Глава 2

Числени методи за ОДУ от първи ред

Твърде малко ОДУ, възникващи в практиката, могат да бъдат решени аналитично. Такива са например уравненията с разделящи се променливи, линейните уравнения и др. (за повече информация – вж. [2, 8]). Ето защо най-често за тяхното решаване е необходимо използването на числени методи. Един начин за численото решаване на ОДУ е да приближим производните в диференциалното уравнение по подходящ начин.

2.1 Приближаване на производни на функция на една променлива.

Производната на функция може да бъде дефинирана като

$$f'(t) = \lim_{\Delta t \to 0} \frac{f(t + \Delta t) - f(t)}{\Delta t}.$$

Тази дефиниция не е възможно да се реализира в компютърната аритметика, тъй като всички числа в компютъра се представят с определен брой значещи цифри след десетичната точка. Следователно бихме могли да приближим производната по следния начин:

$$f'(t) \approx \frac{f(t + \Delta t) - f(t)}{\Delta t}, \tag{2.1}$$

като изберем Δt да бъде фиксирано малко число (например $\Delta t = 0.001$).

Ако положим формално $\Delta t = -\Delta s$ във формула (2.1), то ще получим друг начин за приближаване на производна:

$$f'(t) \approx \frac{f(t + \Delta t) - f(t)}{\Delta t} = \frac{f(t - \Delta s) - f(t)}{-\Delta s} = \frac{f(t) - f(t - \Delta s)}{\Delta s}$$

или

$$f'(t) \approx \frac{f(t) - f(t - \Delta t)}{\Delta t}. \tag{2.2}$$

Като съберем формулите (2.1) и (2.2), получаваме следното:

$$f'(t) \approx \frac{f(t + \Delta t) - f(t - \Delta t)}{2\Delta t}. \tag{2.3}$$

Задача 2. Да се приближи производната на функцията $f(x) = e^x$ в интервала $[0, 10]$, като се използват формули (2.1), (2.2) или (2.3). Да се начертаят на една графика производната и нейните приближения при $\Delta t = 0.5$, $\Delta t = 0.001$ и $\Delta t = 10^{-15}$.

Решение. На графиките по-долу са изобразени производната на e^x (т.е. e^x) в синьо и приближенията ѝ, използвайки съответно формули (2.1), (2.2) и (2.3) с $\Delta t = 0.5$, в оранжево:

- **(f[t+Δt]-f[t])/Δt**

In[81]:=
```
Δt = 0.5;
f[t_] = Exp[t];
DerF[t_] = D[f[t], t];
plotd1 = Plot[{DerF[t], (f[t + Δt] – f[t]) / Δt}, {t, 0, 10}];
```

- **(f[t] - f[t-Δt])/Δt**

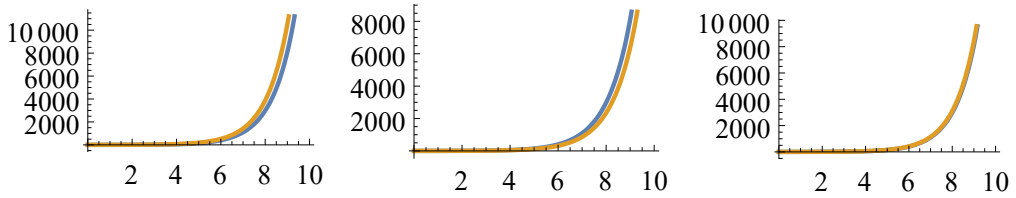
In[85]:=
```
plotd2 = Plot[{DerF[t], (f[t] – f[t – Δt]) / Δt}, {t, 0, 10}];
```

- **(f[x+Δt] - f[x-Δt])/(2Δt)**

In[86]:=
```
plotd3 = Plot[{DerF[t], (f[t + Δt] – f[t – Δt]) / (2 Δt)}, {t, 0, 10}];
```

In[87]:=
```
GraphicsRow[{plotd1, plotd2, plotd3}]
```

Out[87]=



За $\Delta t = 0.05$ формула (2.3) да дава най-добра апроксимация на e^x. Също така в случая формула (2.1) приближава отгоре функцията, а формула (2.2) – отдолу. За произволна функция, ако формула (2.1) приближава функцията отгоре в даден интервал, то формула (2.2) ще я приближава отдолу.

