Софиийски университет "Св. Климент Охридски"
Факултет по математика и информатика

Проект

по

Дифиренциални уравнения и приложения

спец. Информатика, 2 курс, зимен семестър,

учебна година 2017/18

Тема И17-3

15 януари 2018 г.

София

Изготвил:

Иво Алексеев Стратев

Фак. номер: 45342

Група: 3

Оценка: ....

Съдържание

Списък на фигурите

Тема (Задание) на проекта

Задача 1. Дадена е задачата на Коши

$$xy' = 5y + 3x, \quad y(2) = 1$$

1. Напишете интегрално уравнение еквивалентно на дадената задача и по метода на Пикар дефинирайте редица от последователни приближения на решението на тази задача.
2. Начертайте с различни цветове графиките на първото, третото и петотоприближение в интервала $[1,\ 3]$. С подходящ числен метод, вграден в Matlab, намерете числено решение на дадената задача и начертайте с черен цвят неговата графика в същия интервал.

Задача 2. Дадена е системата

$$\left|\begin{array}{l} \dot{x} = -2x + 4y \\ \dot{y} = x(x+4) \end{array}\right.$$

1. Намерете нейните равновесни точки. Начертайте векторно поле на тази система в правоъгълник, който съдържа намерените равновесни точки и с негова помощ изследвайте за устойчивост равновесните положения.
2. За решението на задачата на Коши за дадената система с начални условия $x(0) = x_0,\ y(0) = y_0$ направете анимация на движението на точката $(x(t),\ y(t))$ във фазовата равнина за $t \in [0,\ 8]$, като точката $(x_0,\ y_0)$ се въвежда чрез кликване с мишката в избрания правоъгълник.

Решение на Задача 1

Теоретична част

При $x = 0$ получаваме $0 = 5y$ или $y = 0$, тоест получаваме точката $(0,\ 0)$.

Нека $x \neq 0$ тогава можем да разделим на x и така получаваме $y' = \frac{5}{x}y + 3$. Ще сменим името на променливата от x на s. Тогава уравнението добива вида:

$$\frac{\mathrm{d}}{\mathrm{d}s}y = \frac{5}{s}y + 3$$

Интегрираме уравението от двете страни в граници от 2 до x и получаваме

$$\int_2^x \frac{\mathrm{d}}{\mathrm{d}s}y(s)\ \mathrm{d}s = \int_2^x \frac{5}{s}y(s) + 3\ \mathrm{d}s \implies$$

$$\int_2^x \mathrm{d}y = \int_2^x \frac{5}{s}y(s) + 3\ \mathrm{d}s \implies$$

$$y(x) - y(2) = \int_2^x \frac{5}{s}y(s) + 3\ \mathrm{d}s \implies$$

$$y(x) = y(2) + \int_2^x \frac{5}{s}y(s) + 3\ \mathrm{d}s \implies$$

$$y(x) = 1 + \int_2^x \frac{5}{s}y(s) + 3\ \mathrm{d}s$$

Полученото интегрално уравнение е еквивалентно на дадената задача на Коши за $x \neq 0$. Така сведохме задачата на Коши

$$xy' = 5y + 3x, \quad y(2) = 1$$

до следната еквивалентна на нея задача

$$y(x) = 1 + \int_2^x \frac{5}{s}y(s) + 3\ \mathrm{d}s, \quad y(0) = 0.$$

Тоест от дифиренциално уравнение преминахме към интегрално.

Дефинираме следата редица от последователни приближения на решението на тази задача по метода на Пикар

$$y_0(x) \equiv y(2) = 1$$

$$\forall n \in \mathbb{N} \quad y_{n+1}(x) = 1 + \int_2^x \frac{5}{s}y_n(s) + 3\ \mathrm{d}s$$

Символно решение на дадената задача на Коши

Първо ще намерим общия вид на решенията на даденото диференциялно уравнение. Както вече установихме при $x = 0$ получаваме точката $(0,\ 0)$. При $x \neq 0$

получихме уравнението $y' = \frac{5}{x}y + 3$, което е линейно диференциално уравнение и неговото общо решение е:

$$y = e^{\int \frac{5}{x}\,\mathrm{d}x}\left(C + \int 3e^{-\int \frac{5}{x}\,\mathrm{d}x}\,\mathrm{d}x\right)$$

Използвайки факта, че ние търсим решение на дадената задача дефинирано в точката 2, получаваме

$$y = x^5\left(C + 3\int x^{-5}\,\mathrm{d}x\right) \implies$$

$$y = x^5(C - \frac{3}{4}x^{-4}) = Cx^5 - \frac{3}{4}x$$

Използвайки началното условие намираме стойността на константа C

$$y(2) = C.2^5 - \frac{3}{4}2 = C.2^5 - \frac{3}{2} = 1 \implies C = \frac{5}{2^6}$$

