Analýza Nelineárnych Dynamických Systémov v MATLAB
Zadanie č. 2

Roiko Oleksii

11. januára 2024

Definícia Problému

Analyzujte tri nelineárne dynamické systémy (NDS): a) pružina a tlmič (prednášky) b) van der Polov oscilátor (SimSys) c) matematické kyvadlo (SimSys) Pre každý typ nelineárneho dynamického systému vyriešte nasledujúce úlohy: - v rámci analytického riešenia: a) prepíšte NDS do substitučného kanonického tvaru a vypočítajte jeho rovnovážne stavy b) vykonajte linearizáciu nelineárneho systému v každom vypočítanom rovnovážnom stave (= realizujte výpočet prvkov matice stavu A - Jakobiánu) a ku každému NDS uveďte lineárnu/e aproximáciu/cie vo všetkých rovnovážnych stavoch c) určte typy rovnovážnych stavov (charakter singulárnych bodov) a posúďte stabilitu v malom pre daný rovnovážny stav (aplikujte metódu 1. priblíženia podľa Ljapunova - t.j. zostavenie a výpočet koreňov CHR LDS). Vykonajte záver ohľadom stability daných typov NDS na základe výsledkov stability ich lineárnej aproximácie v rovnovážnych stavoch. - v rámci algoritmicko-simulačného riešenia: a) napíšte program na získanie časových priebehov x1(t), x2(t) z daného typu NDS a jeho lineárnej aproximácie pomocou vstavanej funkcie ode45 a vlastnej naprogramovanej funkcie Runge-Kutta b) napíšte funkciu, ktorá generuje zakreslenie fázového portrétu (pre zvolený typ NDS/LDS) s uvažovaním rôznych kombinácií parametrov a cyklickej zmeny počiatočných podmienok. Fázové portréty generujte za predpokladu, že uvažovaný NDS/LDS je a) bez budenia (u=0) b) s budením (u(t) – je potrebné zvoliť vhodný budiaci signál).

Moje Riešenie (van der Polov oscilátor)

```
file unlin.m

function xder = unlin(~, u, mu)
    u1 = u(1);
    u2 = u(2);

    u1dt = u2;
    u2dt = mu*(1 - u1.^2).*u2 - u1;

    xder=[u1dt; u2dt];
end
```

file main.m

```
mu = 0.8; % Define the parameter for the Van der Pol oscillator

tspan = [0 10];

figure;

for i = -5:1:5
    for j = -5:1:5
        x0 = [i j];
        [t, x] = ode45(@(t,x) unlin(t,x,mu), tspan, x0);
        plot(x(:,1), x(:,2),'r');
        hold on;
    end
end

title('Phase Portrait of the Van der Pol Oscillator');
xlabel('x');
ylabel('y');
[X, Y] = meshgrid(-10:10, -10:10);
U = Y;
V = mu*(1 - (X.^2)).*Y - X;
% Create a mask for the center of the phase portrait
centerMask = (abs(X) < 3) & (abs(Y) < 3);

% Calculate U and V for the center
U_center = U .* centerMask;
V_center = V .* centerMask;

% Calculate U and V for the rest of the plot
U_rest = U .* ~centerMask;
V_rest = V .* ~centerMask;

% Create the quiver plot for the center with a smaller scale
  factor
quiver(X, Y, U_center, V_center, 1.7, 'g');

hold on;

% Create the quiver plot for the rest of the plot with a larger
  scale factor
quiver(X, Y, U_rest, V_rest, 2);

xlim([-10,10])
ylim([-10,10])
```

```
file main2.m

x0 = [0 1];
tspan = [0 50];
mu = 1;

[t,x] = ode45(@(t,x) unlin(t,x,mu),tspan,x0);

plot(t,x);
```

