Національний Технічний Університет України
Навчально-науковий комплекс
«Інститут прикладного системного аналізу»

ЛАБОРАТОРНАЯ РАБОТА № 1

з дисципліни «Моделювання складних систем»

Тема: «Динаміка процесів, що описуються диференціальними рівняннями першого та другого порядку»

Виконали:
студенти гр. КА-41
Мельничук Валентин
Лочман Ярослава
Снігірьова Валерія

Прийняв:
професор кафедри ММСА, д.т.н.
Степашко В.С.

Київ 2018

Математичні моделі

Модель Ферґюльста

Описує динаміку росту популяції з урахуванням обмеженості ресурсів середовища:

$$N' = \mu N(k - N)$$

N – чисельність популяції,
μ – коефіцієнти народжуваності (приріст популяції за одиницю часу),
k – ємність середовища (гранична чисельність популяції),
N_0 – початкова чисельність популяції.

Модель Солоу

Односекторна модель економічної динаміки:

$$k' = sak^{\alpha} - (\mu + q)k$$

α – частка капіталу в продукції
m – норма витрат на амортизацію
a – рівень розвитку економіки
s – норма накопичення
q – темп приросту робочої сили
k_0 – початковий капітал

Рівняння вимушених коливань

$$x'' + 2\delta x' + \omega_0^2 x = f_0 cos(\omega t)$$

δ - коефіцієнт згасання
ω_0 - власна частота коливань
ω – частота коливань зовнішньої сили
f_0 – амплітуда зовнішньої сили

Модель динаміки двох популяцій „хижак-жертва”. Найпростіша модель (Лотткі-) Вольтерра

$$\begin{cases} x' = (\alpha_x y - \beta_x)x \\ y' = (\alpha_y - \beta_y x)y \end{cases}$$

α_x - "Норма споживання" жертв
α_y - Природна народжуваність жертв
β_x - Природна смертність хижаків
β_y - "Норма споживаності" жертв

x_0 - Початкова кількість хижаків
y_0 - Початкова кількість жертв

Розв’язання задач Коші для усіх розглянутих вище диференціальних рівнянь здійснювалось за допомогою **функції LSODA**, написаної на FORTRAN. LSODA – це метод Адамса/ BDF, з автоматичним виявленням жорсткості та переключенням на відповідний метод.

Результати

Модель Фергюльста

Модель Фергюльста – це логістичне рівняння.

Стаціонарні значення чисельності популяції (коли $N=0$): $N_{1,\text{ст}} = 0$, $N_{2,\text{ст}} = k$. Його перша стаціонарна точка $N_{1,\text{ст}} = 0$ – *нестійка,* друга $N_{2,\text{ст}} = k$ – *стійка,* що можна простежити на (*рис. 1 – 4*).

Нагадаємо, що величина k – ємність екологічної ніші популяції – відповідає такій чисельності популяції, за якої фактична швидкість відтворення в результаті конкуренції настільки знижена, що популяція в цілому може тільки відновлювати в кожному поколінні свою чисельність. У цей момент кількість народжених особин врівноважується кількістю загиблих.

Отже, при невеликих N_0 ($N_0<k/2$) чисельність швидко зростає, але по мірі зростання сповільнює темпи росту, при цьому прямуючи до границі k (*рис. 1*).

При середніх $N0$ ($k/2<N0<k$) чисельність плавно зростає, при цьому прямуючи знизу до границі k (*рис. 2*). Точка перегину графіка розв'язку рівняння Фергюльста лежить в області від'ємних значень часу.

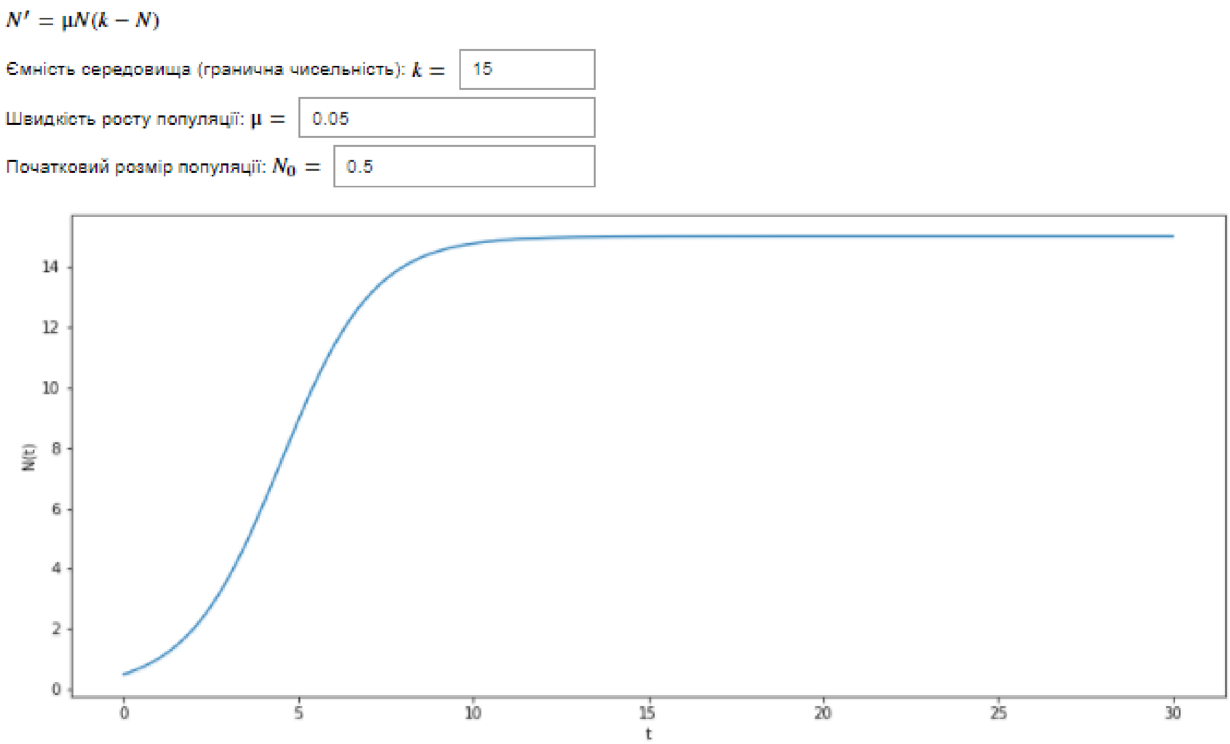

Рис. 1. Розв'язок рівняння Фергюльста ($N0<k/2$)

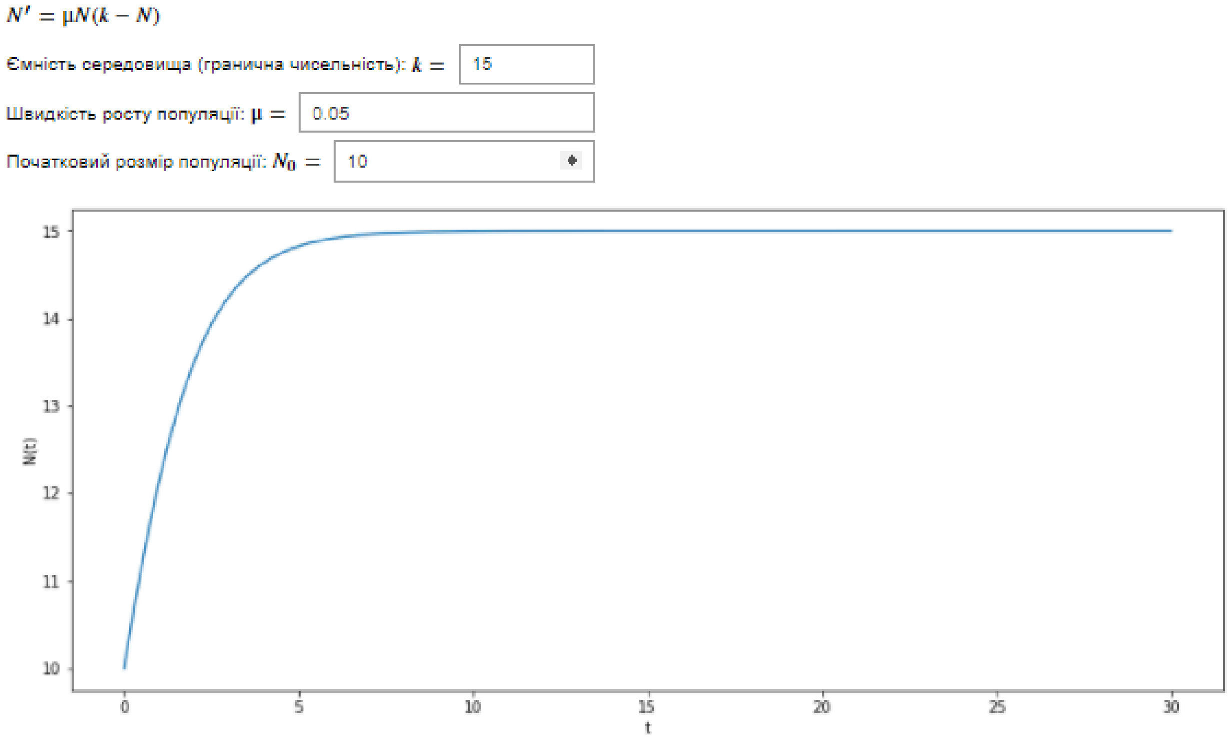

Рис. 2. Розв'язок рівняння Ферґюльста (k/2<N0<k)