Сега да видим какво се случва за различни стойности на Δt. От дефиницията на производна следва, че при намаляване на Δt би трябвало да се получи по-добро приближение (в дефиницията за производна $\Delta t \to 0$). Резултатите от експериментите за $\Delta t = 0.5$, $\Delta t = 0.001$ и $\Delta t = 10^{-15}$ са показани по-долу:

- **Δt=0.5**

In[8]:=
```
plotStep5 = Plot[{DerF[t], (f[t + 0.5] − f[t]) / 0.5}, {t, 0, 10}];
```

- **Δt=0.001**

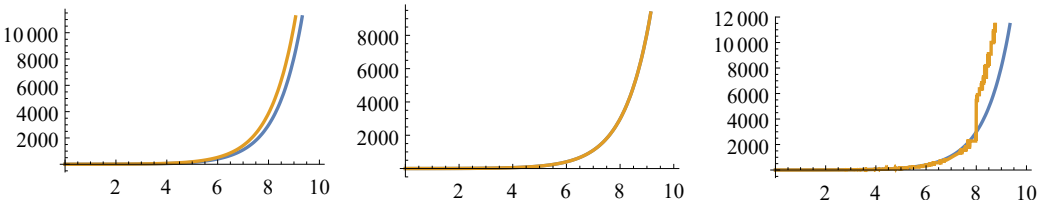
In[9]:=
```
plotStep001 = Plot[{DerF[t], (f[t + 0.001] − f[t]) / 0.001}, {t, 0, 10}];
```

- **Δt=10^(-15)**

In[10]:=
```
plotStepMin15 = Plot[{DerF[t], (f[t + 10^(−15)] − f[t]) / 10^(−15)}, {t, 0, 10}];
```

In[54]:=
```
GraphicsRow[{plotStep5, plotStep001, plotStepMin15}]
```

Out[54]=



Намаляването на грешката се забелява на първите две графики. Но при $\Delta t = 10^{-15}$ полученото приближение не е добро. Причината за това е, че числата в компютъра се представят с фиксиран брой значещи цифри. Това означава, че грешката първоначално ще намалява при намаляване на стъпката, но когато машинната точност бъде достигната (и следователно намаляването на стъпката не може повече да подобрява точността), грешката ще започне да расте заради увеличаването на броя на операциите. Това е и причината за появата на "шума" на третата графика. □

2.2 Абсолютна и относителна грешка. Полином на Тейлър и оценка на грешката.

Както отбелязахме, повечето числените методи включват някаква апроксимация. Ето защо разбирането на идеята за грешка е от много голяма важност за ефективното им използване. Нека да въведем следните дефиниции:

Дефиниция 1. *Абсолютна грешка наричаме разликата между точната и приближената стойност при дадена апроксимация:*

$$\varepsilon_a := exact\ value - approximation.$$

Дефиниция 2. *Относителна грешка дефинираме по следния начин:*

$$\varepsilon_r := \frac{exact\ value - approximation}{exact\ value} = \frac{\varepsilon_a}{exact\ value}.$$

Задача 3. Пресметнете абсолютните и относителните грешке за задача 2.

Решение. Абсолютните и относителните грешки при използването на формули (2.1), (2.2), (2.3) с $\Delta t = 0.001$ за приближение на функцията e^x са изобразени на графиките

- **(f[t+Δt] - f[t])/Δt**

In[114]:=
```
Δt = 0.001;
absErrorD1 = Plot[DerF[t] – (f[t + Δt] – f[t]) / Δt, {t, 0, 10}];
relErrorD1 = Plot[(DerF[t] – (f[t + Δt] – f[t]) / Δt) / DerF[t], {t, 0, 10}];
```

- **(f[t] - f[t-Δt])/Δt**

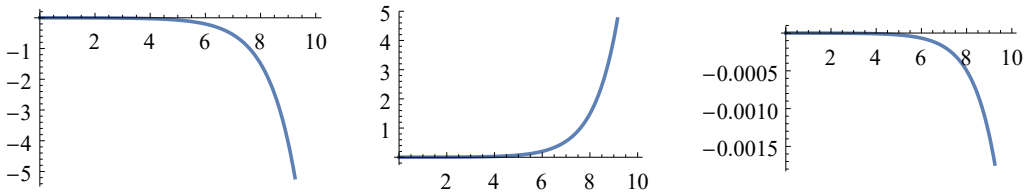
In[117]:=
```
absErrorD2 = Plot[DerF[t] – (f[t] – f[t – Δt]) / Δt, {t, 0, 10}];
relErrorD2 = Plot[(DerF[t] – (f[t] – f[t – Δt]) / Δt) / DerF[t], {t, 0, 10}];
```

