**Софийски университет „Св. Климент Охридски“**

*Факултет по математика и информатика*

*Специалност: “Информационни системи ”*

*Курс: 2, Група: 3*

*Дисциплина: “Диференциални уравнения”*

***Проект***

*Тема: 32*

***Изготвил:***

Никола Петров Кирилов(71986)

*Зимен семестър 2020/2021 година*

**Задача 1.** Използвайки диференчна схема решете числено задачата на Коши.

y′′ +(2−sinx)y′ +12y = 0, −1 ≤ x ≤ 7, y(0) = −2, y′(0) = −1.

Използвайте стъпки 0.1, 0.01, 0.001 и изобразете резултатите в една и съща координатна система.

**Решение:**

*zad1.m*

clear all

h = -0.1;

x = 0:h:-1;

N = length(x);

y(1) = -2;

z0 = -1;

y(2) = y(1) + h\*z0 + h^2/2\*(-12\*y(1) - (2- sin(2))\*z0);

for n=2:N-1

y(n+1) = 2\*y(n) - y(n-1) + h^2\*(-12\*y(n) - (2- sin(x(n))\*(y(n) - y(n-1))/h));

end

plot(x,y,'g');

axis([-2,8,-2,8])

hold on;

h = 0.1;

x = 0:h:7;

N = length(x);

y(1) = -2;

z0 = -1;

y(2) = y(1) + h\*z0 + h^2/2\*(-12\*y(1) - (2- sin(2)\*z0));

for n=2:N-1

y(n+1) = 2\*y(n) - y(n-1) + h^2\*(-12\*y(n) - (2- sin(x(n))\*(y(n) - y(n-1))/h));

end

plot(x,y,'g');

axis([-2,8,-2,8])

clear all

%2stupka

h = -0.01;

x = 0:h:-1;

N = length(x);

y(1) = -2;

z0 = -1;

y(2) = y(1) + h\*z0 + h^2/2\*(-12\*y(1) - (2- sin(2))\*z0);

for n=2:N-1

y(n+1) = 2\*y(n) - y(n-1) + h^2\*(-12\*y(n) - (2- sin(x(n))\*(y(n) - y(n-1))/h));

end

plot(x,y,'b');

axis([-2,8,-2,8])

hold on;

h = 0.01;

x = 0:h:7;

N = length(x);

y(1) = -2;

z0 = -1;

y(2) = y(1) + h\*z0 + h^2/2\*(-12\*y(1) - (2- sin(2)\*z0));

for n=2:N-1

y(n+1) = 2\*y(n) - y(n-1) + h^2\*(-12\*y(n) - (2- sin(x(n))\*(y(n) - y(n-1))/h));

end

plot(x,y,'b');

axis([-2,8,-2,8])

%3stupka

h = -0.001;

x = 0:h:-1;

N = length(x);

y(1) = -2;

z0 = -1;

y(2) = y(1) + h\*z0 + h^2/2\*(-12\*y(1) - (2- sin(2))\*z0);

for n=2:N-1

y(n+1) = 2\*y(n) - y(n-1) + h^2\*(-12\*y(n) - (2 - sin(x(n))\*(y(n) - y(n-1))/h));

end

plot(x,y,'r');

axis([-2,8,-2,8])

hold on;

h = 0.001;

x = 0:h:7;

N = length(x);

y(1) = -2;

z0 = -1;

y(2) = y(1) + h\*z0 + h^2/2\*(-12\*y(1) - (2- sin(2)\*z0));