Намерихме аналитично решение да дадената задача на Коши в квадратури

$$y = \frac{5}{2^6}x^5 - \frac{3}{4}x$$

Матлаб код

```
% plots the first, third and fifth approximations by using
    Picard method
% for the Cauchy problem:
% xdy/dx = 5y + 3x, y(2) = 1
% in blue, green, red color for x in [1; 3]
% than solves numerically the given problem and plots the
    solution in black
function tema3_zad1
    % x for all plots is in the interval [1; 3]
    xmin = 1;
    xmax = 3;

    % from many runs and experiments for the given equation
        y in [-8; 17]
    % fits perfectly from the plots
    ymin = -8;
    ymax = 17;

    % the starting condition y(2) = 1 = y0 = y(x0)
    x0 = 2;
    y0 = 1;

    % draw x-axis for x in [xmin; xmax] and y-axis for y in
        [ymin; ymax]
    axis([xmin xmax ymin ymax]);
    % ensure all plots remain
    hold on;
```

```
    % draw grid
    grid on;
    % label the x-axis
    xlabel('x');
    % label the y-axis
    ylabel('y');

    % the number of intervals used for spliting the inteval for x-es left
    % to x0 and right to x0 (since both ode45 and the recursive integral
    % from Picard method starts from x0)
    count = 100;
    % intervals for x-es left to x0
    xleft = x0:((xmin - x0) / count):xmin;
    % intervals for x-es right to x0
    xright = x0:((xmax - x0) / count):xmax;

    % cumtrapz is used for approximating the recursive integral from Picard
    % method wich takes interval splitted at sub-intervals represented by a
    % single vector and returns as an output vector with partial sums
    % this is why we will add y0 to each element of the partial sums vector
    % creating vector with all 1 multiplied by y0 is the same as creating
    % vector of all y0 with length equal to the length of points left to x0
    % and right to x0
    y0left = y0 * ones(1, length(xleft));
    y0right = y0 * ones(1, length(xright));

    % define variables to store the previous approximation in Picard method
    yleftPrev = y0left;
    yrightPrev = y0right;

    % the right hand side of the given equation
    % xdy/dx = 5y + 3x
    % rewritten as dy/dx = 5(y/x) + 3 when x /= 0
    % (0 is not in our interval so no point is missed)
    f = @(x, y) (5 * (y / x) + 3);

    % colors for the first (blue), third (green) and fifth (red)
    % approximations
    colors = ['b' 'g' 'r'];
    % starting with the color for first approximation to plot
    c = 1;
```

```
    % preallocate memory for graph handles
    g = gobjects(1, 4);

    % since we want to plot only the first, third and fifth approximations
    % we can calculate only the first five approximations
    for k=1:5
        % approximate the k'th Picard approximation
        % for the given equation for x-es left to x0
        yleftk = y0left + cumtrapz(xleft, arrayfun(f, xleft, yleftPrev));
        % approximate the k'th Picard approximation
        % for the given equation for x-es left to x0
        yrightk = y0right + cumtrapz(xright, arrayfun(f, xright, yrightPrev));

        % if k is odd than k is either 1, 3 or 5 and that approximation
        % must be plotted with the proper color
        if mod(k, 2) == 1
            % plot-ing both the points left to x0 and right to x0 in the
            % same color
            p = plot(xleft, yleftk, colors(c), xright, yrightk, colors(c));
            % make line thicker
            p(1).LineWidth = 2;
            % make line thicker
            p(2).LineWidth = 2;
            % save the graph handle for the graph of x-es left of x0
            g(c) = p(1);
            % switch color for the next plot
            c = c + 1;
        end

        % set the prev y-es to the current for the next iteration
        yleftPrev = yleftk;
        yrightPrev = yrightk;
    end

    % solve the given equation numerically with ode45
    % (the given equation is nonstiff so it's OK)
    % ode45 solves dy/dx = f(x, y) from x0 to xmax and requires the
    % required number of initial conditions (in this case only one)
    % at the x0 this is why a split to left x-es and right x-es is
    % required
```

```
    % solve numerically the given equation for x-es left to x0
    [xleft, yleft] = ode45(f, [x0 xmin], y0);
    % solve numerically the given equation for x-es right to x0
    [xright, yright] = ode45(f, [x0 xmax], y0);
    % plot the numeric solution by ploting both the one
    % for x-es left to x0 and right to x0 in black
    p =  plot(xleft, yleft, 'k', xright, yright, 'k');
    % make line thicker
    p(1).LineWidth = 2;
    % make line thicker
    p(2).LineWidth = 2;
    % save the graph handle for the graph of x-es left of x0
    g(c) = p(1);
    % draw legend for the four graphs
    % match colors from the graph handles
    % set legend location to the down-right corner
    % and increase the font size
    legend(g, {'y1', 'y3', 'y5', 'y'}, 'Location', 'southeast', 'FontSize', 14);
    % ading text for each plot
    text(2.7, 3.85, 'y1', 'Color', 'blue', 'FontSize', 18);
    text(2.75, 8.55, 'y3', 'Color', 'green', 'FontSize', 18);
    text(2.95, 14.52, 'y5', 'Color', 'red', 'FontSize', 18);
    text(2.85, 14, 'y', 'Color', 'black', 'FontSize', 18);
end
```

Коментар към използваните вградени в Matlab функции

За численето решение на задачата и за пресмятането на интегралите от последователните приближение са използвани вградени в Matlab функции: ***ode45*** и ***cumtrapz***, които числено пресмятат подадените им дифиренциално уравнение и определен интеграл започвайки от точката на началното условие. Понеже тази точка за дадената задача е 2, която е средата на дадения интервал $[1,\ 3]$ то, това налага последователно да се пресметнат числено решенията в интервалите $[1,\ 2]$ и $[2,\ 3]$ и след това двете решения да бъдат долепени едно за друго.

