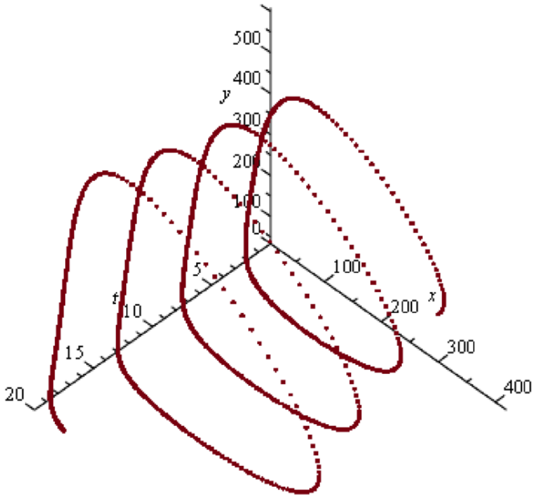
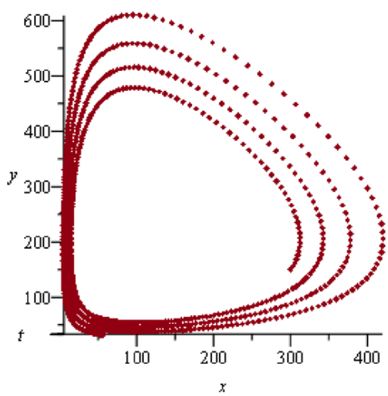
Risolviamo il problema 8.

```
restart:
with(plots):
Digits:=15:
a:=2: b:=0.01: c:=1: d:=0.01:
LVpp := diff(x(t), t) = x(t)*(a-b*y(t)), diff(y(t), t) = y(t)*(-c+d*x(t)):
vars := x(t), y(t):
ics := x(0)=300, y(0)=150:
tmax:=20: h:=0.02: N:=1000:
sol := dsolve({LVpp, ics}, numeric, {vars}, method = classical[foreuler],
              stepsize = h):
odeplot(sol, [t, x(t), y(t)], 0 .. tmax, axes = normal, numpoints = N,
        style = point, labels = [t, x, y]);
```

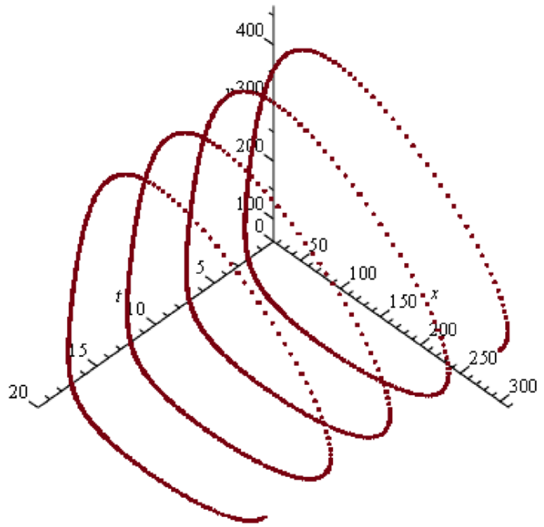



```
odeplot(sol, [t, x(t), y(t)], 0 .. tmax, axes = normal, numpoints = N,
        style = point, labels = [t, x, y], orientation = [0, 90]);
```

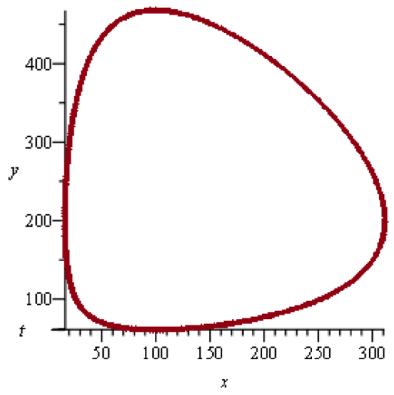


Si ottiene lo stesso risultato del problema precedente con gli stessi valori di `h` e `N` (*cfr*. con le figure di p. 35). Usando invece l'algoritmo di *default*, basato su una migliore approssimazione delle funzioni, si ottiene un risultato più preciso:

```
sol := dsolve({LVpp, ics}, numeric, {vars}, method = rkf45, stepsize = h):
odeplot(sol, [t, x(t), y(t)], 0 .. tmax, axes = normal, numpoints = N,
        style = point, labels = [t, x, y]);
```



```
odeplot(sol, [t, x(t), y(t)], 0 .. tmax, axes = normal, numpoints = N,
        style = point, labels = [t, x, y], orientation = [0, 90]);
```



I grafici ottenuti con i comandi `pointplot3d` (si veda il problema 7) e `odeplot`, nelle tre dimensioni (t, *x*, *y*), rappresentano una curva, soluzione del sistema con le condizioni iniziali considerate. Una precisazione − che tuttavia non dovrebbe essere sfuggita − è doverosa: il diagramma che riporta la traiettoria sul piano delle fasi non è dunque altro che una proiezione bidimensionale di tale curva sul piano (*x*, *y*), come si è visto; e così pure i grafici relativi all'evoluzione temporale della popolazione delle prede e di quella dei predatori sono le proiezioni della stessa curva sul piano (t, *x*) e sul piano (t, *y*), rispettivamente. Ciò sarà messo in luce risolvendo il prossimo problema.