

L'algue tueuse
TD n ° 3
Modélisation mathématique
Q4

Sibylle Roux

Juliette Arazo
Tanguy Thomas

Nicolas Le Gallo

17 octobre 2017

Table des matières

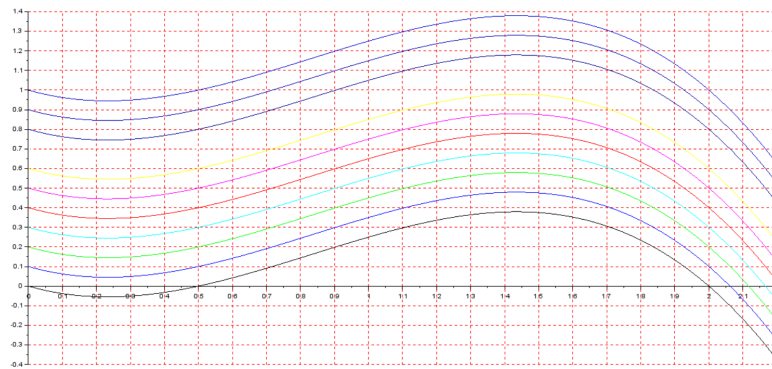
1 Etude du modèle logistique avec effet Allee et immigration	3
1.1 Etude numérique	3
1.1.1 Modèle avec variation de I	3
1.1.2 Modèle avec variation de K	4
1.1.3 Modèle avec variation de A	5
1.1.4 Modèle avec variation de la population initiale	6
1.2 Etude mathématique	7
1.3 Bilan	7
2 Etude du modèle logistique avec prédation	8
2.1 Etude numérique	8
2.1.1 Modèle	8
2.1.2 Modèle avec variation de B	9
2.1.3 Modèle avec variation de C	9
2.2 Etude mathématique	10
2.3 Bilan	10
A Etude du modèle logistique avec effet Allee et immigration- Scripts Scilab	11
A.1 Modèle avec variation de I	11
A.1.1 Vitesse d'accroissement	11
A.1.2 Discretisation	11
A.2 Modèle avec variation de K	12
A.2.1 Vitesse d'accroissement	12
A.2.2 Discretisation	12
A.3 Modèle avec variation de A	13
A.3.1 Vitesse d'accroissement	13
A.3.2 Discretisation	13
A.4 Modèle avec variation de la population initiale	14
B Etude du modèle logistique avec prédation - Scripts Scilab	15
B.1 Modèle	15
B.1.1 Vitesse d'accroissement	15
B.1.2 Discretisation	15
B.2 Modèle avec variation de B	16
B.2.1 Vitesse d'accroissement	16
B.2.2 Discretisation	16
B.3 Modèle avec variation de C	17
B.3.1 Vitesse d'accroissement	17
B.3.2 Discretisation	18

1 Etude du modèle logistique avec effet Allee et immigration

1.1 Etude numérique

1.1.1 Modèle avec variation de I

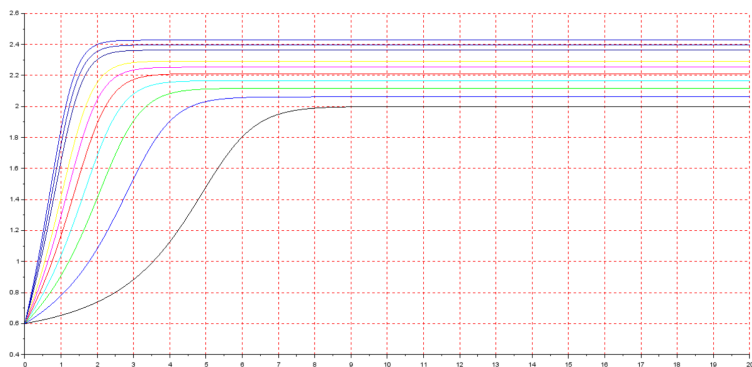
Courbes de la vitesse d'accroissement



Paramètres de modélisation : $K = 2$; $r = 0.5$; $A = 0.5$; I varie de 0 à 1 avec un pas de 0.1

Sur ce graphique on peut voir les différentes courbes de la vitesse d'accroissement pour différentes valeurs de I . On remarque que avec un i assez grand, l'effet Allee n'éteint plus la population mais ralentit seulement sa croissance. En effet 2 points d'équilibres peuvent disparaître pour certaines valeurs de I .