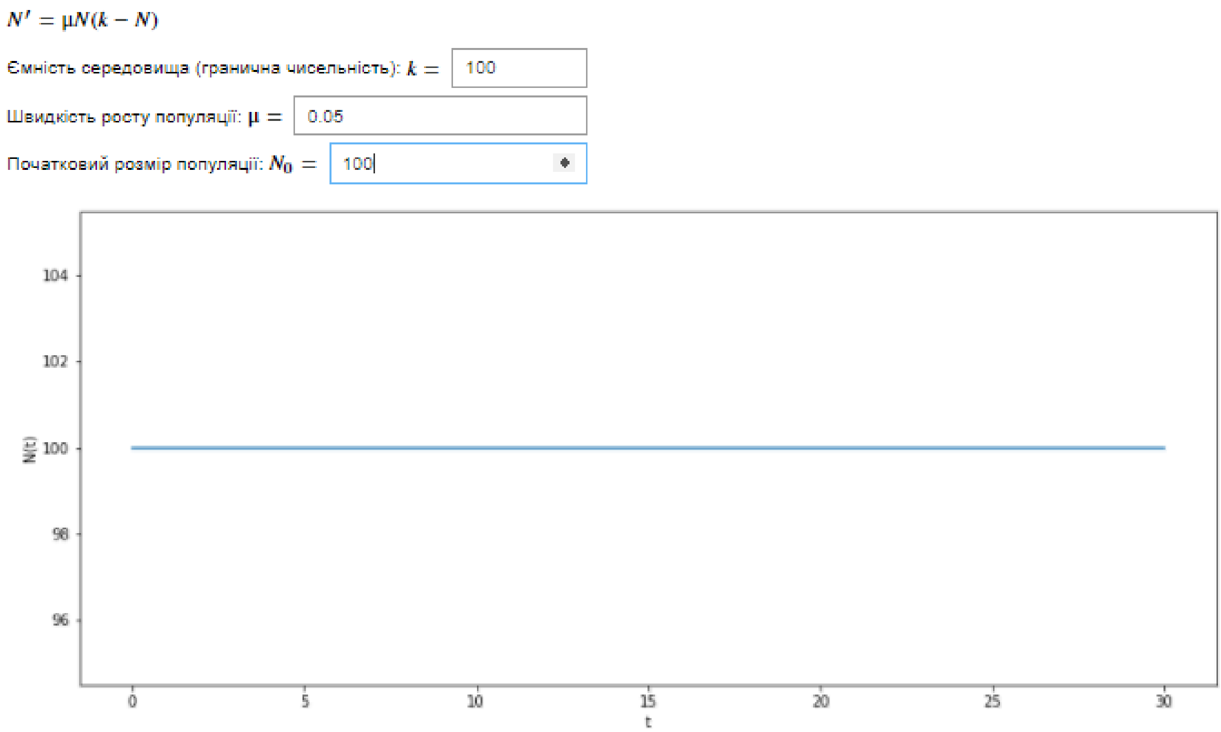

Рис. 3. Логістична крива у випадку, коли N0=k

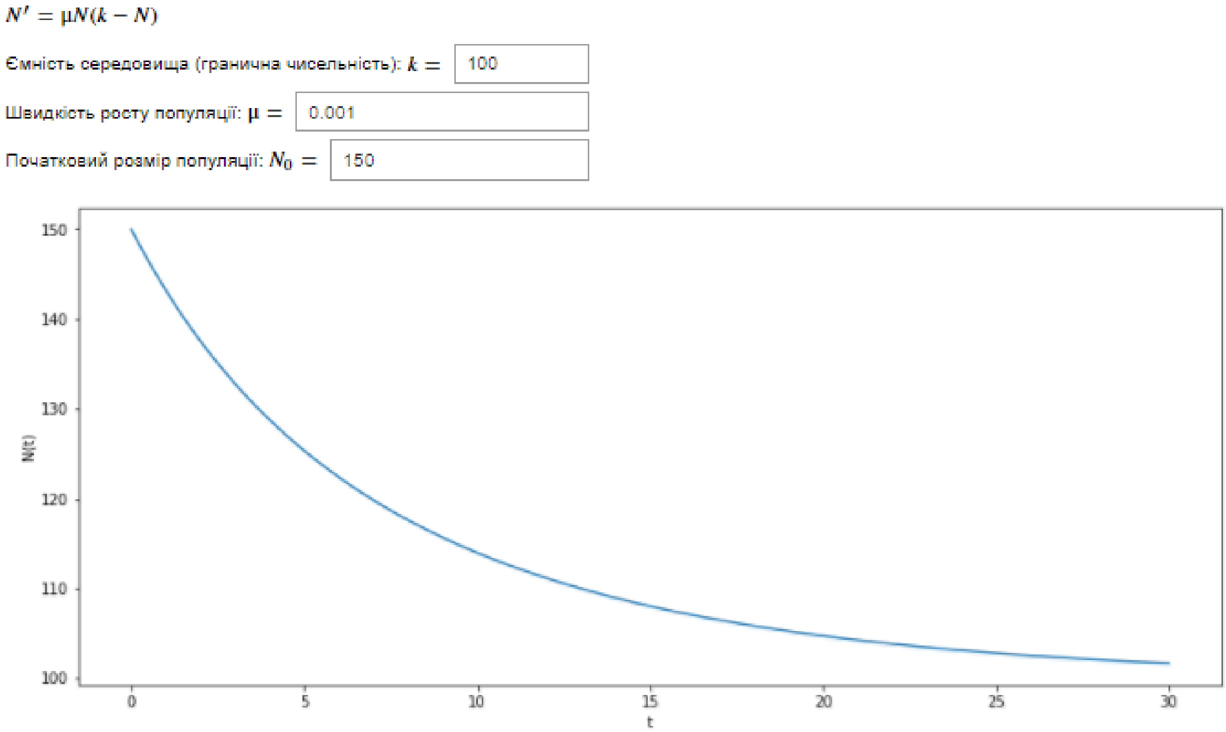

Рис. 4. Логістична крива у випадку, коли N0>k

Якщо *N0=k*, чисельність популяції залишається сталою величиною *(рис. 3*). У випадку, коли *N0>k*, чисельність стрімко зменшується до граничного значення *k* (*рис. 4*).

Отже, сформулюємо підсумок дослідження. Якщо початкова чисельність популяції менше величини екологічної ємності популяції, то з плином часу її розмір буде рости, наближаючись до свого граничного значення *k*. При цьому, якщо початкова чисельність становить менше половини ємності екологічної ніші, на початковому етапі швидкість росту популяції зростатиме, поки чисельність не досягне значення *k/2*, а потім почне знижуватися, прямуючи до нуля.

Якщо початкова чисельність популяції становить більше половини ємності екологічної ніші, то розмір популяції буде збільшуватися, прагнучи до значення *k*, а швидкість її зростання буде неухильно знижуватися. Зміна характеру розвитку популяції (перехід від зростання швидкості росту до зниження в точці *x(t)* = *k/2*) відбулося до того, як дослідник почав за нею спостерігати (тобто до моменту часу *t* = 0).

Якщо ж розмір популяції у початковий момент часу більше гранично можливого значення, то чисельність популяції буде знижуватися.

Модель Солоу – односекторна макроекономічна модель економічної динаміки. Економічна система при цьому виробляє один продукт, який як споживається, так і інвестується. Експорт і імпорт не враховуються. Стан економіки в моделі Солоу визначається за допомогою пяти ендогенних змінних, що змінюються з часом.

$$k' = sak^{\alpha} - (\mu + q)k$$

α – частка капіталу в продукції
m – норма витрат на амортизацію
a – рівень розвитку економіки
s – норма накопичення
q – темп приросту робочої сили
k_0 – початковий капітал

Очевидно, що стаціонарний розв’язок моделі економічної динаміки залежить від параметрів рівняння, початкових умов, а також конкретного варіанта виробничої функції (в нашому випадку - Кобба-Дугласа).

На рис. 5 – 8 наведені приклади розв’язання задачі Коші для наведеного вище диференціального рівняння за різних початкових умов і при різних значеннях вхідних параметрів.

