**Міністерство освіти і науки України**

**Черкаський національний університет імені Богдана Хмельницького**

**Інститут фізики, математики та комп'ютерно-інформаційних систем**

**Кафедра прикладної математики та інформатики**

Дисципліна: **Математичне моделювання**

**З В І Т**

**з лабораторної роботи № 3**

**Тема: „ Математичні моделі,**

**які є системами звичайних диференціальних рівнянь”**

студента 4-го курсу спеціальності «Прикладна математика»

***Годованюк Матвія Ігоровича***

**Черкаси – 2016 р.**

**Частина – 1**

У завданнях №1-2 необхідно виконати та представити основні етапи комп’ютерного моделювання за наступною схемою:

1. Постановка задачі.

2. Побудова математичної моделі.

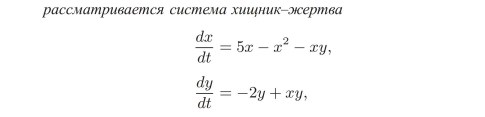
3. Розробка алгоритму і написання програми.

4. Проведення обчислювального експерименту.

5. Аналіз та інтерпретація результатів.

**Завдання 1.** Модель «хижак-жертва». Навести графіки залежності розміру популяцій від часу та фазову площину. Початкові умови обрати самостійно.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

X – популяція жертв

Y - популяція хижаків

a = 5 - коефіцієнт приросту жертв

b = 2 - швидкість приросту популяції хижаків

p = 1 - коефіцієнт взаємодій

q = 1 - коефіцієнт взаємодій

**3. Розробка алгоритму і написання програми.**

% predator-pray

function Predator\_prey

dt = 0.1;

N = 100;

t(1) = 0;

x(1) = 6; % population of victims

y(1) = 6; % population of predators

a = 5; % growth rate of the victims

b = 2; % growth rate of predator population

p = 1; % ratio of interactions

q = 1; % ratio of interactions

for i = 1:N

x(i+1) = x(i) + (a\*x(i) -x(i)\*x(i) - p\*x(i)\*y(i))\*dt;

y(i+1) = y(i) + (-b\*y(i) + q\*y(i)\*x(i))\*dt;

t(i+1) = t(i) + dt;

end

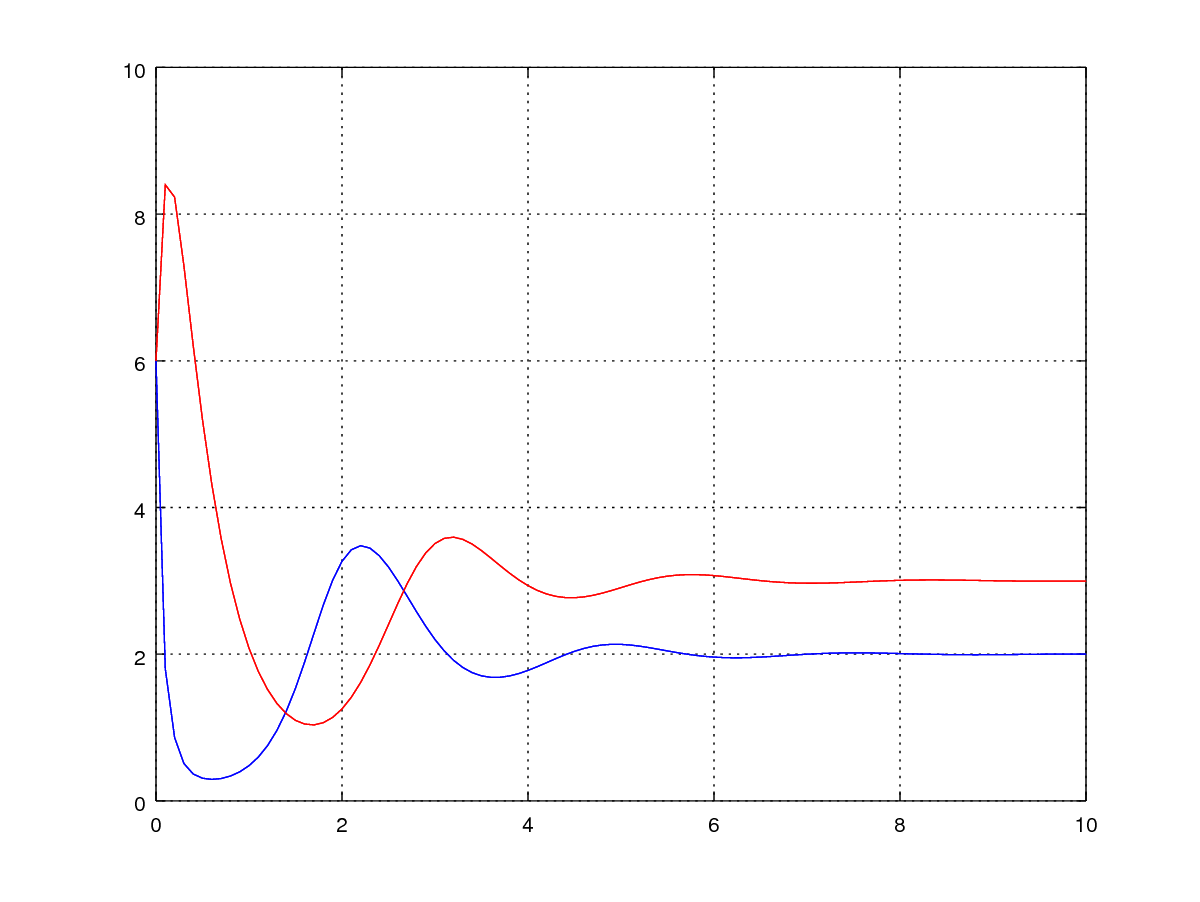
plot(t,x, 'b') % population of victims

grid on

hold on

plot(t,y, 'r') % population of predators

**4. Проведення обчислювального експерименту.**

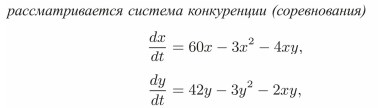
****

**5. Аналіз та інтерпретація результатів.**

Система є саморегульованою та за поагних початкових умов переходить всеж переходить у стійкий стан.