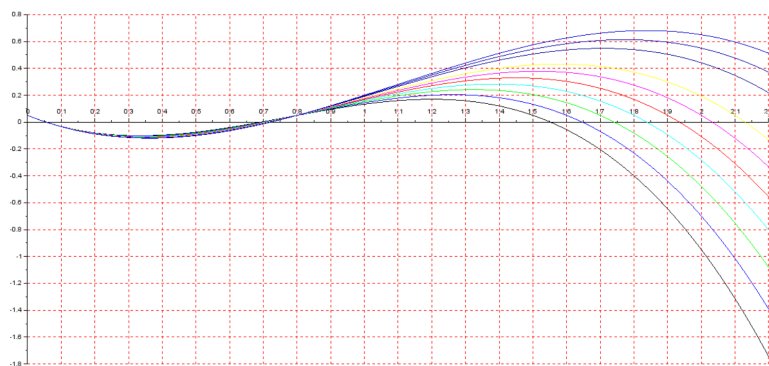
Discrétisation du modèle



Paramètres de modélisation : $a = 0.6$; $K = 2$; $r = 0.5$; $A = 0.5$; I varie de 0 à 1 avec un pas de 0.1

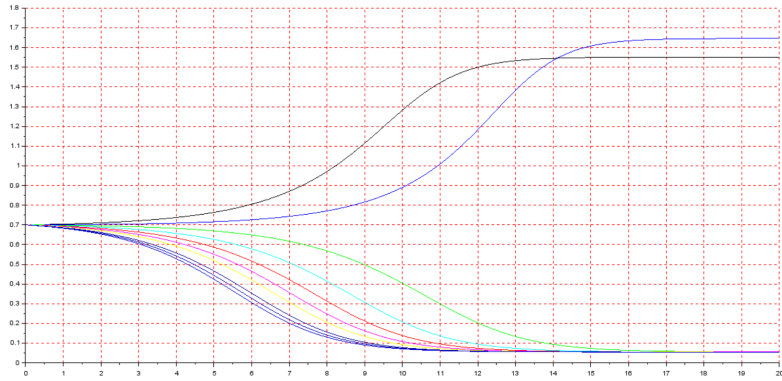
1.1.2 Modèle avec variation de K

Courbes de la vitesse d'accroissement



Paramètres de modélisation : $r = 1$; $A = 0.8$; $I = 0.05$; K varie de 1.5 à 2.5 avec un pas de 0.1

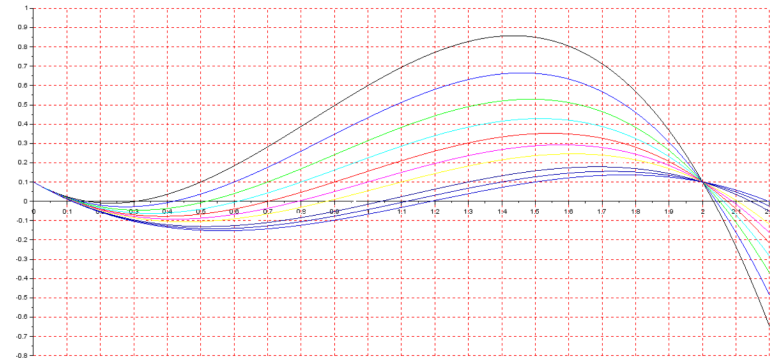
Discrétisation du modèle



Paramètres de modélisation : $a = 0.7$; $r = 1$; $A = 0.8$; $I = 0.05$; K varie de 1.5 à 2.5 avec un pas de 0.1