- **(f[t+Δt] - f[t-Δt])/(2Δt)**

In[119]:=
```
absErrorD3 = Plot[DerF[t] – (f[t + Δt] – f[t – Δt]) / (2 Δt), {t, 0, 10}];
relErrorD3 = Plot[(DerF[t] – (f[t + Δt] – f[t – Δt]) / (2 Δt)) / DerF[t], {t, 0, 10}];
```

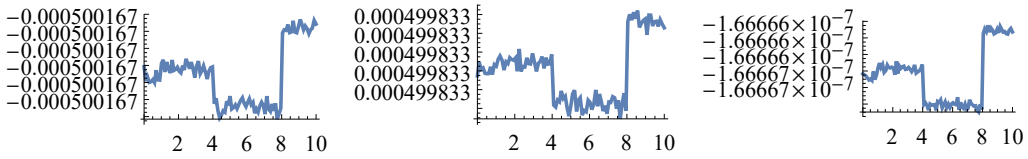
In[121]:=
```
GraphicsRow[{absErrorD1, absErrorD2, absErrorD3}]
```

Out[121]=



In[122]:=
```
GraphicsRow[{relErrorD1, relErrorD2, relErrorD3}]
```

Out[122]=



При една и съща стойност на Δt формула (2.3) дава най-малка абсолютна и относителна грешка, т.е. тя приближава най-добре производната на функция.

Често, като измервател за грешката в даден интервал, се използва максимална абсолютна грешка – колко най-много се различава функцията от нейното приближение в даден интервал. В конкретния случай абсолютната грешка е най-голяма в десния край на интервала. Причината за това е, че функцията e^x расте много бързо. Относителната грешка (или процентната грешка) обаче е приблизително еднаква за целия интервал. □

Абсолютната и относителната грешка представляват добри измерители за това колко е добро едно приближение. В една реална задача обаче обикновено е невъзможно да ги пресметнем, тъй като точното решение не е известно. Затова е важно да можем да оценим грешката, която допускаме при дадена апроксимация. Често в числените методи за диференциални уравнения за оценка

от този вид се използва ред на Тейлър, тъй като много функции могат да се представят чрез развитието си в ред на Тейлър

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k, \tag{2.4}$$

където x_0 е точката, около която развиваме. Тази формула не е възможно да се релизира на компютър, тъй като в нея има сума с безброй много събираеми. Това, което бихме могли да направим, е да приближим функцията с полином от достатъчно голяма степен.

Задача 4. Приближете функцията $x \sin x \cos x$ в интервала $[0, 10]$, като използвате полином на Тейлър от степен 2, 3, 4 около $x_0 = 5$. Начертайте функцията и приближенията на една графика.

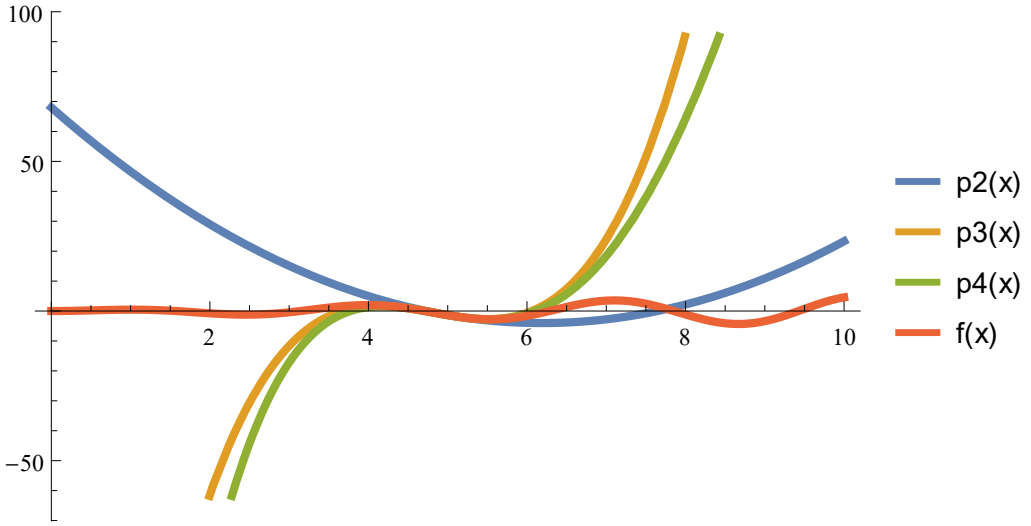
Решение. За целта бихме могли да използваме следния код:

In[47]:=
```
f[x_] := Sin[x] * Cos[x] * x
Df1[x_] = D[f[x], {x, 1}];
Df2[x_] = D[f[x], {x, 1}];
Df3[x_] = D[f[x], {x, 1}];

p2[x_, x0_] := f[x0] + f'[x0] (x - x0) + f''[x0] (x - x0)^2 / 2!
p3[x_, x0_] := p2[x, x0] + f'''[x0] (x - x0)^3 / 3!
p4[x_, x0_] := p3[x, x0] + f''''[x0] (x - x0)^4 / 4!

Plot[{p2[x, 5], p3[x, 5], p4[x, 5], f[x]}, {x, 0, 10},
     PlotLegends → {"p2(x)", "p3(x)", "p4(x)", "f(x)"},
     PlotStyle → Table[Thickness[0.01], {i, 4}]]
```

Out[54]=



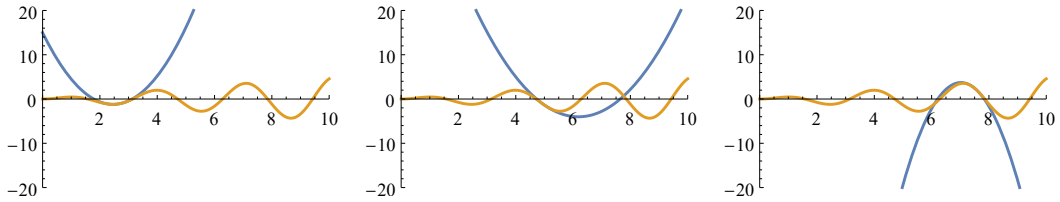
Това, което е важно да отбележим, че при увеличане на степента на полинома, той се „залепва“ все по-добре за функцията около точката, в която развиваме.

Друг интересен въпрос е какво се случва, ако x_0 се променя. Отговор на този въпрос дават следващите графики. На тях е изобразена функцията (в

оранжево) полинома на Тейлър от втора степен, построен с развитие около точките $x_0 = 2.5$, $x_0 = 5$ и $x_0 = 7.5$ (в синьо):

```
In[58]:= GraphicsRow[{Plot[{p2[x, 2.5], f[x]}, {x, 0, 10}, PlotRange → {{0, 10}, {-20, 20}}],
                      Plot[{p2[x, 5], f[x]}, {x, 0, 10}, PlotRange → {{0, 10}, {-20, 20}}],
                      Plot[{p2[x, 7.5], f[x]}, {x, 0, 10}, PlotRange → {{0, 10}, {-20, 20}}]}]
```

Out[58]=



От графиките може да се заключи, че апроксимацията е най-добра за точки, които са близо до точката x_0 (като в нея грешката е 0). Следователно, ако искаме да получим добри резултати в даден интервал, е добра идея да развиваме около средата на този интервал. □

Задача 5 (За магистрите). Коя е минималната степен на полином на Тейлър около $x = 5$, при която абсолютната грешка по модул при приближаването на $x \sin x \cos x$ във всяка точка от интервала $[0, 10]$ е по-малка от 0.1?

