

# Relatório 5

## Dinâmica Populacional

Aluno Mateus Piovezan Otto  
Nº USP 10260480  
Disciplina Introdução à Física Computacional  
Professor Eric de Castro e Andrade

### Introdução

Neste trabalho, estamos interessados em descrever a dinâmica populacional através de modelos matemáticos simples, para os quais uma solução numérica é facilmente obtida por iterações ou métodos mais elaborados, como o de *Euler-Cromer*.

Esses modelos são interessantes à medida que incluem restrições ambientais e ecológicas ao crescimento das espécies, mimetizando o comportamento real de como populações modificam-se e interagem na natureza.

Em particular, abordaremos dois tópicos: o crescimento de uma espécie isolada, modelado através do *mapa logístico* e a oscilação populacional de presas e predadores, com interação descrita pelas equações de *Lotka-Volterra*. Em ambas situações, empregaremos ferramentas da teoria de sistemas dinâmicos, como os expoentes de Lyapunov, para estudar a estabilidade das soluções obtidas.

## 1 Métodos

### Pontos fixos de período 1

Neste exercício, queremos analisar o crescimento populacional de uma espécie isolada e na ausência de predadores. Para isso, vamos construir um modelo matemático que abstraia nossa percepção da dinâmica dessa população, satisfazendo, para tanto, duas hipóteses fundamentais:

1. Mais animais implica em número maior de descendentes.
2. Existe um limite superior para o número de animais que podem habitar sustentavelmente o ambiente.

A princípio, poderíamos utilizar o seguinte modelo:

$$\frac{dN(t)}{dt} = \alpha N(t) \tag{1}$$

onde  $N(t)$  é o número de animais em um determinado tempo e  $\alpha$  quantifica a taxa de reprodução. Como estamos interessados em resolver numericamente as equações do modelo, discretizamos o tempo em unidades características do

sistema (nesse caso, o intervalo entre gerações  $\Delta t$ ) e reescrevemos a derivada em termos da derivada para frente. Com isso, vem que:

$$N_{i+1} = \underbrace{(1 + \alpha\Delta t)}_r N_i = rN_i \quad (2)$$

onde  $N_i$  é o número de animais na  $i$ -ésima geração. Com efeito, sabemos que a solução para tal modelo é uma exponencial crescente e que, por isso, nossa hipótese de uma população finita não está contemplada. Para corrigir o modelo, inserimos um novo termo, de modo que  $N_{i+1}$  pode ser escrita como:

$$N_{i+1} = rN_i \left(1 - \frac{N_i}{N_{max}}\right) \quad (3)$$

Com essa correção impomos um limite superior ( $N_{max}$ ) para a população, a partir do qual ela não cresce mais. Para mantermos a independência relativa às dimensões do problema, normalizamos o termo  $N_i$ , definindo  $x_i \equiv N_i/N_{max}$ :

$$x_{i+1} = rx_i(1 - x_i) \quad (4)$$

Essa equação é denominada *mapa logístico* e descreve um crescimento populacional sob as hipóteses restritivas que enumeramos. Podemos reescrever a Equação 4 do seguinte modo:

$$x_{i+1} = G(x_i) = rx_i(1 - x_i) \quad (5)$$

Onde definimos o *mapa*  $G(x)$ . Estaremos interessados, para efeito de análise do modelo, em determinar para quais valores de  $x_i$  a população não se altera em gerações subsequentes, isto é, para quais  $\bar{x}$ :

$$\bar{x} = G(\bar{x}) \quad (6)$$

Esses  $\bar{x}$  são denominados *pontos fixos* do *mapa*. Para o mapa logístico  $G$ , as soluções são:

$$\begin{aligned} \bar{x} &= 0 \\ \bar{x} &= 1 - \frac{1}{r} \end{aligned}$$

Como  $r > 1$ ,  $\bar{x} \in [0, 1]$ , como esperávamos. Note que ambas soluções independem de  $x_i$  em qualquer geração, sendo completamente ditadas pelo parâmetro  $r$ .

Feita esta introdução, podemos particularizar os métodos de solução para cada letra do Exercício.

## Letra A

Precisamos calcular as soluções para a Equação 6, que nos fornece os pontos fixos para um determinado valor de  $r$ . Para isso, plotamos, utilizando a biblioteca *matplotlib* da instalação científica padrão do *Python*, os gráficos das seguintes funções:

$$f(x) = x \tag{7}$$

$$g(x) = rx(1 - x) \tag{8}$$

E verificamos as interseções para cada valor de  $r \in \{1, 2, 2.5\}$ .

## Letra B

Neste exercício, precisamos calcular a fração da população máxima ( $x_i$ ) existente em cada geração para distintos valores de população inicial ( $x_0$ ) e do parâmetro  $r$ , que dita a reprodução da população. Para simular, escolhemos valores de  $x_0 \in \{0.3, 0.6, 0.9\}$  para cada  $r \in \{1, 2, 2.5\}$ .

Podemos agora, descrever a implementação do programa. Primeiro, definimos as seguintes variáveis:

```
real*8 r, x1, x2, x3
integer niter, i
parameter (niter = 20)
```

Associadas ao  $r$ , aos valores de  $x_0$  escolhidos e a parâmetros da simulação, como o índice da geração ( $i$ ) e o número de iterações ( $N_{iter}$ ). Aqui, o número de iterações foi escolhido como  $N_{iter} = 20$  porque a população tende a um valor fixo após poucas iterações ( $\approx 10$ ). Após isso, abrimos o arquivo de saída e inicializamos as variáveis:

```
!Output
open(10, file='exerc1bR2.dat')

!Condição inicial
x1 = 0.3d0
x2 = 0.6d0
x3 = 0.9d0
i = 0
r = 2.0d0
```

E iteramos  $x_i$  usando o mapa logístico  $G$ , armazenando o resultado em cada geração:

```
do while (i .lt. niter)
  write(10,*)i, x1, x2, x3
  x1 = x1*r*(1.0d0-x1)
  x2 = x2*r*(1.0d0-x2)
  x3 = x3*r*(1.0d0-x3)
  i = i + 1
enddo
```

Com os dados gerados, plotamos a fração da população máxima na  $i$ -ésima geração ( $x_i$ ) em função do número de iterações. A partir destes, verificamos a evolução da população, atestando se o valor final correspondia aos pontos fixos calculados na *Letra A* para cada  $r$  e se estes eram, de fato, independentes da população inicial ( $x_0$ ).

### Letra C

Neste exercício, queremos estudar a estabilidade do mapa logístico. Podemos fazer isso, calculando a distância entre trajetórias adjacentes, cujos valores iniciais diferem por um real diminuto, que denominaremos  $\varepsilon$ . Essa distância pode ser matematicamente expressa como:

$$d_i = \left| G^{(i)}(x_0 + \varepsilon) - G^{(i)}(x_0) \right| \quad (9)$$

onde  $G^{(i)}(x_j) = G(G^{(i-1)}(x_j))$  é uma composição sucessiva do mapa aplicada ao ponto  $x_j$ .

