Universidade de São Paulo
Instituto de Física de São Carlos
Introdução à Física Computacional

Projeto 5: Dinâmica Populacional

Beatriz de Camargo Castex Ferreira - 10728077

São Carlos - 10/06/2019

Sumário:

1. INTRODUÇÃO

É muito comum ouvir falar que tudo está um caos, mas o que realmente é o caos? Ele é quantitativo? Como chegamos ao caos? O caos foi identificado em muitas situações da vida real, como circuitos eletrônicos, reações químicas, turbulência e oscilações de células cardíacas. Sendo assim, é um fenômeno que merece ser estudado cuidadosamente. Para tal nesse projeto é discutido o problema de dinâmica populacional, em que se observa o crescimento da densidade populacional de indivíduos dependendo das suas condições, enquanto um sistema "organizado" descende ao caos.

2. METODOLOGIA

Como será tratada a questão de dinâmica populacional, nesta prática iremos observar mapas logísticos para o crescimento de uma população de indivíduos, primeiro sem nenhuma outra espécie, e depois havendo competição de predadorismo.

2.1. Ponto fixo de período 1

Neste exercício será estudado o crescimento populacional de uma única espécie, utilizando o seguinte mapa logístico:

$$G(x_i) = rx_i(1 - x_i) \quad (1)$$

onde x é a densidade populacional, r é a taxa de crescimento populacional e G é a geração em que o crescimento acontece, sendo que a máxima densidade populacional possível é $x_i = 1$, após a qual o crescimento da espécie se torna insustentável e a população (que estaria crescendo) volta a diminuir.

A. Ponto Fixo

Ao observar o gráfico para diferentes valores de r é possível perceber alguns pontos especiais. Entende-se então que a equação (1) tem uma quantidade de soluções que varia de acordo com o valor do r. Fazendo um gráfico para r = 0,5 , 1 e 2 percebe-se que nos valores menores do que 1 a inclinação do gráfico é muito baixa para que se consiga ver mais do que a solução (0,0), porém quando temos uma taxa r maior do que 1 percebem-se duas soluções, a solução (0,0) e um ponto fixo x*. Desse modo, podemos entender melhor o funcionamento do mapa logístico e refletir sobre o que isso representa para o crescimento de uma população.

B. Dependência do ponto fixo.

Para sabermos se o ponto fixo depende do ponto inicial ou se da taxa de variância escreve-se um programa em FORTRAN que executa o mapa logístico, permitindo que estudemos melhor o fenômeno. Isso é feito usando um sistema de integração de Euler e escolhendo múltiplos valores para a taxa de crescimento e para a densidade de população inicial.

```
        ! Programa que calcula a densidade populacional de uma população sem
predadores em função do número de gerações para 5 densidades iniciais diferentes
utilizando diferentes taxas de variação populacional

    program populacao
    IMPLICIT NONE

    ! Variáveis
    REAL*8 :: pop                   ! densidade populacional (x)
    REAL*8 :: pop_inicial           !densidade populacional inicial
    REAL*8 :: dpop_inicial          ! Variação da densidade populacional inicial
```

```
REAL*8 :: taxa                  ! taxa de variação populacional atual
REAL*8, dimension(3) :: taxa_var ! Taxa de variação populacional
INTEGER*8 :: G                  ! número de gerações
INTEGER*8 :: i                  ! contador
INTEGER*8 :: j                  ! contador

! Inicializando variáveis
taxa_var(1) = 1.d0
taxa_var(2) = 2.d0
taxa_var(3) = 2.5d0


! Abrindo arquivos para impressão de variáveis
open(11, file = "populacao_r1_x018.dat")
open(12, file = "populacao_r1_x036.dat")
open(13, file = "populacao_r1_x054.dat")
open(14, file = "populacao_r1_x072.dat")
open(15, file = "populacao_r1_x090.dat")
open(21, file = "populacao_r2_x018.dat")
open(22, file = "populacao_r2_x036.dat")
open(23, file = "populacao_r2_x054.dat")
open(24, file = "populacao_r2_x072.dat")
open(25, file = "populacao_r2_x090.dat")
open(31, file = "populacao_r3_x018.dat")
open(32, file = "populacao_r3_x036.dat")
open(33, file = "populacao_r3_x054.dat")
open(34, file = "populacao_r3_x072.dat")
open(35, file = "populacao_r3_x090.dat")


! Queremos graficar a densidade para 3 taxas de variação diferentes
DO i =1,3

      taxa  = taxa_var(i)
      dpop_inicial = 0.18d0
      pop_inicial = 0.d0

      ! Queremos graficar a densidade para 5 x iniciais diferentes entre 0
e 1
      DO j=1,5

            ! calculando valor inicial novo da densidade pop.
            pop_inicial = pop_inicial + dpop_inicial
            pop = pop_inicial

            ! Calculando a variação da densidade populacional, utilizando
um número arbitrário de gerações como geração máxima
            DO G = 1, 30

                  pop = taxa * pop * (1.d0 - pop)

                  write((10*i)+j,*)G,pop
            enddo
      enddo
enddo
```

```
end program populacao
```

C. Distância

Escolhendo um novo valor de população inicial muito próximo do anterior permite que se avalie a diferença entre uma e outra, assim é possível notar quando elas divergem uma da outra, demonstrando como apenas uma leve diferença poderia causar mudanças.

Isto pode ser obtido com uma leve mudança ao código do exercício anterior:

```
! Programa que calcula a distância entre as gerações de dois valores ligeiramente diferentes.

        ! Queremos graficar a densidade para 3 taxas de variação diferentes
    DO i =1,3

        taxa  = taxa_var(i)
        dx_inicial = 0.18d0
        x_inicial = 0.d0

        ! Queremos graficar a densidade para 5 x iniciais diferentes entre 0 e 1
        DO j=1,5

            ! calculando valor inicial novo da densidade pop.
            x_inicial = x_inicial + dx_inicial
            x = x_inicial
            y = x_inicial + E

            ! Calculando a variação da densidade populacional, utilizando um número arbitrário de gerações como geração máxima
            DO G = 1, 30

                x = taxa * x * (1.d0 - x)
                y = taxa * y * (1.d0 - y)

                d = ABS(y - x)

                write((10*i)+j,*)G,d
            enddo
        enddo
    enddo
    end program distancia
```

