valore limite per le prede

fp=diag([a*(1-f(1)/K)-b*f(2) -c+d*f(1)])*f;
```

5 Conclusioni

Per assicurare la sostenibilità nel lungo termine dell'attività di pesca, è essenziale, da un lato, riuscire a valutare le condizioni degli stock sovrasfruttati e, dall'altro, che i risultati di queste analisi siano considerati nel processo decisionale relativo alla gestione della pesca.

Attraverso le procedure chiamate di "stock assessment", si tenta di stimare il numero di pesci correntemente in uno stock e il tasso di mortalità dovuto alle attività di pesca; si cerca, inoltre, di estrapolare predizioni su come lo stock risponderà all'attuale e futura modalità di gestione della pesca.

I modelli utilizzati nello stock assessment sono tradizionalmente modelli basati su una sola specie. Tuttavia esistono anche approcci che considerano più specie contemporaneamente e anche l'interazione predatore-prede; in questi casi possono essere utilizzati dei modelli preda-predatore o estensioni del modello di Lotka-Volterra nell'ambito di modelli più estesi.