

# Семинар 06

Ивайло Андреев

27 март 2025

## 1 Увод към упражнението

### 1.1 Коплекситетът от непрекъснатостта

Ще започнем днешното упражнението с отворен въпрос.  
Кое е най-основополагащото нещо от ДИС-а?

Кое е най-важното нещо от ДИС-а с една дума?

Разбира се, вероятно верният отговор е нещо доста основно като например аритметични операции или понятието за функция. По-скоро въпросът се отнася за тези неща, които се учат от ДИС-а. Примерни отговори: интеграли, производни, порядъци, сходимост, ред, редица, криволинеен трапец и други.

Отговорът, който аз ще дам, е *граница*.

Защо?

1. Производна е по дефиниция граница.
2. Неопределеният интеграл може да се разглежда като обратна операция на диференцирането (което казахме, че по дефиниция е просто граница), а определеният интеграл по дефиниция е граница на Риманови суми.
3. Търсим сходимост чрез граници.
4. Редици и редове са интересни заради граници.

Границите обаче имат един проблем - не са тривиални.

## 1.2 Сведения от числения анализ

Нека разгледаме стандартната дефиниция за производна

$$f'(x) = \lim_{s \rightarrow 0} \frac{f(x+s) - f(x)}{s}$$

Имаме неопределеност от вида  $\frac{0}{0}$  в общия случай и съответно диференцирането изисква доста повече отколкото просто елементарни математически операции. Численият анализ ни казва, че можем да приближим точната стойност на производната (и в това число на всички нетривиални математически операции) с произволно малка грешка.

Нека  $h$  е произволно малко

Тогава можем да приближим производната по следния начин

$$f'(x) = \lim_{s \rightarrow 0} \frac{f(x+s) - f(x)}{s} \approx \frac{f(x+h) - f(x)}{h}$$

За сега ще кажем само, че колкото по-малко е  $h$ , толкова по-близо сме до точната стойност и съответно толкова е по-малка грешката от приближението.

Имаме различни еквивалентни дефиниции за производна и когато използваме същата логика за приближение получаваме аналогично, но не еквивалентни приближения. Ето 3 подхода за числено диференциране:

1. Разлика напред:  $f'(x) = \lim_{s \rightarrow 0} \frac{f(x+s) - f(x)}{s} \approx \frac{f(x+h) - f(x)}{h}$
2. Разлика назад:  $f'(x) = \lim_{s \rightarrow 0} \frac{f(x) - f(x-s)}{s} \approx \frac{f(x) - f(x-h)}{h}$
3. Централна разлика:  $f'(x) = \lim_{s \rightarrow 0} \frac{f(x+s) - f(x-s)}{2s} \approx \frac{f(x+h) - f(x-h)}{2h}$

Нека пак припомним, че всички тези числени приближение са точно това - **приближения**. Доста съществен въпрос е каква е връзката между това колко малко  $h$  използваме и каква грешка очакваме от приближението. По-конкретен въпрос е: "Ако намалим  $h$  с дадена единица количество, то колко ще намалее грешката от приближението?". Така можем да говорим в порядъци.

Набързо само ще споменем грешката от вече споменатите числени диференцирания

1. Разлика напред: грешка от порядъка на  $O(h)$
2. Разлика назад: грешка от порядъка на  $O(h)$
3. Централна разлика: грешка от порядъка на  $O(h^2)$

Повече подробности около грешки от числени методи се коментират в избираеми курсове за числени методи, включително "Теория на апроксимациите" "Числени методи за диференциални уравнения част първа" "Числени методи за диференциални уравнения част втора" и други.

### 1.3 Възли и стойности

Важно нещо, което трябва да отбележим е, че когато използваме числени методи за намиране на приближение на функция (например за намиране на интерполационен полином или намиране на решение на диференциално уравнение), то те НЕ ни дават (приближена) функция, на която да подадем произволни аргументи, а ни дават приближени стойности на решението в точни определени точки, които ще наричаме възли.

Затова въвеждаме означенията:

1. Възли - точки по абсцисата (обикновено равноотдалечени), в които точки ще намираме приближение на решението.
2. Стойности - приближената стойност на решението във възлите.