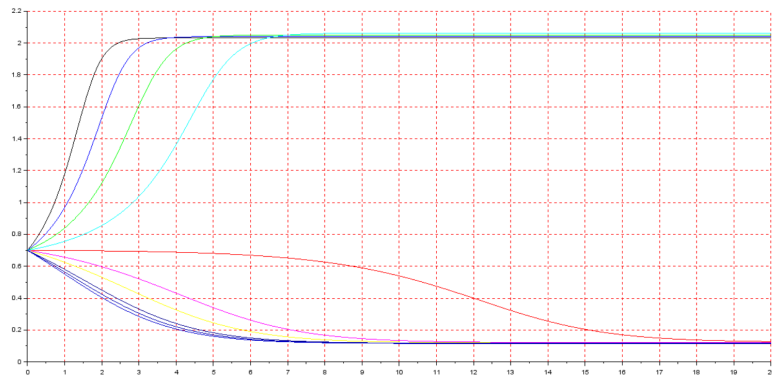
1.1.3 Modèle avec variation de A

Courbes de la vitesse d'accroissement



Paramètres de modélisation : $I = 0.1$; $K = 2$; $r = 1$; A varie de 0.5 à 1.5 avec un pas de 0.1

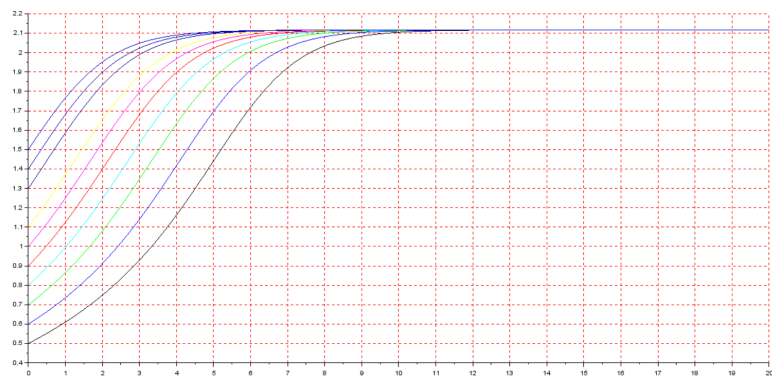
Discretisation du modèle



Paramètres de modélisation : $a = 0.7$; $I = 0.1$; $K = 2$; $r = 1$; A varie de 0.5 à 1.5 avec un pas de 0.1

1.1.4 Modèle avec variation de la population initiale

Discretisation du modèle



Paramètres de modélisation : $A = 0.5$; $I = 0.1$; $K = 2$; $r = 0.25$; a varie de 0.5 à 1.5 avec un pas de 0.1