Pode-se então achar o expoente de Lyapunov para comparar com o decaimento da distância. O expoente de Lyapunov é um quantificador do caos determinístico. Isso é feito realizando um ajuste exponencial no gráfico da distância e considerando que, quando i >> 1:

$$d_i \sim e^{-\lambda i} \quad (2)$$

D. Lyapunov

Pode-se estimar o expoente de Lyapunov de outra forma, fazendo a derivada do mapa logístico (1) de acordo com a equação:

$$\lambda \approx \frac{1}{n}\sum_{j=0}^{n-1} \quad ln\left|G'\left(x_j\right)\right|$$

(3)

Para melhorar a precisão pode-se fazer um programa em FORTRAN que calcula o valor do expoente para r = 2,5 utilizando cinco valores de densidade populacional inicial diferentes e calculando o seu valor médio:

```
      ! Programa que calcula a estimativa do expoente de Lyapunov pelo ln|G'(xo)|
-> G'(xo) = r (1 - 2x)

        program lyapunov
        IMPLICIT NONE

        ! Variáveis
        REAL*8 :: pop            ! densidade populacional (x)
        REAL*8 :: pop_inicial  ! densidade populacional inicial
        REAL*8 :: dpop_inicial ! Variação da densidade populacional inicial
        REAL*8 :: pop_linha    ! derivada da densidade populacional
        REAL*8 :: taxa         ! taxa de variação populacional atual
        INTEGER*8 :: n         ! número de gerações
        INTEGER*8 :: i         ! contador

        ! Abrindo arquivos para impressão de variáveis
        open(11, file = "lyapunov_x018.dat")
        open(12, file = "lyapunov_x036.dat")
        open(13, file = "lyapunov_x054.dat")
        open(14, file = "lyapunov_x072.dat")
        open(15, file = "lyapunov_x090.dat")

        ! Inicializando variáveis
        taxa = 2.5d0
        dpop_inicial = 0.18d0
        pop_inicial = 0.d0

        ! Queremos graficar a derivada da densidade para 5 x iniciais
diferentes entre 0 e 1
        DO i = 1,5

            ! calculando valor inicial novo da densidade pop.
            pop_inicial = pop_inicial + dpop_inicial
            pop = pop_inicial

            ! Calculando a derivada da variação da densidade populacional,
utilizando um número arbitrário de gerações como geração máxima
            DO n = 1, 30

                pop = taxa * pop * (1.d0 - pop)
                pop_linha = taxa * (1 - (2.d0 * pop))
```

```
                write(10+i,*)n,LOG(ABS(pop_linha))
          enddo
enddo

end program lyapunov
```

2.2. Dobras de período e caos.

Ao utilizar uma taxa de variação maior do que 3 percebe-se uma mudança. Essa mudança é uma nova instabilidade nas respostas obtidas pelos programas e percebem-se novos pontos fixos. Estes pontos estão em uma segunda ordem do mapa logśtico (1), ou seja:

$$x^* \equiv G(G(x^*)) \tag{4}$$

A. Pontos FIxos

Novamente é feita uma análise gráfica do mapa logístico, considerando a equação (4), dessa vez observando o que acontece quando r > 3, utilizando r = 2,9; r = 3 e r = 3,1.

B. Diagrama de Bifurcação

Fazendo um gráfico do ponto fixo em função da taxa de crescimento r pode-se observar a introdução de novos pontos fixos eventualmente levando ao caos. Como discutido em sala, pode ser utilizado o seguinte código para chegar no gráfico:

```
! Programa que constroí o diagrama de bifurcação do mapa logístico do crescimento
de uma população

      program bifurcacao
      IMPLICIT NONE

      ! Variáveis
      REAL*8 :: pop ! Densidade populacional (x)
      REAL*8 :: pop_inicial !densidade populacional inicial
      REAL*8 :: taxa ! taxa de variação populacional
      REAL*8 :: var_taxa ! Variação da taxa populacional
      INTEGER*8 :: Nconverge ! número de iterações para ter convergência para x*
      INTEGER*8 :: Ncoleta ! número de pontos coletados para cada iteração
      INTEGER*8 :: i ! contador
      INTEGER*8 :: j ! contador

      ! Abrindo arquivos para impressão de variáveis
      open(20, file = "bifurcacao.dat")

      ! Inicializando variáveis
      var_taxa = 0.001d0
      Nconverge = 50
      Ncoleta = 1000
```

```
      taxa = 2.8d0

      ! Começamos do valor que seria x* não convencional, diminuindo as iterações
necessárias para convergir
      pop_inicial = 1 - (1/taxa)
      pop = pop_inicial

      do while (taxa < 3.99)

      ! Iteramos o mapa até obtermos a convergência para x*
      do i = 1,Nconverge

            pop = taxa * pop * (1 - pop)

      enddo

      ! Coletamos então o ponto x* (coletamos vários pois não sabemos quantos
pontos divergentes temos para cada r)
      do j = 1, Ncoleta

            write(20,*)taxa, pop
            pop = taxa * pop * ( 1 - pop)

      enddo

      ! Iteramos a taxa de variação:
      taxa = taxa + var_taxa

      enddo

      end program bifurcacao
```

C. Constante de Feigenbaum

Analisando cascatas de duplicação em equações logísticas percebe-se que os valores onde ocorrem as bifurcações convergem geometricamente, de modo que a sua razão se aproxima da constante de Feigenbaum δ. Verificar isso para o gráfico de bifurcações do crescimento populacional pode ser feito encontrando-se os valores de r em que passamos a obter quatro pontos fixos (período 2^2, r_2) e oito pontos fixos (período 2^3, r_3), e calculando:

$$\delta = \frac{r_2 - r_1}{r_3 - r_2} \tag{5}$$

D. Caos

Neste item deve-se novamente analisar o crescimento populacional, a distância e a constante de Iyapunov dentro da área do caos, localizada após r_c = 3,569946.