Най-често възлите са равноотдалечени на разстояние от  $h$  единици. Тоест когато намаляваме стъпката  $h$  с цел да намалим грешката на приближенията, най-често ще се наложи да увеличим и броя възли и съответно броя стойности. Това ни подтиква да използваме софтуер за реализиране на числени методи. В рамките на този курс ще използваме MATLAB/Octave за тези цели.

### 1.4 Octave за числата на Фибоначи

Сега ще видим как можем да смятаме числата на Фибоначи с Octave. Ще направим това, защото числените методи за диференциални уравнения, които ще прилагаме после, са идейно много подобни.

Задачата е да намерим първите  $N$  числа на Фибоначи, като напишем походящ на Octave.

Няма да го направим рекурсивно, а итеративно. По-конкретно:

1. инициализираме масив *nodes* с първите 20 индекса, за който искаме да намерим числата на Фибоначи (т.е. от 0 до  $N - 1$ )
2. инициализираме масив *fibonacci-values* с празни елементи, всеки съответстващ на числото на Фибоначи за съответния индекс
3. запълваме  $F = \text{fibonacci-values}$  по следната математическа схема:
  - а)  $F_0 = 0$
  - б)  $F_1 = 1$
  - в)  $F_{i+1} = F_i + F_{i-1} \quad \forall i \geq 2$

```
N = 20;  
  
nodes = 0:1:N-1;  
  
fibonacci_values = zeros(size(nodes));
```

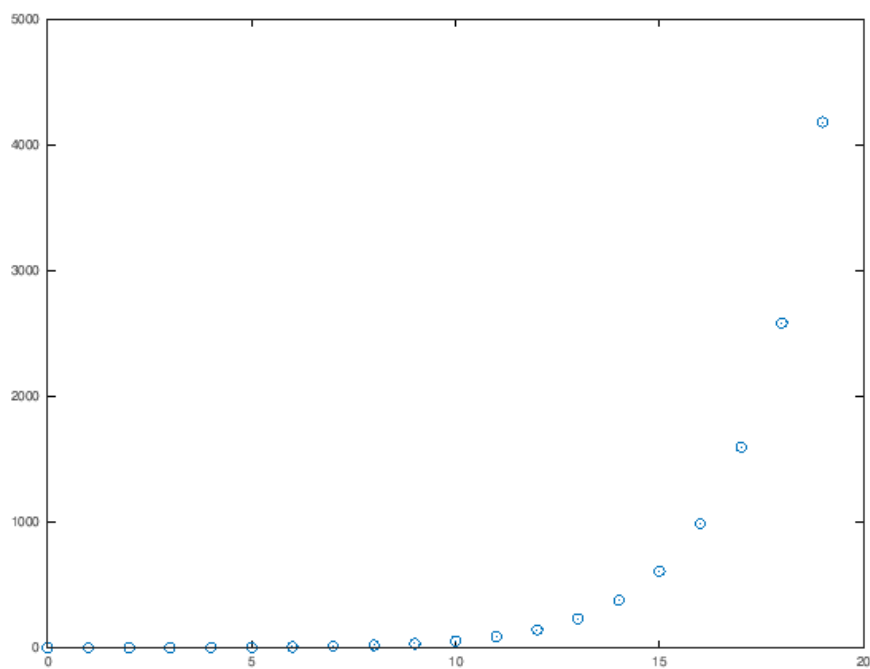
```

fibonacci_values(1) = 0;
fibonacci_values(2) = 1;

for i = 2:length(nodes)-1
    fibonacci_values(i + 1) = fibonacci_values(i) + fibonacci_values(i - 1);
end

plot(nodes, fibonacci_values, "o")

```



Фигура 1: Графика от кода

Общо взето това е нивото на код, нужно за имплементиране на числените методи за диференциални уравнения.

## 2 Метод на Ойлер

### 2.1 Общ случай

Нека имаме следната задача на Коши

$$\begin{cases} y' = f(x, y) \\ y(x_0) = z_0 \end{cases}$$

Ще я решим числено със следната схема.

Нека  $x \in [x_0, A]$ ,  $x_0 < A \in \mathbb{R}$

Дефинираме следната мрежа:  $x_i = x_0 + ih$ ,  $i = 0, \dots, N-1$ , където  $N \in \mathbb{N}$  и  $h = \frac{A - x_0}{N}$

$N$  е броят възли и колкото по-голямо е  $N$ , толкова по-малко е  $h$ .