**Завдання 2.** Модель конкуренції видів. Навести графіки залежності розміру популяцій від часу та фазову площину. Початкові умови обрати самостійно.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

X – популяція 1

Y - популяція 2

a1 = 60 - коефіцієнт швидкості приросту популяції 1

b1 = 3 - коефіцієнт швидкості природнього приросту популяції 1

c1 = 4 - коефіцієнт, який описує конуренцію видів

a2 = 42 - коефіцієнт швидкості приросту популяції 2

b2 = 3 - коефіцієнт швидкості природнього приросту популяції 2

c2 = 2 - коефіцієнт, який описує конуренцію видів

**3. Розробка алгоритму і написання програми.**

% competition types

function competition\_model

dt = 0.01;

N = 250;

t(1) = 0;

x(1) = 10; % population first

y(1) = 20; % population druih

a1 = 60; % speed ratio of the population 1

b1 = 3; % speed ratio of natural increase of population 1

c1 = 4; % coefficient that describes the konurentsiya types

a2 = 42; % speed ratio of the population 2

b2 = 3; % speed ratio of natural increase of population 2

c2 = 2; % coefficient that describes the konurentsiya types

for i = 1:N

x(i+1) = x(i) + (x(i)\*(a1 - b1\*x(i) - c1\*y(i)))\*dt;

y(i+1) = y(i) + (y(i)\*(a2 - b2\*y(i) - c2\*x(i)))\*dt;

t(i+1) = t(i) + dt;

end

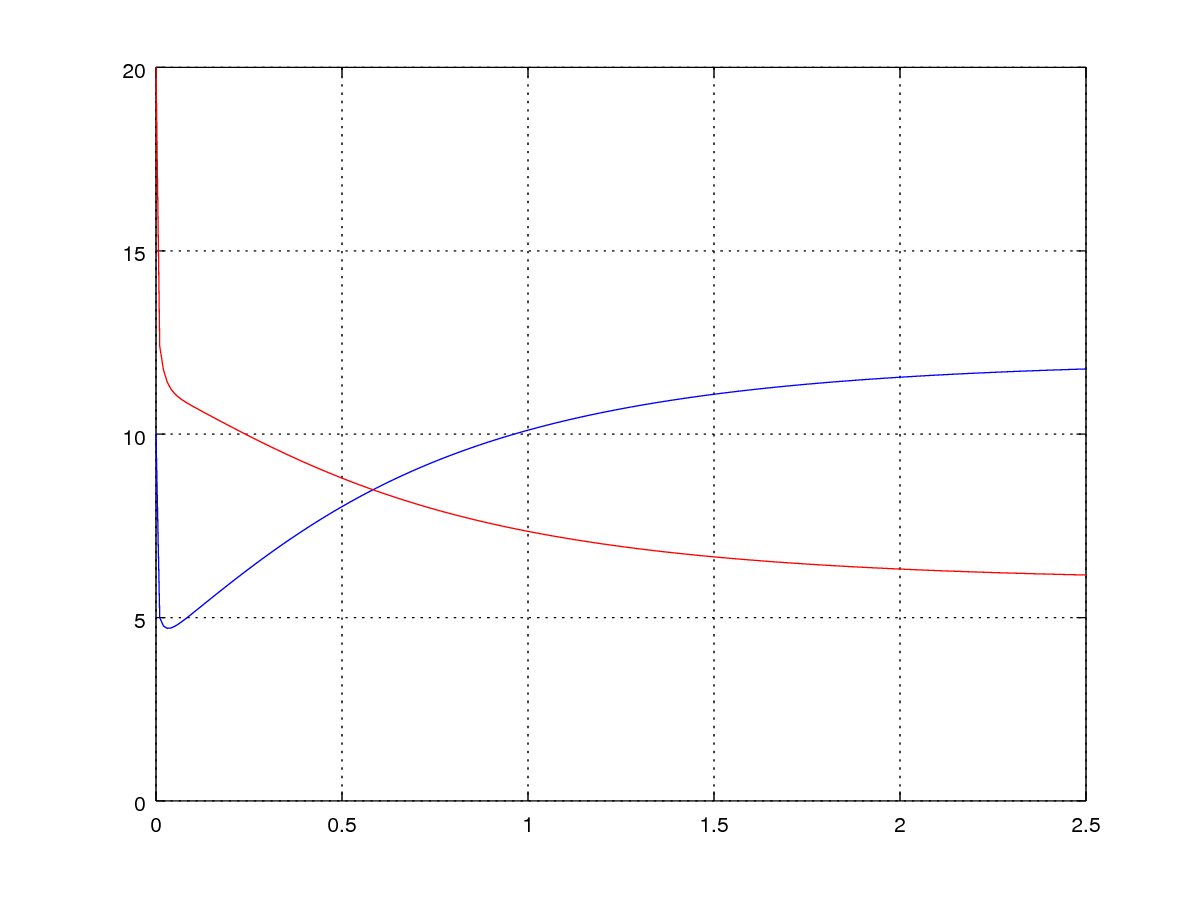
plot(t,x, 'b') % population 1

grid on

hold on

plot(t,y, 'r') % population 2

**4. Проведення обчислювального експерименту.**



**5. Аналіз та інтерпретація результатів.**

При різних початкових даних ми отримаємо різні зміни в популяції, так, наприклад, якщо ми візьмемо X0 = 20 – популяція 1 а Y0 = 10 – популяція 2, то перші Х повністю витісняють других У за проміжок часу, який рівний 1.5. А при X0 = 10 – популяція 1 а Y0 = 20 – популяція 2, бачимо що ці дві популяції можуть співіснувати, при кому їхня кількість прирівнюється до певного фіксованого значення.