file main3.m

```
mu = 1; % Define the parameter for the Van der Pol oscillator

tspan = [0 10];

figure;

syms x1 x2 real

f = [x2;mu*(1 - x1.^2).*x2 - x1];

J = jacobian(f, [x1; x2]);

J_at_equilibrium = subs(J, [x1; x2], [0; 0]);

J_at_equilibrium = double(J_at_equilibrium);

f_linear = @(t, Y) J_at_equilibrium * Y;

 % x0 = [1 1];
 % [t, x] = ode45(f_linear, tspan, x0);
 % plot(t, x,'r');

for i = -5:1:5
    for j = -5:1:5
        x0 = [i j];
        [t, x] = ode45(f_linear, tspan, x0);
        plot(x(:,1), x(:,2),'r');
        hold on;
    end
end

title('Phase Portrait of linear Van der Pol Oscillator');
xlabel('x');
ylabel('y');
[X, Y] = meshgrid(-10:10, -10:10);
U = J_at_equilibrium(1,1)*X + J_at_equilibrium(1,2)*Y;
V = J_at_equilibrium(2,1)*X + J_at_equilibrium(2,2)*Y;
quiver(X, Y, U, V,1.7)
xlim([-10,10])
ylim([-10,10])
```

file checking.m

```
% Define the parameters for the pendulum
mu = 1;

% Define the symbols for the variables and parameter
syms u1 u2 real

% Time span for the simulation
tspan = [0 10];

% Initial conditions close to the stable equilibrium point
x0 = [0 1];

% Simulate the nonlinear system
[t_nl, x_nl] = ode45(@(t, x) unlin(t, x, mu), tspan, x0);


% Define the linearized system at the stable equilibrium point
A_stable = [0, 1; -1, 1];
f_linear_stable = @(t, x) A_stable * x;

% Simulate the linearized system
[t_l, x_l] = ode45(f_linear_stable, tspan, x0);

% Plot the results
figure;
hold on
plot(t_nl, x_nl, 'b');
plot(t_l, x_l, 'r--');
hold off
legend('Nonlinear System(1)','Nonlinear System(2)', 'Linearized
    System(1)','Linearized System(2)');
title('Comparison of Nonlinear and Linearized System at Stable
    Equilibrium');
xlabel('Time (s)');
ylabel('x');


% Define the system of equations
f = [u2; mu*(1 - u1.^2).*u2 - u1];

% Compute the Jacobian matrix
J = jacobian(f, [u1, u2]);

% Evaluate the Jacobian matrix at the equilibrium points
J_stable = double(subs(J, [u1, u2], [0, 0]));

% Display the evaluated Jacobian matrices
```

```
disp('Jacobian at stable equilibrium (theta = 0):');
disp(J_stable);

% Check eigenvalues of each Jacobian to determine the type of
    equilibrium
eig_stable = eig(J_stable);

disp('Eigenvalues at stable equilibrium (theta = 0):');
disp(eig_stable);
```

Van der Polov Oscilátor

a) Prepíšte NDS do Substitučného Kanonického Tvaru a Vypočítajte Jeho Rovnovážne Stavy

Rovnice Van der Polovho oscilátora v substitučnom kanonickom tvare sú:

$$\begin{aligned}\dot{u}_1 &= u_2 \\ \dot{u}_2 &= \mu(1 - u_1^2)u_2 - u_1\end{aligned}$$

Na výpočet rovnovážnych stavov položíme obe derivácie na nulu:

$$\begin{aligned}0 &= u_2 \\ 0 &= \mu(1 - u_1^2)u_2 - u_1\end{aligned}$$

Z prvého rovnice plynie, že $u_2 = 0$. Dosadíme to do druhej rovnice:

$$0 = \mu(1 - u_1^2) \cdot 0 - u_1$$

Odtiaľ dostávame, že $u_1 = 0$. Takže rovnovážny stav pre Van der Polov oscilátor je (0, 0).

b) Vykonajte Linearizáciu Nelineárneho Systému

Linearizácia pre Van der Polov oscilátor v rovnovážnom stave (0, 0) pre mu = 1 je:

$$A_{\text{stable}} = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}$$

c) Určte Typy Rovnovážnych Stavov a Posúdte Stabilitu

Rovnovážny stav (0, 0) je nestabilne ohnisko. Pre posúdenie stability môžeme vypočítať vlastné čísla Jakobiánovej matice v tomto bode.