A convergência ou divergência de  $d_i$  reflete nossa expectativa de que trajetórias adjacentes são, ou não, estáveis quanto a perturbações de  $x_0$ . Para calcular  $d_i$ , fixamos  $r = 2.5$  e tomamos  $x_0 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ , calculando  $d_i$  em cada iteração e armazenando o resultado num arquivo externo.

Agora, descrevemos os detalhes da implementação. Declaramos as variáveis do problema:

```
real*8 r, x1, x2, x3, x4, x5, g
real*8 d1, d2, d3, d4, d5, eps
real*8 x10, x20, x30, x40, x50
parameter (eps = 1.0d-4)

integer niter, i
parameter (niter = 100)
```

onde  $x1$  e  $x10$  são a população perturbada e não perturbada, respectivamente,  $d1$  é a distância, como definida anteriormente e  $eps$  é a magnitude ( $\varepsilon$ ) de perturbação. A escolha de  $\varepsilon$  é arbitrária, uma vez que a *taxa de convergência* de  $d_i$  independe da magnitude da perturbação. Isso ficará claro na *Letra D*, quando abordarmos os *expoentes de Lyapunov*.

Após isso, inicializamos as variáveis (apenas uma inicialização é mostrada) e iteramos as distâncias, atentando às composições sucessivas do mapa  $G$ :

```
open(10, file='exerc1cR25.dat')
x1 = 0.1d0
x10 = x1 + eps
i = 1
r = 2.5d0

do while (i .lt. niter)
```

```

        d1 = dabs(g(x10, r) - g(x1, r))
        x1 = g(x1, r)
        x10 = g(x10, r)

        write(10, *)i, d1
        i = i +1
    enddo

```

Onde  $g(x, r)$  é o mapa, definido como a seguinte função:

```

real*8 function g(x, r)

real*8 x,r
g = r*x*(1.0d0-x)

return

end function g

```

Com os dados gerados, plotamos os gráficos da distância  $d_i$  em função do número de iterações e verificamos que  $d_i$  decai exponencialmente.

Feito isso, ajustamos à cada curva um modelo exponencial da forma:

$$d_i = \alpha e^{\beta i} \quad (10)$$

Utilizando mínimos-quadrados não-linear, mais especificamente o *método de Levenberg-Marquardt*. Nesse caso, como nosso modelo é representativo do decaimento apenas após um número suficientemente elevado de iterações, desconsideramos os 10 primeiros pontos de  $d_i$ , ajustando aos 90 pontos remanescentes.

Para estimar o erro associado à cada coeficiente do ajuste, utilizamos o desvio padrão de  $1\sigma$ , calculado pela raiz quadrada dos termos diagonais da matriz de covariância do ajuste.

## Letra D

Outra maneira de compreender o decaimento de  $d_i$  e analisar a estabilidade (convergência ou divergência) das trajetórias é através do *expoente de Lyapunov*, definido como:

$$\lambda = \frac{1}{n} \sum_{j=0}^{n-1} \ln |G'(x_j)| \quad (11)$$

onde  $G'(x_j)$  é a derivada do mapa aplicada ao ponto  $x_j$ , com  $x_j = G^j(x_0)$ . Com efeito, este  $\lambda$  é aproximadamente a taxa de decaimento de  $d_i$ , ou seja, deve corresponder aos coeficientes  $\beta$  que obtemos nos ajustes da *letra C*.

Para calcular  $\lambda$ , adotamos procedimento similar à *letra C*: fixamos  $r = 2.5$  e tomamos  $x_0$  diversos no conjunto  $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ .

Quanto à implementação, declaramos as variáveis e parâmetros do problema:

```

!Parâmetros do problema
real*8 r, x1, x2, x3, x4, x5, g, gl
real*8 l1, l2, l3, l4, l5
integer niter, i
parameter (niter = 100000)

```

Abrimos o arquivo de saída e especificamos as condições iniciais:

```

!Output
open(10, file='exerc1dR25.dat')

```

```

!Condição inicial
i = 0
x1 = 0.1d0
x2 = 0.3d0
x3 = 0.5d0
x4 = 0.7d0
x5 = 0.9d0
r = 2.5d0

```

E iteramos  $N_{iter} = 10^5$  vezes segundo a Equação 11:

```

do while (i .le. niter-1)
    x1 = g(x1, r)
    l1 = l1 + dlog(dabs(gl(x1, r)))

    x2 = g(x2, r)
    l2 = l2 + dlog(dabs(gl(x2, r)))

    x3 = g(x3, r)
    l3 = l3 + dlog(dabs(gl(x3, r)))

    x4 = g(x4, r)
    l4 = l4 + dlog(dabs(gl(x4, r)))

    x5 = g(x5, r)
    l5 = l5 + dlog(dabs(gl(x5, r)))

    i = i + 1
enddo

write(*, *)l1/niter, l2/niter, l3/niter, l4/niter, l5/niter

```

A escolha de  $N_{iter}$  foi guiada pelo fato de que  $d_i$  só apresenta a forma exponencial para um número suficientemente grande de iterações. Desse modo,  $\lambda$  só se aproxima do coeficiente  $\beta$  de decaimento de  $d_i$  tomando um número de iterações elevado.

Após isso, comparamos os valores  $\lambda$  aos coeficientes  $\beta$  ajustados na *letra C*, atestando se nossas escolhas metodológicas produzem resultantes mutuamente consistentes.

## Dobras de período e caos

Neste exercício, nosso intuito é aprofundar o entendimento do mapa logístico, estendendo as técnicas descritas no *Exercício 1* para valores de  $r > 3$ . Para esses casos, o mapa logístico passa a ter pontos fixos de período dois e superiores com soluções não triviais.

Definimos um ponto fixo de período dois como a solução da equação:

$$\bar{x} = G(G(\bar{x})) \quad (12)$$

Que podemos expandir do seguinte modo:

$$r\bar{x}^3 - 2r\bar{x}^2 + (r-1)\bar{x} + \frac{1}{r^2} - 1 = 0 \quad (13)$$

E que tem as seguintes soluções:

$$\bar{x}_0 = 0 \quad (14)$$

$$\bar{x}_1 = 1 - \frac{1}{r} \quad (15)$$

$$\bar{x}_{2,\pm} = \frac{1}{2r} \left( 1 + r \pm \sqrt{-3 - 2r + r^2} \right) \quad (16)$$

Note que a existência de duas soluções (período dois) está condicionada à seguinte restrição:

$$r^2 - 2r - 3 > 0 \quad (17)$$

Que só é satisfeita se  $r > 3$ .

### Letra A

Neste exercício, adotamos procedimento similar àquele da *letra A* do Exercício anterior. Definimos a função identidade e a composição dupla do mapa e verificamos a interseção das duas funções para  $r = 3.2$ . Após isso, comparamos o resultado às soluções previstas para a Equação 13.

### Letra B

Neste exercício, queremos construir o *diagrama de bifurcação* do mapa logístico, que evidencia e clarifica o significado dos pontos fixos de período  $2^n$  e o surgimento do caos determinístico.