Para tal, modifica-se levemente os códigos da questão 1, como a seguir:

```
! Queremos graficar a densidade para 3 taxas de variação diferentes
      DO i =1,5
```

```
      taxa  = taxa_var(i)
      dpop_inicial = 0.18d0
      pop_inicial = 0.d0

      ! Queremos graficar a densidade para 5 x iniciais diferentes entre 0 e 1
      DO j=1,5

            ! calculando valor inicial novo da densidade pop.
            pop_inicial = pop_inicial + dpop_inicial
            pop = pop_inicial

            ! Calculando a variação da densidade populacional, utilizando um
número arbitrário de gerações como geração máxima
            DO G = 1, 30

                  pop = taxa * pop * (1.d0 - pop)

                  write((10*i)+j,*)G,pop
      enddo
      enddo

      end program populacao


! Queremos graficar a densidade para 3 taxas de variação diferentes
      DO i =1,5

      taxa  = taxa_var(i)
      dx_inicial = 0.18d0
      x_inicial = 0.d0

      ! Queremos graficar a densidade para 5 x iniciais diferentes entre 0 e 1
      DO j=1,5

            ! calculando valor inicial novo da densidade pop.
            x_inicial = x_inicial + dx_inicial
            x = x_inicial
            y = x_inicial + E

            ! Calculando a variação da densidade populacional, utilizando um
número arbitrário de gerações como geração máxima
            DO G = 1, 30

                  x = taxa * x * (1.d0 - x)
                  y = taxa * y * (1.d0 - y)

                  d = ABS(y - x)

                  write((10*i)+j,*)G,d
            enddo
      enddo
      enddo
      end program distancia
```

2.3. Modelo predador-presa

Considera-se agora um sistema em que há duas espécies, uma de predadores e outra de presas, em que a população de uma está diretamente relacionada com a outra.

Um exemplo seria o modelo Lotka-Volterra:

$$x' = ax - byx \tag{6}$$
$$y' = -cy + dxy \tag{7}$$

onde x são as presas e y os predadores.

Este é um modelo interessante, mesmo que restritivo e não completamente realístico. Neste exercício serão considerados a = ⅔, b = 4/3 e c = d =1 .

B. Lotka-Volterra

Pode-se utilizar o método de Runge-Kutta na quarta ordem para integrar as equações de Lotka-Volterra (6) e (7), de forma que suas formas discretas se tornam:

$$x_{i+1} = x_i + \frac{\Delta t}{6}(F_x^1 + 2(F_x^2 + F_x^3) + F_x^4) \tag{8}$$
$$y_{i+1} = y_i + \frac{\Delta t}{6}(F_y^1 + 2(F_y^2 + F_y^3) + F_y^4)$$

(9)

sendo:

$$F_x^1 = ax - bxy \tag{10}$$
$$F_y^1 = -cy + dxy \tag{11}$$
$$F_x^j = a\,T_x^j - bT_x^jT_y^j \tag{12}$$
$$F_x^j = -c\,T_y^j + dT_x^jT_y^j \tag{13}$$
$$T_x^j = x_i + \Delta t C^j\,F_x^{(j-1)} \tag{14}$$
$$T_y^j = y_i + \Delta t C^j\,F_y^{(j-1)} \tag{15}$$

Visto isso é possível escrever um programa como o abaixo:

```
! Programa que utiliza o método RK4 para integrar as equações de Lotka-Volterra para a interação entre predadores e presas

program presa_predador
IMPLICIT NONE

!Variáveis
real*8 :: presa          ! Número de presas na geração atual
real*8 :: predador            ! Número de predadores na geração atual
real*8, dimension(4) :: Fps   ! Aproximação de taxa de crescimento para presas
real*8, dimension(4) :: Fpd   ! Aproximação de taxa de crescimento para predadores
```

```
      real*8, dimension(4) :: AJUps ! Número de presas ajustado para aproximação
      real*8, dimension(4) :: AJUpd ! Número de presas ajustado para aproximação
      real*8, dimension(4) :: C     ! Constantes de integração
      real*8 :: t            ! Tempo decorrido
      real*8 :: t_max        ! Tempo máximo decorrido
      real*8 :: dt                  ! Variação do tempo
      integer*8 :: j                ! Contador
      real*8, external :: f         ! Função de evolução da população da presa
      real*8, external :: g         ! Função de evolução da população do predador

      ! Abrindo arquivos para imprimir variáveis
      open(1, file = "presas_vs_tempo.dat")
      open(2, file = "predadores_vs_tempo.dat")
      open(3, file = "presas_vs_predadores.dat")


      ! Inicializando variáveis
      presa = 2
      predador = 1
      t = 0.d0
      t_max = 100.d0
      dt = 0.001d0
      C(2) = 0.5d0
      C(3) = 0.5d0
      C(4) = 1.d0

      DO WHILE (t < t_max)

      ! Calculando as aproximações de taxa de variação iniciais
      Fps(1) = f(presa,predador)
      Fpd(1) = g(presa,predador)

      ! Calculando os outros ajustes e aproximações de taxa de variação
      DO j = 2,4

            AJUps(j) = presa + ( dt * C(j) * Fps(j-1) )
            AJUpd(j) = predador + ( dt * C(j) * Fpd(j-1) )

            Fps(j) = f(AJUps(j),AJUpd(j))
            Fpd(j) = g(AJUps(j), AJUpd(j))

      enddo

      ! Calculando populações em determinado tempo
      presa = presa + ( ( dt / 6.d0) * ( Fps(1) + ( 2 * ( Fps(2) + Fps(3) ) ) +
Fps(4) ) )
      predador = predador + ( ( dt / 6.d0) * ( Fpd(1) + ( 2 * ( Fpd(2) + Fpd(3) )
) + Fpd(4) ) )

      ! Calulando tempo
      t = t + dt

      ! Imprimindo resultados
      write(1,*)t, presa
      write(2,*)t,predador
```

```
write(3,*)presa,predador


enddo


end program presa_predador

real*8 function f(x,y)
IMPLICIT NONE

!Variáveis:
real*8, intent(in)  :: x !Variável de entrada das presas
real*8, intent(in)  :: y !Variável de entrada dos predadores
real*8 :: a !Taxa de crescimento das presas
real*8 :: b ! Taxa de crescimento dos predadores em relação as presas

a = 2.d0/3.d0
b = 4.d0/3.d0

f = ( a * x ) - (b * x * y )

RETURN
end

real*8 function g(x,y)
IMPLICIT NONE

!Variáveis:
real*8, intent(in)  :: x !Variável de entrada das presas
real*8, intent(in)  :: y !Variável de entrada dos predadores
real*8 :: c !Taxa de crescimento dos pedadores
real*8 :: d ! Taxa de crescimento das presas em relação aos predadores

c = 1.d0
d = 1.d0

g = - ( c * y ) + (d * x * y )

RETURN
end
```

D. Espaço de fase

É possível calcular o espaço de fase do sistema utilizando o código do ítem b:

E. Lebres e linces

Pode-se comparar resultados obtidos pelo modelo criado com resultados observados na vida real. Assim, com dados obtidos no Canadá sobre linces e lebres inicia-se o programa com a = 0,481; b = 0,025; c = 0,927; d = 0,028; x_o = 30 e y_o = 4.