CHR vyzera tak to:

$$lambda^2 - lambda + 1$$

Vyuzili sme na vypocet tuto formulu:

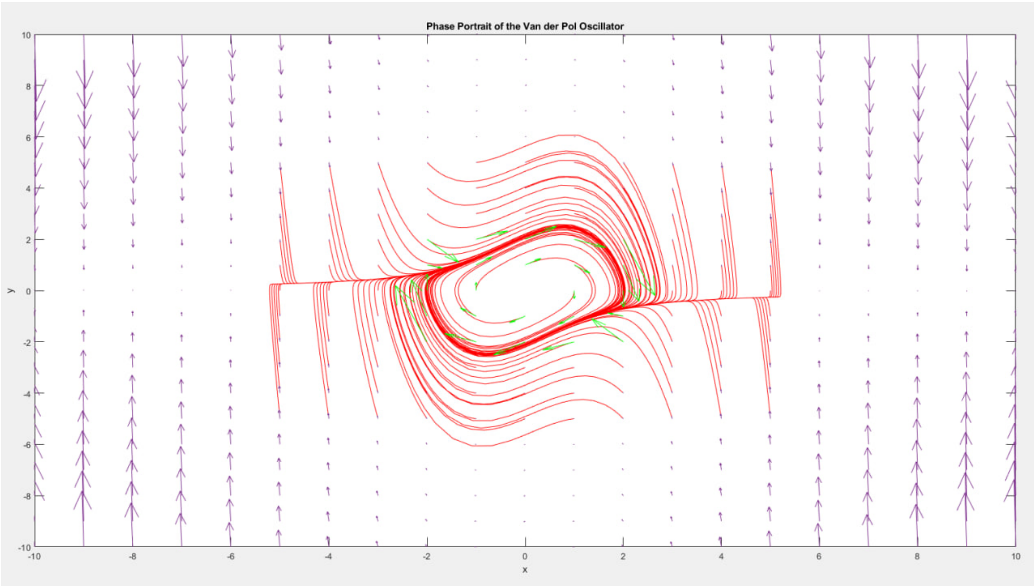
$$det(J - lambda * I)$$

$$\text{Jacobian at stable equilibrium (0, 0):} \quad J_{\text{stable}} = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}$$

Eigenvalues at stable equilibrium (0, 0):

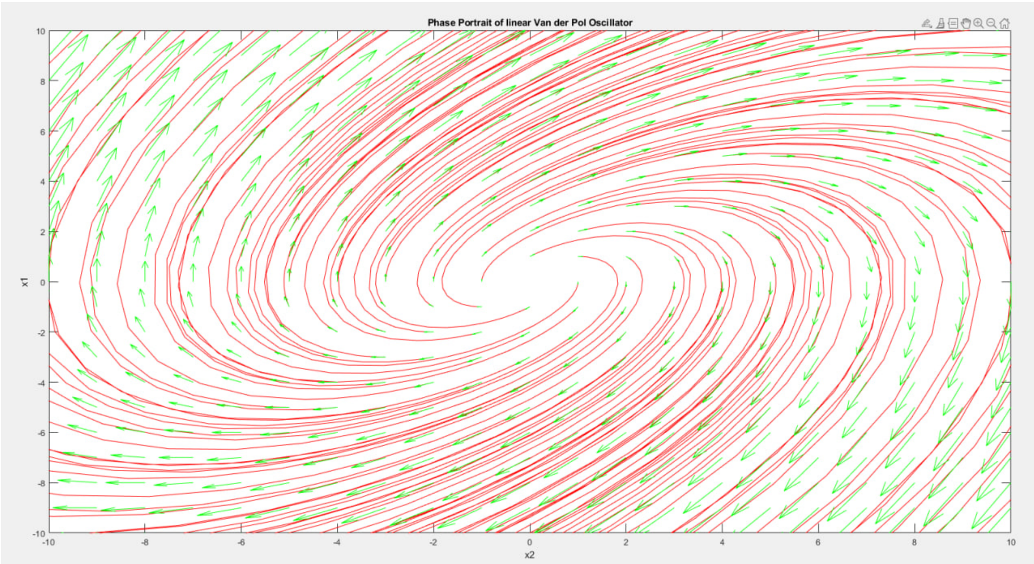
$$\text{Eigenvalues:} \quad \lambda_{1,2} = 0.5 + 0.8660i, \, 0.5 - 0.8660i$$