Para obter o diagrama, adotamos o procedimento convencional das implementações. Declaramos as variáveis utilizadas:

```

real*8 r, x, rmin, rmax
real*8 g
integer i,j, nr, niter
parameter (niter= 10000)
parameter (nr = 100000)
parameter (rmin = 2.5d0)
parameter (rmax = 4.0d0)

```

onde  $r$ ,  $r_{min}$  e  $r_{max}$ , especificam o valor de  $r$  em cada iteração e seus valores mínimo e máximo,  $g$  é a função do mapa,  $n_r$  é o número de  $r$  gerados e  $n_{iter}$  é o número de vezes que executamos o mapa  $g$  sobre um ponto  $x$ , para obter um ponto fixo.

Como queremos manter o diagrama uniforme, obtendo diversos  $x$  no intervalo  $[0, 1]$ , temos de inicializar a simulação com um  $x_0$  diferente à cada passo. Caso contrário, obteríamos um diagrama de bifurcação apenas com alguns ramos, sem a característica distribuição no eixo  $y$ . Utilizamos, para tal, um gerador de números pseudoaleatórios. Ele é baseado na classe dos geradores conhecidos como *Xorshift*, de período  $2^{1024} - 1$ , suficientemente elevado para nossos propósitos. Com ele, somos capaz de gerar  $x_0$  distintos para cada etapa da simulação.

Assim, iteramos para os diversos valores de  $r$ , obtendo, em cada iteração, um ponto fixo do mapa, fazendo:

```

do i=1,nr
  r = rmin + (rmax-rmin)/dfloat(nr) * dfloat(i)
  ! Gera um x inicial
  call random_number(x)
  do j=1,niter
    x = g(x,r)
  enddo
  write(10,*)r, x
enddo

```

Com os dados gerados, plotamos o diagrama de bifurcação, sobre o qual discutiremos na Seção de Resultados.

### Letra C

O objetivo deste exercício é *estimar* a chamada *constante de Feigenbaum*. Essa constante, definida como:

$$\delta = \lim_{k \rightarrow \infty} \frac{r_k - r_{k-1}}{r_{k+1} - r_k} \quad (18)$$

Está relacionada com a taxa de bifurcação de mapas com um mínimo quadrático. No nosso caso, vamos apenas estimá-la porque somos capazes de calcular apenas o termo com  $k = 2$ , de modo que obteremos:

$$\delta_1 = \frac{r_2 - r_1}{r_3 - r_2} \quad (19)$$

Para fazer isso, tomamos o diagrama de bifurcação da *Letra C* e delimitamos, visualmente, os pontos onde as dobras de período se iniciavam. A incerteza nesse procedimento é baixa porque, dada escolha de  $N_r = 100000$ , os  $r$ 's estão bem próximos e a delimitação das bifurcações é clara.

### Letra D

Neste exercício, queremos estudar a região do caos determinístico acessada por valores de  $r > 3.569946$ .

Primeiro, calculamos e plotamos a fração do valor máximo da população ( $x_i$ ) em função do número de gerações ( $i$ ). O procedimento aqui é similar àquele descrito na *Letra B* do Exercício 1, exceto pela escolha de  $r = 3.75$  acima do limiar do caos.

Após isso, produzimos um gráfico da distância ( $d_i$ ) em função do número de gerações ( $i$ ), mantendo  $\varepsilon = 10^{-4}$ . Aqui, como o comportamento do sistema é caótico, esperamos que as trajetórias diverjam. Seguindo o método delineado no Exercício 1, ajustamos um modelo exponencial da forma:

$$d_i = \alpha e^{\beta i}$$

Aos  $n = 16$  primeiros pares  $(d_i, i)$ . Essa escolha de  $n$  baseia-se no fato de que, para os valores iniciais escolhidos  $\{0.1, 0.3, 0.5\}$ , antes de 16 interações, a distância  $d_i$  é aproximadamente regular. Após isso, observamos diversas irregularidades bruscas que contaminam o ajuste e tornam-o insatisfatório.

Por último, calculamos os *expoentes de Lyapunov* para cada valor de  $x_0 \in \{0.1, 0.3, 0.5\}$  seguindo a mesma metodologia do Exercício 1, e comparamos os expoentes aos coeficientes  $\beta$  do ajuste exponencial às distâncias.

### Modelo predador-presa

Queremos aprimorar o modelo ecológico explorado nos exercícios anteriores, incluindo uma segunda espécie e modelando a influência recíproca de cada uma das populações. Um dos modelos mais simples a ser considerado é o modelo predador-presa ou Lotka-Volterra. Nesse modelo, as populações das duas espécies respeitam às seguintes equações diferenciais

$$\frac{dx}{dt} = ax - bxy \quad (20)$$

$$\frac{dy}{dt} = -cy + dxy \quad (21)$$

$$(22)$$

onde  $x$  é a população de presas,  $y$  a população de predadores e  $a, b, c$  e  $d$  são constantes associadas ao crescimento individual de cada população e à interação

entre elas. Vamos precisar o significado dessas constantes na metodologia da *letra A*

Apesar de mais realístico, esse modelo possui hipóteses que limitam a sua capacidade preditiva:

1. Sempre há comida para as presas.
2. Não há outra fonte de alimento disponível aos predadores, que nunca se saciam.
3. As taxas de variação das populações só dependem do seu tamanho. Interações ou mudanças do meio ambiente não são levadas em conta.

### **Letra A**

Queremos, neste exercício, discutir as constantes  $a$ ,  $b$ ,  $c$  e  $d$  que aparecem no modelo de Lotka-Volterra e seu significado no escopo das simulações. Por analogia ao mapa logístico, podemos traçar alguns paralelos:

- $a$ : caracteriza a taxa reprodutiva da população de presas na ausência de predadores. Sua unidade é  $T^{-1}$ , indicando que tem características de uma frequência reprodutiva.
- $b$ : caracteriza o declínio populacional de presas induzido pela predação. Tem unidades de  $T^{-1}I^{-1}$ , onde  $I$  é o número de indivíduos. Isso significa que ela mensura o declínio populacional de presas relativamente ao tamanho da população de predadores.
- $c$ : caracteriza o declínio populacional natural de predadores, quando na ausência de presas. Tem as unidades de  $a$ .
- $d$ : caracteriza o aumento populacional da população de predadores induzido pela oferta de presas. Tem as unidades de  $b$ .

### **Letra B**

Nesse exercício, vamos utilizar o *método de Euler-Cromer* para resolver as equações diferenciais do modelo de Lotka-Volterra. A justificativa para a utilização deste método e não qualquer outro será clarificada posteriormente, quando abordarmos a periodicidade da solução para o modelo.