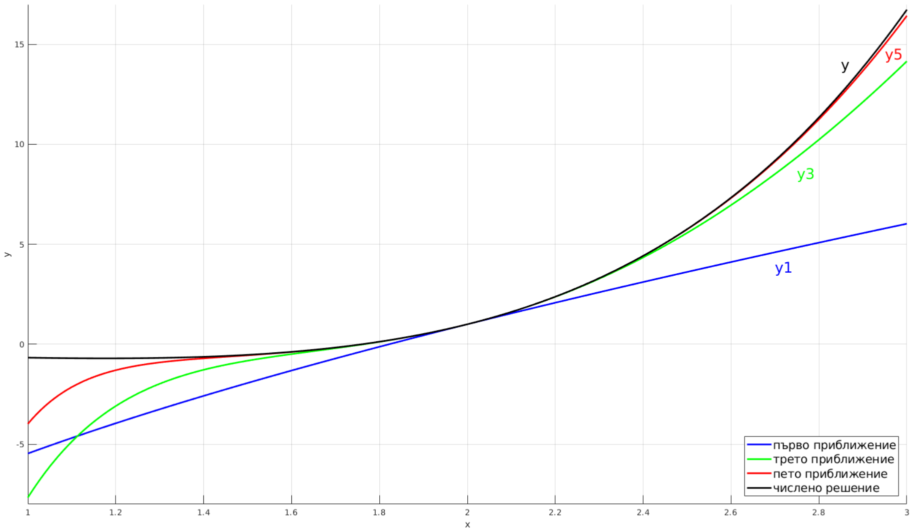
Аналогично от математическа гледна точка графиката на решението в интервала $[1,\ 3]$ също може да бъде разделена на две графики. Тоест

$$
\begin{aligned}
G(x,y)|_{x\in[1,\ 3]} &= \{(x,\ y(x)) \mid x \in [1,\ 3]\} = \\
= \{(x,\ y(x)) \mid x \in [1,\ 2]\} &\cup \{(x,\ y(x)) \mid x \in [2,\ 3]\} = \\
&= G(x,y)|_{x\in[1,\ 2]} \cup G(x,y)|_{x\in[2,\ 3]}
\end{aligned}
$$

Графики

Коментари към получените с Matlab резултати

На графиката са начертани графиките на първото приближение y_1 със син цвят, третото приближение y_3 със зелен цвят и с петото приближение y_5 с червен цвят, получени по метода на Пикар в интервала $[1,\ 3]$. С черен цвят е начертана



Фигура 1: Графки на първото, третото и петото приближение по метода на Пикар и графика на численото решение в интервала $[1,\ 3]$

графиката на полученото числено решение на дадената задача на Коши също в интервала $[1,\ 3]$.

Решение на Задача 2

Теоретична част

Търсим равновесните точки на системата

$$\left|\begin{array}{l}\dot{x} = -2x + 4y\\ \dot{y} = x(x+4).\end{array}\right.$$

Това са точките за нулевите вектори от векторното поле, породено от системата.

$$\left|\begin{array}{l}\dot{x} = 0\\ \dot{y} = 0\end{array}\right. \implies \left|\begin{array}{l}-2x + 4y = 0\\ x(x+4) = 0\end{array}\right. \implies \left|\begin{array}{l}y = \frac{x}{2}\\ x(x+4) = 0\end{array}\right. \implies \{(0,\ 0),\ (-4,\ -2)\}$$

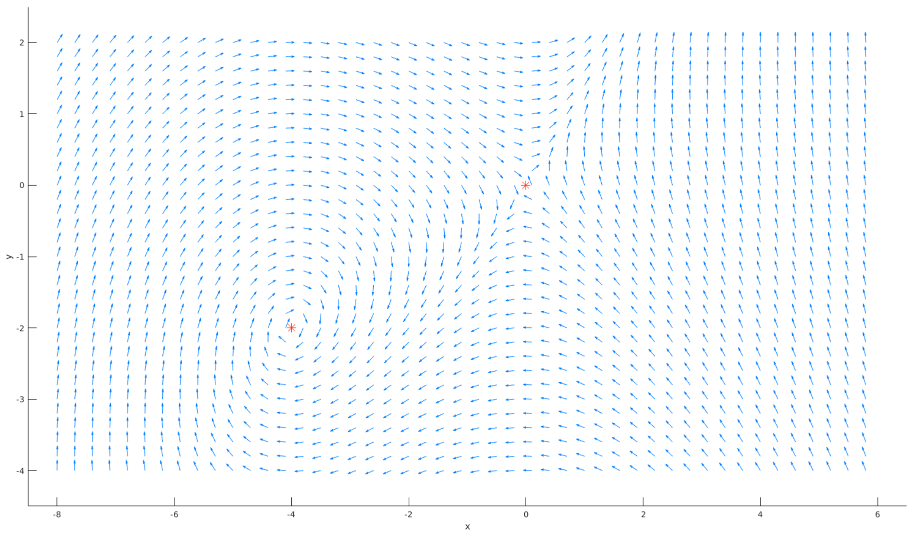
С помощта на Matlab ще начертаем векторното поле на дадената система и чрез него ще изследваме за устойчивост намерените равновесни точки.