Предполагаме, че  $y(x_i) \approx y_i$ , тоест, че нашето приближение е приблизително равно на истинската стойност на решението в същата точка (с някаква грешка).

Численото решение, което решава задачата се задава със следната диференчна схема:

$$y_0 = z_0$$

$$y_{i+1} = y_i + hf(x_i, y_i)$$

## 2.2 Извеждане на диференчната схема

Ще я изведем с най-елементарните методи от числения анализ - числено приближение на производна. Хубаво е да отбележим, че има и подходяща геометрична интерпретация, която я има разписана в репото, но не и в този документ.

Имаме следното

$$\begin{aligned} f(x_i, y_i) &= y'(x_i) \\ &\stackrel{\text{def}}{=} \lim_{s \rightarrow 0} \frac{y(x_i + s) - y(x_i)}{s} \\ &\approx \frac{y(x_i + h) - y(x_i)}{h} \\ &= \frac{y(x_{i+1}) - y(x_i)}{h} \\ &= \frac{y_{i+1} - y_i}{h} \end{aligned}$$

Така

$$y_{i+1} = y_i + hf(x_i, y_i)$$

## 2.3 Конкретен пример

Дадена е следната задача на Коши

$$\begin{cases} y' = x^2 - y^2 \\ y(0) = 6 \end{cases}$$

Да се намери приближение на решението с метода на Ойлер в интервала  $x \in [0, 5]$  със стъпки  $h = 0.1$ ,  $h = 0.02$  и  $h = 0.001$ . Да се начертаят графиките на трите приближенията в една и съща координатна система.

## 2.4 Решение на код

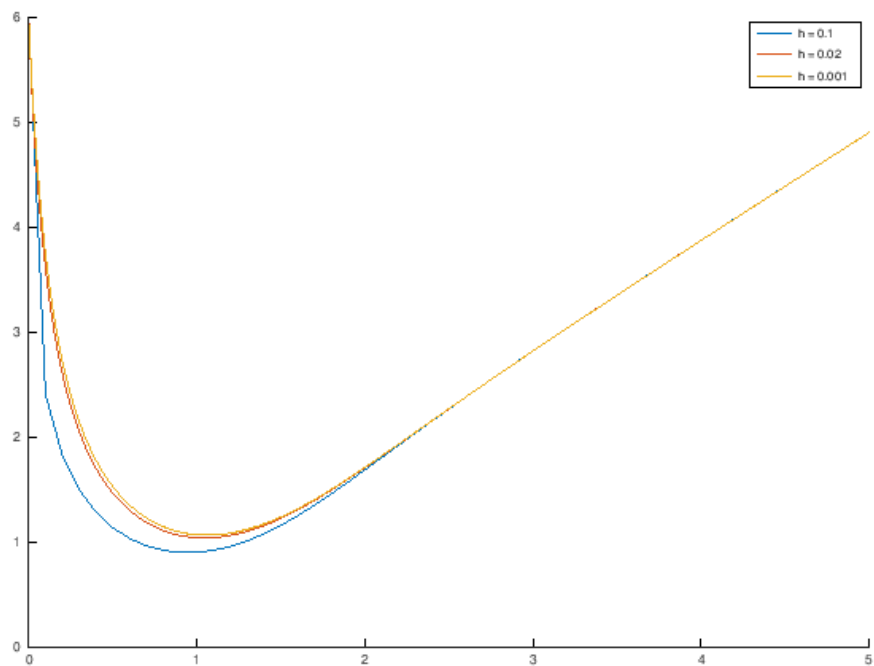
```
warning('off','all');
% pkg install -forge symbolic
pkg load symbolic;
graphics_toolkit("gnuplot");

function [nodes, values] = euler_numerical_method(f, x_0, A, y_0, h)
    N = round((A - x_0)/h);
    nodes = x_0 + h * (0:N-1);
    values = zeros(1, N);
    values(1) = y_0;
    for i = 1:N-1
        values(i+1) = values(i) + h * f(nodes(i), values(i));
    end
end

f = @(x, y) x^2 - y^2;
A = 5; x_0 = 0; y_0 = 6;

hold on
for h = [0.1, 0.02, 0.001]
    [nodes, values] = euler_numerical_method(f, x_0, A, y_0, h);
    plot(nodes, values)
end

legend("h = 0.1", "h = 0.02", "h = 0.001")
```



Фигура 2: Графики на приближените решения на задачата на Коши

### 3 Числен метод за ОДУ от втори ред

#### 3.1 Общ случай

Нека имаме следната задача на Коши

$$\begin{cases} y'' = f(x, y, y') \\ y(x_0) = y_0 \\ y'(x_0) = z_0 \end{cases}$$

Ще я решим числено със следната схема.