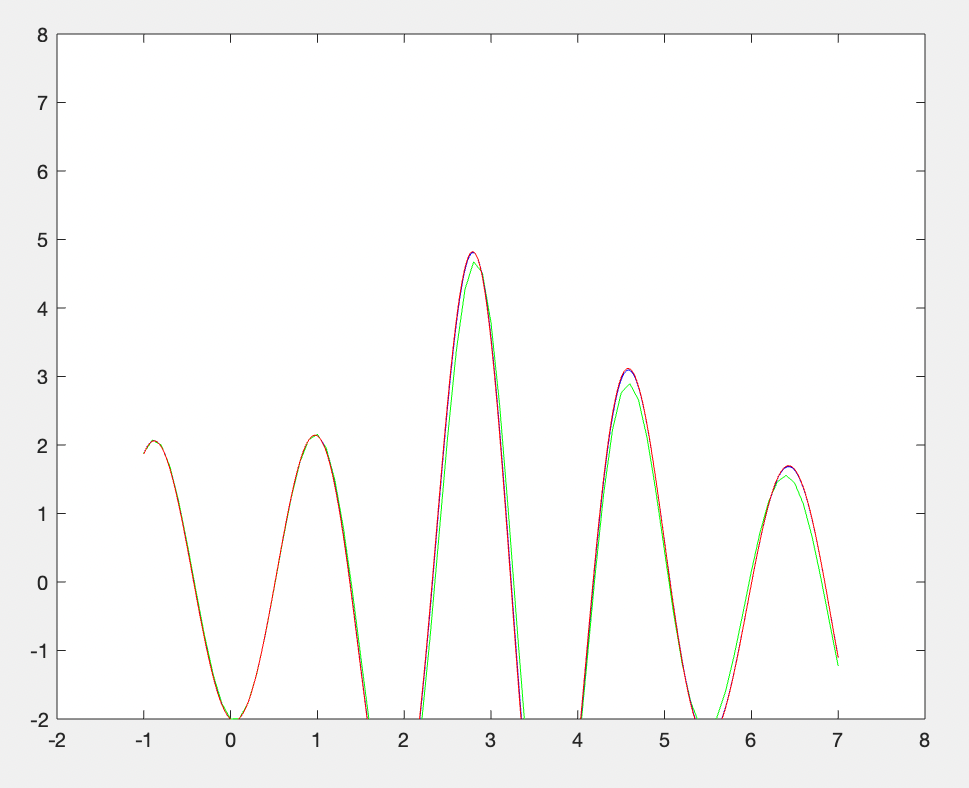
for n=2:N-1

y(n+1) = 2\*y(n) - y(n-1) + h^2\*(-12\*y(n) - (2 - sin(x(n))\*(y(n) - y(n-1))/h));

end

plot(x,y,'r');

axis([-2,8,-2,8])

****

# Задача 2.

Дадена е системата

А) Намерете равновесните точки на системата.

Б) Начертайте фазов портрет на системата.

В) Определете, кои равновесни точки са устойчиви.

Г) За решението на задачата на Коши за системата с начални данни x(0) = 6, y(0) = 4, направете анимация на движението на точката (x(t), y(t)) във фазовото пространство, когато времето t се мени от 0 до 1.

**Решение:**

\*Тъй като първоначално намерих по-стара версия на MATLAB, която беше от 2015 и трябваше да правя функциите на отделен func скрипт, но после намерих 2019 издание, на което е направено така, че всичко да работи дори и на едно място, но не съм ги променял.

*sys1.m*

function res = sys1(t,x)

res = [-x(2)^3 + x(1); -x(1)\*x(2) + 1];

end

*ravnovesni.m*

function res = ravnovesni(x,y)

res = solve('(-y)\*(-y)\*(-y) + x, (-x)\*y + 1')

end

*phase.m*

*\*първо чертаем координатните оси\**

plot([-3, 3], [0, 0])

hold on

plot([0, 0], [-3, 3])

hold on

axis ([-3, 3, -3, 3])

[x0, y0] = ginput(1);

while x0 >= -3 && x0 <= 3 && y0 >= -3 && y0 <= 3

[~, x] = ode45(@sys1, [0, 3], [x0, y0]);

plot(x(:, 1), x(:, 2), 'r')

[t, x] = ode45(@sys1, [0, -3], [x0, y0]);

plot(x(:, 1), x(:, 2), 'b')

[x0, y0] = ginput(1);

end

\*Използваме ode45(базирано на явния метод на Рунге-Кута от 4 и 5 ред)

*animacia.m*

[t,x] = ode45(@sys1, [0,1], [6,4]);

for k = 1:length(x)

plot(x(1:k,1),x(1:k,2))

hold on

plot(x(k,1),x(k,2),'\*')

axis([-3,3,-3,3]);