$k' = sak^{\alpha} - (\mu + q)k$

Норма накопичення: $s =$ 0.2

Рівень розвитку економіки: $a =$ 2.5

Частка капіталу у сукупній продукції: $\alpha =$ 0.3

Норма витрат на амортизацію: $\mu =$ 0.1

Темп приросту робочої сили: $q =$ 0.1

Початковий капітал: $k_0 =$ 0.8

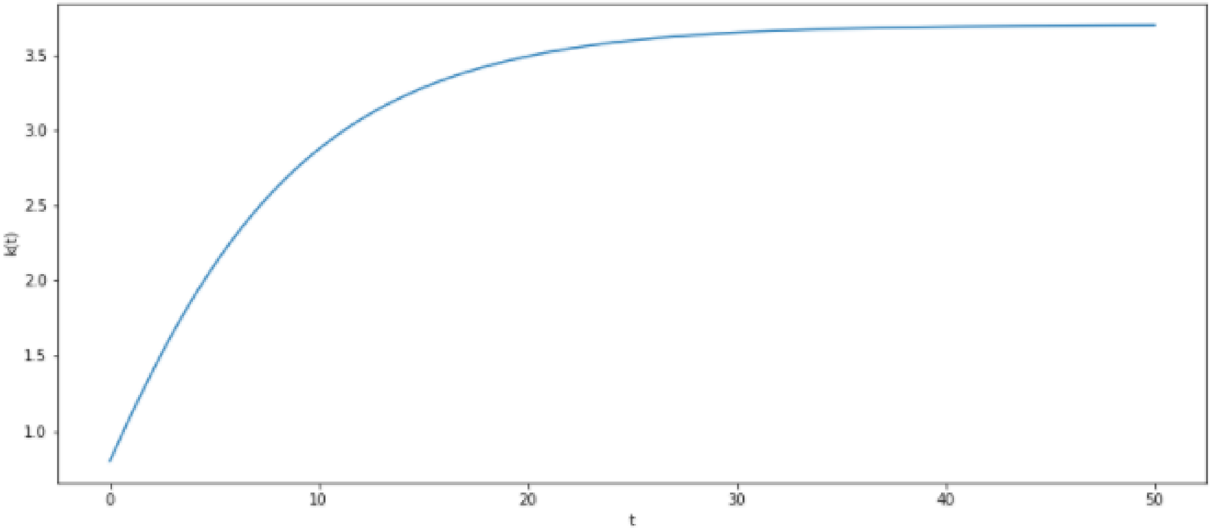

Рис. 5. Інтегральна крива в моделі Солоу при значеннях вхідних параметрів для України

$k' = sak^{\alpha} - (\mu + q)k$

Норма накопичення: $s =$ 0.2

Рівень розвитку економіки: $a =$ 2.5

Частка капіталу у сукупній продукції: $\alpha =$ 0.8

Норма витрат на амортизацію: $\mu =$ 0.1

Темп приросту робочої сили: $q =$ 0.1

Початковий капітал: $k_0 =$ 0.8

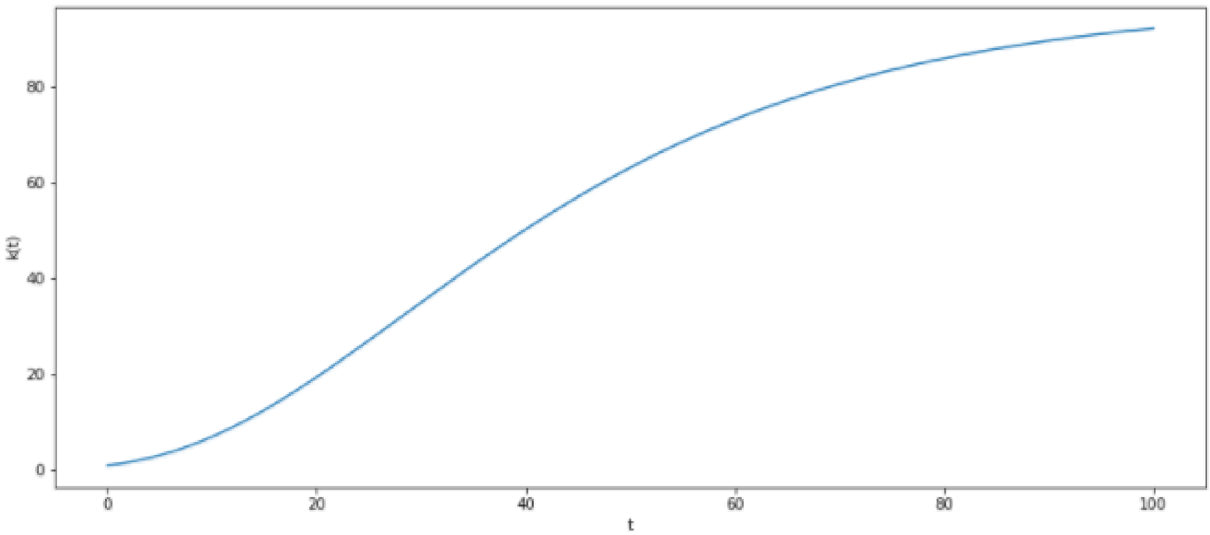

Рис. 6. Розв'язок моделі Солоу при значеннях вхідних параметрів, що забезпечують сигмоїдальну логістичну криву

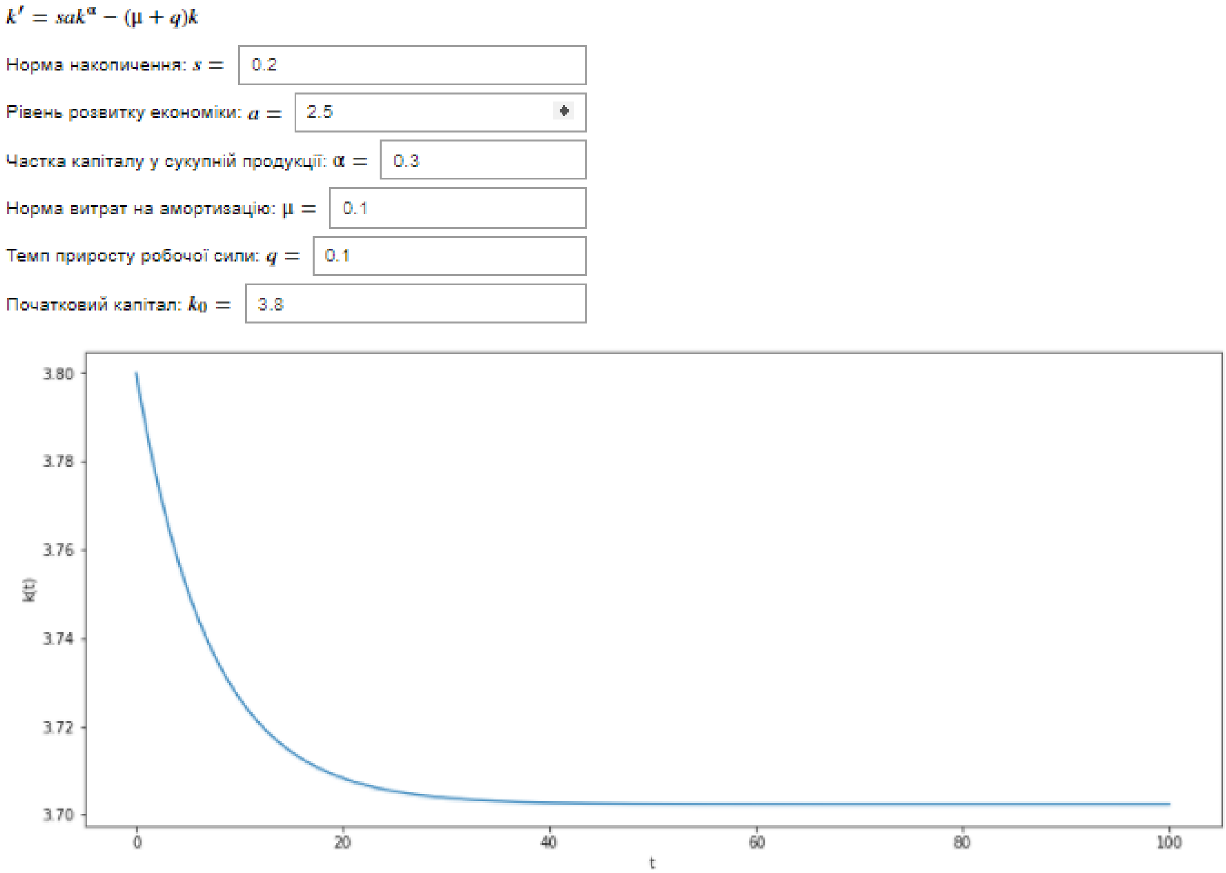

Рис. 7. Інтегральна крива в моделі Солоу при значеннях вхідних параметрів, що визначають спадний економічний процес