Para reescrever as equações, vamos fazer um paralelo com as simulações dinâmicas realizadas nos projetos anteriores. Nelas, tínhamos um par de equações na forma:

$$\frac{dx}{dt} = v(t) \quad (23)$$

$$\frac{dv}{dt} = a(t) \quad (24)$$

$$(25)$$

Para o modelo de Lotka-Volterra, podemos considerar então a influência de  $y$  sobre  $x$ , vendo  $\frac{dy}{dt}$  como o análogo à derivada da velocidade ou fazer o procedimento inverso. Desse modo, podemos, empregando o *método de Euler-Cromer*, escrever as equações que descrevem a população de presas (por isso iteramos os predadores antes):

$$y_{i+1} = (1 - c\Delta t)y_i + d\Delta t x_i y_i \quad (26)$$

$$x_{i+1} = (1 + a\Delta t)x_i - b\Delta t x_i y_{i+1} \quad (27)$$

E, similarmente, iterando a população de presas antes, descrever a população de predadores:

$$x_{i+1} = (1 + a\Delta t)x_i - b\Delta t x_i y_i \quad (28)$$

$$y_{i+1} = (1 - c\Delta t)y_i + d\Delta t x_{i+1} y_i \quad (29)$$

Para implementar a solução em *FORTRAN*, declaramos, inicialmente, as variáveis e parâmetros do modelo:

```
! Parametros do modelo
real*8 a, b, c, d
parameter (a=2.0d0/3.0d0)
parameter (b=4.0d0/3.0d0)
parameter (c=1.0d0)
parameter (d=1.0d0)

! Parametros e variaveis da simulacao
real*8 dt, dur, t
parameter (dt=1.0d-4)
parameter (dur=100.0d0)

! Variaveis
! Populacao de presas
real*8 x0, xi, xf
! Populacao de predadores
real*8 y0, yi, yf
```

Onde  $dt$  é o passo da simulação, que discutiremos na *letra C*. Após isso, especificamos as condições iniciais do sistema:

```

! Condições iniciais
t = 0.0d0
x0 = 0.4d0
y0 = 0.8d0

```

Abrimos os arquivos de saída e iteramos, utilizando o procedimento descrito:

```

! Saida
open(20, file='exerc3b-0.4-0.8.dat')

do while (t .le. dur)
  yi = (1-c*dt)*y0 + d*dt*x0*y0
  xi = (1+a*dt)*x0 - b*dt*x0*yi

  xf = (1+a*dt)*x0 - b*dt*x0*y0
  yf = (1-c*dt)*y0 + d*dt*xf*y0

  write(20, *)t, xi, yf

  t = t + dt
  x0 = xi
  y0 = yf
enddo

```

Isso conclui o nosso algoritmo padrão para resolução das equações de Lotka-Volterra para diversas escolhas de parâmetros e condições iniciais.

### Letra C

Nesse exercício, estamos interessados em utilizar o modelo de Lotka-Volterra para verificar a dinâmica temporal das populações. Para tanto, uma escolha cuidadosa do intervalo temporal dos passos ( $\Delta t$ ) é necessária.

Nossa escolha de  $\Delta t$  é baseada na seguinte heurística:

1. Fixe condições iniciais  $(x_0, y_0)$ . No nosso caso  $(x_0, y_0) = (0.1, 0.1)$ .
2. Tome um  $\Delta t$ . No nosso caso, tomamos  $\Delta t \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ .
3. Construa, para cada  $\Delta t$ , o espaço de fase dos pontos  $x_i, y_i$ .
4. O  $\Delta t$  ótimo é aquele para o qual, qualquer outro  $\Delta t$  menor que ele, produza uma trajetória indistinguível.

Esse procedimento garante que a escolha de  $\Delta t$  combine precisão dos resultados com tempo de execução razoável.

Com  $\Delta t$  determinado, simulamos, para diversos conjuntos de valores iniciais, o comportamento das populações de presas e predadores. Em particular, adotamos as condições iniciais como  $(x_0, y_0) = \{(0.4, 0.4), (0.8, 0.4), (1.6, 0.4), (1.6, 3.2)\}$ , que refletem situações distintas de dominância de predadores ou presas.

Após isso, comparamos a dinâmica das populações interagentes àquela estudada nos Exercícios 1 e 2, que trata de populações isoladas.

### Letra D

Com o algoritmo descrito na *letra B*, calculamos as trajetórias para as 4 condições iniciais especificadas na *letra C*. Como ficará evidente posteriormente, o modelo de Lotka-Volterra também possui um ponto fixo estacionário. Podemos obter as coordenadas desse ponto  $(x_p, y_p)$  por raciocínio similar ao empregado para o mapa logístico: os pontos fixos devem ser limites a partir do qual iterações sucessivas não alteram o estado da população.

Nesse sentido, estipulamos que os pontos  $x_p$  e  $y_p$  sejam as soluções para as equações:

$$x_p = (1 + a\Delta t)x_p - b\Delta t x_p y_p \quad (30)$$

$$y_p = (1 - c\Delta t)y_p + d\Delta t x_p y_p \quad (31)$$

Rearranjando os termos, encontramos a seguinte solução:

$$x_p = \frac{c}{d} \quad y_p = \frac{a}{b} \quad (32)$$

Que verificaremos estar de acordo com os gráficos produzidos para as trajetórias no espaço de fase.

### Letra E

Neste exercício, estamos interessados em aplicar o modelo de Lotka-Volterra a uma situação real, comparando as predições do modelo a dados sobre as populações de lebres e linces no Canadá no início do século XX. Para tanto, utilizaremos os parâmetros de simulação como:

$$(a, b, c, d) = (0.481, 0.025, 0.927, 0.028)$$

E plotaremos tanto como as populações modificam-se como função do tempo quanto a trajetória que elas descrevem no espaço de fases.

Para atestar a qualidade das predições do modelo, utilizaremos uma estatística simples, que é o desvio populacional ( $\Delta P$ ) definido como:

$$\Delta P = P_{data} - P_{LV} \quad (33)$$

onde  $P_{data}$  é a população, num determinado ano, segundo os dados fornecidos e  $P_{LV}$  é a população predita, para o mesmo ano, pelo modelo das equações de Lotka-Volterra.

Outros aspectos sobre o modelo, como o aparecimento e manutenção de populações muito pequenas e a capacidade regenerativa das espécies, serão discutidos na Seção de Resultados.

## 2 Resultados

### Pontos fixos de período 1

#### Letra A

Calculamos graficamente a solução para  $r \in \{1, 2, 2.5\}$ , conforme evidenciado na Figura 1. O *grid* de fundo deixa claro as soluções  $\bar{x} = 0$ ,  $\bar{x} = 0.5$  e  $\bar{x} = 0.6$  para  $r = 1$ ,  $r = 2$  e  $r = 2.5$ , respectivamente.

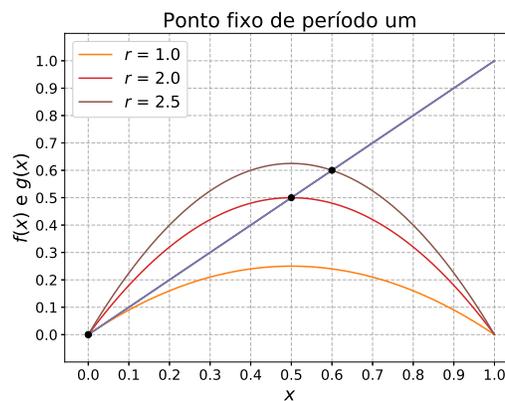


Figura 1: Pontos fixos de período um.