Na obrázku môžeme vidieť fázový portrét nelineárneho Van der Polovho oscilátora.



Obr. 1: Fázový portrét nelineárneho Van der Polovho oscilátora

Na obrázku môžeme vidieť fázový portrét lineárneho Van der Polovho oscilátora.



Obr. 2: Fázový portrét lineárneho Van der Polovho oscilátora

Moje Riešenie (Matematické Kyvadlo)

```
function dxdt = pendulum(~, x, m, l, B, g, MM)
    theta = x(1);
    omega = x(2);

    dtheta_dt = omega;
    domega_dt = -(B/m).*omega - (g/l)*sin(theta) + MM/(m*l^2);

    dxdt = [dtheta_dt; domega_dt];
end
```

```
% Parameters
l = 2; % Length of the pendulum
g = 9.8; % Acceleration due to gravity
m = 0.3038; % Mass of the pendulum bob
B = 0.5; % Damping coefficient adjusted for damping
MM = 0; % No external driving torque

tspan = [0 10]; % Time span for the simulation

figure; % Create a new figure

% Use a much denser grid for initial conditions to capture more
    details
for i = linspace(-8, 8, 10) % Increased density of initial
    angles
    for j = linspace(-3, 3, 10) % Increased density of initial
        angular velocities
        x0 = [i j]; % Initial conditions [theta, omega]
        [t, x] = ode45(@(t,x) pendulum(t,x,m,l,B,g,MM), tspan,
            x0);
        plot(x(:,1), x(:,2), 'r', 'LineWidth', 0.5); % Red
            trajectories with thinner lines
        hold on;
    end
end

% Set the title and labels for the plot
title('Phase Portrait of the mathematical pendulum');
xlabel('Angle (rad)');
ylabel('Angular velocity (rad/s)');

% Create a denser grid for the vector field
[X, Y] = meshgrid(linspace(-8, 8, 10), linspace(-3, 3, 10));
U = Y;
V = -B/m.*Y - g/l * sin(X) + MM/(m*l^2); % Adjusted for damping

% Create a denser quiver plot for the vector field
quiver(X, Y, U, V, 'b', 'LineWidth', 1);

% Set the limits for the axes
xlim([-8, 8])
ylim([-3, 3]);
```

```matlab
x0 = [7 7];
tspan = [0 10];
l = 2; % Length of the pendulum
g = 9.8; % Acceleration due to gravity
m = 0.3038; % Mass of the pendulum bob
B = 0.5; % Damping coefficient adjusted for damping
MM = 0; % No external driving torque


[t,x] = ode45(@(t,x) pendulum(t,x,m,l,B,g,MM),tspan,x0);

plot(t,x(:,1));
```

```
% Define the parameters for the pendulum
l = 2; % Length of the pendulum
g = 9.8; % Acceleration due to gravity
m = 0.3038; % Mass of the pendulum bob
B = 0.6; % Damping coefficient adjusted for damping
MM = 1; % No external driving torque

tspan = [0 10];

figure;

% Define symbols for the variables
syms x1 x2 real

% Define the system of equations
f = [x2; -(B/m).*x2 - (g/l)*sin(x1) + MM/(m*l^2)];

% Calculate the Jacobian matrix
J = jacobian(f, [x1, x2]);

% Evaluate the Jacobian matrix at the first equilibrium point
    (0, 0)
J_at_equilibrium1 = double(subs(J, [x1, x2], [0, 0]));

% Evaluate the Jacobian matrix at the second equilibrium point (
    pi, 0)
J_at_equilibrium2 = double(subs(J, [x1, x2], [pi, 0]));

% Define the linearized system of differential equations for the
     first equilibrium
f_linear1 = @(t, Y) J_at_equilibrium1 * Y;

% Define the linearized system of differential equations for the
     second equilibrium
f_linear2 = @(t, Y) J_at_equilibrium2 * Y;


% x0 = [7 7];
% [t, x] = ode45(f_linear1, tspan, x0);
% plot(t, x,'r');
%Plot the phase portrait for the first equilibrium point
for i = linspace(-8, 8, 10)
    for j = linspace(-3, 3, 10)
        x0 = [i; j];
        [t, x] = ode45(f_linear1, tspan, x0);
        plot(x(:,1), x(:,2), 'r', 'LineWidth', 0.5);
        hold on;
        [t1, x1] = ode45(f_linear2, tspan, x0);
