Ako (ne)riešiť diferenciálne rovnice numericky

Jakub Šťavina

Department of Physics and Astronomy,
The University of Manchester

Úvodné sústredenie TMF

4. Október 2024

Vektory a derivácie

- **Vektor** je usporiadaná n-tica reálnych čísel
- Je možné ich sčítavať a násobiť skalárom:

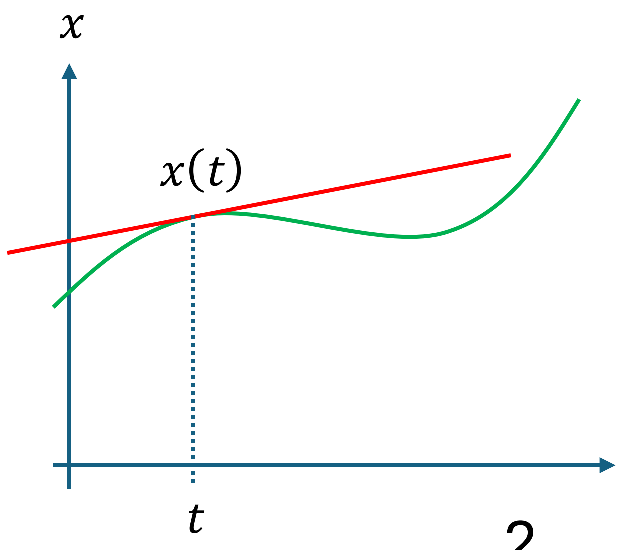
$$\text{Ak} \quad \mathbf{a} = \begin{pmatrix} a_x \\ a_y \\ \vdots \end{pmatrix} \quad \text{a} \quad \mathbf{b} = \begin{pmatrix} b_x \\ b_y \\ \vdots \end{pmatrix} \quad \text{potom} \quad \mathbf{a} + \mathbf{b} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \\ \vdots \end{pmatrix}$$

$$\text{a navyše} \quad k\mathbf{a} = \begin{pmatrix} ka_x \\ ka_y \\ \vdots \end{pmatrix}.$$

- **Derivácia** funkcie x v bode t:

= **sklon dotyčnice** ku grafu funkcie x v bode t:

$$\frac{dx}{dt} = \lim_{h \to 0} \left(\frac{x(t+h) - x(t)}{h} \right)$$



Čo je to diferenciálna rovnica?

- Vzťah medzi funkciami $x, y, \ldots$ a ich deriváciami

$$\mathcal{F}\left(x, y, \ldots; \frac{dx}{dt}, \frac{dy}{dt}, \ldots, \right) = 0$$

- Štandardný tvar

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}), \qquad \mathbf{x}(0) = \mathbf{x}_0$$

kde:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ \vdots \end{pmatrix}, \qquad \mathbf{f}(t, \mathbf{x}) = \begin{pmatrix} f_x(t, x, y, \ldots) \\ f_y(t, x, y, \ldots) \\ \vdots \end{pmatrix}, \qquad \frac{d\mathbf{x}}{dt} = \begin{pmatrix} dx/dt \\ dy/dt \\ \vdots \end{pmatrix}$$

RIEŠENÍM NIE SÚ ČÍSLA, ALE FUNKCIE: $\mathbf{x}(t; \mathbf{x}_0)$

Príklady diferenciálnych rovníc:

- Newtonov II zákon

$$\frac{d}{dt}\begin{pmatrix}\mathbf{p}\\ \mathbf{x}\end{pmatrix} = \begin{pmatrix}\mathbf{F}(\mathbf{x},t)\\ \mathbf{p}/m\end{pmatrix} \longleftarrow \frac{d^2\boldsymbol{x}}{dt^2} = \mathbf{a} = \frac{1}{m}\mathbf{F}(\mathbf{x})$$

- Zákon chladnutia

$$\frac{dT}{dt} = r(T_0 - T)$$

- RLC obvody

$$\frac{d^2I}{dt^2} + \frac{R}{L}\frac{dI}{dt} + \frac{1}{LC}I = 0$$

- Dynamika populácií

$$\frac{d}{dt}\begin{pmatrix}x\\ y\end{pmatrix} = \begin{pmatrix}\alpha x - \beta xy\\ -\gamma y + \delta xy\end{pmatrix}$$

- A nespočet ďalších...

Ako numericky derivovať?