#### Letra B

Agora, vamos plotar, para cada  $r \in \{1, 2, 2.5\}$ , a fração máxima da população ( $x_i$ ) em função do número de iterações ( $i$ ).

Inicialmente, para  $r = 1$  e  $x_0 \in \{0.3, 0.6, 0.9\}$ , obtivemos o gráfico da Figura 2.

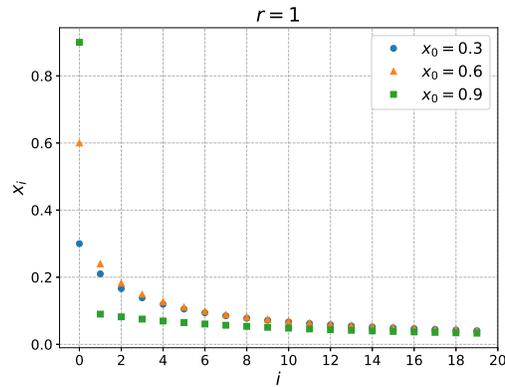


Figura 2: População em função do número de iterações para diferentes valores iniciais.

Note que, para esse caso,  $x_i$  tende assintoticamente para  $x_i = 0$ , como esperávamos para essa escolha de  $r$ .

Agora, para  $r = 2$ , com a mesma escolha de populações iniciais, obtivemos o gráfico disposto na Figura 3

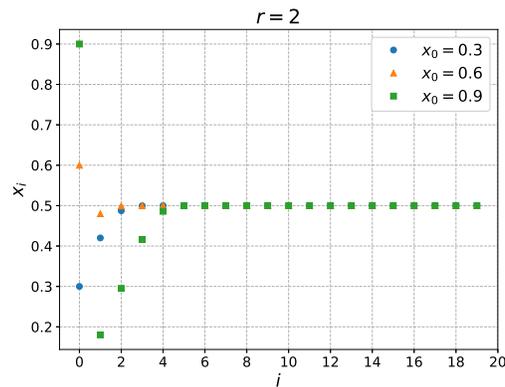


Figura 3: População em função do número de iterações para diferentes valores iniciais.

Novamente, atestamos que  $x_i$  tende à 0.5, ponto fixo para  $r = 2$ , independentemente da escolha de  $x_0$ .

Por fim, para  $r = 2.5$ , com a mesma escolha para  $x_0$ , obtivemos o gráfico da Figura 4.

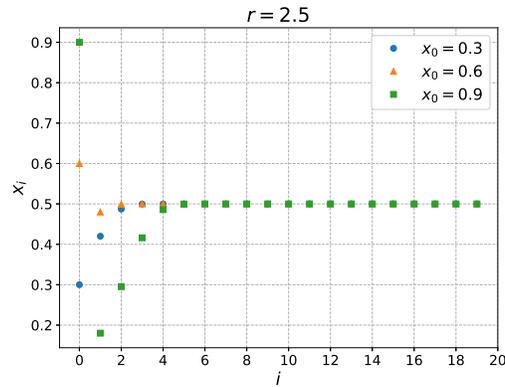


Figura 4: População em função do número de iterações para diferentes valores iniciais.

Que conclui nossa validação da *letra A*, mostrando que a população tende ao ponto fixo independentemente da escolha de  $x_0$  e cai bruscamente se  $x_0$  é maior que  $\bar{x}$ , o ponto fixo do mapa.

### Letra C

Agora, vamos tratar da distância entre trajetórias adjacentes (para  $\varepsilon = 10^{-4}$ ). Seguindo a metodologia descrita, plotamos a distância como função do número de iterações, a partir da décima. Importante notar que, embora a escala vá até a trigésima iteração (para melhorar a visualização), o ajuste é feito com os 90 pontos remanescentes após a décima iteração.

Para  $r = 2.5$ ,  $x_0 = 0.1$  e  $x_0 = 0.3$ , obtivemos os gráficos dispostos na Figura 5 e na Figura 6

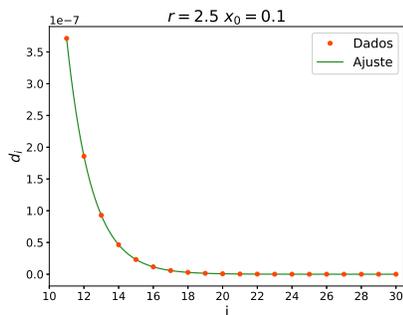


Figura 5: Distância em função do número de iterações para  $x_0 = 0.1$ .

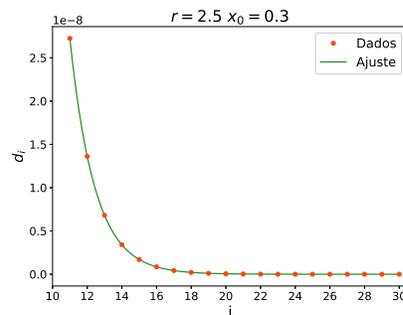


Figura 6: Distância em função do número de iterações para  $x_0 = 0.3$ .

Note que, em ambos os casos  $d_i$  decai exponencialmente. Isso significa que, para o  $r$  escolhido, as trajetórias são convergentes e independem da população

inicial.

Com intuito de aprimorar a disposição das informações, abordaremos os coeficientes exponenciais do ajuste na *letra D*, quando analisarmos os coeficientes de Lyapunov para cada condição inicial escolhida.

Por completudeza, seguindo o guia de execução do projeto, apresentamos também as distâncias calculadas para a  $r = 2.5$  e as condições iniciais  $x_0 = 0.5$ ,  $x_0 = 0.7$  e  $x_0 = 0.9$ , como dispostas nas Figuras 7, 8 e 9.

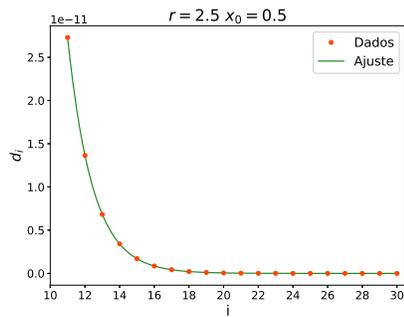


Figura 7: Distância em função do número de iterações para  $x_0 = 0.5$ .

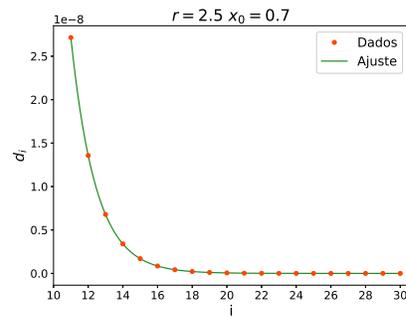


Figura 8: Distância em função do número de iterações para  $x_0 = 0.7$ .