**Частина – 2**

У завданнях №3-5 необхідно виконати та представити основні етапи комп’ютерного моделювання за наступною схемою:

1. Постановка задачі.

2. Побудова математичної моделі.

3. Розробка алгоритму і написання програми.

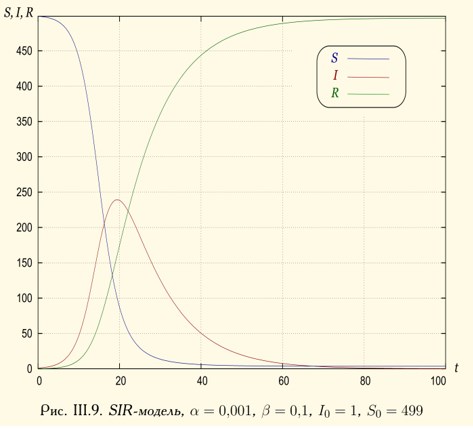
4. Проведення обчислювального експерименту.

5. Аналіз та інтерпретація результатів

**Завдання 3.** Модель «SIR».

Навести графіки залежності кількості людей кожного класу від часу. Початкові умови та параметри обрати самостійно. Підібрати та продемонструвати на графіках, при яких початкових умовах та параметрах а) епідемія хвороби відбудеться, б) епідемії не відбудеться.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

S(1) = 499 - неінфіковані

R(1) = 0 - ізольовані, мають імунітет, вилікувані

I(1) = 1 - інфіковані

alpha = 0.001 - ймовірність зараження

beta = 0.1 - відсоток вилікувані і ізольованих

**3. Розробка алгоритму і написання програми.**

% model of infection

function virus\_SIR\_model

dt = 0.1;

N = 1000;

t(1) = 0;

S(1) = 499; % uninfected

R(1) = 0; % insulated, are immune, cured

I(1) = 1; % infected

alpha = 0.001; % probability of infection

beta = 0.1; % percentage of cured isolated

for i = 1:N

S(i+1) = S(i) + (-alpha\*S(i) \* I(i))\*dt;

I(i+1) = I(i) + (alpha\*S(i) \* I(i) - beta\*I(i))\*dt;

R(i+1) = R(i) + (beta\*I(i))\*dt;

t(i+1) = t(i) + dt;