- Z definície:

Dostatočne malé h:
$|h| \ll 1$

$$\frac{dx}{dt} = \lim_{h \to 0}\left(\frac{x(t+h)-x(t)}{h}\right) \approx \frac{x(t+h)-x(t)}{h}$$

- Čo teda hovorí diferenciálna rovnica v štandardnom tvare?

$$\frac{\mathbf{x}(t+h)-\mathbf{x}(t)}{h} \approx \mathbf{f}(\mathbf{x},t)$$

Eulerova metóda:

$$\frac{\mathbf{x}(t+h)-\mathbf{x}(t)}{h} \approx \mathbf{f}(\mathbf{x},t)$$

- Upravme ju na tvar

$$\mathbf{x}(t+h) \approx \mathbf{x}(t) + h\mathbf{f}(\mathbf{x},t)$$

- Zadefinujme $\mathbf{x}_n \approx \mathbf{x}(hn)$ podľa vzťahu

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n, t)$$

EXPLICITNÁ EULEROVA METÓDA

Príklad:

- Diferenciálna rovnica

$$\frac{dx}{dt} + x = 0, \quad x(0) = 1$$

- Identifikuj $f(x)$

$$f(x) = -x$$

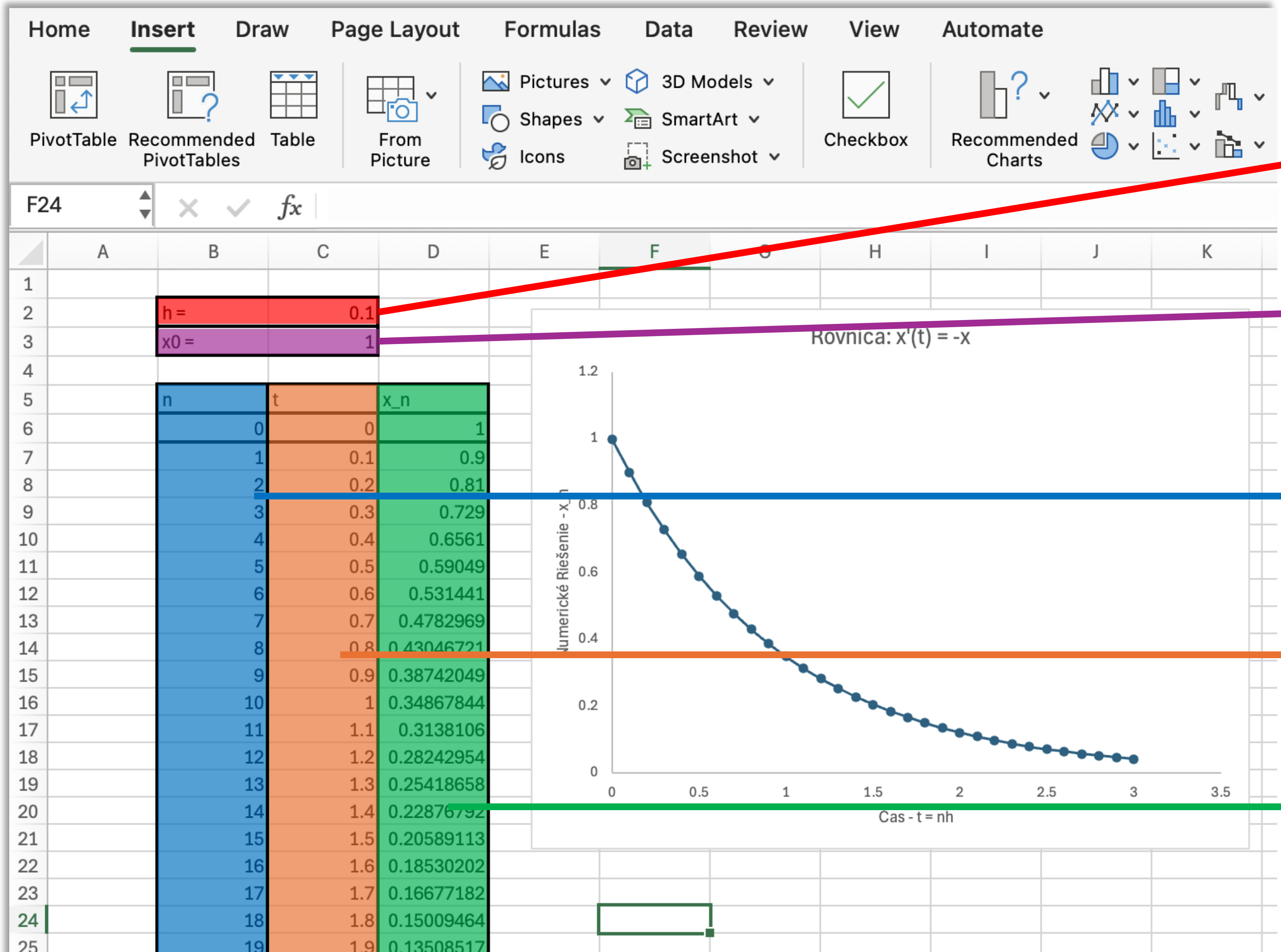
- Eulerova metóda

$$x_{n+1} = x_n + hf(x_n)$$

- Dosadiť za $f(x_n)$ a uplatni počiatočnú podmienku

$$x_{n+1} = x_n - hx_n, \quad x_0 = 1$$

Jednoduchá implementácia v Exceli



Časový krok: $h = 0.1$

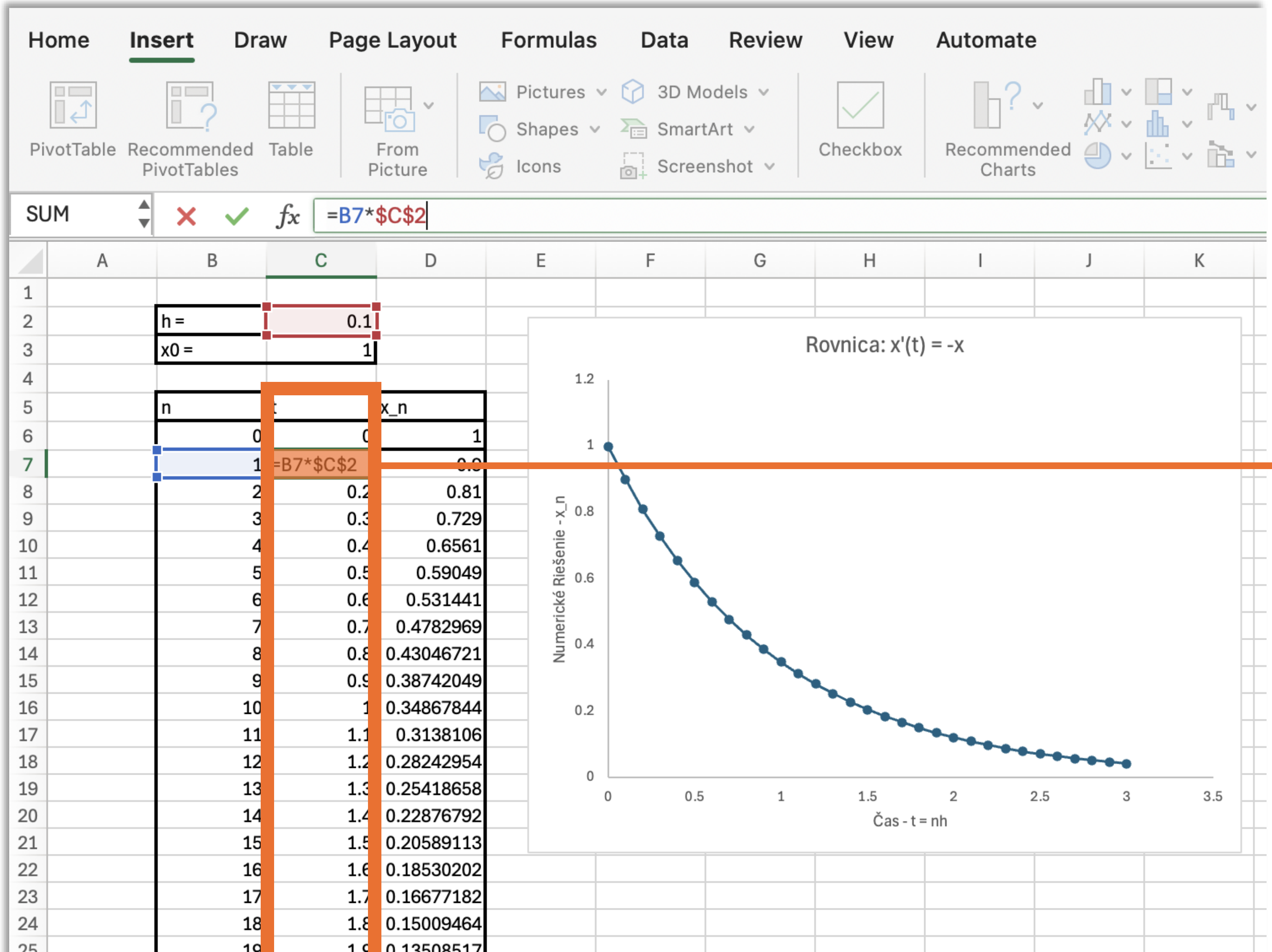
Počiatočná podmienka: $x_0 = 1$