Код на Matlab изчертващ векторното поле на системата

```
xmin = -8;
xmax = 6;

ymin = -4;
ymax = 2;

x = xmin:0.3:xmax;
y = ymin:0.2:ymax;

offset = 0.5;
```

```
axis([xmin - offset xmax + offset ymin - offset ymax +
    offset]);
hold on;
xlabel('x');
ylabel('y');

plot(0, 0, 'Marker', '*', 'MarkerSize', 11, 'Color', [1,
    0.2, 0.1]);
plot(-4, -2, 'Marker', '*', 'MarkerSize', 11, 'Color', [1,
    0.2, 0.1]);

f1 = @(x, y) (-2 * x + 4 * y);
f2 = @(x, y) (x.^2 + 4 * x);

[X, Y] = meshgrid(x, y);
P = f1(X, Y);
Q = f2(X, Y);
Length = sqrt(P.^2 + Q.^2);
quiver(X, Y, P ./ Length, Q ./ Length, 0.42, 'Color', [0,
    0.5, 1]);
```

Графика на изчертаното векторно поле на системата



Фигура 2: Графкика на векторното поле на дадената система изчертано в подходящ правоъгълник съдържащ равновесните точки на системата

Коментари към изчератаното с Matlab векторно поле

Както лесно можем да забележим в околност на точката $(0,\ 0)$ фазовия портрет на системата има вид на седло. Тоест точката $(0,\ 0)$ е неустойчива. в околност на точката $(-4,\ -2)$ фазовия портрет системата има вид на устойчив фокус и следователно точката $(-4,\ -2)$ е асимптотически устойчива.

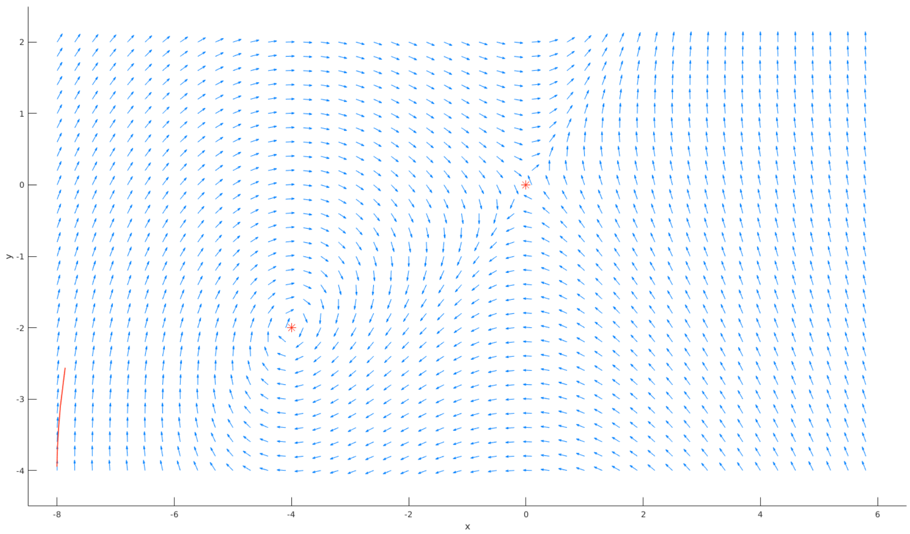
Матлаб код

```
% plots the fixed points of the system of ODEs:
% dx/dt = -2x + 4y, dy/dt = x^2 + 4x
% plays animation of point moving in the phase plane
% at the curve starting at (x(0), y(0)) = (x0, y0),
% which are inputed via mouse click
% the curve ends at (x(8), y(8)) (t is in [0, 8])
% the curve is actually the solution of the given system
% with inital conditions (x(0), y(0)) = (x0, y0)
% the system is time independent (Autonomous system) and it's solved
% numerically with ode15s
function tema3_zad2
    % maximum t, t is in [0, 8];
    tmax = 8;

    % fixed points are (0, 0) and (-4, -2)
    % D = [-8, 6] x [-4, 2] seems like the perfect rectangle
    % x in [-8, 6]
    xmin = -8;
    xmax = 6;
    % y in [-4, 2]
    ymin = -4;
    ymax = 2;

    % x in [xmin, xmax] with step 0.3
    x = xmin:0.3:xmax;
    % y in [ymin, ymax] with step 0.2
    y = ymin:0.2:ymax;

    % axis offset for beauty
    offset = 0.5;
    % draw x-axis and y-axis
    axis([xmin - offset xmax + offset ymin - offset ymax + offset]);
    % ensure all plots remain
    hold on;
    % label the x-axis
    xlabel('x');
    % label the y-axis
    ylabel('y');

    % plot the fixed point (0, 0) with * (Asterisk) as marker in red
    plot(0, 0, 'Marker', '*', 'MarkerSize', 11, 'Color', [1, 0.2, 0.1]);
    % plot the fixed point (-4, -2) with * (Asterisk) as marker in red
    plot(-4, -2, 'Marker', '*', 'MarkerSize', 11, 'Color', [1, 0.2, 0.1]);

    % dx/dt = f1(x, y) = -2x + 4y
    f1 = @(x, y) (-2 * x + 4 * y);
```

```
    % dy/dt = f2(x, y) = x^2 + 4x
    f2 = @(x, y) (x.^2 + 4 * x);

    % Create 2-D grid coordinates
    % with x-coordinates defined by the vector x
    % and y-coordinates defined by the vector y.
    [X, Y] = meshgrid(x, y);
    % calcualte the velocity vectors x ends (or dx/dt for each X)
    P = f1(X, Y);
    % calcualte the velocity vectors y ends (or dy/dt for each Y)
    Q = f2(X, Y);
    % calculate each velocity vector length (threat them as radius vectors)
    Length = sqrt(P.^2 + Q.^2);
    % plot each normalizate velocity vector using quiver and scale them
    % each radius vector (P(k), Q(k)) is transition at (X(k), Y(k))
    quiver(X, Y, P ./ Length, Q ./ Length, 0.42, 'Color', [0, 0.5, 1]);

    % clear varaibles that are no longer needed (only array ones)
    clear x;
    clear y;
    clear X;
    clear Y;
    clear P;
    clear Q;
    clear Length;

    % get initial conditions via mouse click
    [x0, y0] = ginput(1);
    % solve the given system of ODEs wich is time independent (Autonomous
    % system) wich turns out to be stiff so ode15s solver was choosen
    % dsolve dose not even solve the system ...
    % solving for t in [0, tmax] with initial conditions
    % x(0) = x0 and y(0) = y0
    [T, S] = ode15s(@(t, s) [f1(s(1), s(2)); f2(s(1), s(2))], [0, tmax], [x0; y0]);
    % save the length of solution vectors lengths wich are the same as T
    timeLength = length(T);
    % release memory for T
    clear T;

    for k=1:timeLength
```

```
        % plot each point of the point movement curve starting at (x0, y0)
        % and ending at (x(tmax), y(tmax)) with (x, y) beeing the solution
        % also style the plot with color and thicker line width
        plot(S(1:k,1), S(1:k,2), 'Color', [1, 0.2, 0.1], 'LineWidth', 1.2);
        % capture the current axes as a movie frame
        getframe;
    end
end
```