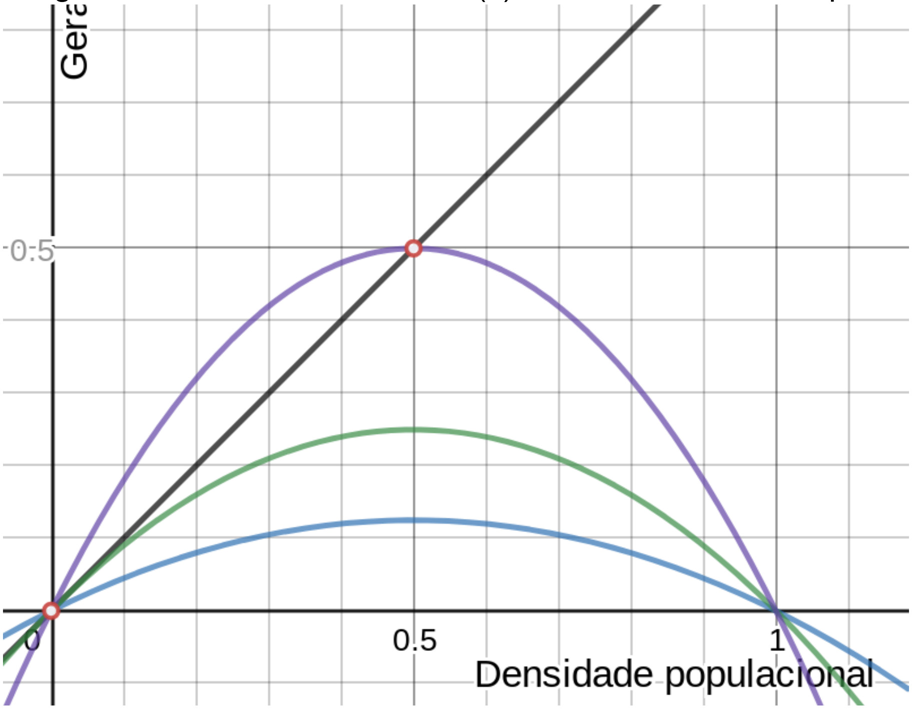
3. RESULTADOS

3.1. Ponto fixo de período 1

A. Ponto Fixo

Fazendo um gráfico das curvas $f(x) = x$ e o mapa logístico (1) e obtém-se:

Imagem 1 - Gráfico de f(x) = x e do mapa logístico (1)



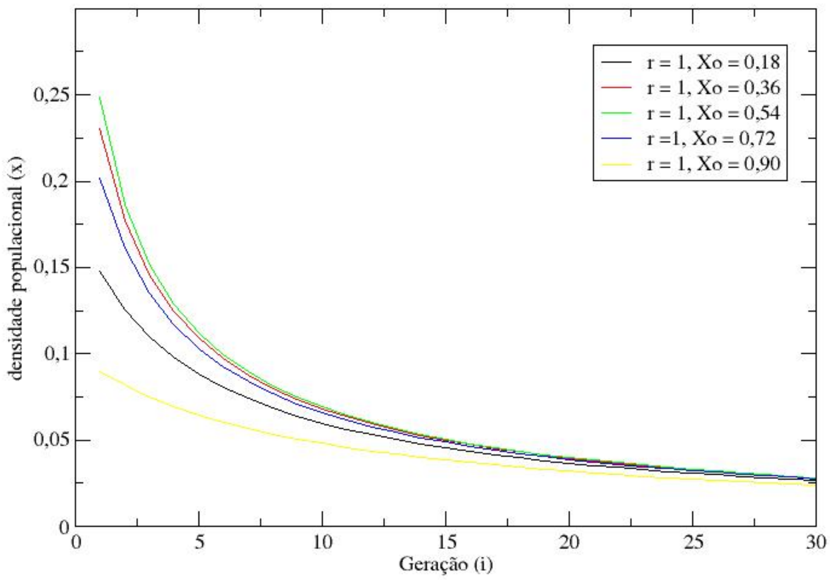
Fonte: Elaborado pelo compilador utilizando DESMOS[6]

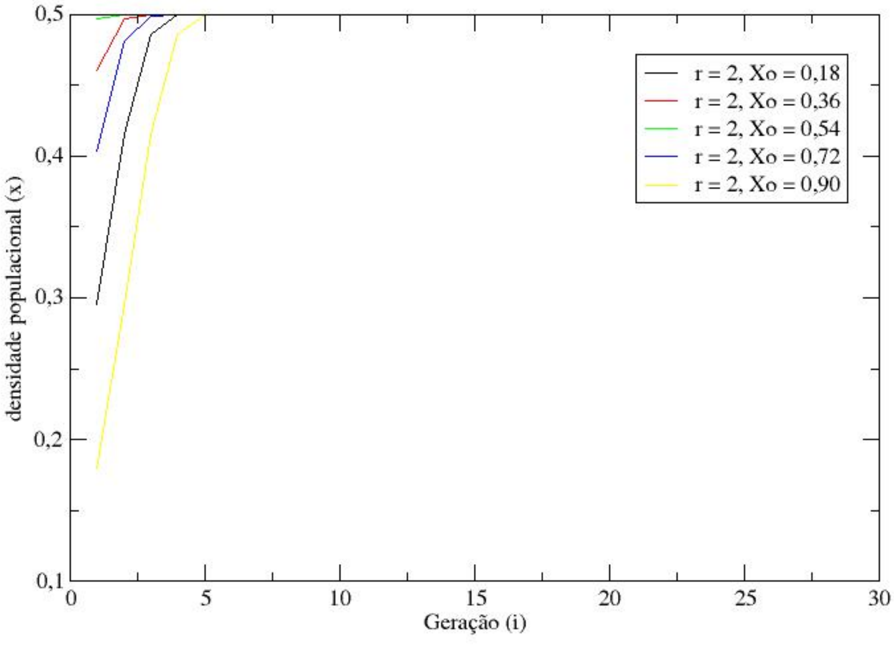
Nota-se que apenas na curva de maior r (lilás) as curvas se cruzam mais de uma vez. Para as curvas r = 0,5 (azul) e r = 1 (verde) apenas existe a solução 0. Portanto percebe-se que apenas existe uma solução não trivial para r > 1.