Нека  $x \in [x_0, A]$ ,  $x_0 < A \in \mathbb{R}$

Дефинираме следната мрежа:  $x_i = x_0 + ih$ ,  $i = 0, \dots, N-1$ , където  $N \in \mathbb{N}$  и  $h = \frac{A - x_0}{N}$

$N$  е броят възли и колкото по-голямо е  $N$ , толкова по-малко е  $h$ .

Предполагаме, че  $y(x_i) \approx y_i$ , тоест, че нашето приближение е приблизително равно на истинската стойност на решението в същата точка (с някаква грешка).

#### 3.2 Извеждане на диференчната схема

Използваме следната формула за числено диференциране (разлика назад)

$$\begin{aligned} y'(x_i) &= \lim_{s \rightarrow 0} \frac{y(x_i) - y(x_i - s)}{s} \\ &\approx \frac{y(x_i) - y(x_i - h)}{h} \\ &= \frac{y(x_i) - y(x_{i-1}))}{h} \end{aligned}$$

Така получаваме

$$y'(x_i) = \frac{y(x_i) - y(x_{i-1}))}{h}$$

Диференцираме по  $x_i$

$$y''(x_i) = \frac{y'(x_i) - y'(x_{i-1}))}{h}$$

$y'(x_i)$  и  $y'(x_{i-1})$  приближаваме с числено диференциране с разлика напред

$$\begin{aligned}
y''(x_i) &= \frac{\lim_{s \rightarrow 0} \frac{y(x_i + s) - y(x_i)}{s} - \lim_{s \rightarrow 0} \frac{y(x_{i-1} + s) - y(x_{i-1})}{s}}{h} \\
&\approx \frac{\frac{y(x_i + h) - y(x_i)}{h} - \frac{y(x_{i-1} + h) - y(x_{i-1})}{h}}{h} \\
&= \frac{y(x_i + h) - y(x_i) - y(x_{i-1} + h) + y(x_{i-1})}{h^2} \\
&= \frac{y(x_i + h) - y(x_i) - y(x_i) + y(x_{i-1})}{h^2} \\
&= \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} \\
&\approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}
\end{aligned}$$

Така

$$y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f(x_i, y_i, y'_i)$$

Откъдето

$$\begin{aligned}
y_{i+1} &= 2y_i - y_{i-1} + h^2 f(x_i, y_i, y'_i) \\
y_{i+1} &= 2y_i - y_{i-1} + h^2 f\left(x_i, y_i, \frac{y_i - y_{i-1}}{h}\right)
\end{aligned}$$

Също имам начални условия

$$y_0 = y_0$$

$$\frac{y_1 - y_0}{h} = z_0$$

$$y_1 = h z_0 + y_0$$

Окончателно численото приближение, което решава задачата се задава със следната диференчна схема:

$$y_0 = y_0$$

$$y_1 = h z_0 + y_0$$

$$y_{i+1} = 2y_i - y_{i-1} + h^2 f\left(x_i, y_i, \frac{y_i - y_{i-1}}{h}\right) \quad i = 1, \dots, N-2$$



### 3.3 Конкретен пример

Дадена е задачата на Коши

$$\begin{cases} y''(x) = x^2 + xy y' + y' \\ y(0) = 1 \\ y'(0) = 2 \end{cases}$$

Да се намери приближение на решението с числен метод за ОДУ от втори ред в интервала  $x \in [0, 5]$  със стъпки  $h = 0.1$ ,  $h = 0.02$  и  $h = 0.001$ . Да се начертаят графиките на трите приближенията в една и съща координатна система.