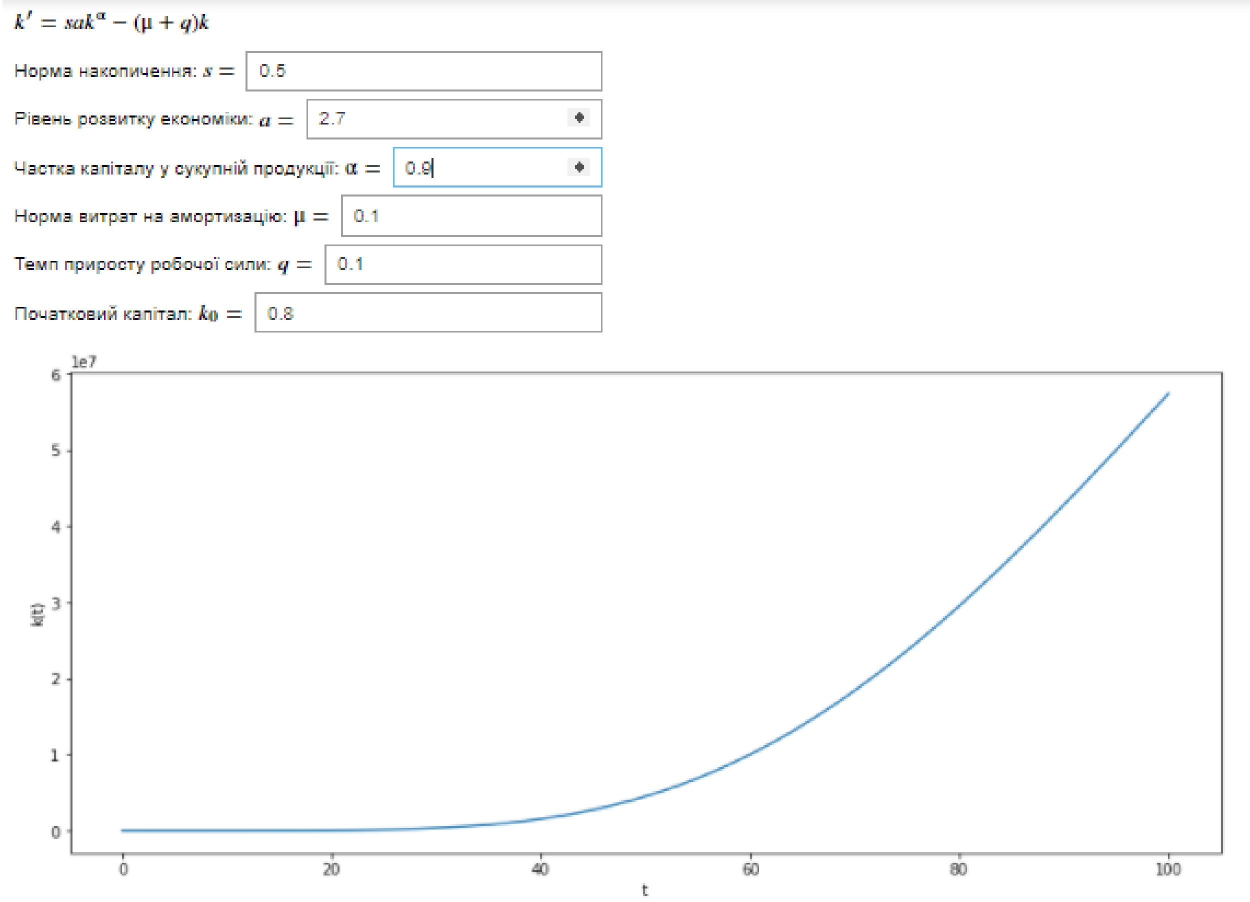

Рис. 8. Інтегральна крива в моделі Солоу при інших значеннях вхідних параметрів

На рис. 7 спостерігаємо, що за відповідних вхідних параметрів має місце спад питомої капіталоозброєності, тобто «проїдання» фондів.

Взагалі, залежно від значення початкової питомої капіталоозброєності мають місце три різновиду перехідного процесу:

1) Якщо $k_0 < k^{**}$, k^{**} – капіталоозброєність за золотим правилом (це точка, в якій дотична до графіка виробничої функції паралельна до прямої вибуття капіталу y = -(m+q) k), то капіталоозброєність зростає прискореними темпами;

2) Якщо $k^{*} > k_0 > k^{**}$, (де k^{*} – стійкий рівень капіталоозброєності), то має місце повільний спад капіталоозброєності;

3) Якщо $k^{*} < k_0$, то має місце проїдання фондів.

Отже, можна зробити наступні висновки:

1. За показниками для України ($k_0 = 0.8$, $s = 0.2$, $a = 2.5$, $\mu = q = 0.1$, $\alpha = 0.3$) капіталоозброєність поступово зростає до усталеного значення протягом певного часу, а тоді залишається сталою.

2. Якщо частка капіталу у продукції збільшиться, показник капіталоозброєності сповільнить темпи зростання.

3. При підвищенні витрат на амортизацію і при меншому темпі приросту робочої сили спостерігатимемо зниження рівня капіталоозброєності.

4. При незначних темпах приросту робочої сили, невеликих витратах на амортизацію, але за підвищеної норми накопичення простежується стрімке зростання рівня капіталоозброєності.

Рівняння вимушених коливань

$x'' + 2\delta x' + \omega_0^2 x = f_0 cos(\omega t)$

Коефіцієнт згасання: $\delta =$ 0

Власна частота: $\omega_0 =$ 0.4

Частота зовнішньої сили: $\omega =$ 0

Амплітуда зовнішньої сили: $f_0 =$ 0

Початкове положення: $x_0 =$ 0.3

Початкова швидкість: $x'_0 =$ 0.2

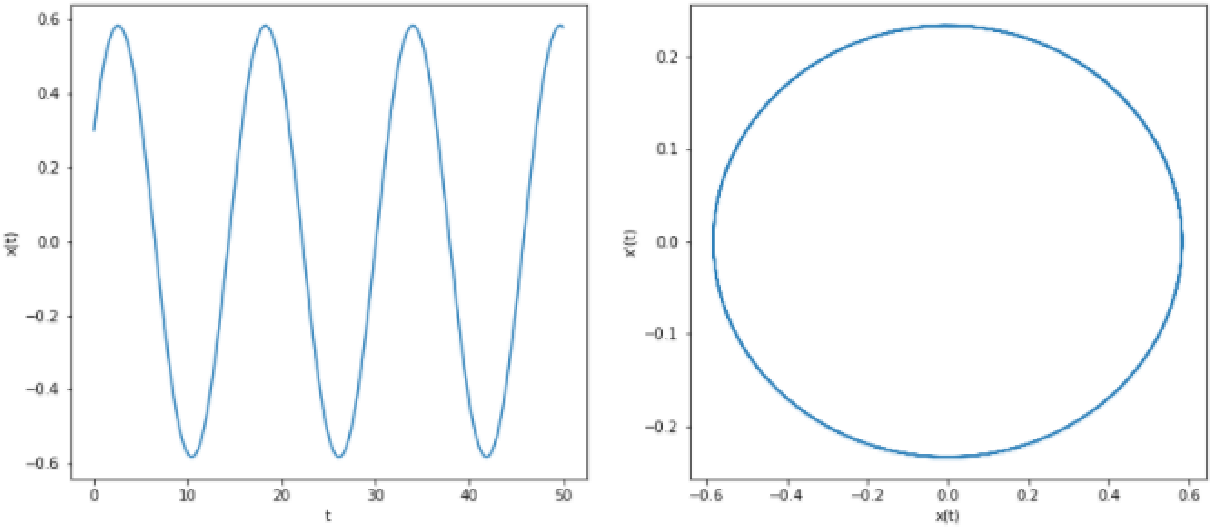

Рис. 9. Гармонічні коливання ($\delta = f_0 = 0$)

$x'' + 2\delta x' + \omega_0^2 x = f_0 cos(\omega t)$

Коефіцієнт згасання: $\delta =$ 0.1

Власна частота: $\omega_0 =$ 0.7

Частота зовнішньої сили: $\omega =$ 0

Амплітуда зовнішньої сили: $f_0 =$ 0

Початкове положення: $x_0 =$ 0.4

Початкова швидкість: $x'_0 =$ 1.4

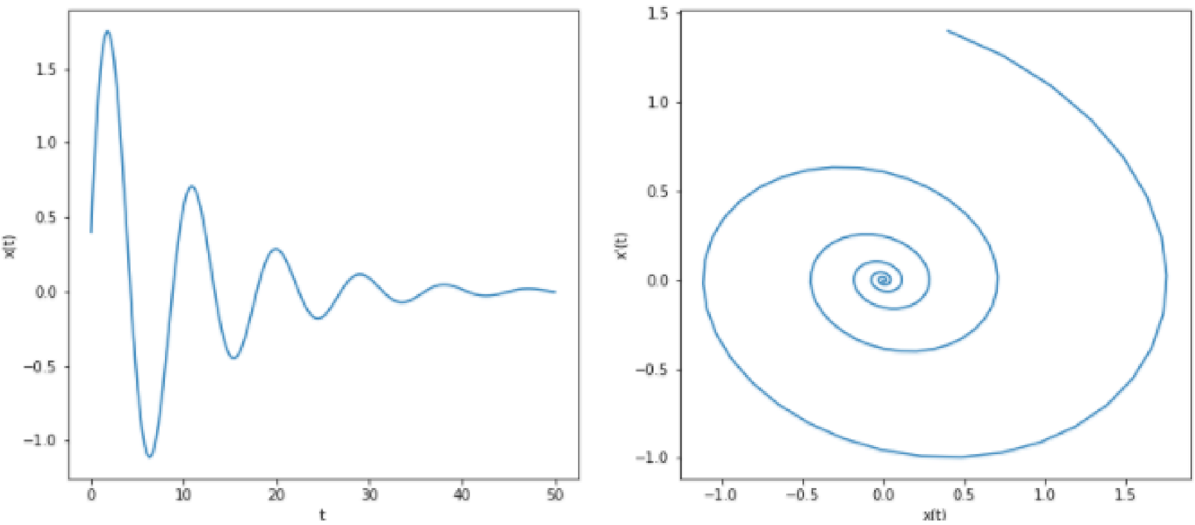

Рис. 10. Згасаючі коливання (f_0 = 0)