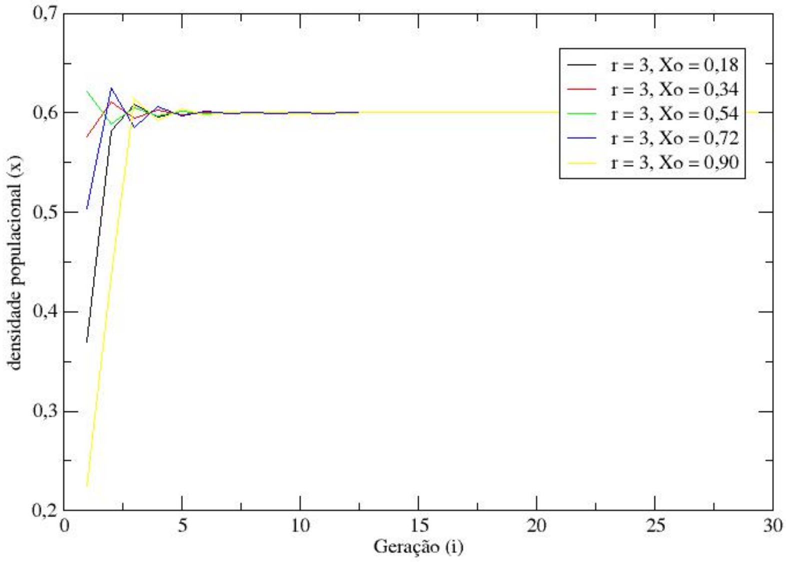
B. Dependência do ponto fixo.

Ao executar o programa descrito obtemos:

Imagem 2 - Gráfico da densidade populacional em função das gerações com diferentes taxas de variação.


r = 1, Xo = 0,18
r = 1, Xo = 0,36
r = 1, Xo = 0,54
r =1, Xo = 0,72
r = 1, Xo = 0,90
densidade populacional (x)
Geração (i)


r = 2, Xo = 0,18
r = 2, Xo = 0,36
r = 2, Xo = 0,54
r = 2, Xo = 0,72
r = 2, Xo = 0,90
densidade populacional (x)
Geração (i)



Fonte: Elaborado pelo compilador utilizando xmgrace[1]

Analisando o gráfico pode-se ver que não importa o valor inicial da população, se a taxa de crescimento é a mesma o crescimento populacional se direciona para o mesmo ponto. Assim é possível observar os pontos fixos encontrados no ítem anterior.

O valor de um ponto fixo deveria ser dado por:

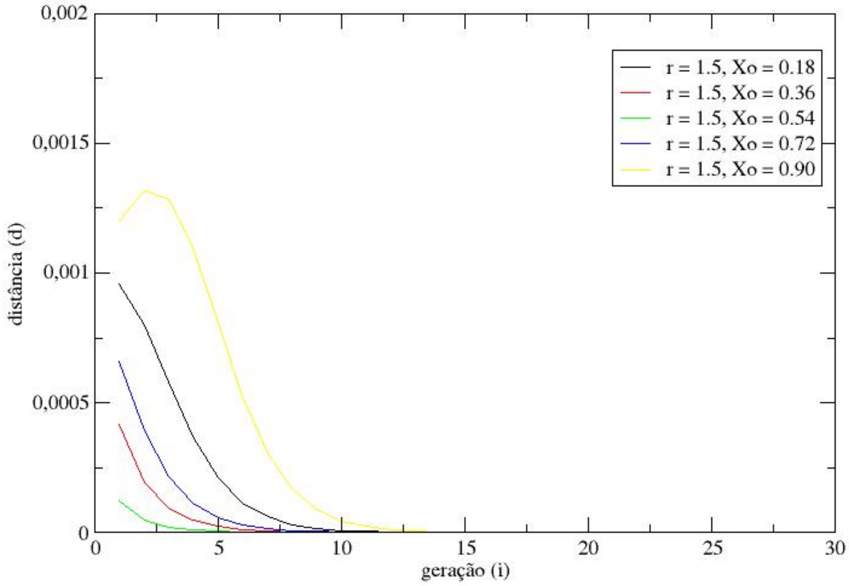
$$x^* = 1 - r^{-1} \quad (16)$$

Pelos gráficos acima vê-se que esta relação é mantida.

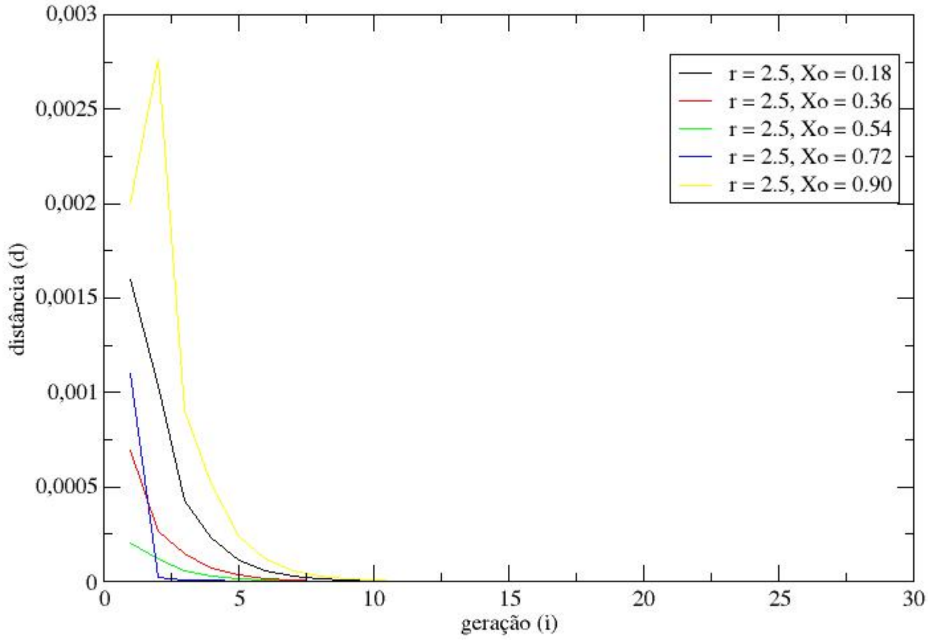
C. Distância

Executando o programa obtém-se:

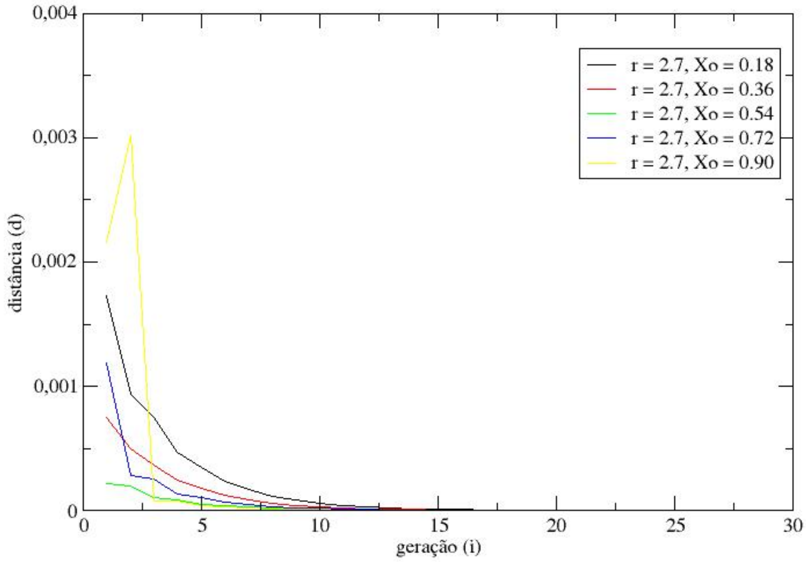
Imagem 3 - Gráfico da diferença de densidade populacional em função das gerações com diferentes taxas de variação.



0,002
0,0015
0,001
0,0005
0
distância (d)
0
5
10
15
20
25
30
geração (i)
r = 1.5, Xo = 0.18
r = 1.5, Xo = 0.36
r = 1.5, Xo = 0.54
r = 1.5, Xo = 0.72
r = 1.5, Xo = 0.90



0,003
0,0025
0,002
0,0015
0,001
0,0005
0
distância (d)
0
5
10
15
20
25
30
geração (i)
r = 2.5, Xo = 0.18
r = 2.5, Xo = 0.36
r = 2.5, Xo = 0.54
r = 2.5, Xo = 0.72
r = 2.5, Xo = 0.90



Fonte: Elaborado pelo compilador utilizando xmgrace[1]

Podemos perceber que de fato, eventualmente a distância começa a diminuir através de uma função exponencial. Fazendo a regressão linear obtiveram-se os resultados:

para r = 1,5, λ= 0,69304; para r = 2,5, λ = 0,69324; para r = 2,7, λ = 0,35714.

D. Lyapunov

Executando o programa e calculando o valor médio obteve-se:

$$\lambda = 0{,}693147181$$

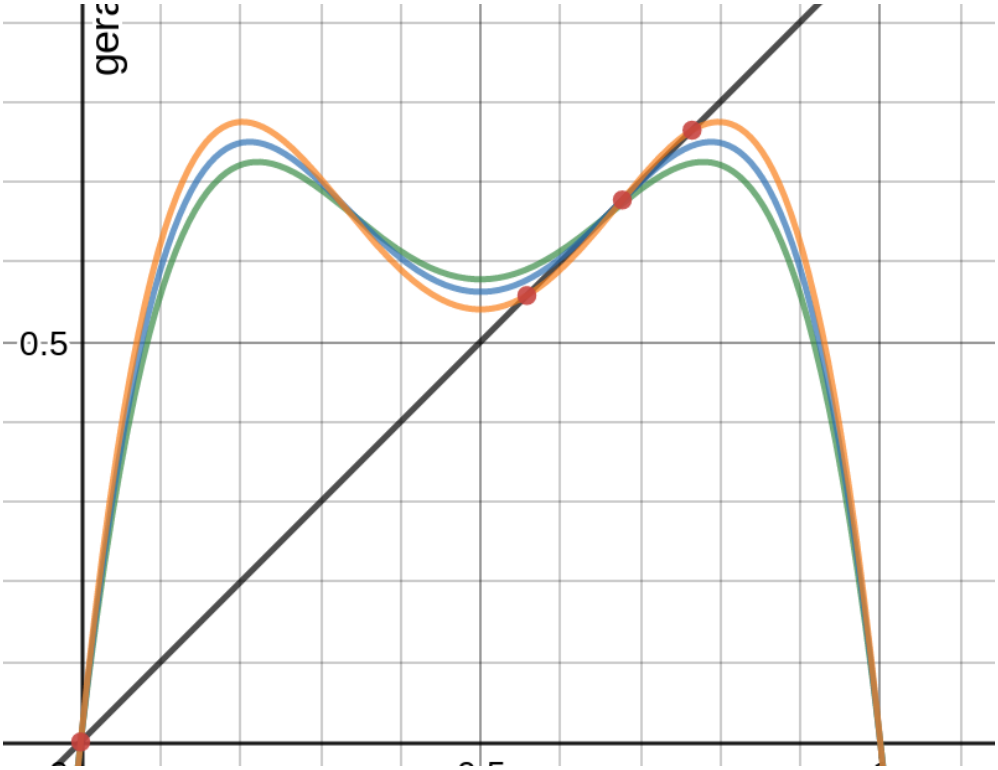
que é um valor próximo aos obtidos nos itens anteriores.

3.2. Dobras de Período e caos

A. Pontos fixos

Foi feito o seguinte gráfico

Imagem 4 - Gráfico de f(x) = x e o mapa logístico (4).