end

plot(t,S, 'b') % uninfected

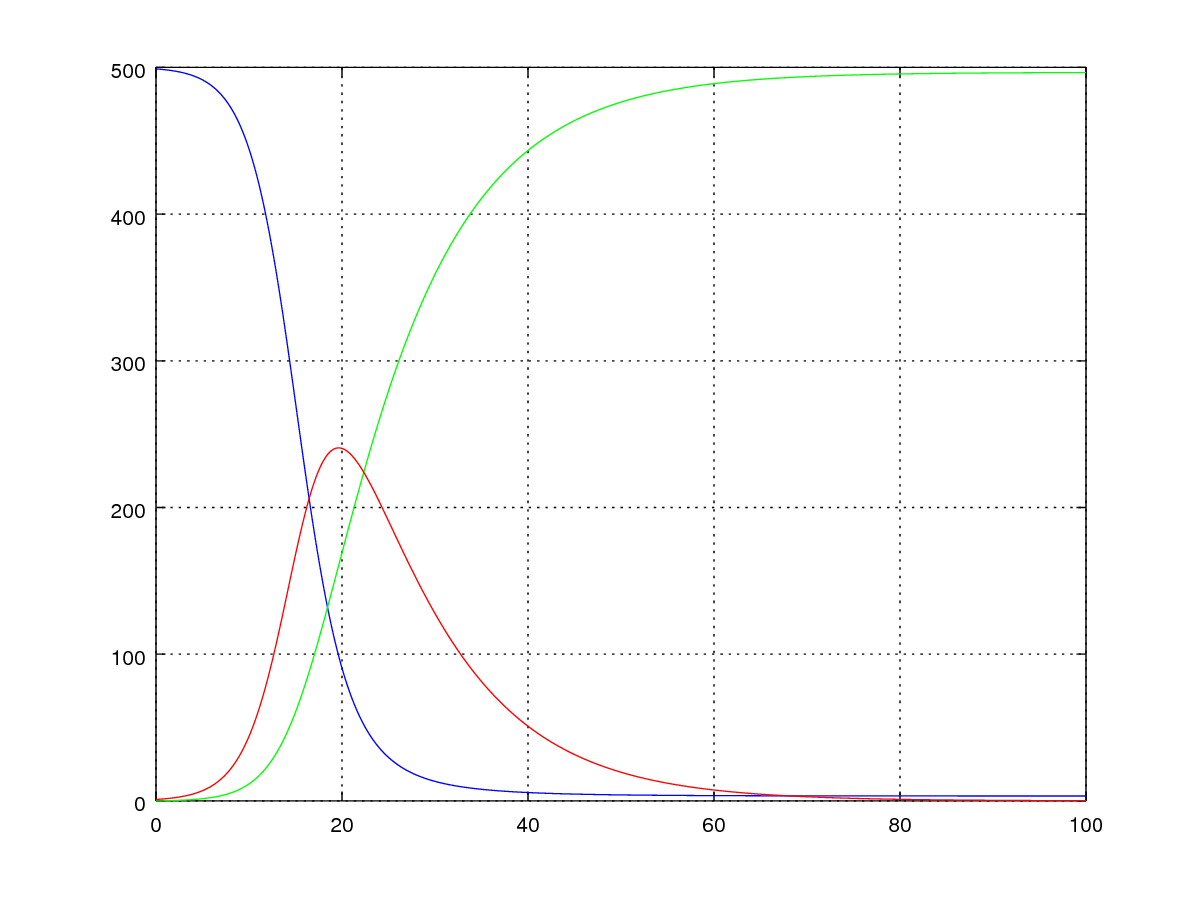
grid on

hold all

plot(t,R, 'g') % isolated, cured

plot(t,I, 'r')% infected

**4. Проведення обчислювального експерименту.**

****

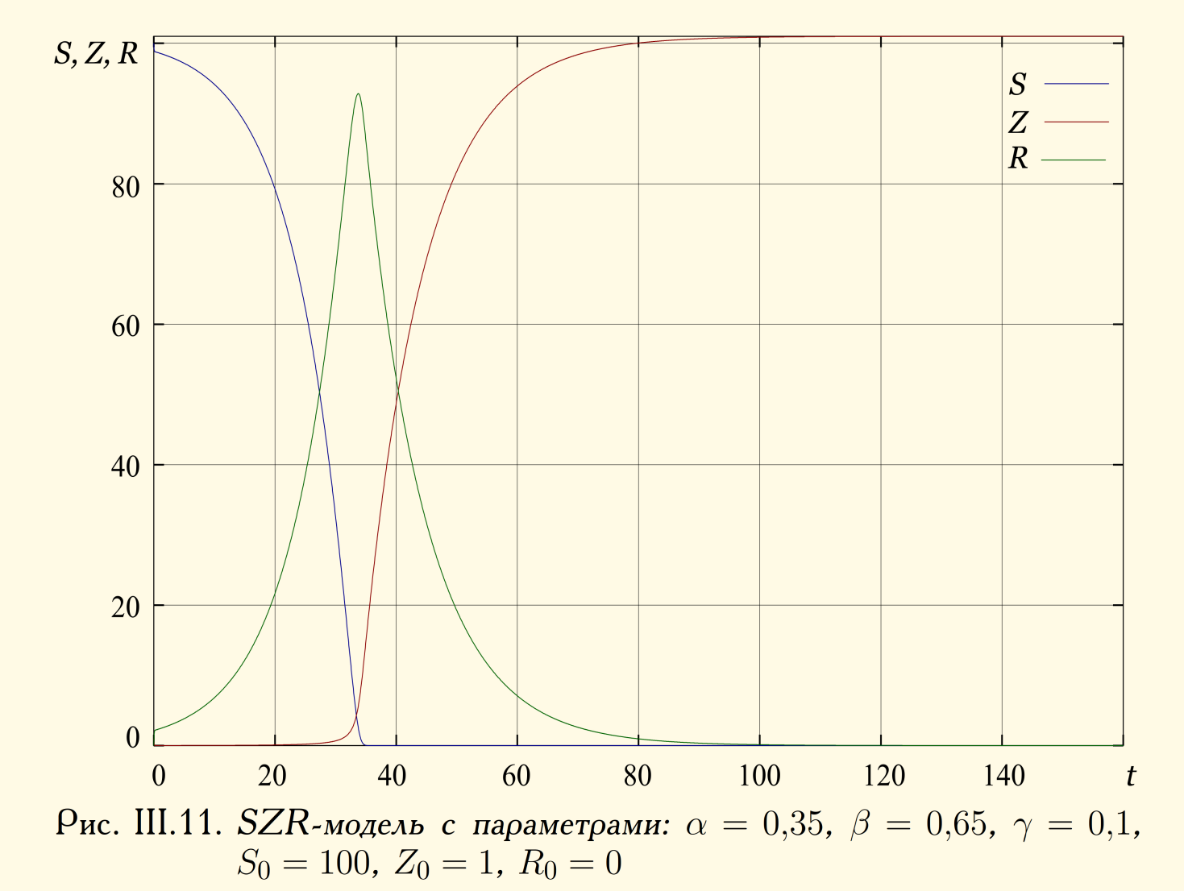
**5. Аналіз та інтерпретація результатів**

З плином часу кожен з хворих зробиться здоровим. У всіх стане якась імунітет який буде боротись з хворобою.