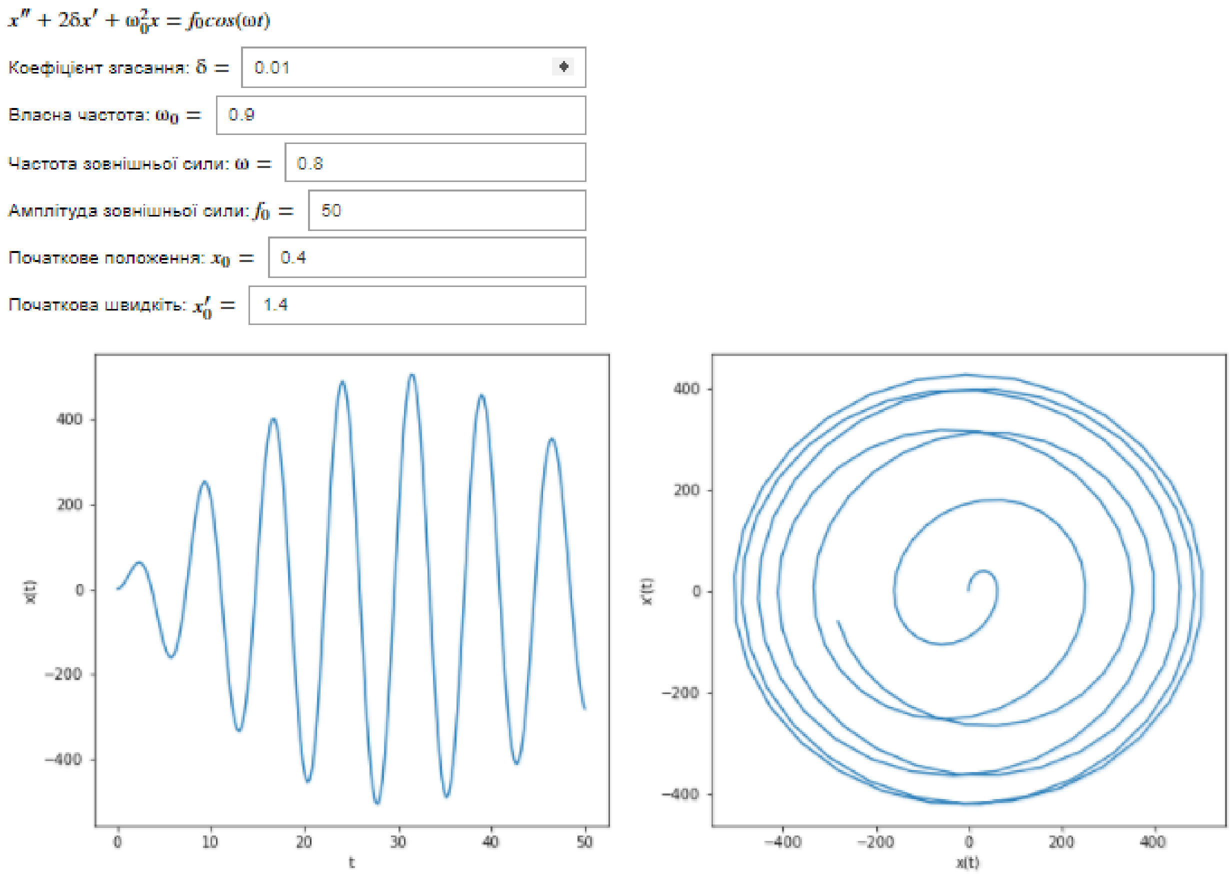

Рис. 11. Вимушені коливання (f_0 =50, δ = 0,01)

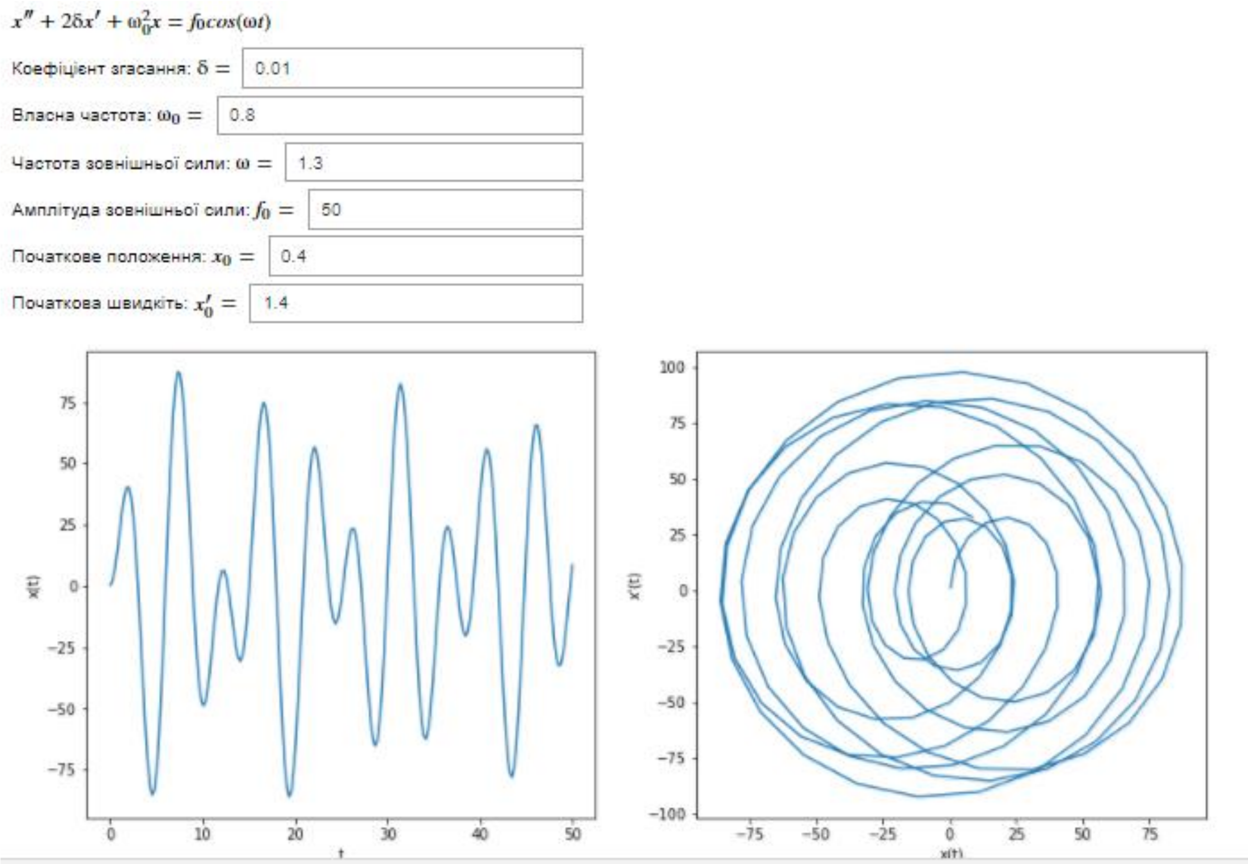

Рис. 12. Вимушені коливання (ω_0 != ω)