### 3.4 Решение на код

```
warning('off','all');
% pkg install -forge symbolic
pkg load symbolic;
graphics_toolkit("gnuplot");

function [nodes, values] = numerical_method(f, x_0, A, y_0, z_0, h)
    N = round((A - x_0)/h);

    nodes = x_0 + h * (0:N-1);

    values = zeros(1, N);

    values(1) = y_0;
    values(2) = h*z_0 + y_0;

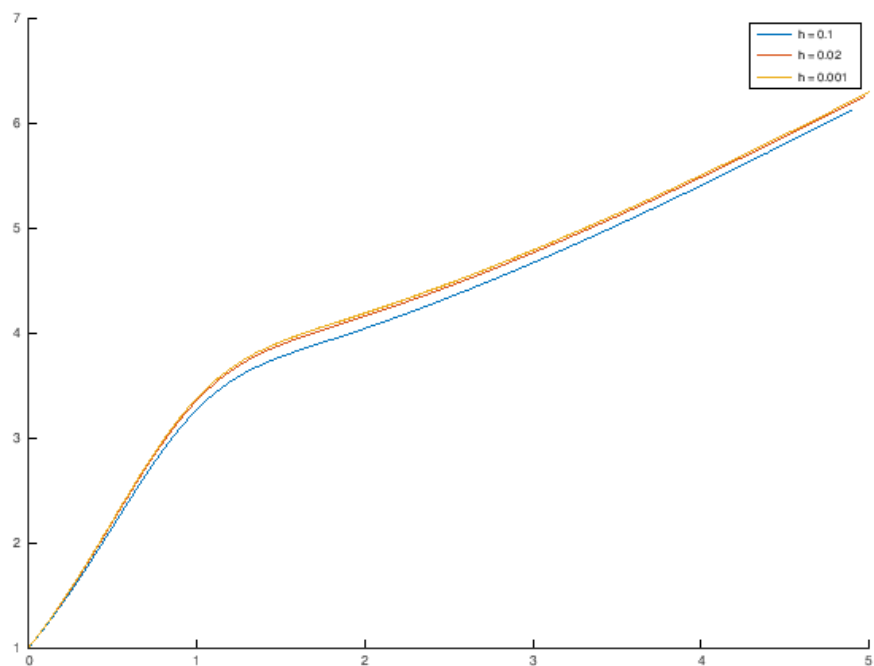
    for i = 2:N-1
        values(i+1) = 2*values(i) - values(i-1) + h^2 * f(nodes(i), values(i), (values(i)
            - values(i-1)) / h);
    end
end

f = @(x, y, y_prim) x^2 - x*y*y_prim + y_prim;
x_0 = 0;
A = 5;
y_0 = 1;
z_0 = 2;

hold on

for h = [0.1, 0.02, 0.001]
    [nodes, values] = numerical_method(f, x_0, A, y_0, z_0, h);
    plot(nodes, values)
end

legend("h = 0.1", "h = 0.02", "h = 0.001")
```



Фигура 3: Графики на приближените решения на задачата на Коши

## 4 Метод на Пикар

### 4.1 Общ случай

Имаме следната задача на Коши

$$\begin{cases} y' = f(y, x) \\ y(x_0) = y_0 \end{cases}$$

Интегрираме уравнението формално, тоест в граници от  $x_0$  до  $x$ .

$$\int_{x_0}^x y'(t) dt = \int_{x_0}^x f(y(t), t) dt$$

$$\int_{x_0}^x dy(t) = \int_{x_0}^x f(y(t), t) dt$$

$$y(x) - y(x_0) = \int_{x_0}^x f(y(t), t) dt$$

$$y(x) = y(x_0) + \int_{x_0}^x f(y(t), t) dt$$

$$y(x) = y_0 + \int_{x_0}^x f(y(t), t) dt$$

Така сведохме задачата на Коши до \*интегрално уравнение\*. Това означава, че вече вместо да имаме уравнение с  $y$  и  $y'$ , имаме уравнение само с  $y$ . Това ни позволява да направим последователни приближения, започвайки с  $y_0$  от задачата на Коши и обща форма, която се задава по следния начин:

$$y_{i+1}(x) = y_0 + \int_{x_0}^x f(y_i(t), t) dt$$

## 4.2 Конкретен пример

Дадена е следната задача на Коши

$$\begin{cases} y' = y \\ y(0) = 1 \end{cases}$$

Да се намерят първите 5 приближения с метода на Пикар аналитично.

Да се начертаят с MATLAB/Octave в една координатна система графиките на първите 5 приближения на задачата на Коши по метода на Пикар в интервала  $x \in [-2, 2]$ .

