

MATEMATICKÉ KYVADLO

J. PÚČEK, L. KOŠÁRKOVÁ, M. FUKSA

ABSTRACT. Zkoumáme řešení rovnici matematického kyvadla pomocí explicitní Eulerovy metody, Runge-Kuttovy metody a

CONTENTS

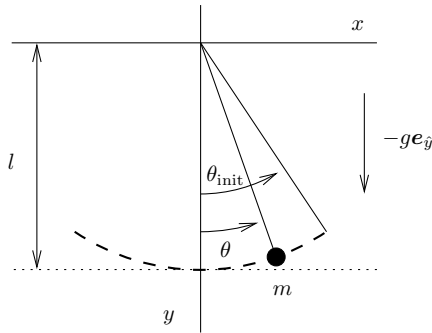
1. Úvod	1	4. Numerický výpočet	3
2. Fyzikální kontext	1	4.1. Explicitní Eulerova metoda	4
3. Perioda kmitů	2	4.2. Rungeova-Kuttova metoda	7
3.1. Přímý výpočet pomocí aproximace eliptického integrálu	2		

1. ÚVOD

V tomto dokukmentu budeme zkoumat pohyb matematického kyvadla, jehož dynamický systém je ukázkovým příkladem systému s periodickým pohybem. Hlavním předmětem zkoumání je, jak se mění chování numerického řešení vůči aproximaci analytického řešení za použití různých numerických metod.

2. FYZIKÁLNÍ KONTEXT

Kyvadlo o kterém budeme pojednávat je definováno jako na obrázku. Délka závěsu kyvadla je l a kyvadlo se hýbe v homogenním gravitačním poli s gravitačním zrychlením g .



Obrázek 1. Matematické kyvadlo.

Rovnice pro popis pohybu kyvadla vypadá následovně:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0. \quad (1)$$

Problém s rovnicí (1) je, že se jedná o rovnici *nelineární*. Pro nelineární rovnice je velmi obtížné najít explicitní řešení, vyjádřené pomocí elementárních funkcí.

Ve speciálním případě malých kmitů, když θ je dostatečně malá ve všech časech, můžeme použít následující aproximaci pro malé úhly $\sin \theta \approx \theta$ a převést naši nelineární rovnici (1) na rovnici lineární:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \theta = 0. \quad (2)$$

Hlavní otázkou je, zdali *dynamika* zlinearizovaného systému je vůbec podobná dynamice nelineárního systému. Ukazuje se, že určité porozumění dynamice plynoucí z rovnice (1) lze získat studováním aproximace (2) či studováním lepších aproximací stejného typu.

3. PERIODA KMITŮ

Jelikož neznáme explicitní vyjádření řešení diferenciální rovnice, tak to vypadá, že nelze nalázt funkci periody oscilace T . Ukazuje se ovšem, že nemusíme vyřešit rovnici, abychom našli periodu oscilací. Perioda oscilace pro linearizovaný systém (2) je

$$T = 2\pi\sqrt{\frac{l}{g}}. \quad (3)$$

Díky linearizaci se dá předpokládat, že perioda oscilace kyvadla vypuštěného s nulovou rychlostí z počáteční pozice danou počátečním úhlem θ_0 je

$$T = 2\pi\sqrt{\frac{l}{g}} (1 + a\theta_0^2 + b\theta_0^4 + \dots), \quad (4)$$

kde $a, b \in \mathbb{R}$ jsou reálné konstanty a navíc díky symetrii problému řada nemá liché členy.

3.1. Přímý výpočet pomocí aproximace eliptického integrálu. Vynásobením rovnice (1) členem $\frac{d\theta}{dt}$ a následnou úpravou dostáváme

$$\frac{d}{dt} \left(\frac{1}{2} \left(\frac{d\theta}{dt} \right)^2 - \frac{g}{l} \cos \theta \right) = 0. \quad (5)$$

Rovnice (5) navíc implikuje

$$\left(\frac{d\theta}{dt} \right)^2 = \frac{2g}{l} \cos \theta + C, \quad (6)$$

kde C je integrační konstanta, která je pevně dána počáteční podmínkou. Když $\theta = \theta_0$, tak chceme aby kyvadlo mělo nulovou úhlovou rychlost $\frac{d\theta}{dt}|_{\theta=\theta_0} = 0$ a tím dostáváme

$$\frac{d\theta}{dt} = \sqrt{\frac{2g}{l}} \sqrt{\cos \theta - \cos \theta_0}. \quad (7)$$

(předpokládejme, že $\frac{d\theta}{dt} \geq 0$.) Všimněme si, že (7) implikuje, že rychlost $\frac{d\theta}{dt}$ zůstává omezená po celou dobu. Nyní využijme větu o derivaci inverzní funkce a tím získáváme

$$\frac{dt}{d\theta} = \frac{1}{\sqrt{\frac{2g}{l}} \sqrt{\cos \theta - \cos \theta_0}}. \quad (8)$$