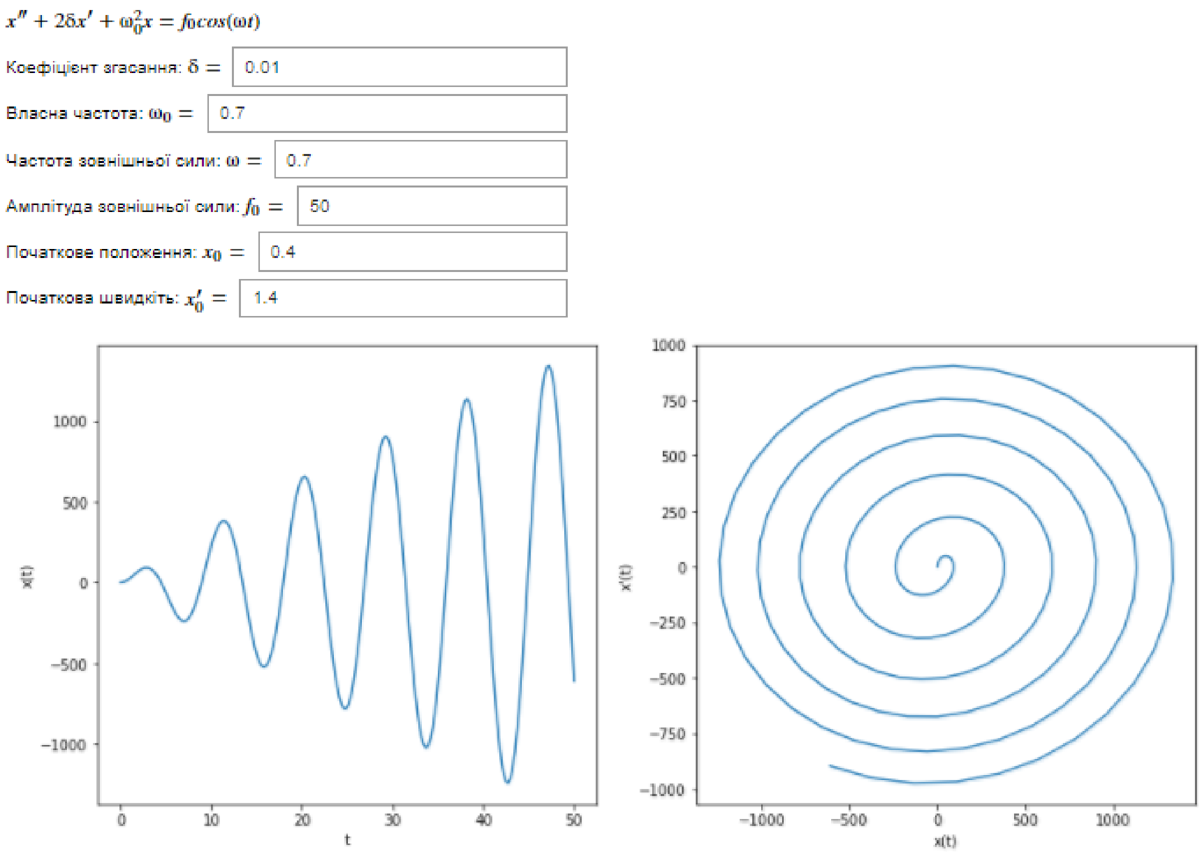

Рис. 13. Вимушені коливання (f_0 =50, δ = 0)

Вимушені коливання мають наступну природу: спочатку, в процесі встановлення вимушених коливань, відбувається накладення вільних згасаючих, а також вимушених коливань. Після того як вільні коливання припиняться, залишається тільки вимушені коливання.

Різке зростання амплітуди вимушених коливань при наближенні частоти зовнішньої сили до власної частоти коливань системи зветься **резонансом**.

Це явище яскраво проілюстровано у третьому і четвертому випадку.

Амплітуда вимушених коливань при резонансі визначається так:

$$A_{\max} \approx \frac{w0 \cdot f0}{2\delta} = q \cdot f0$$

Тобто амплітуда усталених коливань при резонансі в q разів більша за амплітуду при дуже низькій зовнішній частоті w.

За відсутності коефіцієнта згасання амплітуда вимушених коливань перевищує амплітуду зовнішньої сили при всіх частотах в інтервалі між $w = 0$ і $w = \sqrt{2}\, w_0$.

Найпростіша модель динаміки системи «Хижак - жертва»

Наведемо модель в тих позначеннях, які використовуються у розробленій программі:

$$\begin{cases} x' = (\alpha_x y - \beta_x)x \\ y' = (\alpha_y - \beta_y x)y \end{cases}$$

α_x - "Норма споживання" жертв

α_y - Природна народжуваність жертв

β_x - Природна смертність хижаків

β_y - "Норма споживаності" жертв

x_0 - Початкова кількість хижаків

y_0 - Початкова кількість жертв

Кожне рівняння моделі Лотки-Вольтерра є логістичною залежністю. Стаціонарний розв’язок:

$$\begin{cases} x^* = \dfrac{\alpha_y}{\beta_y} \\ y^* = \dfrac{\beta_x}{\alpha_x} \end{cases}$$

Отже, точка рівноваги повністю визначається параметрами «протилежної» популяції.

1) Якщо початкові умови такі, що $x_0 = x^*$; $y_0 = y^*$, то цей стан системи зберігається продовж всього періоду часу, на якому спостерігається система (рис. 14). Фазова крива вироджується в точку.

2) Якщо початкові умови інші, то траєкторії x(t), y(t) є періодичними функціями, про що свідчать результати обчислювальних експериментів, наведених на рис. 15 – 17.

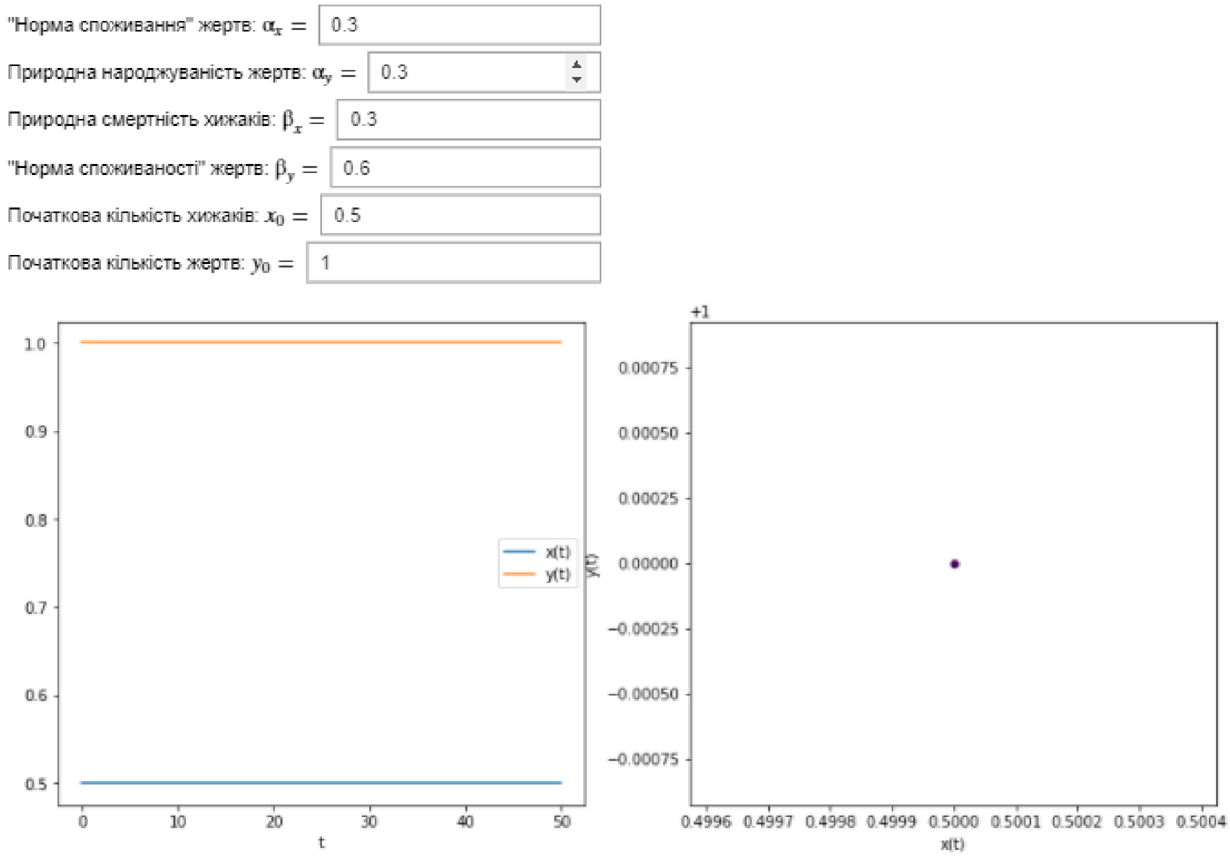

Рис. 14. Розв'язок системи диференціальних рівнянь моделі Лотки-Вольтерра за початкових умов $x_0 = x^*;\ y_0 = y^*$

На фазовій площині фазові траєкторії створюють граничні цикли у вигляді еліпсів (рис. 15) або інших опуклих замкнених фігур в залежності від початкових умов. Для того, щоб простежити відхилення фазових траєкторій від положення рівноваги системи на фазових площинах на рис. 15 – 17, б зображена сама стаціонарна точка.