## 4.3 Аналично решение

$$\begin{cases} y' = y, \\ y(0) = 1. \end{cases}$$

Интегралното уравнение по метода на Пикар е:

$$y_{i+1}(x) = y_0 + \int_{x_0}^x y_i(t) dt$$

$$y_{i+1}(x) = 1 + \int_0^x y_i(t) dt$$

*Първо приближение:*

$$y_0 = 1.$$

Второ приближение:

$$y_1 = 1 + \int_0^x 1 dt = 1 + [t]_0^x.$$

$$y_1 = 1 + x.$$

Трето приближение:

$$y_2 = 1 + \int_0^x (1 + t) dt.$$

$$= 1 + \left[ t + \frac{t^2}{2} \right]_0^x.$$

$$y_2 = 1 + x + \frac{x^2}{2}.$$

Четвърто приближение:

$$y_3 = 1 + \int_0^x \left( 1 + t + \frac{t^2}{2} \right) dt.$$

$$= 1 + \left[ t + \frac{t^2}{2} + \frac{t^3}{6} \right]_0^x.$$

$$y_3 = 1 + x + \frac{x^2}{2} + \frac{x^3}{6}.$$

Пето приближение:

$$y_4 = 1 + \int_0^x \left( 1 + t + \frac{t^2}{2} + \frac{t^3}{6} \right) dt.$$

$$= 1 + \left[ t + \frac{t^2}{2} + \frac{t^3}{6} + \frac{t^4}{24} \right]_0^x.$$

$$y_4 = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}.$$

Така виждаме, че приближенията формират реда на Тейлър на  $e^x$ .

Също е добре да отбележим, че последователните приближения стават все по-точни в околност на точката  $(x_0, y_0)$

## 4.4 Решение на код

```
warning('off','all');
pkg load symbolic;
graphics_toolkit("gnuplot");

syms x y(x)

dsolve(diff(y, x) == y, y(0) == 1)

syms x y f

x_0 = 0;
y_0 = 1;

approximations_count = 5;

f(x, y) = y;

approximations = sym([]);

approximations(1) = 1;

for i = 1:approximations_count-1
    approximations(i+1) = approximations(1) + int(f(x, approximations(i)), x, x_0, x);
end

transpose(approximations)

x_axis = -2:0.2:2;

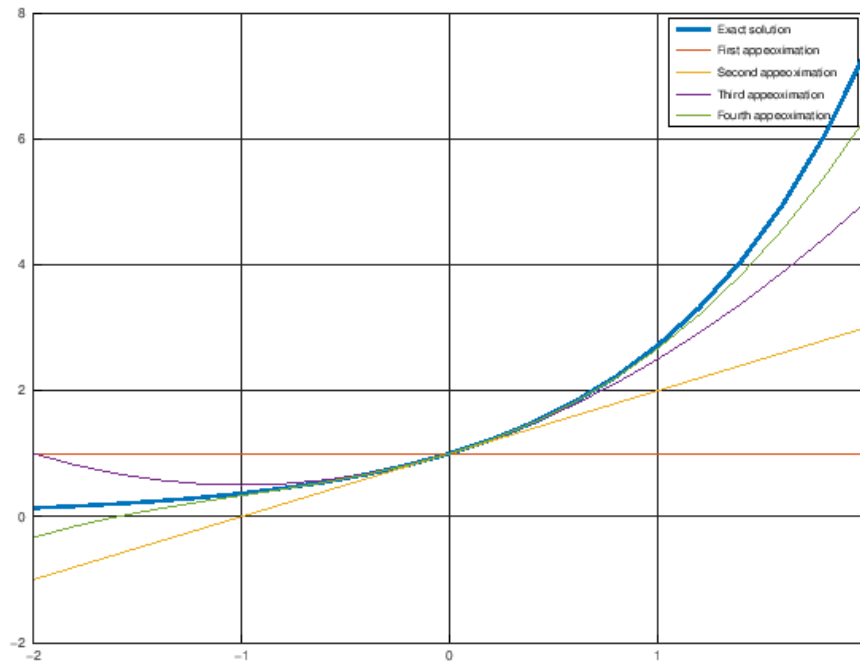
for i = 1:approximations_count-1
    approximations_values(i, :) = double(subs(approximations(i), x, x_axis));
end

hold on;
grid on;

plot(x_axis, exp(x_axis), "LineWidth", 6)

for i = 1:approximations_count-1
    plot(x_axis, approximations_values(i, :), "LineWidth", 2)
end

legend("Exact solution", "First appeoximation", "Second appeoximation", "Third
    appeoximation", "Fourth appeoximation", "Fifth appeoximation")
```



Фигура 4: Графики на последователни приближени решения на задачата на Коши

## 5 Поле от прави

### 5.1 Общ случай

Нека имаме диференциално уравнение от вида  $y' = f(x, y)$

Ще построим поле от прави за него.