**Завдання 4.** Модель «SZR».

Навести графіки залежності кількості людей кожного класу від часу. Початкові умови та параметри обрати самостійно. Підібрати та продемонструвати на графіках, при яких початкових умовах та параметрах переможуть а) люди, б) зомбі.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

S(1) = 100 - не зомбі

R(1) = 0 - вбиті зомбі, що можуть повстати з мертвих

Z(1) = 1 - зомбі

alpha = 0.35 - ймовірність зараження при контакті

beta = 0.65 - ймовірність вбити зомбі

gamma = 0.1 - ймовірність повстати з мертвих

**3. Розробка алгоритму і написання програми.**

% model of the zombie infestation

function virus\_zombi\_SZR\_model

dt = 0.01;

N = 10000;

t(1) = 0;

S(1) = 100; % not a zombie

R(1) = 0; % dead zombies who can rise from the dead

Z(1) = 1; % zombie

alpha = 0.35; % the probability of infection upon contact

beta = 0.65; % chance to kill a zombie

gamma = 0.1; % probability to rise from the dead

for i = 1:N

S(i+1) = S(i) + (-alpha\*S(i) \* Z(i))\*dt;

Z(i+1) = Z(i) + (alpha\*S(i) \* Z(i) + gamma\*R(i)- beta\*Z(i)\*S(i))\*dt;

R(i+1) = R(i) + (beta\*Z(i) \* S(i) - gamma\*R(i))\*dt;

t(i+1) = t(i) + dt;

end

plot(t,S, 'g') % not a zombie

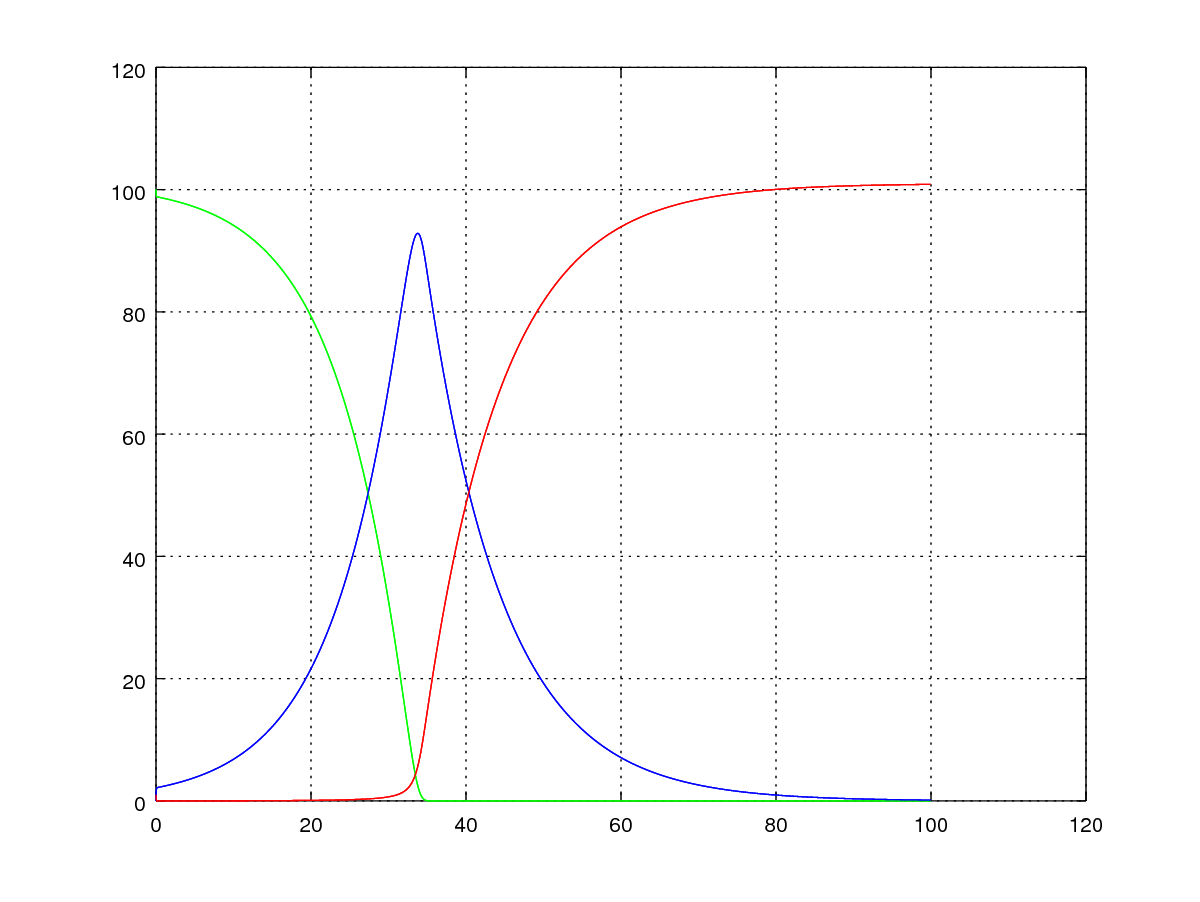
grid on

hold all

plot(t,R, 'b') % dead zombies who can rise from the dead

plot(t,Z, 'r') % zombies

**4. Проведення обчислювального експерименту.**



**5. Аналіз та інтерпретація результатів**

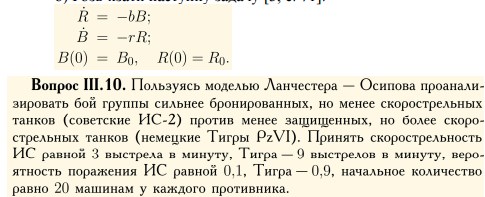
При різних початкових даних буде або зомбі – апокаліпсис, або повна перемога людей, якщо більше ніж 20% від загальної кількості людей стають зомбі – то перемога зомбі, якщо ж кількість зомбі менші ніж 20% то перемагають люди (при сприятливих умовах для людей), інакше перемога зомбі.