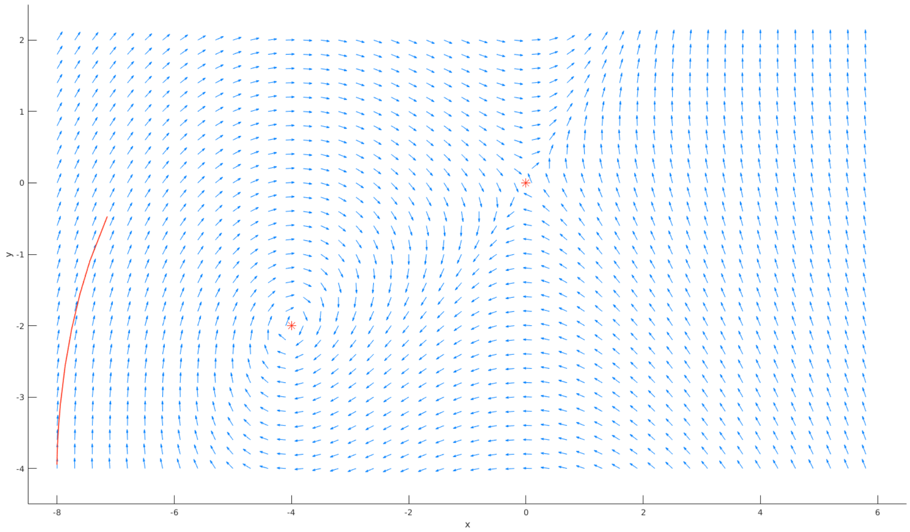
Графики



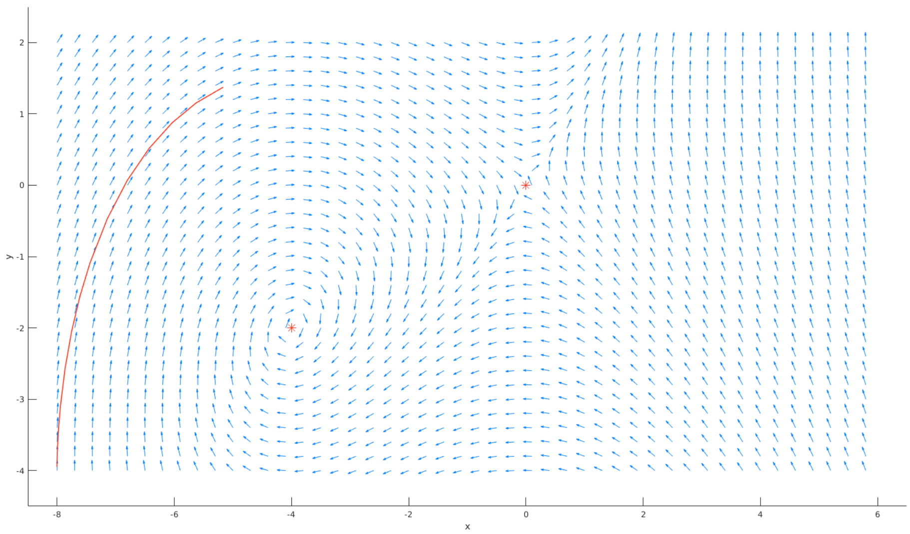
Фигура 3: Кадър 1 от анимация на движение на точка във фазовата равнина на системата