Избираме си мрежа от точки в равнината.

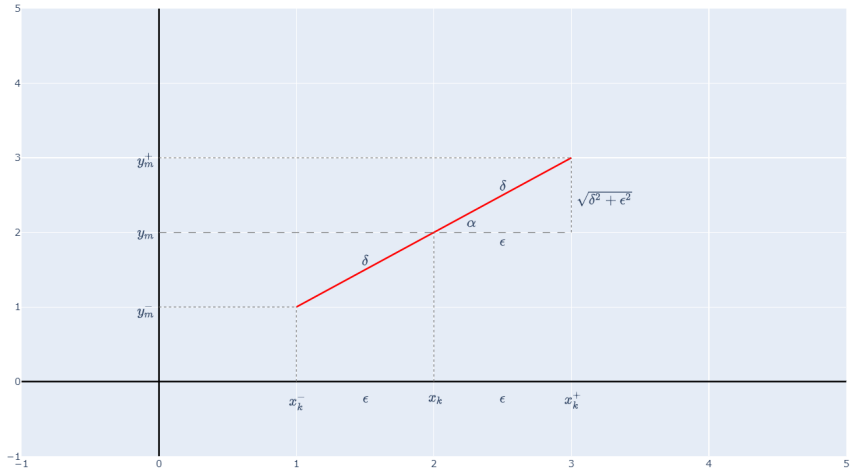
За всяка точка  $(x_k, y_m)$  от мрежата ще разглеждаме следната задача на Коши

$$\begin{cases} y' = f(x, y) \\ y(x_k) = y_m \end{cases}$$

Няма да я решаваме обаче.

Ще построим отсечка с фиксирана дължина, която да минава през точката  $(x_k, y_m)$ .

Нека тази дължина е  $2\delta$ .



Фигура 5: Графики на последователни приближени решения на задачата на Коши

Знаем, че  $y'(x_k) = \tan \alpha$

Така наклонът и дължината на отсечката са еднозначно определени.

Остава да открием точни координати за точките  $(x_k^+, y_m^+)$  и  $(x_k^-, y_m^-)$ . Те са симетрични и е достатъчно да намерим координатите само на една от двете точки - например  $(x_k^+, y_m^+)$

Означаваме с  $\epsilon$  разстоянието между  $x_k$  и  $x_k^+$

С Питагорова теорема намираме третата страна на правоъгълния триъгълник горе дясно.

От тригонометричното определени за тангенс  $\tan \alpha = \frac{\sqrt{\delta^2 + \epsilon^2}}{\epsilon}$

$$\begin{cases} y'(x_k) = f(x_k, y_m) \\ y'(x_k) = \tan \alpha \\ \tan \alpha = \frac{\sqrt{\delta^2 + \epsilon^2}}{\epsilon} \end{cases}$$

Значи

$$f(x_k, y_m) = \frac{\sqrt{\delta^2 + \epsilon^2}}{\epsilon}$$

$$f(x_k, y_m)^2 = \frac{\delta^2 + \epsilon^2}{\epsilon^2}$$

$$f(x_k, y_m)^2 + 1 = \frac{\delta^2}{\epsilon^2}$$

$$\epsilon^2 = \frac{\delta^2}{f(x_k, y_m)^2 + 1}$$

$$\epsilon = \frac{\delta}{\sqrt{f(x_k, y_m)^2 + 1}}$$

Намерихме  $\epsilon$  и съответно можем да намерим  $x$ -координатата  $x_k^+ = x_k + \epsilon$ . Остава да намерим и  $y$ -координатата. Ще я намерим по следния начин

$$y_m^+ - y_m = \sqrt{\delta^2 - \epsilon^2} = \epsilon f(x_k, y_m)$$

Откъдето

$$y_m^+ = y_m + \epsilon f(x_k, y_m)$$

## 5.2 Конкретен пример

Нека имаме следното диференциално уравнение:

$$y' = -\frac{y}{x}$$

1. Да се начертае поле от прави за това ДУ в интервала  $x \in [-5, 5], y \in [-5, 5]$  с черен цвят.
2. В същата координатна система начертайте графиката на решението на задачата на Коши за даденото диференциално уравнение с начално условие  $y(1.5) = -2$  с червен цвят.