1.2 Etude mathématique

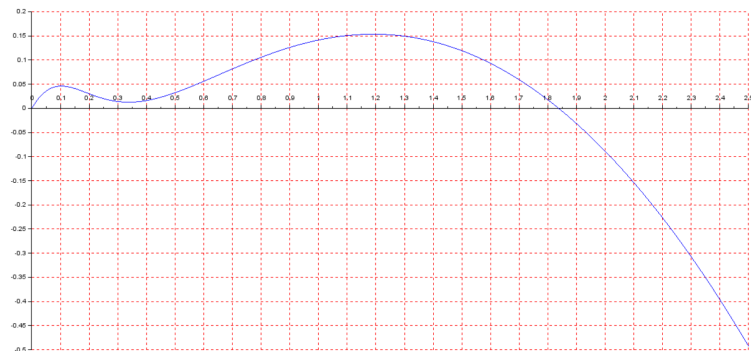
1.3 Bilan

2 Etude du modèle logistique avec prédation

2.1 Etude numérique

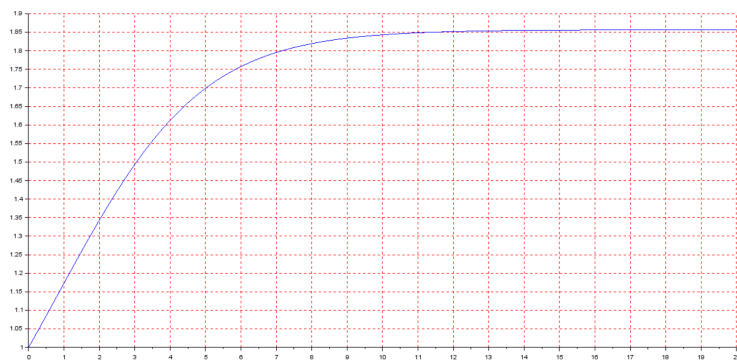
2.1.1 Modèle

Vitesse d'accroissement



Paramètres de modélisation : $r = 1$; $A = 0.5$; $K = 2.5$; $B = 0.5$; $C = 0.3$

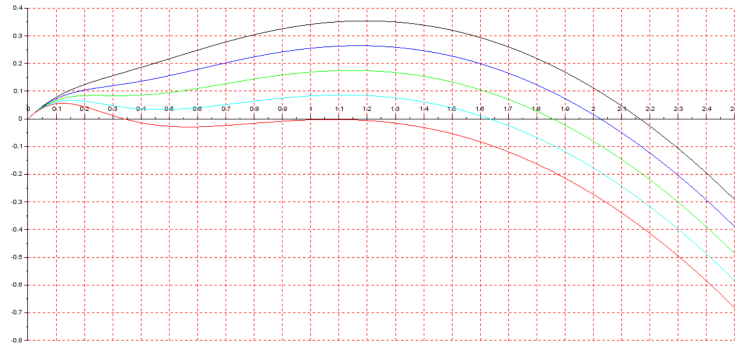
Discretisation



Paramètres de modélisation : $r = 1$; $A = 0.5$; $K = 2.5$; $B = 0.5$; $C = 0.4$;
 $h = 0.05$; $a = 1$

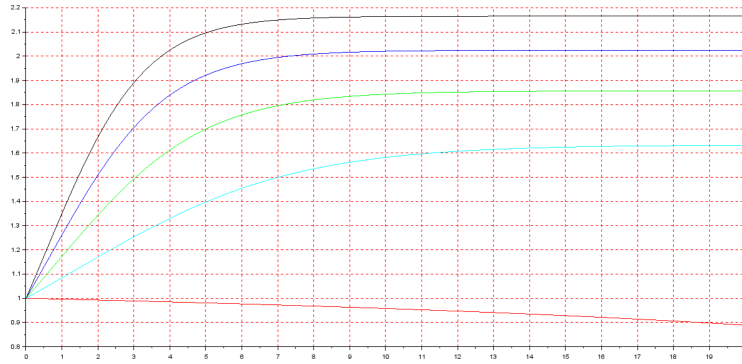
2.1.2 Modèle avec variation de B

Vitesse d'accroissement



Paramètres de modélisation : $r = 1$; $A = 0.5$; $K = 2.5$; $C = 0.4$; B varie de 0.3 à 0.7 avec un pas de 0.1

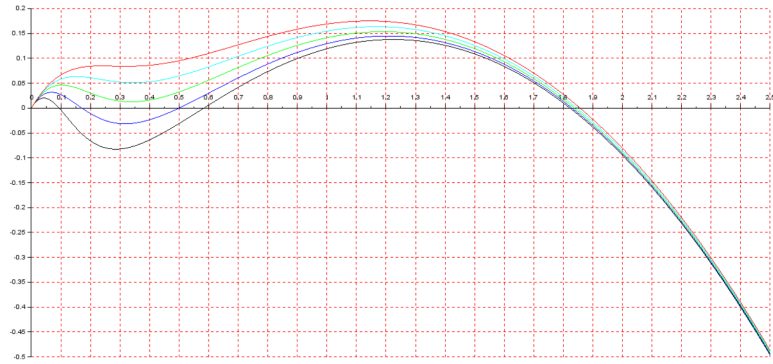
Discretisation



Paramètres de modélisation : $r = 1$; $A = 0.5$; $K = 2.5$; $C = 0.4$; $h = 0.05$; $a = 1$; B varie de 0.3 à 0.7 avec un pas de 0.1