Коментари към получените с Matlab резултати

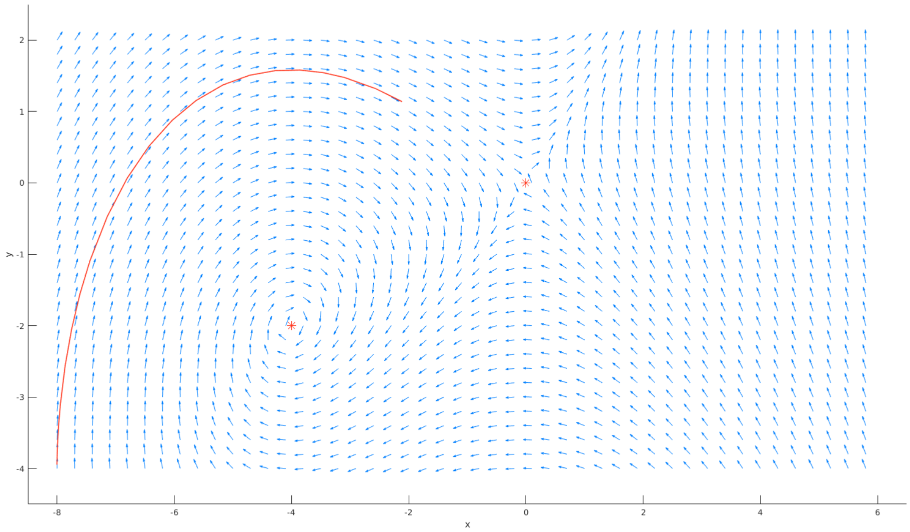
Фигури 3-11 показват кадри от анимацията на движението на точка във фазовата равнина. Фигури 12-14 показват различни фазови криви. Фигура 12 демонстрира асимптотическата устойчивост на точката $(-4,\ -2)$. Фигури 13 и 14 демонстират неустойчивостта на точката $(0,\ 0)$.



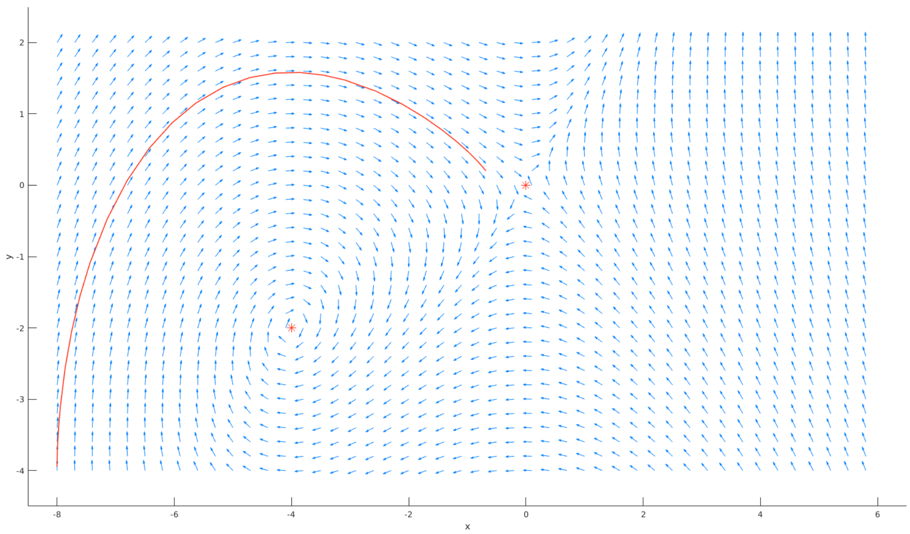
Фигура 4: Кадър 2 от анимация на движение на точка във фазовата равнина на системата



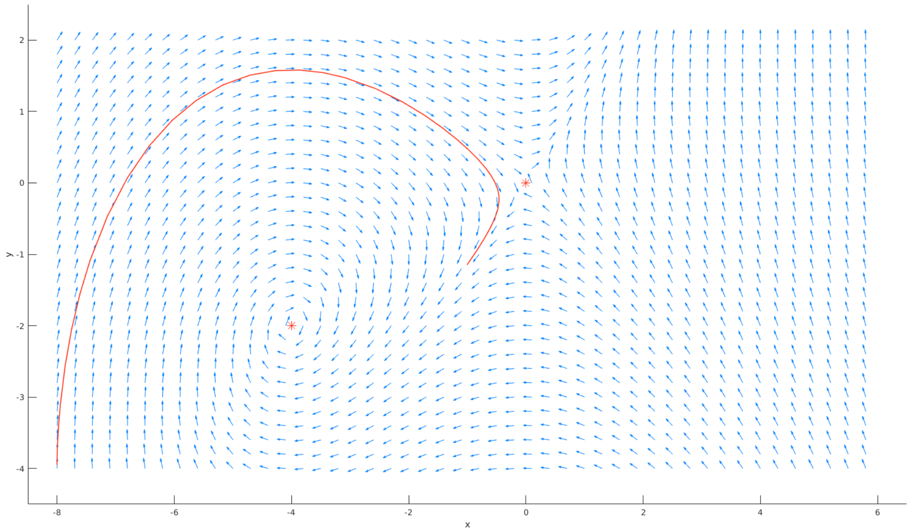
Фигура 5: Кадър 3 от анимация на движение на точка във фазовата равнина на системата