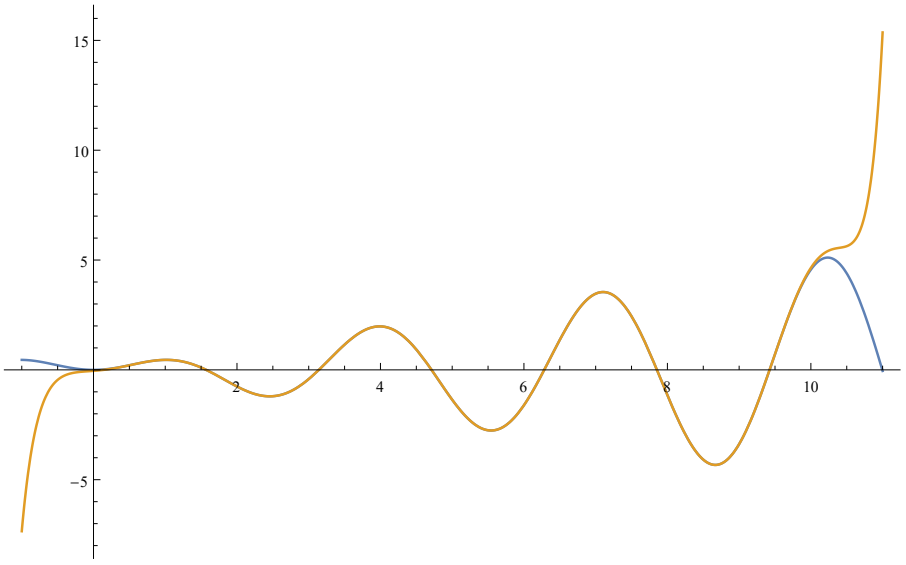
Решение. Прилагаме решение на задачата

```
In[126]:= f[x_] := Sin[x] * Cos[x] * x
          taylorSeries[x_, order_] := f[5] + Sum[ReplaceAll[D[f[y], {y, k}], y → 5.] * (x - 5.)^k / k!, {k, 1, order}];

          For[order = 1, order ≤ 100, order++,
            maxError = Maximize[{Abs[f[x] - taylorSeries[x, order]], 0 ≤ x ≤ 10}, x][[1]];
            If[maxError ≤ 0.1,
                 Print[{order, maxError}];
                 Break[]];
           ];
          {28, 0.0780946}

In[129]:= Plot[{f[x], taylorSeries[x, 28]}, {x, -1, 11}, PlotRange → All]
```

Out[129]=



Може да се види, че при увеличаване на степента на полинома на Тейлър, той се приближава все повече до функцията, която апроксимира. Важно е да се

отбележи, че поради грешките от закръглявания не е възможно в компютърната аритметика да приближим функция с произволна точност. Нещо повече, увеличаването на степента на полинома на Тейлър не винаги ще води до по-добри приближения – след достигане на машинната точност, при по-голям брой аритметични операции грешките от закръгляване ще доведат до по-лоши резултати. □

След този кратък обзор върху формулата на Тейлър, сме готови да преминем към темата за оценка на грешката.

Задача 6. Намерете оценка за абсолютната грешка на апроксимациите в задача 2.

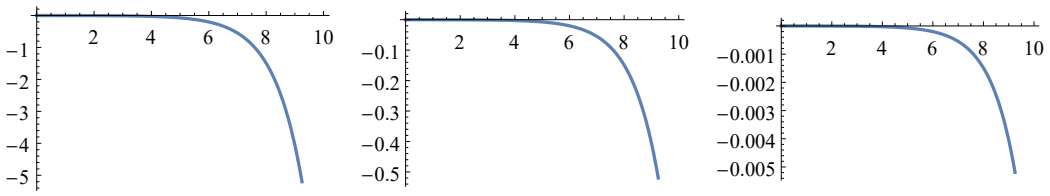
Решение. Първо, ще изведем три теоретични оценки на грешката. За целта, ще намерим разликата между точното и приближеното решение и ще развием в ред на Тейлър около точката t. По този начин получаваме следните оценки за формули (2.1), (2.2) и (2.3):

$$err_1 = f'(t) - \frac{f(t+\Delta t) - f(t)}{\Delta t} = f'(t) - \frac{f(t) + \Delta t f'(t) + O(\Delta t^2) - f(t)}{\Delta t} = O(\Delta t)$$
$$err_2 = f'(t) - \frac{f(t) - f(t-\Delta t)}{\Delta t} = f'(t) - \frac{f(t) - (f(t) - \Delta t f'(t) + O(\Delta t^2))}{\Delta t} = O(\Delta t)$$
$$err_3 = f'(t) - \frac{f(t+\Delta t) - f(t-\Delta t)}{\Delta t} = O(\Delta t^2)$$