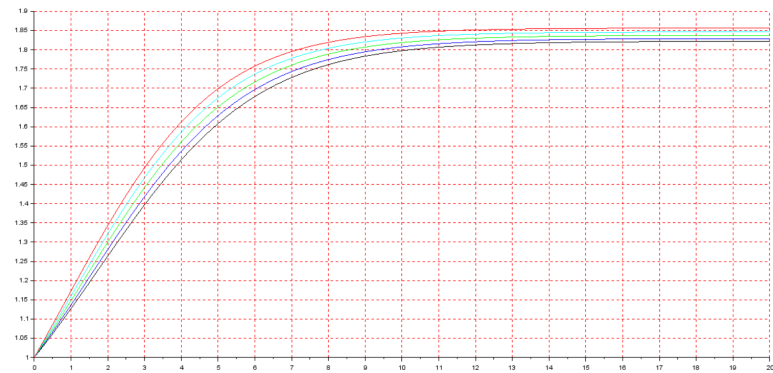
2.1.3 Modèle avec variation de C

Vitesse d'accroissement



Paramètres de modélisation : $r = 1$; $A = 0.5$; $K = 2.5$; $B = 0.5$; C varie de 0.2 à 0.4 avec un pas de 0.05

Discretisation



Paramètres de modélisation : $r = 1$; $A = 0.5$; $K = 2.5$; $B = 0.5$; $h = 0.05$; $a = 1$; C varie de 0.2 à 0.4 avec un pas de 0.05

2.2 Etude mathématique

2.3 Bilan

A Etude du modèle logistique avec effet Allee et immigration- Scripts Scilab

A.1 Modèle avec variation de I

A.1.1 Vitesse d'accroissement

```
clear
clf

r = 0.5 ; A = 0.5 ; K = 2 ; // variables du modèles
Ivect=0:0.1:1; // variable qui varie
x = linspace(0, 2.2, 301); // vecteur contenant les valeurs de la vitesse d'accroissement

function f = allee_imig(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+ I // opérations vectorielles. x est un vecteur
endfunction

for i=1:11; // Bouclae qui va dessiner les différentes courbes
    I=Ivect(i); // Assignment valeur qui varie
    plot2d(x, allee_imig(x), style = i) // Tracé de la vitesse d'accroissement
end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];
```

A.1.2 Discretisation

```
clear
clf

Ivect=0:0.1:1; // variable qui varie
r = 0.5 ; A = 0.5 ; K = 2 ; h = 0.05 ; a = 0.6 ; // variables du modèles + pas de temps
ndate = 0:h:20; // vecteur des instants où on calcule la solution

function f = allee_img(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+I // opérations vectorielles. x est un vecteur
endfunction

x(1)=a; // Initialisation de la population initiale

for i=1:11; // Boucle qui va dessiner les différentes courbes
    I=Ivect(i); // Assignment valeur qui varie
    for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
```

```

        x(n+1) = x(n) + h * allee_img(x(n)); // Calcul de la population
    end
    plot2d(ndate, x, style = i) // Tracé de la discretisation
end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

A.2 Modèle avec variation de K

A.2.1 Vitesse d'accroissement