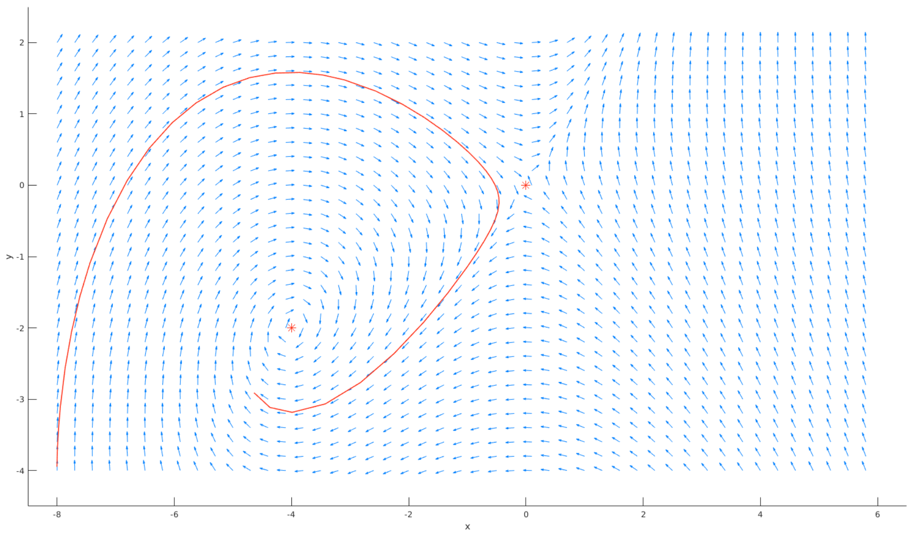
Фигура 6: Кадър 4 от анимация на движение на точка във фазовата равнина на системата



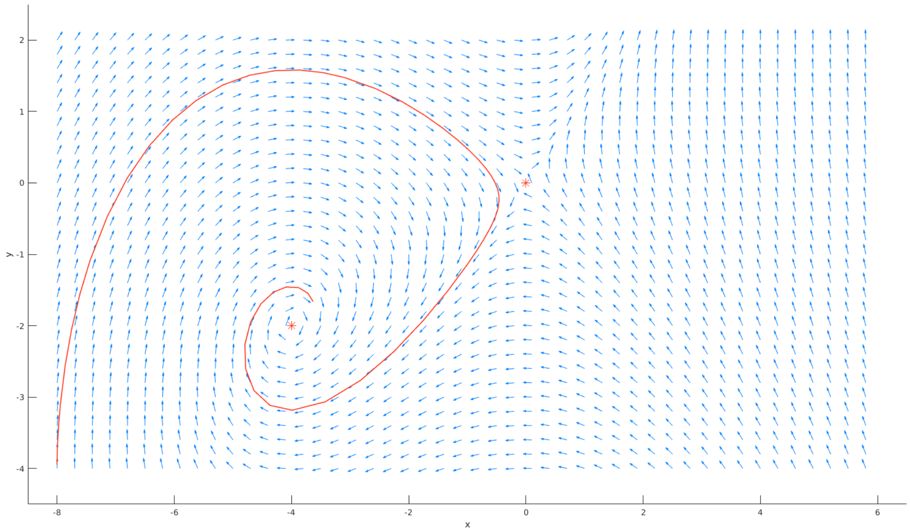
Фигура 7: Кадър 5 от анимация на движение на точка във фазовата равнина на системата



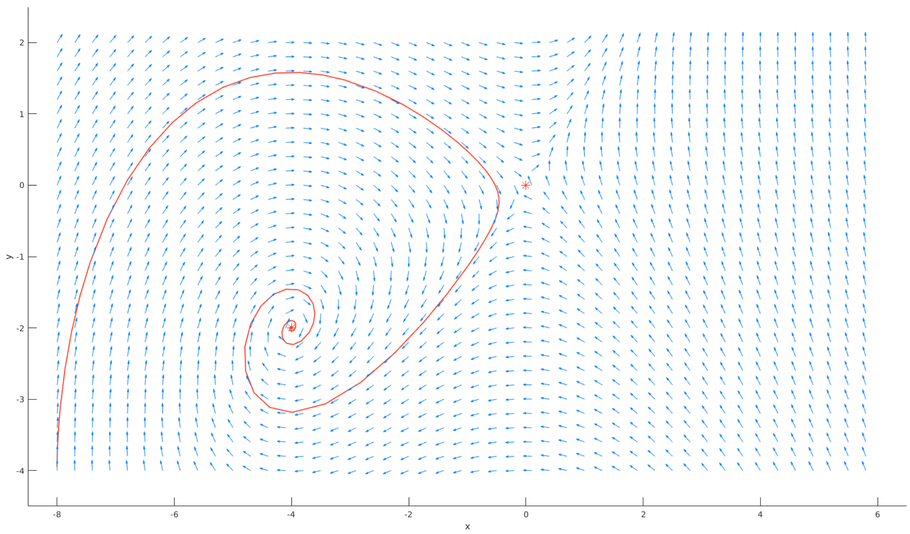
Фигура 8: Кадър 6 от анимация на движение на точка във фазовата равнина на системата