Абсолютната грешка, която получаваме за формули (2.1) и (2.2), е $O(\Delta t)$. Това означава, че ако намалим Δt десет пъти, грешката ще намалее от порядъка на 10 пъти. За да верифицираме този теоретичен резултат, нека да начертаем графиките на абсолютните грешки при $\Delta t = 0.001$, $\Delta t = 0.0001$, $\Delta t = 0.000001$,

```
In[47]:= f[x_] := E^x
         absErrorD1Step001 = Plot[DerF[t] - (f[t + 0.001] - f[t]) / 0.001, {t, 0, 10}];
         absErrorD1Step0001 = Plot[DerF[t] - (f[t + 0.0001] - f[t]) / 0.0001, {t, 0, 10}];
         absErrorD1Step000001 = Plot[DerF[t] - (f[t + 0.000001] - f[t]) / 0.000001, {t, 0, 10}];
         GraphicsRow[{absErrorD1Step001, absErrorD1Step0001, absErrorD1Step000001}]
```

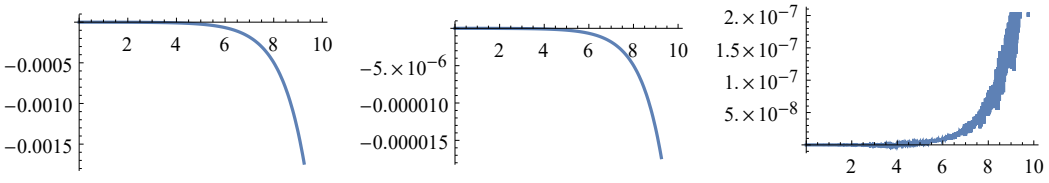
Out[51]=



Аналогично получаваме, че абсолютната грешка при апроксимиране на производната на функцията с формулата (2.3) е $O(\Delta t^2)$. Това означава, че ако намалим Δt десет пъти, грешката ще намалее от порядъка на 100 пъти. При $\Delta t = 0.001$, $\Delta t = 0.0001$, $\Delta t = 0.00001$ получаваме следните графики за абсолютната грешка:

In[60]:=
```
absErrorD3Step001 = Plot[DerF[t] - (f[t + 0.001] - f[t - 0.001]) / 0.002, {t, 0, 10}];
absErrorD3Step0001 = Plot[DerF[t] - (f[t + 0.0001] - f[t - 0.0001]) / 0.0002, {t, 0, 10}];
absErrorD3Step00001 = Plot[DerF[t] - (f[t + 0.00001] - f[t - 0.00001]) / 0.00002, {t, 0, 10}];
GraphicsRow[{absErrorD3Step001, absErrorD3Step0001, absErrorD3Step00001}]
```

Out[63]=



Разбира се, тук е мястото да отбележим, че ако изберем много малко Δt грешките от закръгляване ще започнат да доминират над точността на апроксимацията. □

2.3 Задача на Коши за ОДУ от първи ред. Обща теория.

Започваме изучаването на числените методи за решаване на диференциални уравнения с тези за решаване на ОДУ от първи ред, т.е. ще разглеждаме ОДУ от вида

$$\frac{du}{dt} = f(t, u(t)).$$

Уравнения от по-висок ред можем да сведем до такива от първи ред чрез полагане. Например Законът на Нютон

$$\frac{d^2u}{dt^2} = \frac{F(t)}{m}$$

може да се сведе до система ОДУ от първи ред чрез полагането $du/dt = v$. Получаваме системата

$$\begin{aligned} \frac{du}{dt} &= v, \\ \frac{dv}{dt} &= \frac{F(t)}{m}. \end{aligned}$$

От друга страна, системи уравнения от първи ред можем да разглеждаме като едно векторно уравнение. Горния пример можем да запишем във вида

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} v \\ F(t)/m \end{bmatrix}$$

или още

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}(t)),$$

където $\mathbf{u} = (u, v)^T \in \mathbb{R}^2$ е вектор и $\mathbf{f}(t, \mathbf{u}) = (v, F(t)/m)^T$ е векторна функция $\mathbf{f} : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}^2$.

Казаното дотук ни дава основание да разглеждаме именно уравнения от първи ред. Общият им вид е

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}(t)),$$