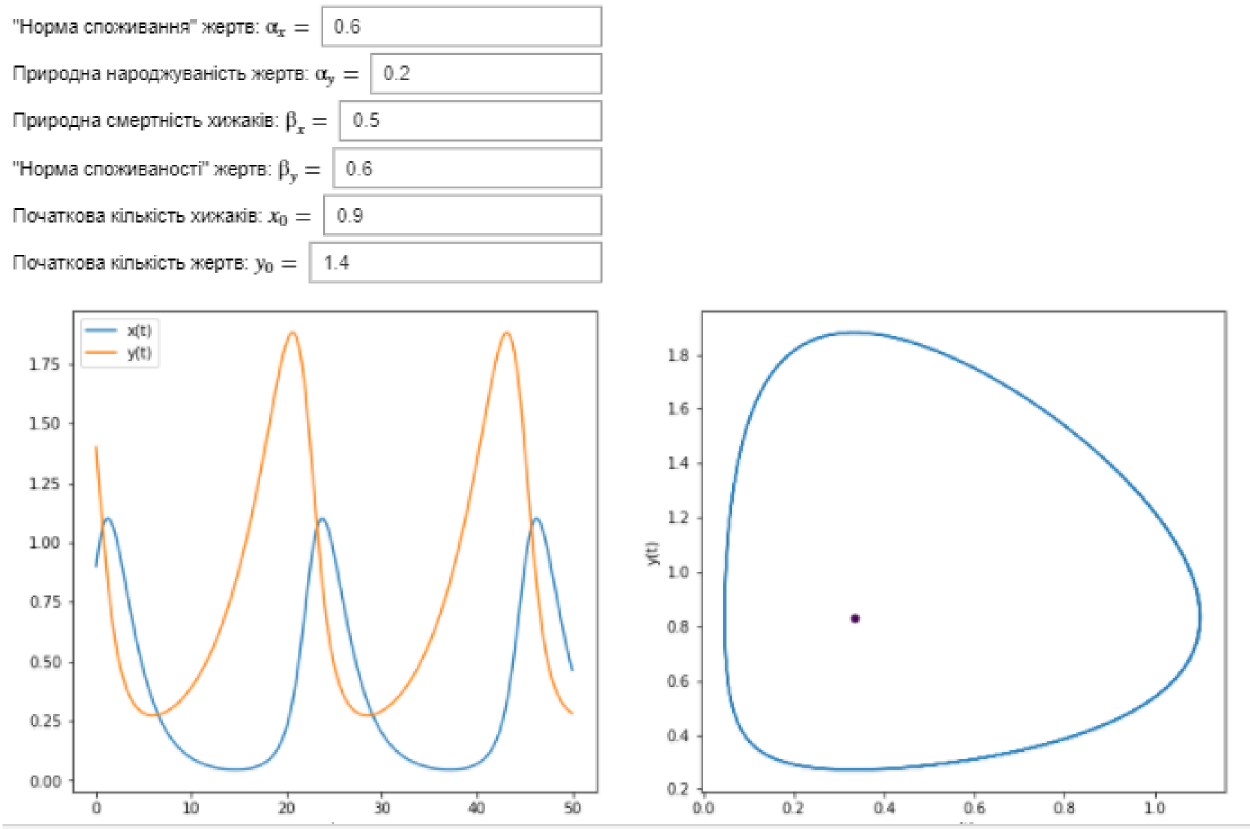

Рис. 15. Розв'язок системи диференціальних рівнянь моделі Лотки-Вольтерра за початкових умов $x_0 = 0.9$; $y_0 = 1.4$

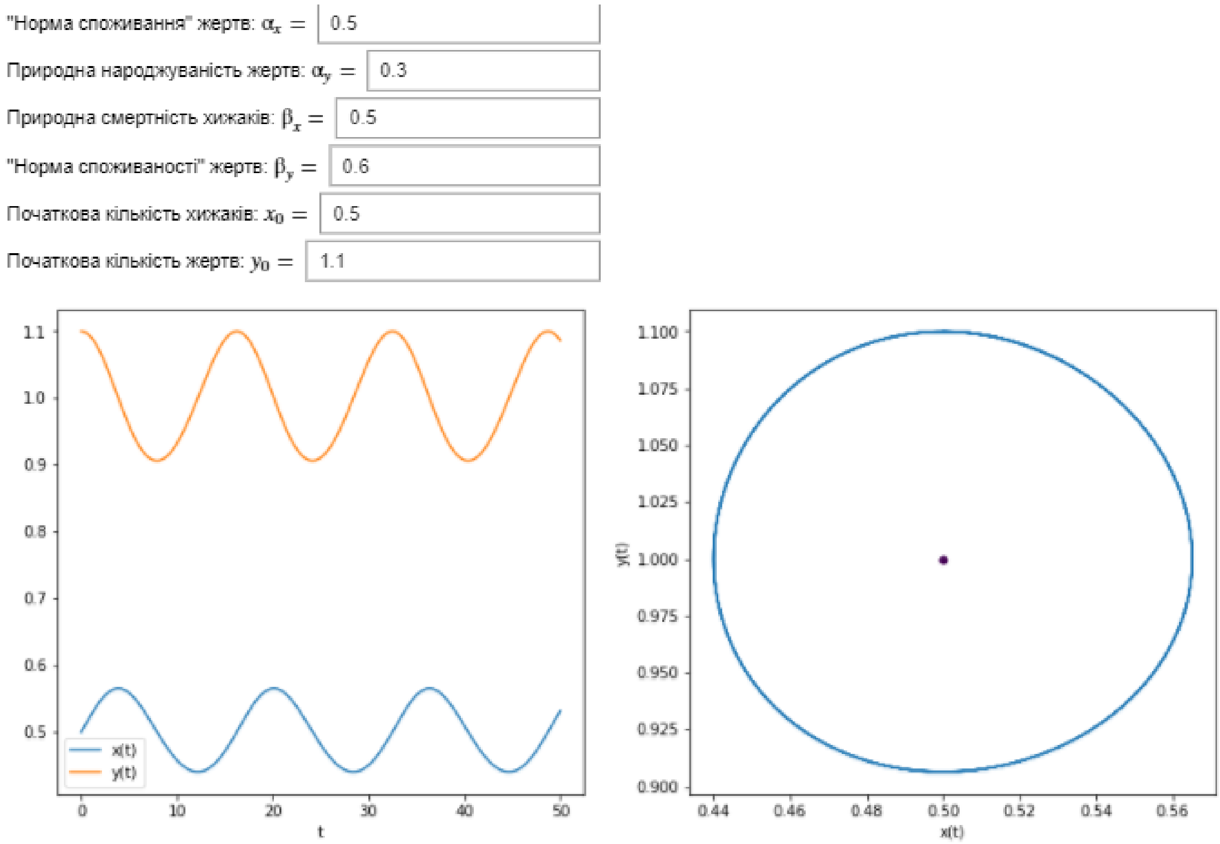

Рис. 16. Розв'язок системи диференціальних рівнянь моделі Лотки-Вольтерра за початкових умов x_0 =0.5; y_0 = 1.1

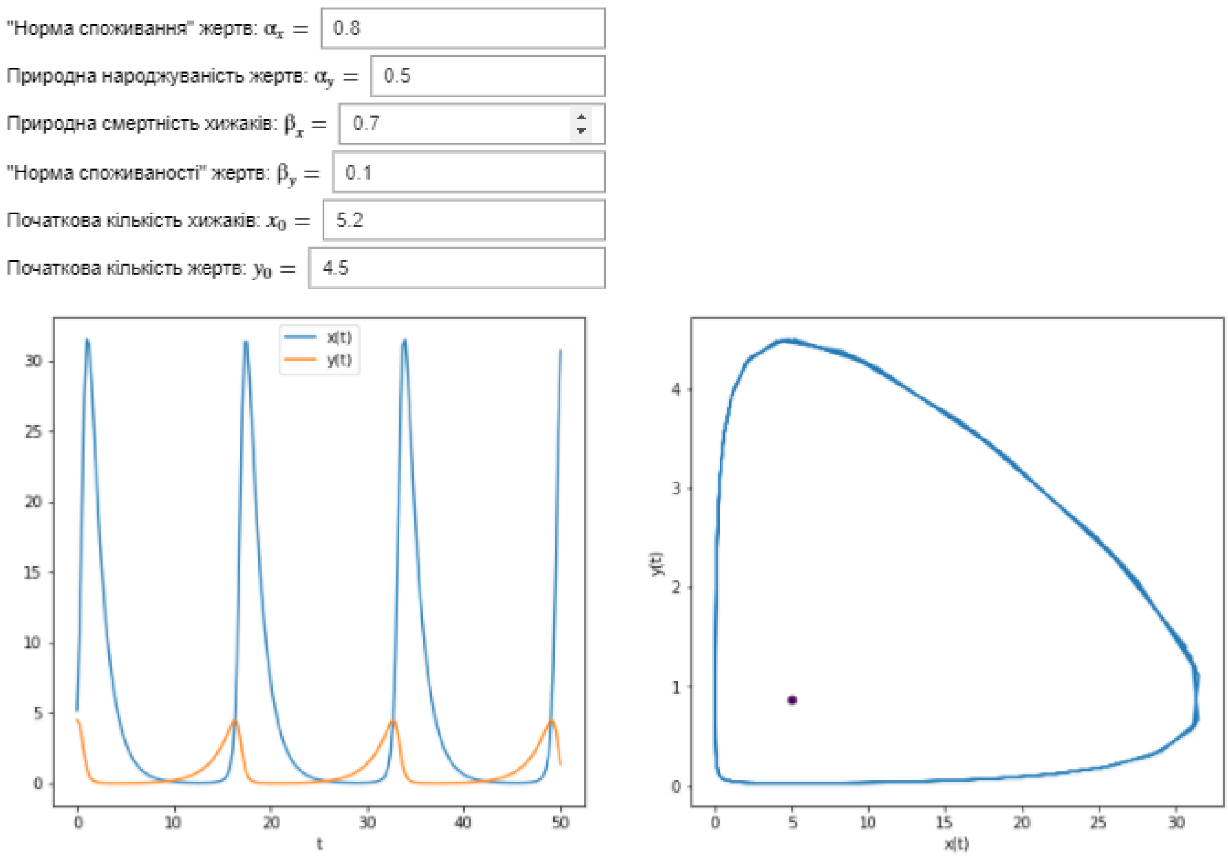

Рис. 17. Розв'язок системи диференціальних рівнянь моделі Лотки-Вольтерра за початкових умов x_0 =5.2; y_0 = 4.5