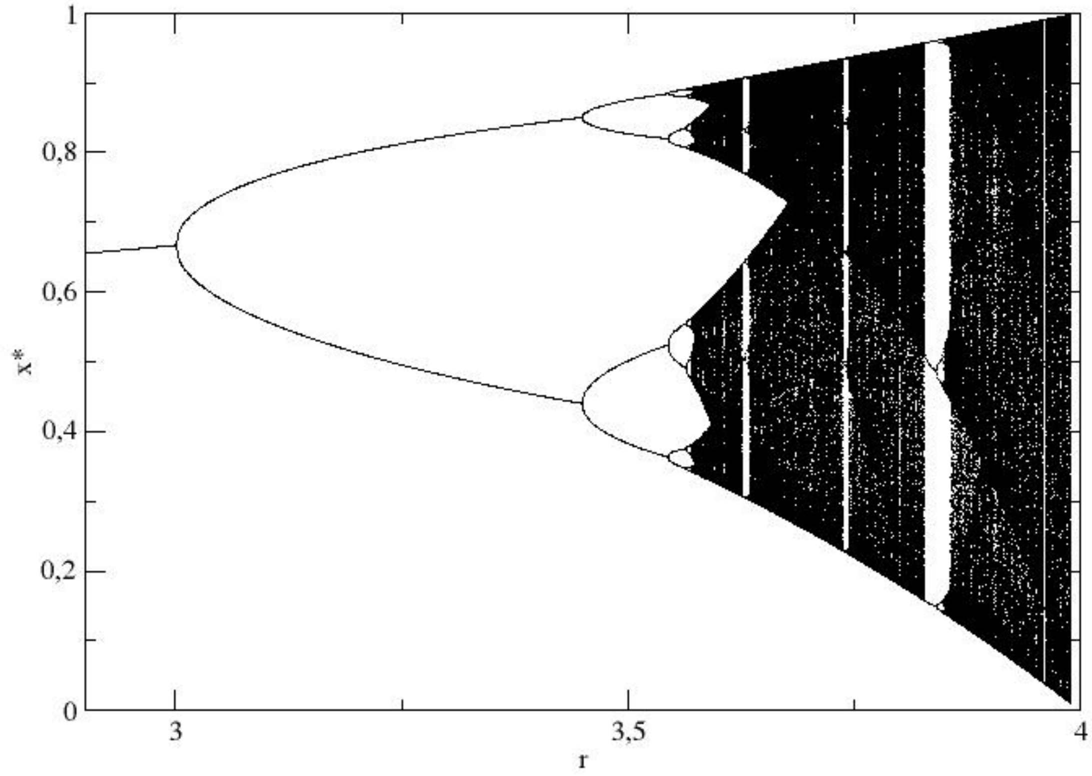
Fonte: Elaborado pelo compilador utilizando DESMOS[6]

Vê-se nesse caso que para valores acima de 3 passamos a ter 4 soluções, o dobro da quantidade anterior, sendo assim, estamos agora no período 2.

B. Diagrama de bifurcação

Ao executar o programa descrito obtivemos:

Imagem 5 - Gráfico do diagrama de bifurcação



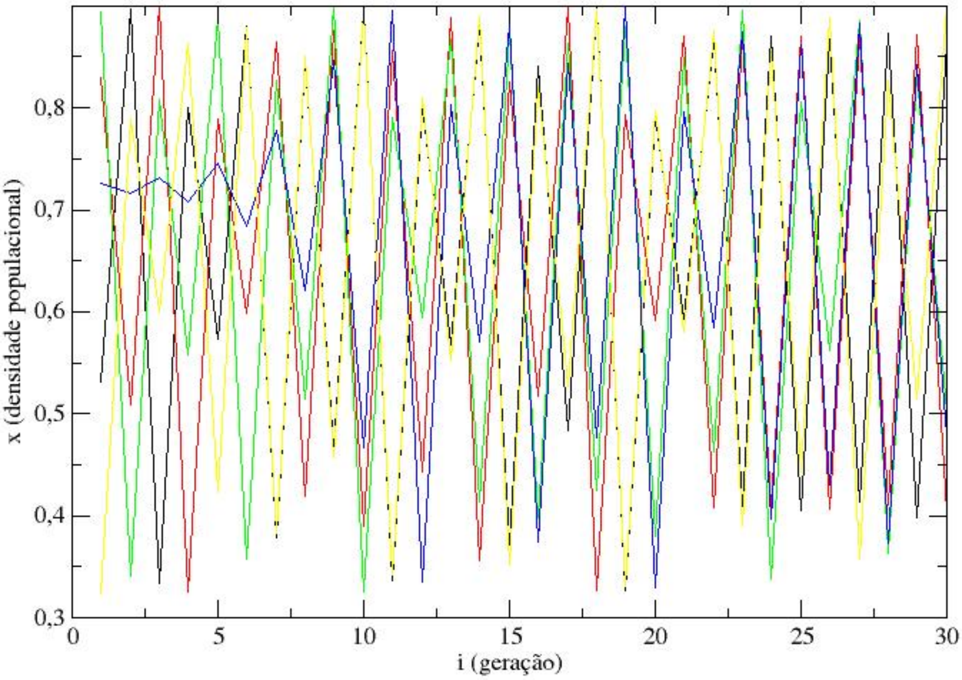
Fonte: Elaborada pelo compilador utilizando xmgrace[1]

C. Constante de Feigenbaum

Observando o gráfico chegamos à: $r_1 = 3$, $r_2 = 3{,}45$ e $r_3 = 3{,}55$.
A partir disso obtém-se:
$\delta = 4{,}5$

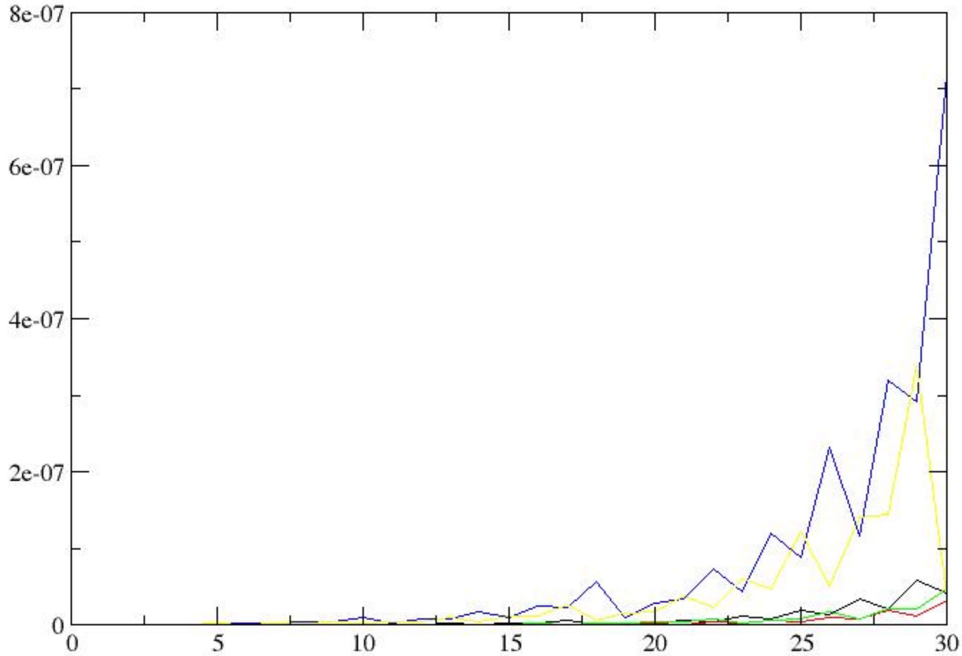
D. Caos

Imagem 6 - Gráfico da densidade populacional em função das gerações para o caos



Fonte: Elaborado pelo compilador utilizando xmgrace[1]

Imagem 7 - Gráfico da diferença de densidade populacional em função das gerações para o caos



Fonte: Elaborado pelo compilador utilizando xmgrace[1]

Através desses gráficos podemos ver que nessa região temos o caos e que quando temos dois valores muito próximos eles podem divergir rápidamente.