```

clear
clf

r = 1 ; A = 0.8 ; I=0.05 ; // variables du modèles
Kvect=1.5:0.1:2.5; // variable qui varie
x = linspace(0, 2.2, 301); // vecteur contenant les valeurs de la vitesse d'accroissement

function f = allee_img(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+ I // opérations vectorielles. x est un vecteur
endfunction

for i=1:11; // Boucle qui va dessiner les différentes courbes
    K=Kvect(i); // Assignment valeur qui varie
    plot2d(x, allee_img(x), style = i) // Tracé de la vitesse d'accroissement
end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

A.2.2 Discretisation

```

clear
clf

Kvect=1.5:0.1:2.5; // variable qui varie
r = 1 ; A = 0.8 ; I=0.05 ; h = 0.05 ; a = 0.7 ; // variables du modèles + pas de temps
ndate = 0:h:20; // vecteur des instants où on calcule la solution

function f = allee_img(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+I // opérations vectorielles. x est un vecteur

```

```

endfunction

x(1)=a; // Initialisation de la population initiale

for i=1:11; // Boucle qui va dessiner les différentes courbes
    K=Kvect(i); // Assignation valeur qui varie
    for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
        x(n+1) = x(n) + h * allee_img(x(n)); // Calcul de la population
    end
    plot2d(ndate, x, style = i) // Tracé de la discretisation
end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

A.3 Modèle avec variation de A

A.3.1 Vitesse d'accroissement

```

clear
clf

r = 1 ; K = 2 ; I=0.1 ; // variables du modèles
Avect=0.5:0.1:1.5; // variable qui varie
x = linspace(0, 2.2, 301); // vecteur contenant les valeurs de la vitesse d'accroissement

function f = allee_img(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+ I // opérations vectorielles. x est un vecteur
endfunction

for i=1:11; // Boucle qui va dessiner les différentes courbes
    A=Avect(i); // Assignation valeur qui varie
    plot2d(x, allee_img(x), style = i) // Tracé de la vitesse d'accroissement
end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

A.3.2 Discretisation

```

clear
clf

```

```

Avect=0.5:0.1:1.5; // variable qui varie
r = 1 ; K = 2 ; I=0.1 ; h = 0.05 ; a = 0.7 ; // variables du modèles + pas de temps
ndate = 0:h:20; // vecteur des instants où on calcule la solution

function f = allee_img(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+I // opérations vectorielles. x est un vecteur
endfunction

x(1)=a; // Initialisation de la population initiale

for i=1:11; // Boucle qui va dessiner les différentes courbes
    A=Avect(i); // Assignment valeur qui varie
    for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
        x(n+1) = x(n) + h * allee_img(x(n)); // Calcul de la population
    end
    plot2d(ndate, x, style = i) // Tracé de la discretisation
end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

A.4 Modèle avec variation de la population initiale

```

clear
clf

avect=0.5:0.1:1.5; // variable qui varie
r = 0.25 ; K = 2 ; I=0.1 ; A = 0.5 ; h = 0.05 ; // variables du modèles + pas de temps
ndate = 0:h:20; // vecteur des instants où on calcule la solution

function f = allee_img(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (x / A - 1 ) .* (1 - x / K)+I // opérations vectorielles. x est un vecteur
endfunction

for i=1:11; // Boucle qui va dessiner les différentes courbes
    x(1)=avect(i); // Assignment valeur qui varie
    for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
        x(n+1) = x(n) + h * allee_img(x(n)); // Calcul de la population
    end
    plot2d(ndate, x, style = i) // Tracé de la discretisation
end

// Définition des paramètres d'affichages

```

```

a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

B Etude du modèle logistique avec prédation - Scripts Scilab

B.1 Modèle

B.1.1 Vitesse d'accroissement

```

clear
clf

// variables du modèles
r = 1 ; A = 0.5 ; K = 2.5 ; B=0.5 ; C=0.3 ;
x = linspace(0, 2.5, 301);

function f = predation(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (1 - x / K) - B * (x.^2 ./ (x.^2 + C^2)) // opération vectorielle
endfunction

plot2d(x, predation(x), style = 2); // Tracé de la vitesse d'accroissement

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

B.1.2 Discretisation