        plot(x1(:,1), x1(:,2), 'b', 'LineWidth', 0.5);
```

```
        hold on
    end
end


% Set the title and labels for the plot
title('Phase Portrait of Linearized Mathematical Pendulum');
xlabel('x');
ylabel('y');

% Create the vector field for the first equilibrium point
[X, Y] = meshgrid(-10:10, -10:10);
U1 = J_at_equilibrium1(1,1)*X + J_at_equilibrium1(1,2)*Y;
V1 = J_at_equilibrium1(2,1)*X + J_at_equilibrium1(2,2)*Y;

% Create the vector field for the second equilibrium point
U2 = J_at_equilibrium2(1,1)*X + J_at_equilibrium2(1,2)*Y;
V2 = J_at_equilibrium2(2,1)*X + J_at_equilibrium2(2,2)*Y;

% Plot the vector fields
quiver(X, Y, U1, V1, 1.7, 'r');
quiver(X, Y, U2, V2, 1.7, 'b');

% Set the limits for the axes
xlim([-10, 10]);
ylim([-10, 10]);
```

```
% Define the parameters for the pendulum
l = 2; % Length of the pendulum
g = 9.8; % Acceleration due to gravity
m = 0.3038; % Mass of the pendulum bob
B = 0.5; % Damping coefficient
MM = 0; % No external driving torque

% Define the symbols for the variables and parameter
syms theta omega real

% Time span for the simulation
tspan = [0 10];

% Initial conditions close to the stable equilibrium point
x0 = [3; 3]; % Small displacement from theta = 0

% Simulate the nonlinear system
[t_nl, x_nl] = ode45(@(t, x) pendulum(t, x, m, l, B, g, MM),
    tspan, x0);

% Define the linearized system at the stable equilibrium point
A_stable = [0, 1; -g/l, -B/m];
f_linear_stable = @(t, x) A_stable * x;

% Simulate the linearized system
[t_l, x_l] = ode45(f_linear_stable, tspan, x0);

% Plot the results
figure;
plot(t_nl, x_nl, 'b', t_l, x_l, 'r--');
legend('Nonlinear System', 'Linearized System');
title('Comparison of Nonlinear and Linearized System at Stable
    Equilibrium');
xlabel('Time (s)');
ylabel('Theta (rad)');

% Now repeat the process for the unstable equilibrium point
% Initial conditions close to the unstable equilibrium point
x0_unstable = [pi + 3; 3]; % Small displacement from theta = pi

% Simulate the nonlinear system
[t_nl_unstable, x_nl_unstable] = ode45(@(t, x) pendulum(t, x, m,
    l, B, g, MM), tspan, x0_unstable);

% Define the linearized system at the unstable equilibrium point
A_unstable = [0, 1; g/l, -B/m];
f_linear_unstable = @(t, x) A_unstable * x;

% Simulate the linearized system
```

```
[t_l_unstable, x_l_unstable] = ode45(f_linear_unstable, tspan,
    x0_unstable);

% Plot the results for the unstable point
figure;
plot(t_nl_unstable, x_nl_unstable, 'b', t_l_unstable,
    x_l_unstable, 'r--');
legend('Nonlinear System', 'Linearized System');
title('Comparison of Nonlinear and Linearized System at Unstable
    Equilibrium');
xlabel('Time (s)');
ylabel('Theta (rad)');



% Define the system of equations
f = [omega; -(B/m).*omega - (g/l)*sin(theta) + MM/(m*l^2)];

% Compute the Jacobian matrix
J = jacobian(f, [theta, omega]);

% Evaluate the Jacobian matrix at the equilibrium points
J_stable = double(subs(J, [theta, omega], [0, 0]));
J_unstable = double(subs(J, [theta, omega], [pi, 0]));

% Display the evaluated Jacobian matrices
disp('Jacobian at stable equilibrium (theta = 0):');
disp(J_stable);

disp('Jacobian at unstable equilibrium (theta = pi):');
disp(J_unstable);

% Check eigenvalues of each Jacobian to determine the type of
    equilibrium
eig_stable = eig(J_stable);
eig_unstable = eig(J_unstable);

disp('Eigenvalues at stable equilibrium (theta = 0):');
disp(eig_stable);

disp('Eigenvalues at unstable equilibrium (theta = pi):');
disp(eig_unstable);
```