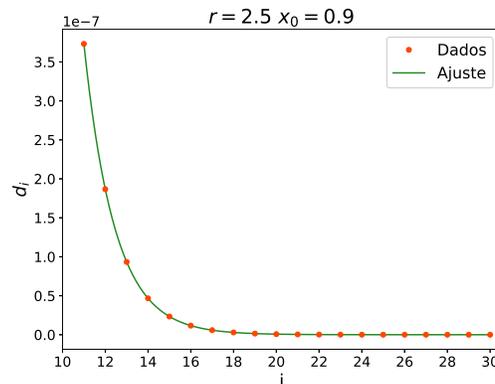


Figura 9: Distância em função do número de iterações para  $x_0 = 0.9$ .

## Letra D

Nosso intuito agora é comparar, para cada  $x_0$  escolhido na *letra C*, o coeficiente exponencial do ajuste ao expoente de Lyapunov. Para cada  $x_0$ , compilamos o coeficiente do ajuste e o expoente de Lyapunov (truncado na mesma quantidade de casas decimais que o coeficiente do ajuste), como consta na Tabela 1.

Tabela 1: Compilação dos coeficientes do ajuste e dos expoentes Lyapunov

$x_0$	Ajuste	Lyapunov
0.1	$-0.69288 \pm 0.00003$	-0.69314
0.3	$-0.69336 \pm 0.00002$	-0.69316
0.5	$-0.69304 \pm 0.00001$	-0.69304
0.7	$-0.69336 \pm 0.00002$	-0.69443
0.9	$-0.69288 \pm 0.00002$	-0.692502

Os dados da Tabela 1 mostram um bom acordo entre os coeficientes do ajuste exponencial e os expoentes de Lyapunov. De fato, eles revelam o claro paralelo que existe entre o decaimento da distância entre trajetórias próximas no mapa logístico e um coeficiente de Lyapunov negativo.

As diferenças nos resultados, que são da ordem de  $10^{-2}$  a  $10^{-3}$  podem estar relacionadas à escolha do passo de iteração para o cálculo do expoente de Lyapunov ou, mais possivelmente, às limitações metodológicas atreladas à nossa escolha de ponto inicial do ajuste e do número de pontos ajustados.

## Dobras de período e caos

### Letra A

Calculamos graficamente as soluções para  $r = 3.2$ , conforme disposto na Figura 10. As linhas tracejadas (em preto) representam as soluções analíticas para os pontos fixos do mapa, como descrevemos nos métodos.

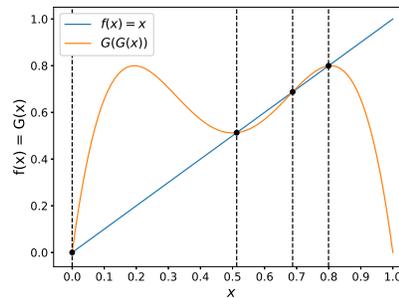


Figura 10: Soluções de período dois para o mapa logístico.

As soluções para os pontos fixos são:

$$(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4) = (0, 0.6875, 0.51304, 0.7994)$$

### Letra B

Plotamos o diagrama de bifurcação, seguindo o método delineado, como exposto na Figura 11. As marcações em vermelho indicam o surgimento das dobras de período e as coordenadas  $r$  associadas serão importantes na determinação do coeficiente de Feigenbaum.

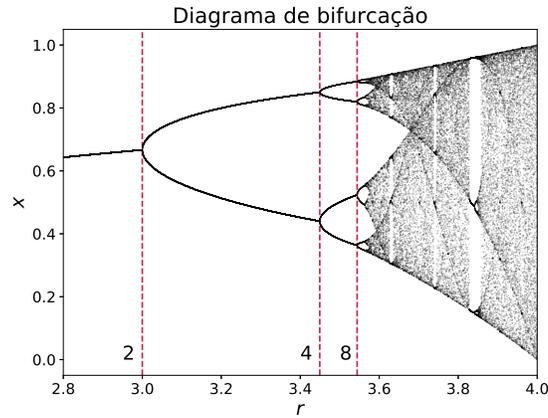


Figura 11: Diagrama de bifurcação.

### Letra C

Com o diagrama de bifurcação da *letra C*, podemos estimar o segundo termo da série que converge para a constante de Feigenbaum. Esse termo pode ser expresso como:

$$\delta_1 = \frac{r_2 - r_1}{r_3 - r_2} \quad (34)$$

Para determinar a constante, seguimos o procedimento de delimitação visual da bifurcação, descrito nos métodos. As Figuras 12 e 13 mostram os gráficos produzidos para auxiliar no processo de determinação de  $r_2$  e  $r_3$ .

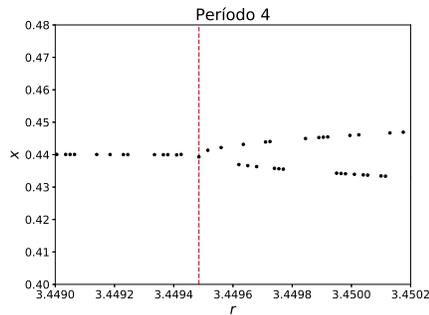


Figura 12: Dobras de período 4.

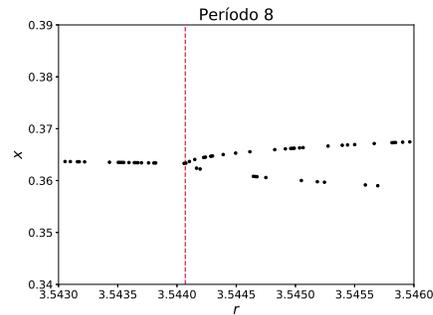


Figura 13: Dobras de período 8.

A partir dos gráficos, encontramos  $r_2 = 3.449485$  e  $r_3 = 3.54407$ , além de  $r_1 = 3.0$  que já era conhecido analiticamente. Com esses valores, podemos calcular a constante  $\delta_1$ , truncando, para facilitar a visualização, no quarto algarismo:

$$\delta_1 = 4.7522$$

Que difere de  $8 \cdot 10^{-4}$ , cerca de uma parte por milésimo, da constante tabulada:

$$\delta_{1,tab} = 4.7514 \quad (35)$$

O resultado encontrado possui um erro diminuto relativo ao esperado, evidenciando que a bifurcação para as dobras de período 2, 4 e 8 seguiu a razão de Feigenbaum.

### Letra D

Acessamos agora a região do casos determinístico, tomando  $r = 3.75$  maior que o limiar de  $r_c = 3.569946$ .

Inicialmente, plotamos a fração máxima da população ( $x_i$ ) em função do número de iterações para três condições iniciais  $x_0 = 0.1, 0.3$  e  $0.5$ , como consta na Figura 14.

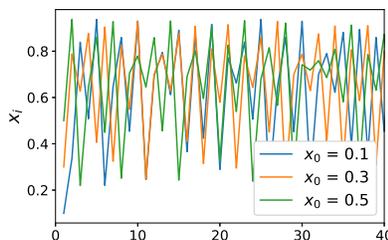


Figura 14: Fração máxima da população em função do número de iterações.