**Частина – 3**

**Завдання 5.** Модель бойових дій Осіпова-Ланчестера.

Навести графіки залежності кількості суперників кожного класу від часу. Початкові умови та параметри обрати самостійно. Підібрати та продемонструвати на графіках, при яких початкових умовах та параметрах переможуть а) перші, б) другі.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

B(1) = 20 - армія радянських

R(1) = 20 - армія німецьких

b = 3\*0.9 - ймовірність розбиття танка 1

r = 9\*0.1 - ймовірність розбиття танка 2

**3. Розробка алгоритму і написання програми.**

% model of hostilities

function war\_model

dt = 0.01;

N = 43;

t(1) = 0;

B(1) = 20; % army Soviet 1

R(1) = 20; % army German 2

b = 3\*0.9; % probability of breaking the tank 1

r = 9\*0.1; % probability of breaking the tank 2

for i = 1:N

R(i+1) = R(i) + (-b\*B(i)) \* dt;

B(i+1) = B(i) + (-r\*R(i)) \* dt;

t(i+1) = t(i) + dt;

end

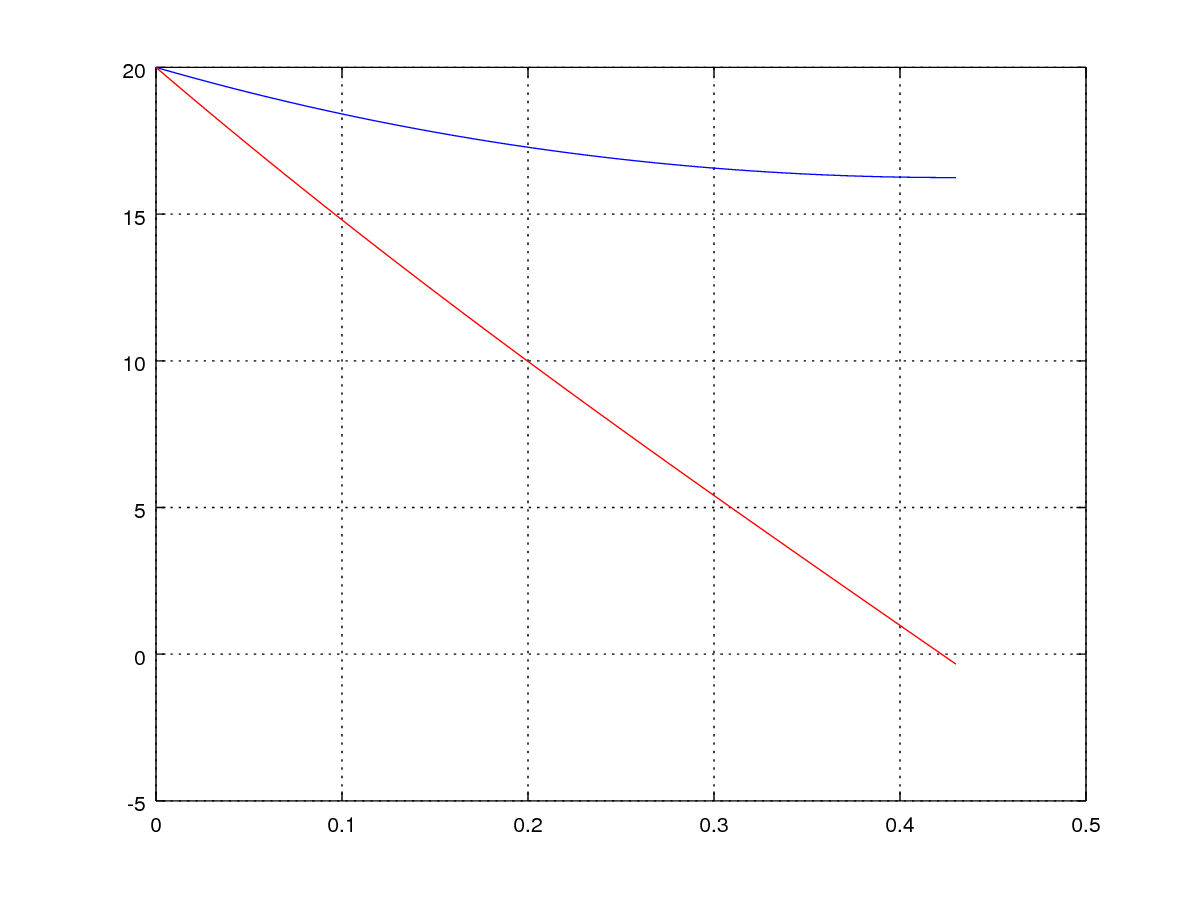
plot(t,B, 'b') % army 1

grid on

hold on

plot(t,R, 'r') % army 2

**4. Проведення обчислювального експерименту.**



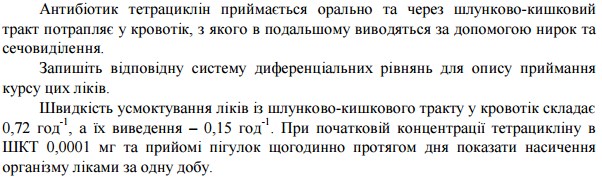
**5. Аналіз та інтерпретація результатів**

Армія синіх перемогла. Армія червоних дуже лінійно померала.