Отже, стаціонарна точка $x_0 = \frac{\alpha_y}{\beta_y}$, $y_0 = \frac{\beta_x}{\alpha_x}$ відповідає системі рівнянь

$$\begin{cases} x' = 0 \\ y' = 0 \end{cases}$$

Тому в цьому випадку чисельність жертв і хижаків залишається сталою. З відхиленням від стаціонарної точки ми спостерігаємо коливальні процеси зміни популяції хижаків і жертв.

Частота коливань чисельності хижаків і жертв одна й та сама і визначається лише власними характеристиками популяції – природною народжуваністю жертв і природною смертністю хижаків, а саме:

$$\omega_0 = \sqrt{\alpha_y \beta_x}$$

Амплітуди і фази коливань різні і залежать від початкових умов.

Програмний код

```
from ipywidgets import *
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from IPython.display import display
get_ipython().magic('matplotlib inline')

from pylab import rcParams
rcParams['figure.figsize'] = 14, 6
style = {'description_width': 'initial'}
layout = Layout(width = '400px')
models = [None] * 4


# ## 1. Динаміка росту популяції

# In[337]:

def model1(k, μ, N0):
    def model1_eq(N,t):
        return μ*N*(k-N)

    t = np.linspace(0, 30, num=200)

    # solve ODE
    N = odeint(model1_eq, N0, t)

    # plot results
    plt.plot(t,N)
    plt.xlabel('t')
    plt.ylabel('N(t)')
    plt.show()


# In[338]:

description1 = Label(value="$N'=μN(k-N)$")
k_val = FloatText(value= 15, description = 'Ємність середовища (гранична чисельність):
$k = $', step=0.1, style = style, layout = layout)
μ_val = FloatText(value = 0.05, description = 'Швидкість росту популяції: $μ = $',
step=0.1, style = style, layout = layout)
N0_val = FloatText(value=0.5, description = 'Початковий розмір популяції: $N_0 = $',
step=0.1, style = style, layout = layout)
models[0] = VBox([description1, interactive(model1, k=k_val, μ=μ_val, N0=N0_val,
continuous_update=True)])


# In[339]:

display(models[0])


# ## 2. Односекторна модель економічної динаміки (модель Солоу)

# In[340]:

def model2(s, a, α, μ, q, k0):
    def model2_eq(k,t):
        return s*a*k**α-(μ+q)*k

    t = np.linspace(0, 100, num=200)

    # solve ODE
    k = odeint(model2_eq, k0, t)

    # plot results
    plt.plot(t, k)
```

```
    plt.xlabel('t')
    plt.ylabel('k(t)')
    plt.show()


# In[341]:

description2 = Label(value="$k'=sak^α-(μ+q)k$")
s_val = FloatText(value=0.2, description = 'Норма накопичення: $s = $', step=0.1,
style = style, layout = layout)
a_val = FloatText(value=2.5, description = 'Рівень розвитку економіки: $a = $',
step=0.1, style = style, layout = layout)
α_val = FloatText(value=0.3, description = 'Частка капіталу у сукупній продукції: $α =
$', step=0.1, style = style, layout = layout)
μ_val = FloatText(value=0.1, description = 'Норма витрат на амортизацію: $μ = $',
step=0.1, style = style, layout = layout)
q_val = FloatText(value=0.1, description = 'Темп приросту робочої сили: $q = $',
step=0.1, style = style, layout = layout)
k0_val = FloatText(value=0.8, description = 'Початковий капітал: $k_0 = $', step=0.1,
style = style, layout = layout)
models[1] = VBox([description2, interactive(model2, s=s_val, a=a_val, α=α_val,
μ=μ_val, q=q_val, k0=k0_val, continuous_update=True)])


# In[342]:

display(models[1])


# ## 3. Рівняння вимушених коливань

# In[343]:

def model3(δ, ω0, ω, f0, x0, x00):

    #x_0' = x_1 = x'
    #x_1' = x'' = f0 * np.cos(ω * t) - 2 * δ * x[1] - (ω0 ** 2) * x[0]
    def model3_eq(x,t):
        return [x[1], f0 * np.cos(ω * t) - 2 * δ * x[1] - (ω0 ** 2) * x[0]]

    t = np.linspace(0, 50, num=200)

    # solve ODE
    x = odeint(model3_eq, np.array([x0, x00]), t)

    # plot results
    fig, ax = plt.subplots(nrows=1, ncols=2)
    ax[0].plot(t, x[:,0])
    ax[0].set_xlabel('t')
    ax[0].set_ylabel('x(t)')
    ax[1].plot(x[:,0], x[:,1])
    ax[1].set_xlabel('x(t)')
    ax[1].set_ylabel("x'(t)")
    plt.show()


# In[344]:

description3 = Label(value="$x''+2δ x'+ω_0^2 x=f_0 cos(ωt)$")
δ_val = FloatText(value=0.1, description = 'Коефіцієнт згасання: $δ = $', step=0.1,
style = style, layout = layout)
ω0_val = FloatText(value=0.3, description = 'Власна частота: $ω_0 = $', step=0.1,
style = style, layout = layout)
ω_val = FloatText(value=0.1, description = 'Частота зовнішньої сили: $ω = $',
step=0.1, style = style, layout = layout)
f0_val = FloatText(value=0.0, description = 'Амплітуда зовнішньої сили: $f_0 = $',
step=0.1, style = style, layout = layout)
x0_val = FloatText(value=0.3, description = 'Початкове положення: $x_0 = $', step=0.1,
style = style, layout = layout)
x00_val = FloatText(value=0.2, description = "Початкова швидкіть: $x_0' = $",
step=0.1, style = style, layout = layout)
```

```
models[2] = VBox([description3, interactive(model3, δ=δ_val, ω0=ω0_val, ω=ω_val,
f0=f0_val, x0=x0_val, x00=x00_val, continuous_update=True)])


# In[345]:

display(models[2])


# ## 4. Рівняння коливань у системі «хижак-жертва»

# In[346]:

def model4(αx, αy, βx, βy, x0, y0):
    def model4_eq(x,t):
        return [(αx * x[1] - βx) * x[0], (αy - x[0] * βy) * x[1]]

    t = np.linspace(0, 50, num=200)

    # solve ODE
    x = odeint(model4_eq, np.array([x0, y0]), t)

    # plot results
    fig, ax = plt.subplots(nrows=1, ncols=2)
    ax[0].plot(t, x[:,0], label = 'x(t)')
    ax[0].plot(t, x[:,1], label = 'y(t)')
    ax[0].legend()
    ax[0].set_xlabel('t')
    ax[1].plot(x[:,0], x[:,1])
    ax[1].set_xlabel('x(t)')
    ax[1].set_ylabel("y(t)")
    plt.show()


# In[347]:

description4 = Label(value=r"""$ \bigg\{ \matrix{x'=(α_x y - β_x)x \cr y'=(α_y - β_y
x)y} $""", layout = Layout(height = '50px'))
αx_val = FloatText(value=0.3, description = '"Норма споживання" жертв: $α_x = $',
step=0.1, style = style, layout = layout)
αy_val = FloatText(value=0.3, description = 'Природна народжуваність жертв: $α_y = $',
step=0.1, style = style, layout = layout)
βx_val = FloatText(value=0.3, description = 'Природна смертність хижаків: $β_x = $',
step=0.1, style = style, layout = layout)
βy_val = FloatText(value=0.6, description = '"Норма споживаності" жертв: $β_y = $',
step=0.1, style = style, layout = layout)
x0_val = FloatText(value=0.3, description = 'Початкова кількість хижаків: $x_0 = $',
step=0.1, style = style, layout = layout)
y0_val = FloatText(value=0.2, description = "Початкова кількість жертв: $y_0 = $",
step=0.1, style = style, layout = layout)
models[3] = VBox([description4, interactive(model4, αx=αx_val, αy=αy_val, βx=βx_val,
βy=βy_val, x0=x0_val, y0=y0_val, continuous_update=True)])


# In[348]:

display(models[3])
```