```

clear
clf

// variables du modèles
r = 1 ; A = 0.5 ; K = 2.5 ; B=0.5 ; C=0.4 ; h = 0.05 ; I=0.2 ; a = 1;
ndate = 0:h:20; // vecteur des instants

function f = predation(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (1 - x / K) - B * (x.^2 ./ (x.^2 + C^2)) // opération vectorielle
endfunction

x(1) = a; // Initialisation de la population initiale

for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
    x(n+1) = x(n) + h * predation(x(n)); // Calcul de la population
endfor

```

```

end

plot2d(ndate, x, style = 2) // Tracé de la trajectoire

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

B.2 Modèle avec variation de B

B.2.1 Vitesse d'accroissement

```

clear
clf

// variables du modèles
r = 1 ; A = 0.5 ; K = 2.5 ; C=0.4 ;
Bvect = 0.3:0.1:0.7; // variable qui varie
x = linspace(0, 2.5, 301);

function f = predation(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (1 - x / K) - B * (x.^2 ./ (x.^2 + C^2)) // opération vectorielle
endfunction

for i=1:5 // Boucle qui va dessiner les différentes courbes

    B=Bvect(i); // Assignation valeur qui varie

    plot2d(x, predation(x), style = i); // Tracé de la vitesse d'accroissement

end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

B.2.2 Discretisation

```

clear
clf

Bvect = 0.3:0.1:0.7; // variable qui varie
// variables du modèles
r = 1 ; A = 0.5 ; K = 2.5 ; C=0.4 ; a = 1 ; h = 0.05 ; I=0.2 ;

```



```

ndate = 0:h:20; // vecteur des instants

function f = predation(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (1 - x / K) - B * (x.^2 ./ (x.^2 + C^2)) // opération vectorielle
endfunction

for i=1:5 // Boucle qui va dessiner les différentes courbes

    B=Bvect(i); // Assignment valeur qui varie
    x(1) = a; // Initialisation de la population initiale

    for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
        x(n+1) = x(n) + h * predation(x(n)); // Calcul de la population
    end

    plot2d(ndate, x, style = i) // Tracé de la trajectoire

end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

B.3 Modèle avec variation de C

B.3.1 Vitesse d'accroissement

```

clear
clf

// variables du modèles
r = 1; A = 0.5; K = 2.5; B=0.5;
Cvect = 0.2:0.05:0.4; // variable qui varie
x = linspace(0, 2.5, 301);

function f = predation(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (1 - x / K) - B * (x.^2 ./ (x.^2 + C^2)) // opération vectorielle
endfunction

for i=1:5 // Boucle qui va dessiner les différentes courbes

    C=Cvect(i); // Assignment valeur qui varie

    plot2d(x, predation(x), style = i); // Tracé de la vitesse d'accroissement

```

```

end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

B.3.2 Discretisation

clear
clf

Cvect = 0.2:0.05:0.4; // variable qui varie
// variables du modèles
r = 1 ; A = 0.5 ; K = 2.5 ; B=0.5 ; h = 0.05 ; I=0.2 ; a = 1 ;
ndate = 0:h:20; // vecteur des instants

function f = predation(x) // fonction qui calcule la vitesse d'accroissement
    f = r * x .* (1 - x / K) - B * (x.^2 ./ (x.^2 + C^2)) // opération vectorielle
endfunction

for i=1:5 // Boucle qui va dessiner les différentes courbes

    C=Cvect(i); // Assignment valeur qui varie
    x(1) = a; // Initialisation de la population initiale

    for n = 1:length(ndate) - 1 // Boucle qui calcule la courbe de la population
        x(n+1) = x(n) + h * predation(x(n)); // Calcul de la population
    end

    plot2d(ndate, x, style = i) // Tracé de la trajectoire

end

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```