Nela notamos, claramente, o surgimento de uma dinâmica irregular, com variações bruscas à cada geração e que é radicalmente alterada por mudanças de condições iniciais. Essa é uma descrição do chamado caos determinístico: a regra de obtenção dos estados (o mapa) é determinista, mas a sensibilidade às condições iniciais mimetiza uma dinâmica caótica.

Para entender melhor o aparecimento do caos, podemos calcular as distâncias entre trajetórias adjacentes (para  $\varepsilon = 10^{-4}$ ), para as condições iniciais da Figura 14, nominalmente  $x_0 = 0.1, 0.3$  e  $0.5$ . Aqui, seguindo a metodologia, ajustamos o modelo exponencial aos 16 primeiros pares  $(i, d_i)$ , devido à irregularidade geral observada nas distâncias a partir desse limiar.

Os gráficos explicitados nas Figuras 15, 16 e 17 evidenciam o procedimento e o resultado do ajuste.

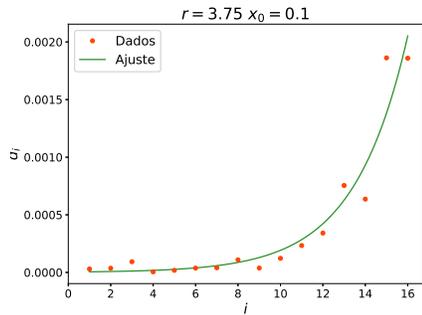


Figura 15: Distância em função do número de iterações.

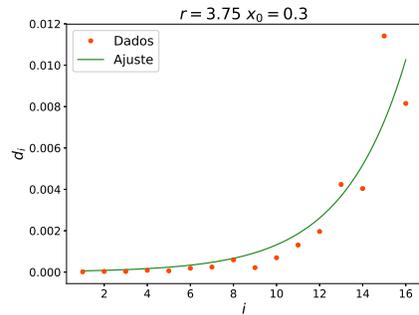


Figura 16: Distância em função do número de iterações.

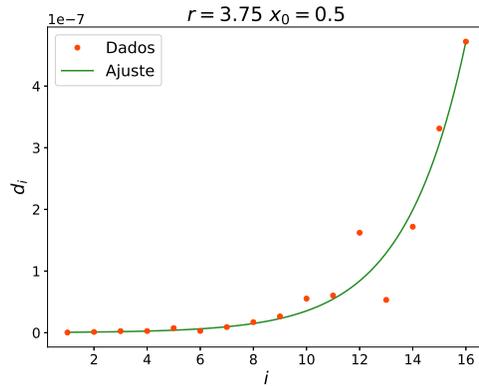


Figura 17: Distância em função do número de iterações.

A partir dos gráficos, observamos que trajetórias com condições iniciais próximas, divergem ao longo das iterações. Isso está de acordo com nossa expectativa, criada a partir da Figura 14, de que haja uma forte dependência das condições iniciais para o  $r$  escolhido.

Quantitativamente, podemos comparar o coeficiente exponencial do ajuste aos expoentes de Lyapunov para  $r = 3.75$ . Para as condições iniciais utilizadas, compilamos os resultados na Tabela 2, onde truncamos o expoente de Lyapunov (calculado com  $n = 10^6$  iterações) na mesma quantidade de algarismo significativos do ajuste.

Tabela 2: Coeficientes do ajuste e expoentes de Lyapunov para  $r = 3.75$ .

$x_0$	Ajuste	Lyapunov
0.1	$0.39 \pm 0.05$	0.36
0.3	$0.31 \pm 0.06$	0.36
0.5	$0.53 \pm 0.04$	0.36

Embora numericamente mais discrepantes que no Exercício 1, possivelmente devido à metodologia do ajuste e à irregularidade de  $d_i$ , o fato de ambos os coeficientes estarem em acordo indica que as trajetórias divergem e que, para a região acessada acima de  $r = 3.569946$ , a dinâmica é extremamente influenciada pela escolha de condições iniciais.

## Modelo predador-presa

### Letra A

Já abordamos o significado das constantes quando descrevemos o modelo na seção de Métodos.

### Letra B

Aqui, a resolução trata da implementação do código que já descrevemos nos métodos.

### Letra C

Antes de prosseguir com as simulações do modelo, temos de estimar o intervalo temporal das iterações ( $\Delta t$ ). Seguindo a metodologia descrita, obtivemos o gráfico disposto na Figura 18.

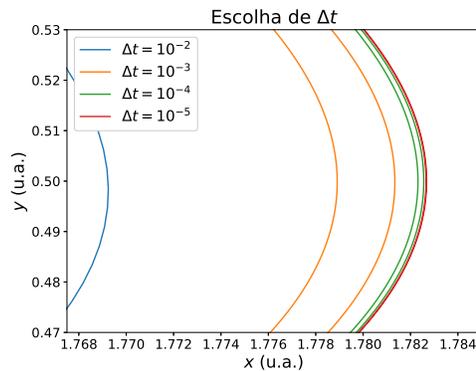


Figura 18: Escolha do passo  $\Delta t$ .

Com o resultado da Figura 18, a escolha de passo temporal ótimo é  $\Delta t = 10^{-4}$ , já que qualquer outra escolha menor que ele produz uma trajetória pouco diferente e acarreta em tempos de execução mais longos.

Com  $\Delta t$  bem determinado, podemos abordar as simulações. Para criar um conjunto de gráficos que representasse satisfatoriamente as previsões do modelo, escolhemos condições iniciais diversas e plotamos as populações de predadores e presas em funções do tempo, como nas Figuras 19, 20, 21 e 22.

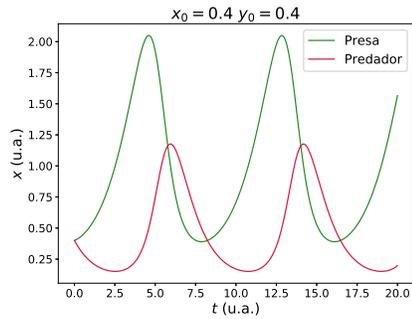


Figura 19: População em função do tempo.

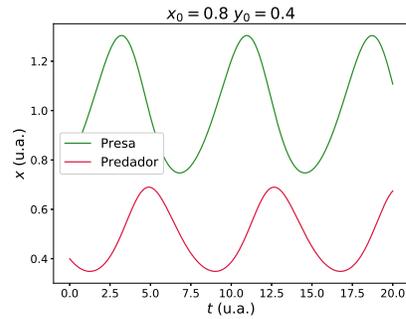


Figura 20: População em função do tempo.

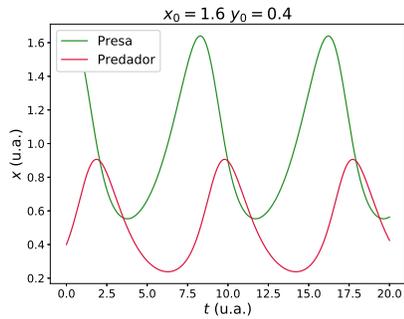


Figura 21: População em função do tempo.