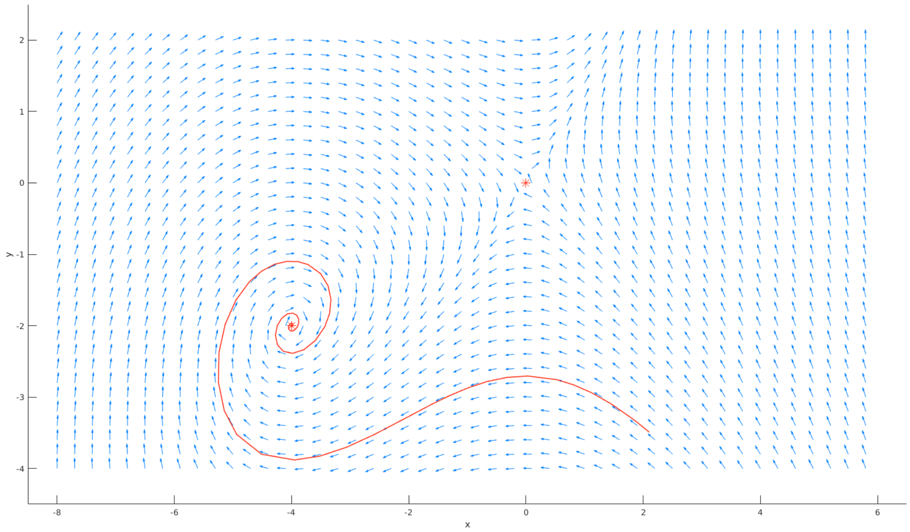
Фигура 9: Кадър 7 от анимация на движение на точка във фазовата равнина на системата



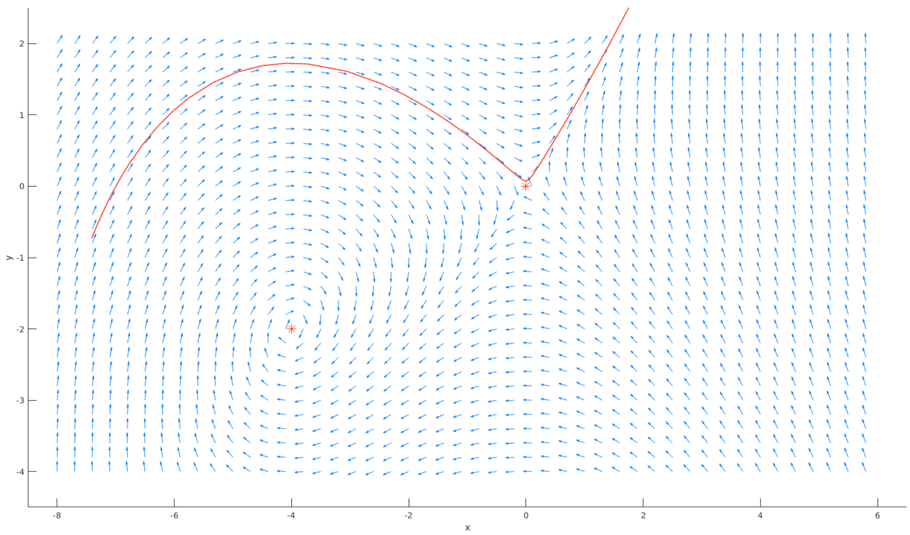
Фигура 10: Кадър 8 от анимация на движение на точка във фазовата равнина на системата



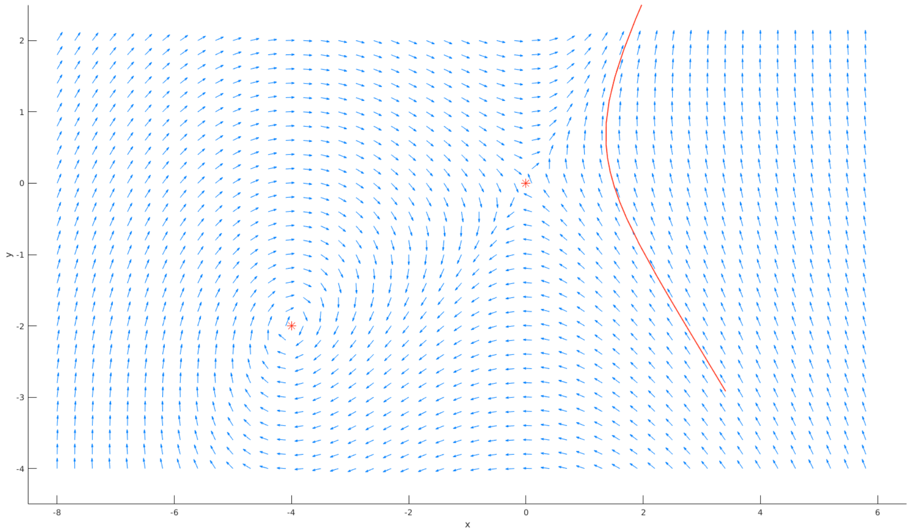
Фигура 11: Кадър 9 от анимация на движение на точка във фазовата равнина на системата



Фигура 12: Фазова крива демонстираща асим. устойчивост на точката $(-4,\ -2)$



Фигура 13: Една фазова крива демонстираща неустойчивостта на точката $(0,\ 0)$



Фигура 14: Друга фазова крива демонстираща неустойчивостта на точката (0, 0)