Matematické Kyvadlo

a) Prepíšte NDS do Substitučného Kanonického Tvaru a Vypočítajte Jeho Rovnovážne Stavy

Rovnice matematického kyvadla v substitučnom kanonickom tvare sú:

$$\dot{\theta} = \omega$$
$$\dot{\omega} = -\frac{g}{l}\sin(\theta)$$

Na výpočet rovnovážnych stavov položíme obe derivácie na nulu:

$$0 = \omega$$
$$0 = -\frac{g}{l}\sin(\theta)$$

Z prvej rovnice plynie, že $\omega = 0$. Z druhej rovnice dostávame možné rovnovážne stavy $\theta = 0$ a $\theta = \pi$.

b) Vykonajte Linearizáciu Nelineárneho Systému

Pre rovnovážny stav $\theta = 0$, Jakobián je:

$$J_{\text{stable}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix}$$

Pre rovnovážny stav $\theta = \pi$, Jakobián je:

$$J_{\text{unstable}} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix}$$

c) Určte Typy Rovnovážnych Stavov a Posúdte Stabilitu

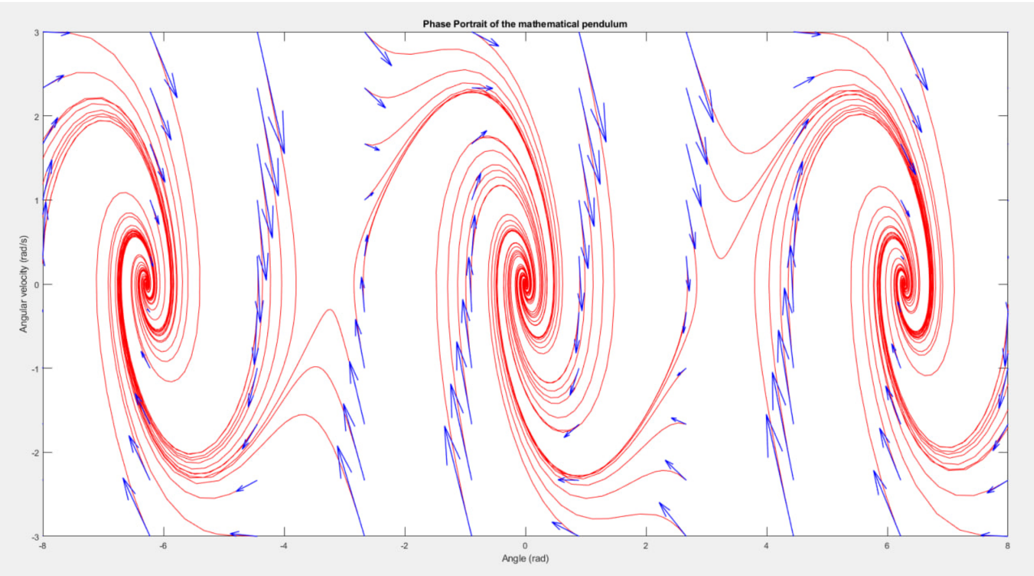
Rovnovážny stav $\theta = 0$ je stabilný ohnisko. Eigenvalues pre tento stav sú:

$$\text{Eigenvalues at stable equilibrium } (\theta = 0): \quad \lambda_{1,2} = -0.8229 \pm 2.0549i$$

Rovnovážny stav $\theta = \pi$ je sedlo. Eigenvalues pre tento stav sú:

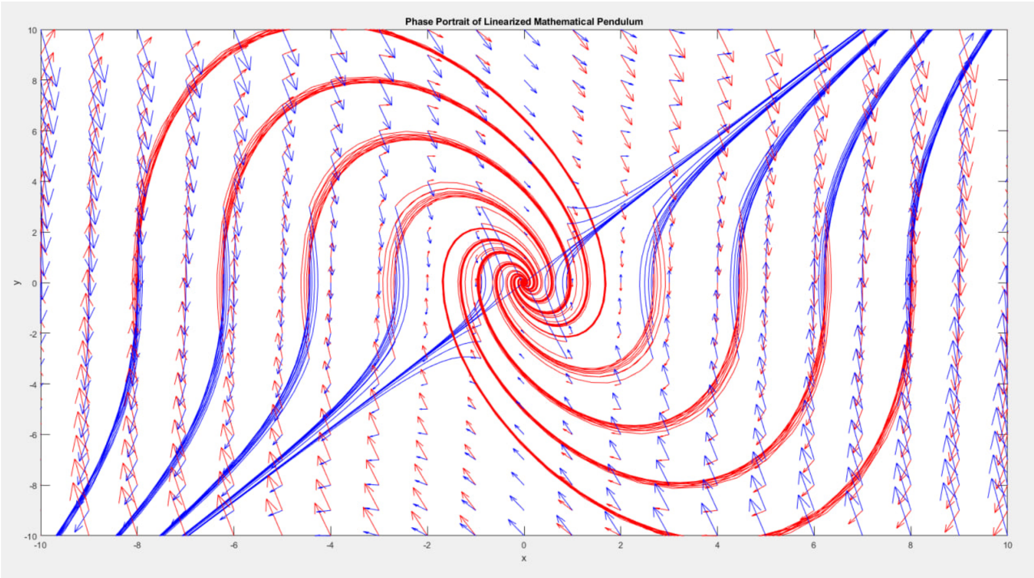
$$\text{Eigenvalues at unstable equilibrium } (\theta = \pi): \quad \lambda_{1,2} = 1.5387, -3.1845$$

Na obrázku môžeme vidieť fázový portrét nelineárneho matematického kyvadla.



Obr. 3: Fázový portrét nelineárneho matematického kyvadla

Na obrázku môžeme vidieť linearizovaného matematického kyvadla.



Obr. 4: Fázový portrét linearizovaného matematického kyvadla