3.3. Modelo predador-presa

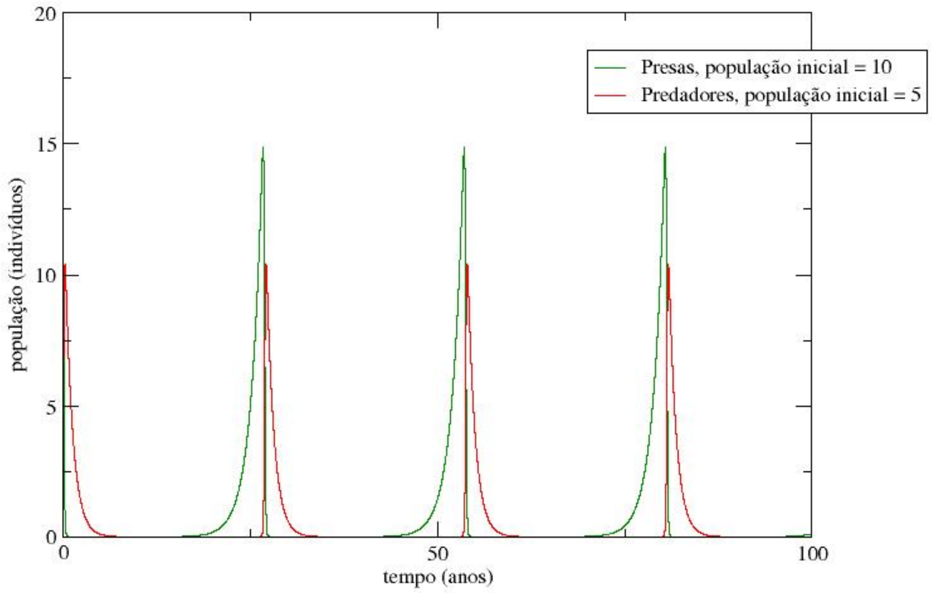
A. Constantes

No modelo de Lotka-Volterra a constante *a* indica a taxa de crescimento da população das presas, *b* indica a taxa de mortalidade das presas com relação aos predadores, *c* indica a taxa de mortalidade dos predadores em relação a mortalidade das presas e *d* indica a taxa de natalidade dos predadores em relação a disponibilidade de presas.

C. presa-predador

Ao rodar o programa do item b, obtemos:

Imagem 7 - Gráfico do crescimento da população de presas e predadores em função do tempo.



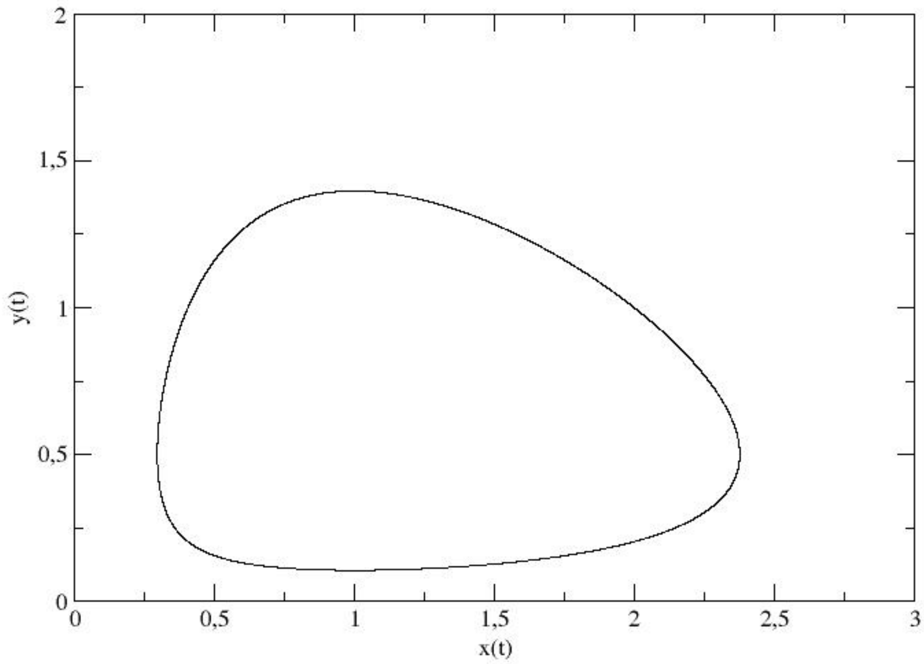
Fonte: Elaborado pelo compilador utilizando xmgrace[1]

Percebe-se que agora que as populações estão interagindo umas com as outras elas oscilam ao invés de estagnar em um ponto fixo. Isso era esperado, pois conforme os predadores aumentam sua população mais presas morrem, até que os predadores começam a morrer. Conforme sua população diminui a população das presas também diminui.

D.Espaço de Fase

Executando o programa obtemos:

Imagem 8 - Gráfico do espaço de fase da relação predador-presa



Fonte: Elaborado pelo compilador utilizando xmgrace[1]

Conforme ajustamos as populações iniciais recebemos formas parecidas para o espaço de fase. Assim o ponto fixo seria o ponto de convergência dos espaços de fase.

4. REFERÊNCIAS

[1]Grace Home. WIS Plasma Lab. Disponível em: <http://plasma-gate.weizmann.ac.il/Grace/>.

[2]Hoyos; José A. **Introdução a programação - Notas de aula** Disponível em:

<http://www.ifsc.usp.br/~hoyos/courses/2019/7600017/intro.pdf>

[3] Hoyos; José A. **Projeto 5 - Dinâmica Populacional** Disponível em:

<http://www.ifsc.usp.br/~hoyos/courses/2019/7600017/p5.pdf >

[4]Caos| Sistemas Determinsticos. Disponível em:

<http://www.if.ufrgs.br/fis01038/caos/caos_det/caos_det.htm>.

[5]Dinmica no linear e caos. Centro de F. Disponível em:

<http://cftc.cii.fc.ul.pt/PRISMA/capitulos/capitulo2/modulo1/topico8.php>.

[6]Explore math with Desmos. Desmos.com. Disponível em:

<https://www.desmos.com/>.