**Завдання 6**. Модель лікування антибіотиком.

Навести графіки зміни концентрації ліків у ШКТ та кровотоці з часом. Початкові умови та параметри обрати самостійно.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

I = 1/24 - швидкість надходження ліків

X(1) = 0.0001 - коефіцієнт ліків в ЖКТ

Y(1) = 0 - концентрація ліків у крові

k1 = 0.72 - коефіцієнт з жкт в кров

k2 = 0.15 - коефіцієнт виводу препарату

**3. Розробка алгоритму і написання програми.**

% model of the drug

function antibiotik\_model

dt = 0.1;

N = 480;

t(1) = 0;

I = 1/24; % the rate of arrival of medicines

X(1) = 0.0001; % ratio of drugs in the gastrointestinal tract

Y(1) = 0; % concentration of medicines in blood

k1 = 0.72; % ratio from the gastrointestinal tract into the blood

k2 = 0.15; % ratio of withdrawal of the drug

for i = 1:N

X(i+1) = X(i) + (I - k1\*X(i)) \* dt;

Y(i+1) = Y(i) + (k1\*X(i) - k2\*Y(i)) \* dt;

t(i+1) = t(i) + dt;

end

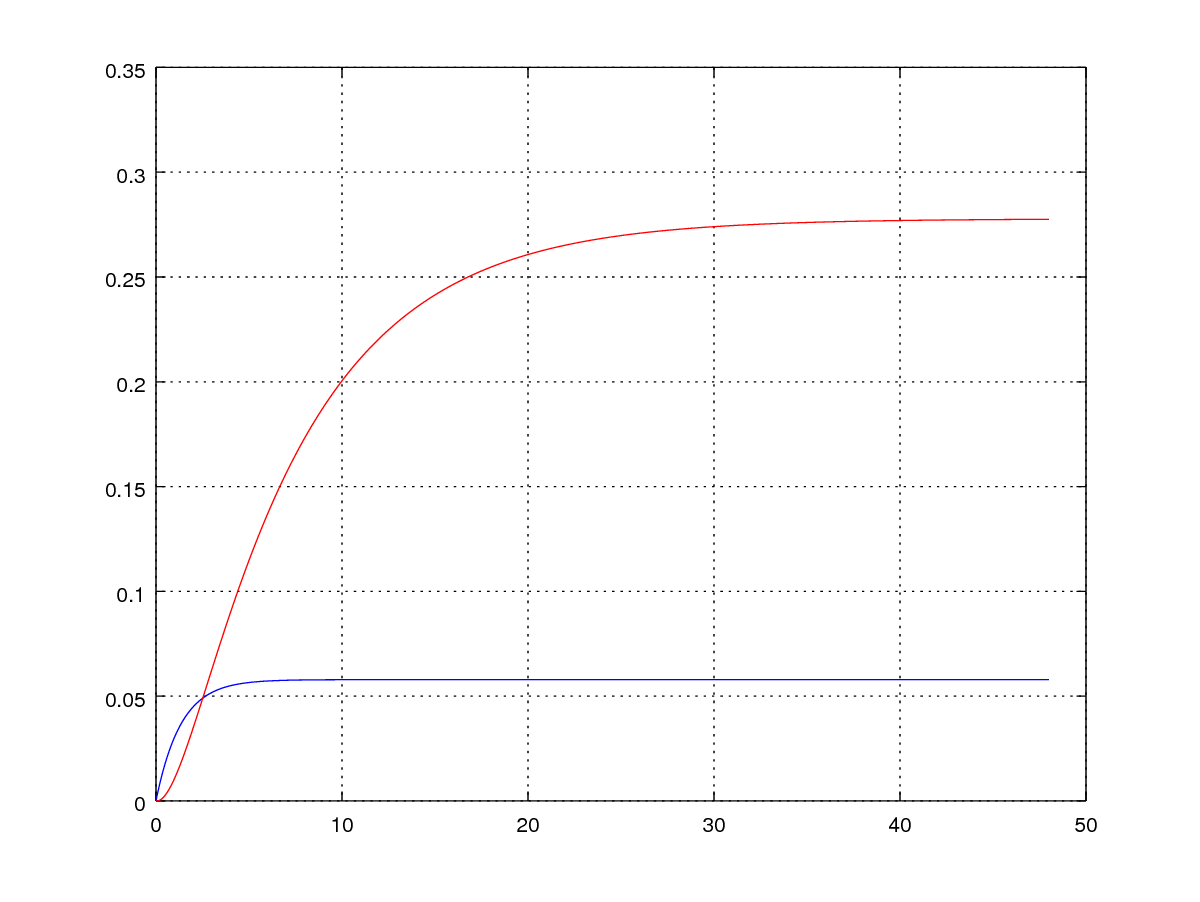
plot(t,X, 'b') % in the gastrointestinal tract

grid on

hold on

plot(t,Y, 'r') % in the blood

**4. Проведення обчислювального експерименту.**



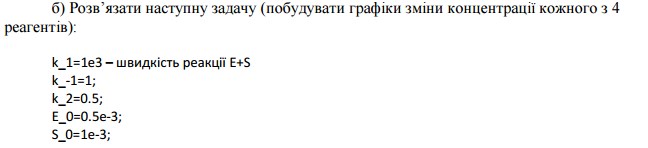
**5. Аналіз та інтерпретація результатів**

З часом концентрація ліків в ЖКТ досягає межі і залишається сталою приблизно 0.05 год-1, а концентрація ліків в крові збільшується, доки не дійде до певної межі, тобто, через 1 добу концентрація препарату буде 0.26 год-1. А М = 0.26 год-1 – цього препарату, при прийомі його кожної години.

**Завдання 7.** Модель ферментних реакцій.

Навести графіки зміни концентрації кожного з 4 реагентів з часом. Початкові умови та параметри обрати самостійно.