Číslo kroku: n

Čas: $t = nh$

Riešenie: $x_n = x_{n-1} - h\, x_{n-1}$

Jednoduchá implementácia v Exceli

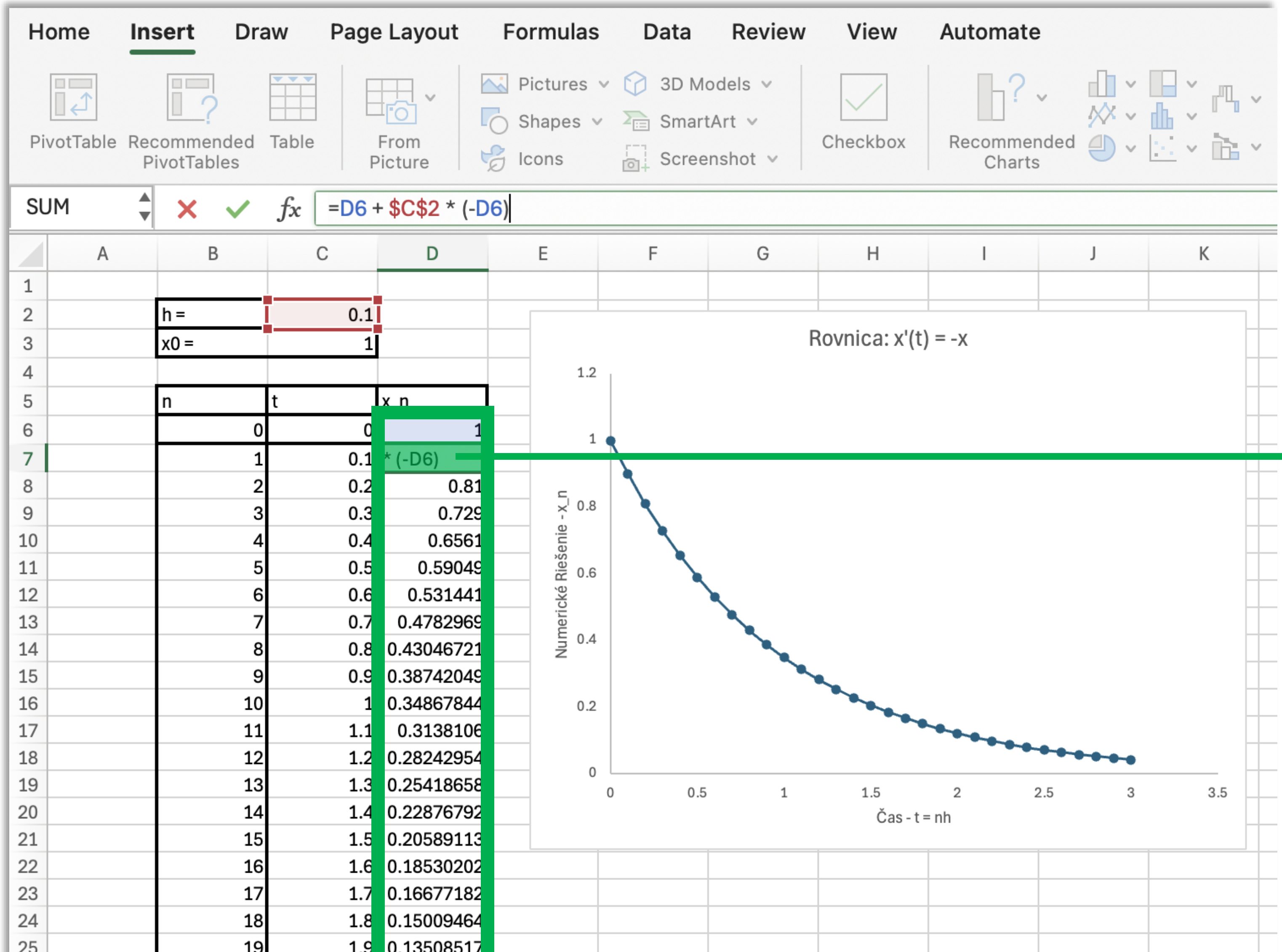


Číslo kroku: n

= B7 * C2

Časový krok: $h = 0.1$

Jednoduchá implementácia v Exceli

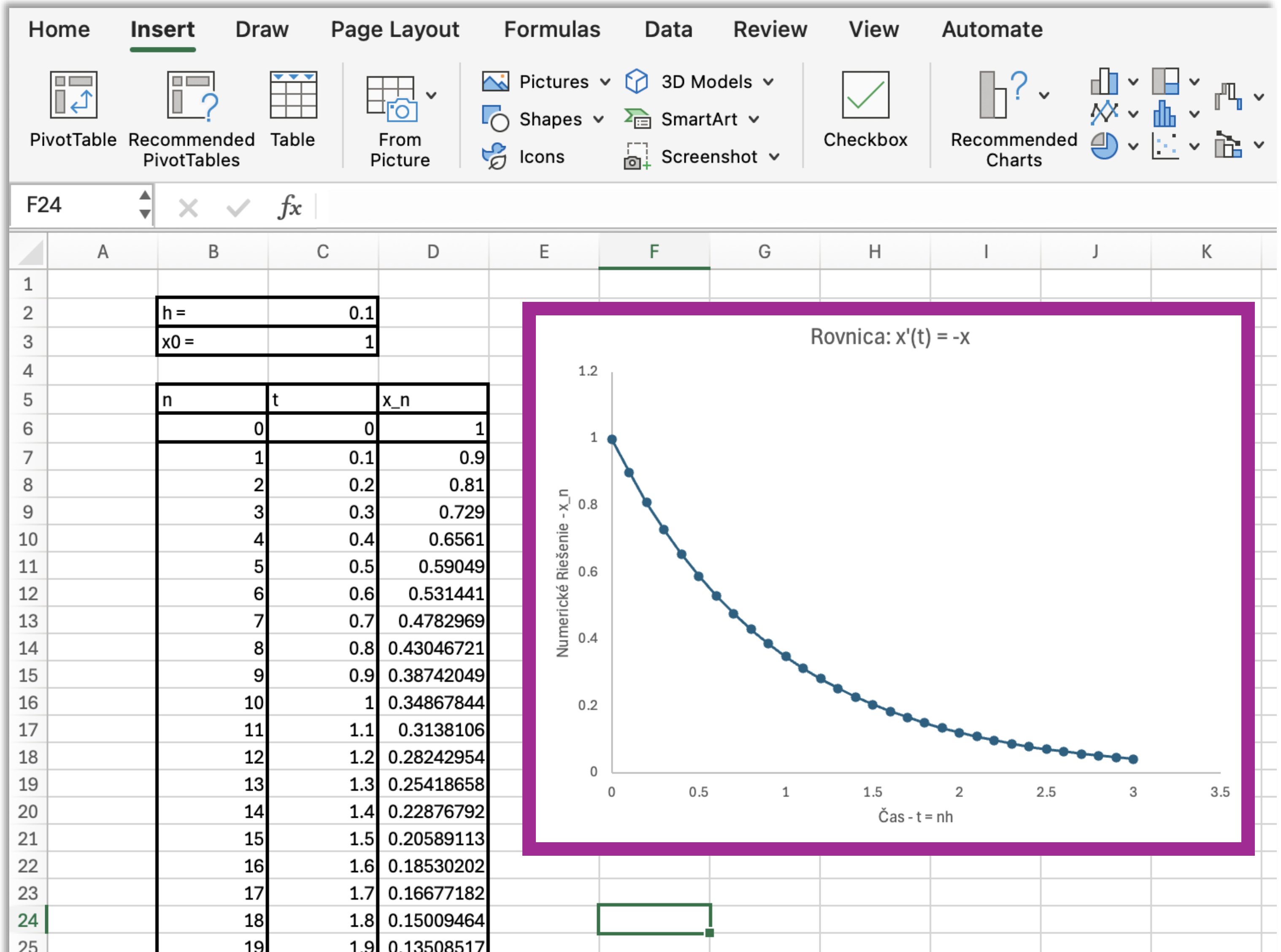


Riešenie:

$$x_n = x_{n-1} + h\,(-x_{n-1})$$

= D6 + C2 * (-D6)