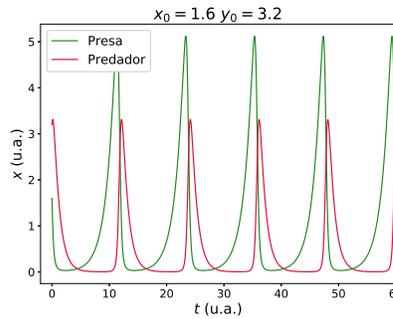


Figura 22: População em função do tempo.

Notemos alguns pontos:

1. As soluções são oscilatórias. Diferentemente do mapa logístico, aqui, a inclusão de uma segunda espécie não só limita o crescimento da outra, como induz o aparecimento de uma dinâmica populacional periódica. Isoladamente, veríamos a população de presas crescer até o limite ambiental, enquanto a população de predadores cairia, uma vez que são dependentes das presas para alimentação e, conseqüentemente, para se reproduzir.

2. As curvas de crescimento das espécies apresentam uma defasagem característica para todos os valores de condições iniciais. Isso está associado à relação predatória: maior número de presas, aumenta a população de predadores, que diminui a disponibilidade de presas, reduzindo a população dos predadores.
3. Para o último caso, em que há predominância de predadores, o modelo prevê que as populações de presas e predadores caíam a níveis bem baixos durante os ciclos. Com efeito, uma das principais críticas a esse modelo ecológico é que ele prevê populações mínimas insustentáveis, que não teriam a capacidade regenerativa predita pelas equações.

### Letra D

Para as diversas condições iniciais da *letra B*, plotamos as “trajetórias” das populações no espaço de fase, como consta na Figura 23.

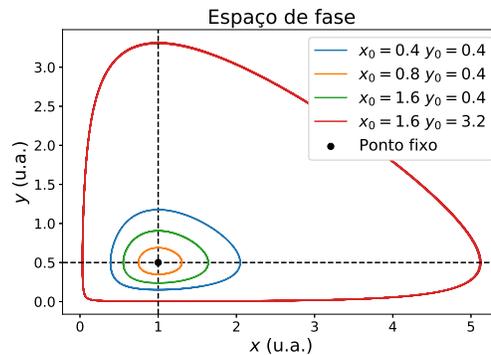


Figura 23: Espaço de fase para o modelo de Lotka-Volterra

Na Figura 23, fica ainda mais evidente a dinâmica oscilatória e regular: as trajetórias no espaço de fase são fechadas. Isso, por si só, justifica a utilização do *método de Euler-Cromer*. Como vimos, o *método de Euler* não conserva a energia do sistema, enquanto a mudança artificial introduzida pelo *método de Euler-Cromer*, faz com que a energia seja também oscilante, mas sempre restituída ao seu valor inicial ao fim do período característico do sistema.

Não obstante, como vimos nos métodos, as trajetórias no modelo de Lotka-Volterra também possuem um ponto fixo estacionário, cujas coordenadas são:

$$x_p = \frac{c}{d} = 1 \quad \text{e} \quad y_p = \frac{a}{b} = \frac{1}{2}$$

Que é o ponto marcado na Figura 23. Note, por exemplo, que as trajetórias circundam o ponto fixo do modelo, mas nunca passam por ele: se passassem, a população ficaria estacionada.

## Letra E

Agora, nosso interesse é utilizar o modelo de Lotka-Volterra para compreender a dinâmica populacional de lebres e lincos no Canadá no início do século XX.

Utilizando o procedimento de solução, para a escolha de coeficientes e  $\Delta t$  especificados nos métodos, obtivemos o gráfico das populações em função do tempo, disposto na Figura 24. Os pontos representam dados enquanto as linhas sólidas representam o modelo.

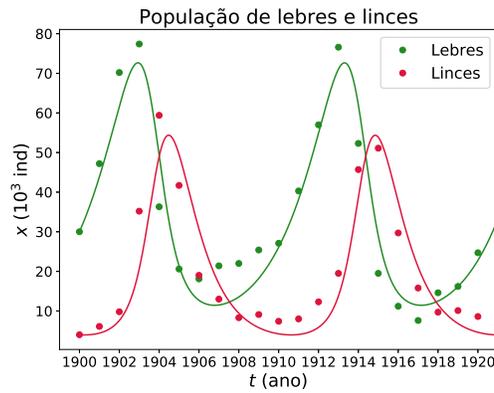


Figura 24: Oscilações nas populações de lebres e lincos, de acordo com o modelo e dados.

Com efeito, o modelo explica qualitativamente bem as oscilações populacionais associadas à interação entre as espécies. Mas há alguns pontos a serem notados: os desvios populacionais relativos ao modelo são elevados, chegando a uma diferença de aproximadamente quinze mil indivíduos, como atesta a Figura 25.

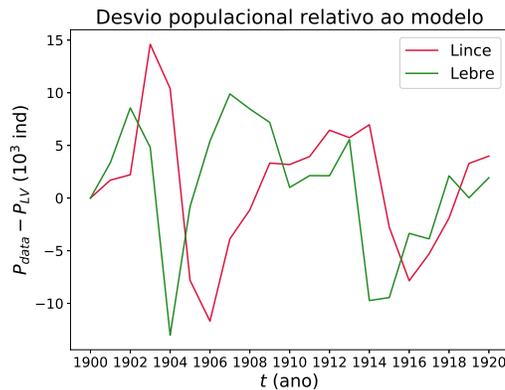


Figura 25: Desvio populacional relativo ao modelo.

Outro fato a ser considerado é que, na natureza, não há interação isolada entre

duas espécies, muito menos interação dissociada de variações ambientais, estações anômalas ou da ocorrência concomitante de diversos ciclos predador-presa.

Para concluir, plotamos a trajetória predita e os dados no espaço de fases do sistema, como disposto na Figura 26.

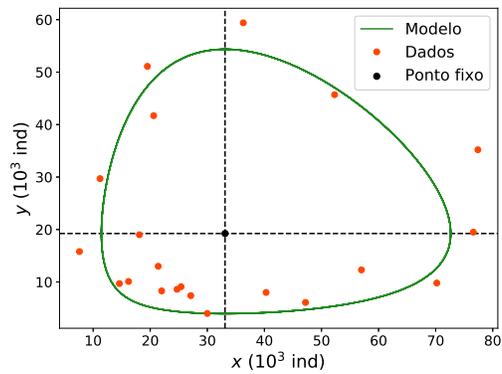


Figura 26: Espaço de fase para a população de lebres e lince.

O ponto fixo nesse caso é:

$$x_p = 33.11 \text{ e } y_p = 19.24$$

Onde utilizamos as unidades como milhares de indivíduos. Aqui, atestamos novamente a qualidade do modelo em reproduzir, pelo menos qualitativamente, o ciclo de predação entre as população de lebres e lince.