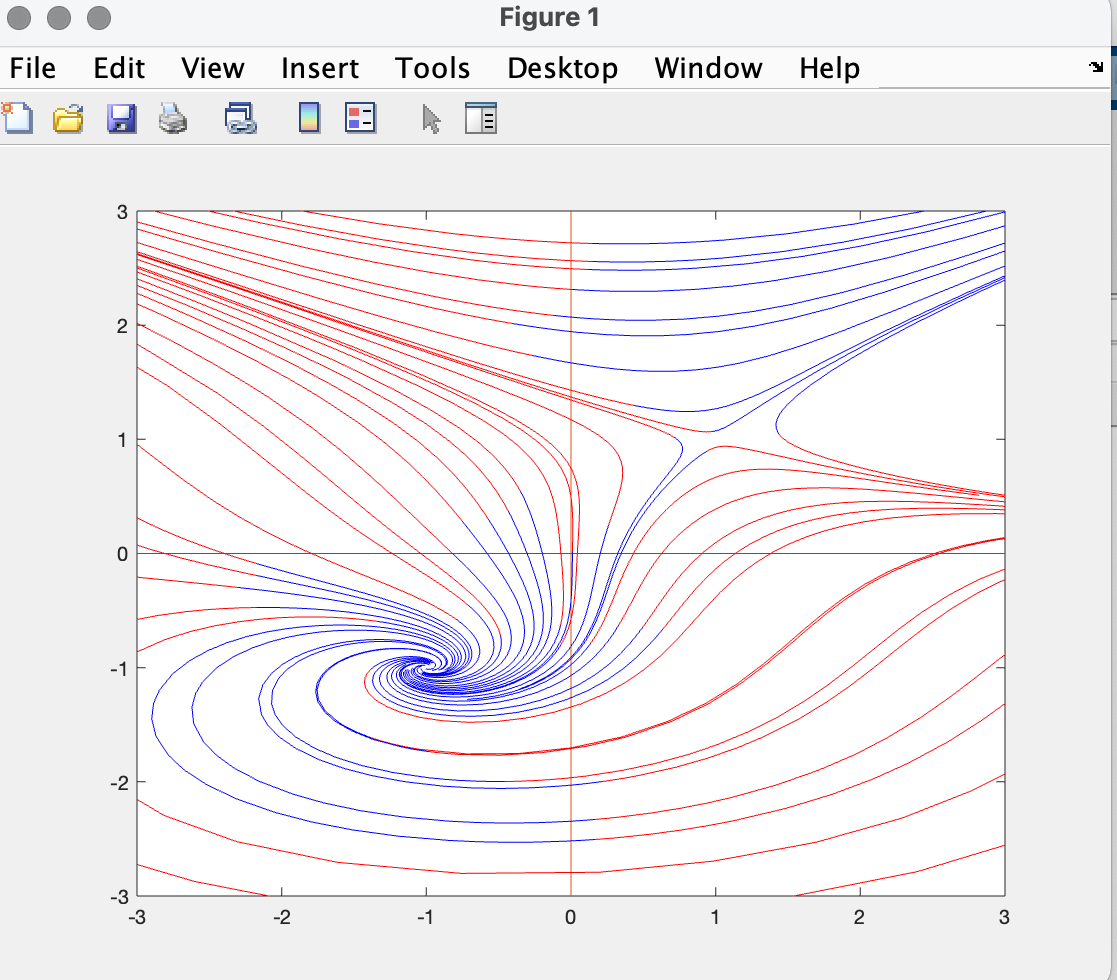
M = getframe;

hold off

end

movie(M)

Изглед от фазовия портрет:



**Задача 3.**

Дадена е смесената задача за уравнението на струната

utt = uxx 0<x<6, t>0

u(0, t) = 0 , t > 0

u(6, t) = 0 , t > 0

u(x,0)=

1+cos(πx), за 1 ≤ t ≤ 2

0, за. 0 < x < 1 и 2<x<6,

ut(x,0) = 0, 0 < х < 6

С помощта на формулата на Д’Аламбер, визуализирайте трептенето на струната за t ∈ [0, 12]

*phi.m*

function res=phi(x)

if x >=1 && x <= 2

res = 1 + cos(pi\*x);

else

res = 0;

end

end

*phi1.m*

function res = phi1(x)

for k = 1:length(x)

if x(k)>= 0 && x(k)<= 6

res(k) = phi(x(k));

elseif x(k) < 0

res(k) = phi1(-x(k));

else

res(k)= -phi1(12-x(k));

end

end

*int.m*

function res=intpsi1(a,b)

for k=1:length(a)

if a(k)==b(k) res(k)=0;

else xx=a(k):(b(k)-a(k))/100:b(k);

yy=psi1(xx);

res(k)=trapz(xx,yy);

end

end

*func.m*

x=0:0.05:6;

t=0:0.05:6;

for k=1:length(t)

u=(phi1(x+t(k))+phi1(x-t(k)))/2;

plot(x,u);

axis([0,6,0,6])

M(k)=getframe;

end

movie(M,3)

**