## 5.3 Решение на код

```
warning('off','all');
% pkg install -forge symbolic
pkg load symbolic;
graphics_toolkit("gnuplot");

x = -5:0.6:5;
y = -5:0.6:5;
delta = 0.2;

f = @(x, y) - y / x;

axis([-6, 6, -6, 6])
hold on

for k=1:length(x)
    for m=1:length(y)
        epsilon = delta / (sqrt(1 + f(x(k), y(m))^2));
        gama = epsilon * f(x(k), y(m));
```



```

    plot([x(k) - epsilon, x(k) + epsilon], [y(m) - gama, y(m) + gama], 'black-');
    plot(x(k), y(m), 'black.')
end
end

x_0 = 1.5;
y_0 = -2;

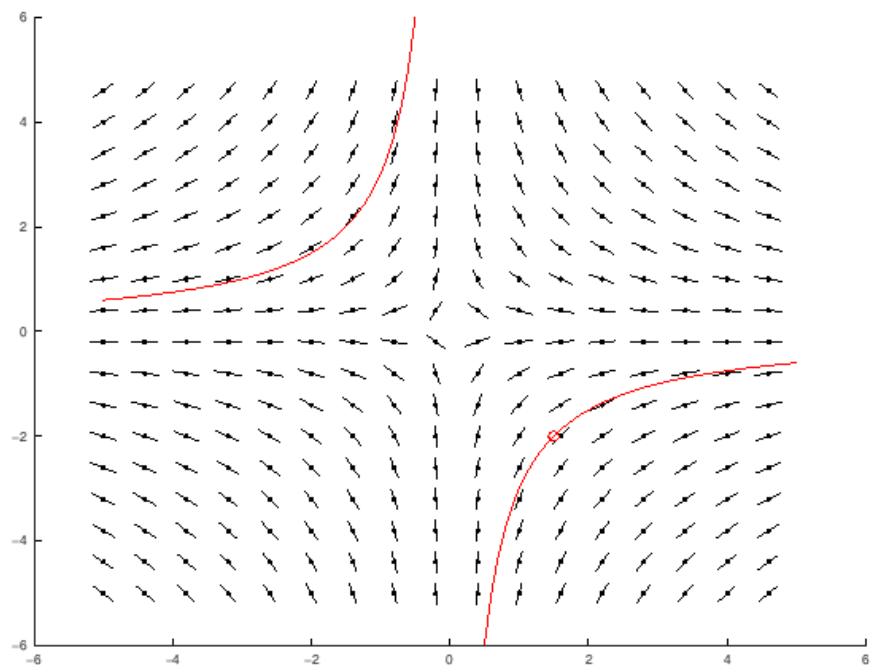
plot(x_0, y_0, 'redo')

syms x y(x)

sol = dsolve(diff(y, x) == - y / x, y(x_0) == y_0)

x_axis = -5:0.1:5;
plot(x_axis, double(subs(sol, x, x_axis)), "red-")

```



Фигура 6: Графики на поле от прави за даденото ДУ

## 5.4 Допирателна към интегрална крива

Нека имаме следната задача на Коши:

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

Производната  $y'(x_0)$  в точката  $x_0$  се изчислява като:

$$y'(x_0) = f(x_0, y_0) = \tan \alpha$$

Това е наклонът  $\tan \alpha$  на допирателната в точката  $(x_0, y_0)$ .  
 Права с наклон  $\tan \alpha$ , минаваща през  $(x_0, y_0)$ , се описва с:

$$y = x \tan \alpha + n$$

$$y = f(x_0, y_0)x + n$$

Тази права минава през точката  $(x_0, y_0)$  и съответно имаме следното равенство:

$$y_0 = f(x_0, y_0)x_0 + n$$

Откъдето

$$n = y_0 - f(x_0, y_0)x_0$$

Заместваме в уравнението на правата:

$$y = f(x_0, y_0)x_0 + y_0 - f(x_0, y_0)x_0$$

Уравнението на допирателната към интегралната крива окончателно е:

$$y = y_0 + f(x_0, y_0)(x - x_0)$$