Jednoduchá implementácia v Exceli

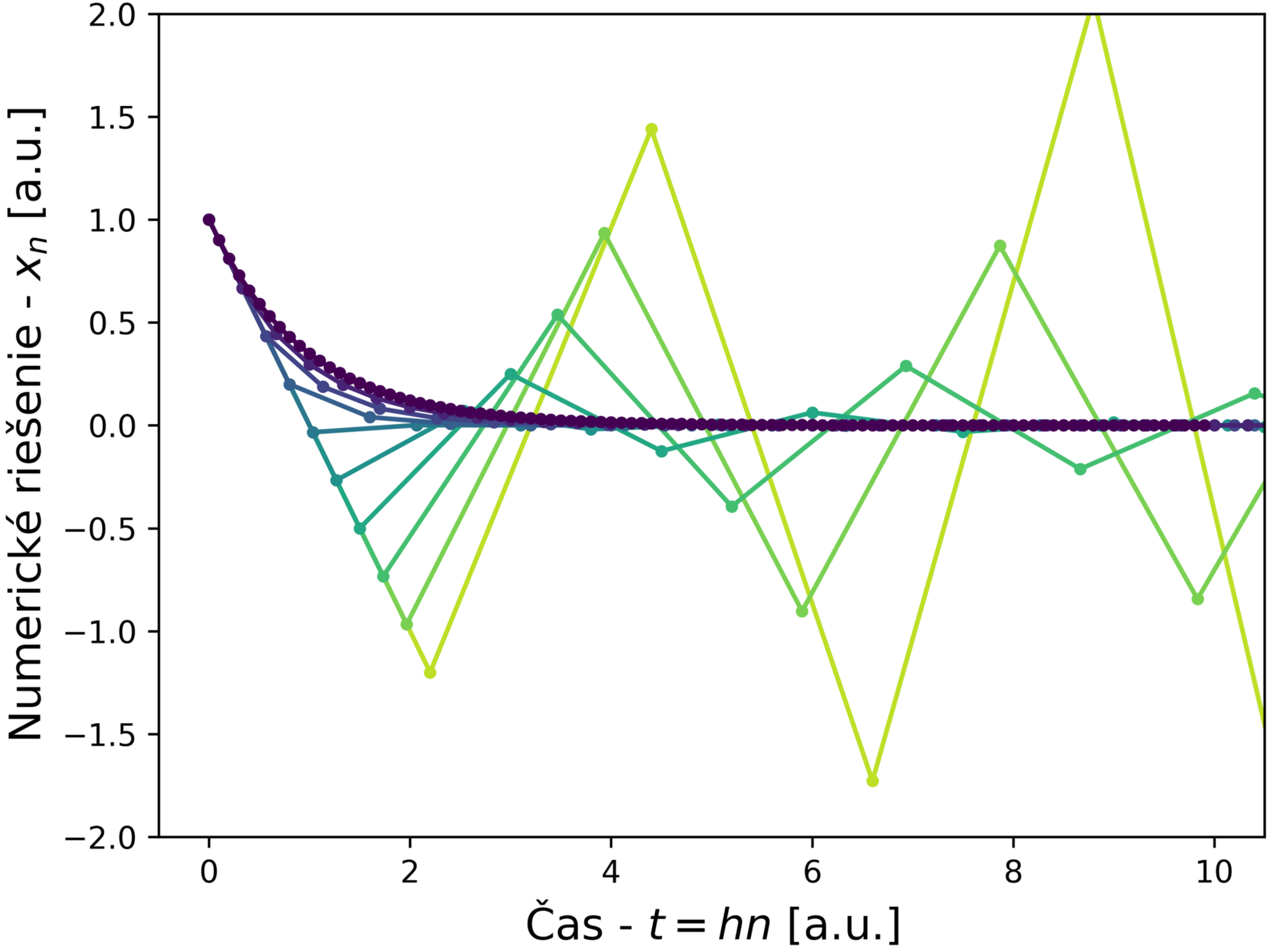


h =	0.1
x0 =	1

n	t	x_n
0	0	1
1	0.1	0.9
2	0.2	0.81
3	0.3	0.729
4	0.4	0.6561
5	0.5	0.59049
6	0.6	0.531441
7	0.7	0.4782969
8	0.8	0.43046721
9	0.9	0.38742049
10	1	0.34867844
11	1.1	0.3138106
12	1.2	0.28242954
13	1.3	0.25418658
14	1.4	0.22876792
15	1.5	0.20589113
16	1.6	0.18530202
17	1.7	0.16677182
18	1.8	0.15009464
19	1.9	0.13508517