**1. Постановка задачі.**



**2. Побудова математичної моделі.**

S(1) = 1e-3 - субстрат

C(1) = 0 - комплекс (S+E)

E(1) = 0.5e-3 - фермент

P(1) = 0 - продукт

k1 = 1e3 - коефіцієнт (S+E) -> (SE)

k\_1 = 1 - коефіцієнт (S+E) <- (SE)

k2 = 0.5 - кофіцієнт SE -> (P+E)

**3. Розробка алгоритму і написання програми.**

% модель ферментних реакцій

function ferment\_model

dt = 0.1;

N = 100;

t(1) = 0;

S(1) = 1e-3; % субстрат

C(1) = 0; % комплекс (S+E)

E(1) = 0.5e-3; % фермент

P(1) = 0; % продукт

k1 = 1e3; % коефіцієнт (S+E) -> (SE)

k\_1 = 1; % коефіцієнт (S+E) <- (SE)

k2 = 0.5; % кофіцієнт SE -> (P+E)

for i = 1:N

S(i+1) = S(i) + ( k\_1 \* C(i) - k1 \* S(i) \* (E(1) - C(i)) ) \* dt;

C(i+1) = C(i) + ( k1 \* S(i) \* (E(1) - C(i)) - (k\_1 + k2) \* C(i) ) \* dt;

t(i+1) = t(i) + dt;

E(i+1) = E(1) - C(i);

P(i+1) = S(1) - S(i) - C(i);

end

plot(t,S, 'b') % субстрат

grid on

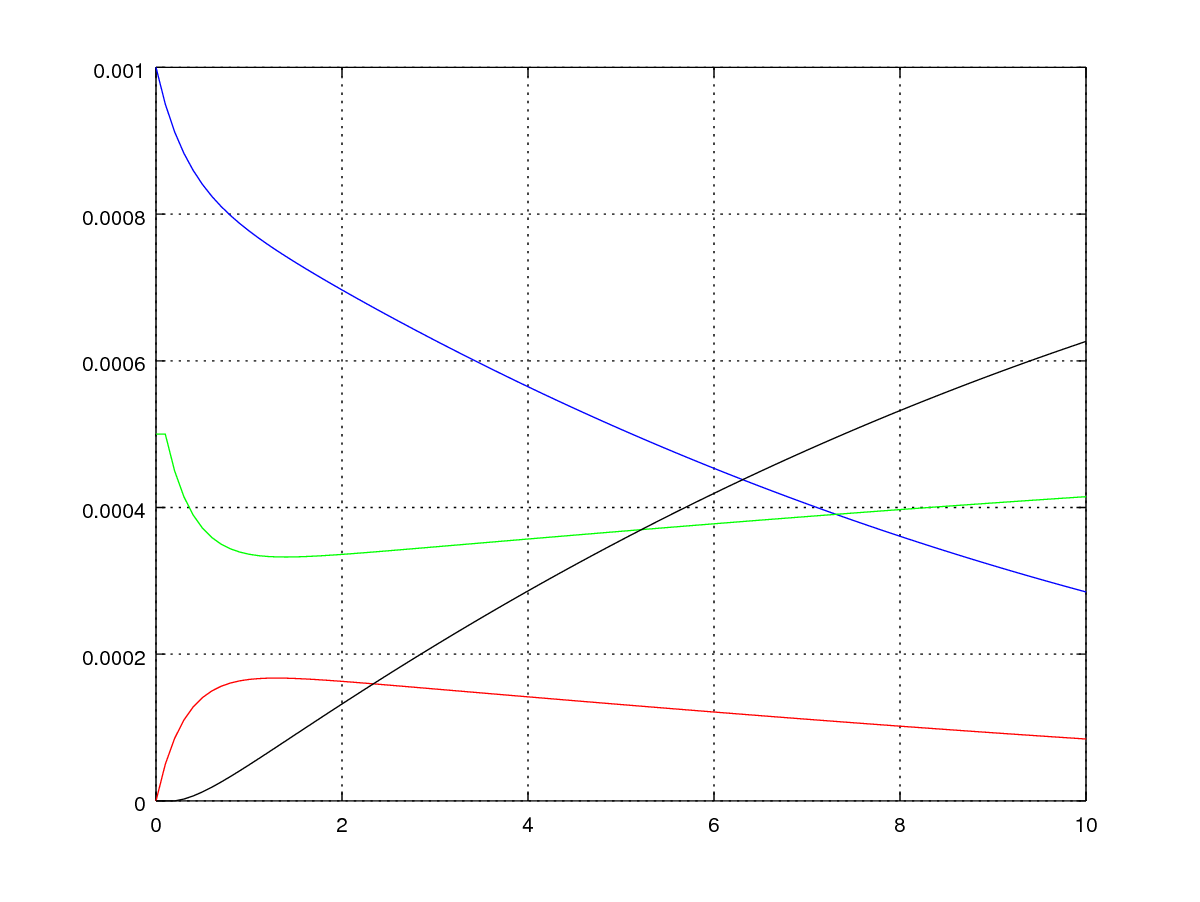
hold all

plot(t,C, 'r') % комплекс

plot(t,E, 'g') % фермент

plot(t,P, 'k') % продукт

**4. Проведення обчислювального експерименту.**



**5. Аналіз та інтерпретація результатів**

З часом Субстрат повністю зникає (він вичерпний), Фермент зменшується, бо витрачається на продукт, але потім знову відновлюється (після реакції), Комплекс після реакції зникає, осільки субстрат рівний 0, отже не утворюється комплекс і реакція не можлива, Продукт збільшується пропорційно до того, як зменшується кількість субстрату, доки той не вичерпається повністю.