Implementácia v Pythone

```
import numpy as np          # Knižnica numpy


N = 100                     # Počet krokov
h = 0.1                     # Časový krok

x = np.zeros(N)             # Pole s num. riešením


def f(x):                   # Zadefinuj funkciu f(x)
        return -x


x[0] = 1                    # Počiatočná podmienka

for i in range(N-1):        # Iteruj N krokov Eulerovej metódy
        x[i+1] = x[i] + h*f(x[i])
```


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
2
4
6
8
10
Numerické riešenie - x_n [a.u.]
Čas - $t = hn$ [a.u.]

Konvergencia, rád a stabilita

$$\boldsymbol{\tau}^h(\mathbf{x}_0) = \mathbf{x}_{1;\mathbf{x}_0} - \mathbf{x}(h;\mathbf{x}_0) \longleftarrow$$ **Lokálna** **chyba**

$$\mathbf{e}_N^h(\mathbf{x}_0) = \mathbf{x}_{N;\mathbf{x}_0} - \mathbf{x}(hN;\mathbf{x}_0) \longleftarrow$$ **Globálna chyba**

Numerická metóda je...

- **konzistentná** ak $\lim_{h\to 0} \frac{\boldsymbol{\tau}^h(\mathbf{x}_0)}{h} = 0.$
- **konvergentná** ak so zmenšujúcim sa krokom h sa numerické riešenie približuje exaktnému riešeniu.
- **rádu** p ak je lokálna chyba

$$\boldsymbol{\tau}^h(\mathbf{x}_0) = \mathcal{O}(h^{p+1}).$$

...potom sa dá ukázať $\mathbf{e}_n^h(\mathbf{x}_0) = \mathcal{O}(h^p)$. $\longleftarrow$ $\mathcal{O}(h^p)$ znamená, že pre $|h| \ll 1$ bude chyba rásť úmerne s h^p.

- **stabilná** ak je rast chyby potláčaný.

Explicitné a implicitné metódy

- **Explicitná Eulerova metóda:**

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(\mathbf{x}_n, t)$$

- **Implicitná Eulerova metóda:**

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(\mathbf{x}_{n+1}, t)$$

  - Je nutné riešiť rovnice pre $\mathbf{x}_{n+1}$
  - Pomáha so stabilitou

Jednoduché spôsoby riešenia rovníc tohoto druhu:

  - hľadanie pevného bodu iteráciou
  - scipy.optimize.fsolve

Harmonický oscilátor

Pohybová rovnica

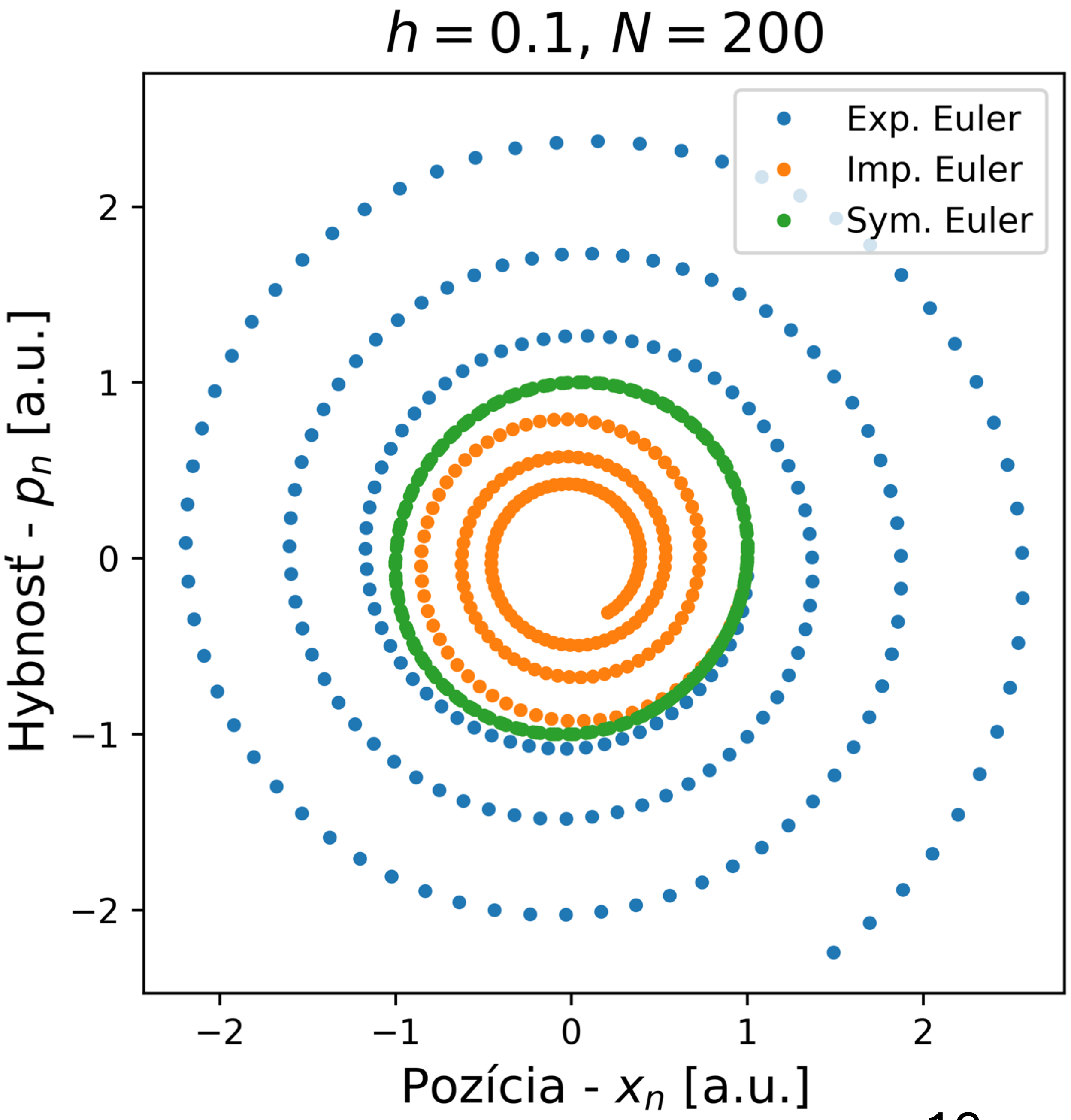
$$\frac{d^2x}{dt^2} = -x,$$

sa dá napísať ako

$$\frac{d}{dt}\binom{x}{p} = \binom{p}{-x}.$$

Platí:

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



Lepšie numerické metódy

- **Pre všeobecné problémy:**
  - Runge-Kutta metódy rádu 4, 5, ...
  - Metódy s adaptívnym krokom
  - BDF metódy

- **Pre pohybové rovnice**
  - Verletova metóda
    (Symplektická pre Hamiltonovské rovnice!)
  - Implicitná metóda stredného bodu
    (Najprv treba napísať rovnice v Hamiltonovskom tvare!)

- **Pre problémy so zákonom zachovania**
  - Projektívne metódy
  - Metóda diskrétnych gradientov

Knižnica scipy v Pythone

- scipy.integrate.solve_ivp

Installing User Guide API reference Building from source Development Release notes 1.14.1 (stable)

SciPy API > Integration and ODEs (`scipy.integrate`) > solve_ivp

On this page
`solve_ivp`

scipy.integrate.solve_ivp

`solve_ivp(fun, t_span, y0, method='RK45', t_eval=None, dense_output=False, events=None, vectorized=False, args=None, **options)`

Solve an initial value problem for a system of ODEs. [source]

This function numerically integrates a system of ordinary differential equations given an initial value:

```
dy / dt = f(t, y)
y(t0) = y0
```

Here t is a 1-D independent variable (time), y(t) is an N-D vector-valued function (state), and an N-D vector-valued function f(t, y) determines the differential equations. The goal is to find y(t) approximately satisfying the differential equations, given an initial value y(t0)=y0.

Rady a varovania

- **Najprvy sa zamysli!**
  - Hľadaj analytické riešenia
  - Fixné body
  - Linearizácia
- **Pracuj s rovnicou v správnom tvare!**
  - Pozor na nelineárne rovnice
  - Pozor na rovnice v neštandardtom tvare

napríklad:

$$\frac{1}{2}m\left(\frac{dx}{dt}\right)^2 + V(x) = E$$

- **Pamätaj na hromadenie chýb a rád!**
  - Skontróluj konvergenciu!
- **Zvoľ vhodnú metódu!**
- **Znovu sa zamysli!**

Priestor na otázky

Ďakujem!