To jest diferenciální rovnice pro čas vyjádřený jako funkce úhlu. Tuto rovnici řešíme separací proměnných.

$$dt = \frac{d\theta}{\sqrt{\frac{2g}{l}} \sqrt{\cos \theta - \cos \theta_0}}. \quad (9)$$

Pokud zintegrujeme rovnici od $\theta = 0$ do $\theta = \theta_0$, tak na levé straně dostáváme $\frac{T}{4}$. Potom vzorec pro periodu oscilace je ve tvaru

$$\frac{T}{4} = \int_{\theta=0}^{\theta_0} \frac{d\theta}{\sqrt{\frac{2g}{l}} \sqrt{\cos \theta - \cos \theta_0}}. \quad (10)$$

Integrál na pravé straně nelze analyticky vypočítat, ale získat její aproximaci možné je. Přepišme si nejdřív integrál do uchopitelnější formy, kde využijeme, že platí známá rovnost $\sin^2 \frac{\alpha}{2} = \frac{1 - \cos \alpha}{2}$.

$$T = 2\sqrt{\frac{l}{g}} \int_{\theta=0}^{\theta_0} \frac{d\theta}{\sqrt{\sin^2 \frac{\theta_0}{2} - \sin^2 \frac{\theta}{2}}}. \quad (11)$$

Integrand je nyní typu $\frac{1}{\sqrt{1-y^2}}$ a dobrá substituce pro $\int \frac{dy}{\sqrt{1-y^2}}$ je $y = \sin \varphi$. Toho nyní využijme.

$$T = 2\sqrt{\frac{l}{g}} \int_{\theta=0}^{\theta_0} \frac{d\theta}{\sqrt{\sin^2 \frac{\theta_0}{2} - \sin^2 \frac{\theta}{2}}} = \left| \begin{array}{l} \sin \varphi = \frac{\sin \frac{\theta}{2}}{\sin \frac{\theta_0}{2}} \\ \cos \varphi d\varphi = \frac{1}{2} \frac{\cos \frac{\theta}{2}}{\sin \frac{\theta_0}{2}} d\theta \\ \sqrt{1 - \frac{\sin^2 \frac{\theta}{2}}{\sin^2 \frac{\theta_0}{2}}} d\varphi = \frac{1}{2} \frac{\sqrt{1 - \sin^2 \frac{\theta}{2}}}{\sin \frac{\theta_0}{2}} d\theta \\ \sqrt{\sin^2 \frac{\theta_0}{2} - \sin^2 \frac{\theta}{2}} d\varphi = \frac{1}{2} \sqrt{1 - \sin^2 \frac{\theta_0}{2}} \sin^2 \frac{\varphi}{2} d\theta \end{array} \right| = 4\sqrt{\frac{l}{g}} \int_{\varphi=0}^{\frac{\pi}{2}} \frac{d\varphi}{\sqrt{1 - k^2 \sin^2 \varphi}}, \quad (12)$$

Kde jsme definovali $k =_{\text{def}} \sin \frac{\theta_0}{2}$. Integrál je nyní ve tvaru *eliptického integrálu*, který je definovaný pro všechny $\varphi \in [0, \frac{\pi}{2}]$ a tedy dostáváme

$$T = 4\sqrt{\frac{l}{g}} \int_{\varphi=0}^{\frac{\pi}{2}} \frac{d\varphi}{\sqrt{1 - k^2 \sin^2 \varphi}}, \quad (13a)$$

kde

$$k =_{\text{def}} \sin \frac{\theta_0}{2}. \quad (13b)$$

Na integrant (13a) použijeme nyní jen Taylorův rozvoj a dostáváme

$$\frac{1}{\sqrt{1 - k^2 \sin^2 \varphi}} \approx 1 + \frac{1}{2} k^2 \sin^2 \varphi \approx 1 + \frac{1}{8} \theta_0^2 \sin^2 \varphi, \quad (14)$$

a následnou integrací a využití vztahů (14) a (13a) dostáváme

$$T = 4 \sqrt{\frac{l}{g}} \int_{\varphi=0}^{\frac{\pi}{2}} \frac{d\varphi}{\sqrt{1 - k^2 \sin^2 \varphi}} \approx 4 \sqrt{\frac{l}{g}} \int_{\varphi=0}^{\frac{\pi}{2}} \left(1 + \frac{1}{8} \theta_0^2 \sin^2 \varphi \right) d\varphi = 2\pi \sqrt{\frac{l}{g}} \left(1 + \frac{\theta_0^2}{16} \right). \quad (15)$$

Případně můžeme využít na výpočet řady software (například *WolframMathematica* nebo jiné). Konkrétně pro *WolframMathematica* můžeme využít příkaz

```
k=Sin[\[Theta]/2]
Series[1/(Sqrt[1-k^2*(Sin[x])^2]),{\[Theta],0,4}]
Integrate[4*Sqrt[l/g]*Series[1/(Sqrt[1-k^2*(Sin[x])^2]),{\[Theta],0,n}],{x,0,Pi/2}]
```

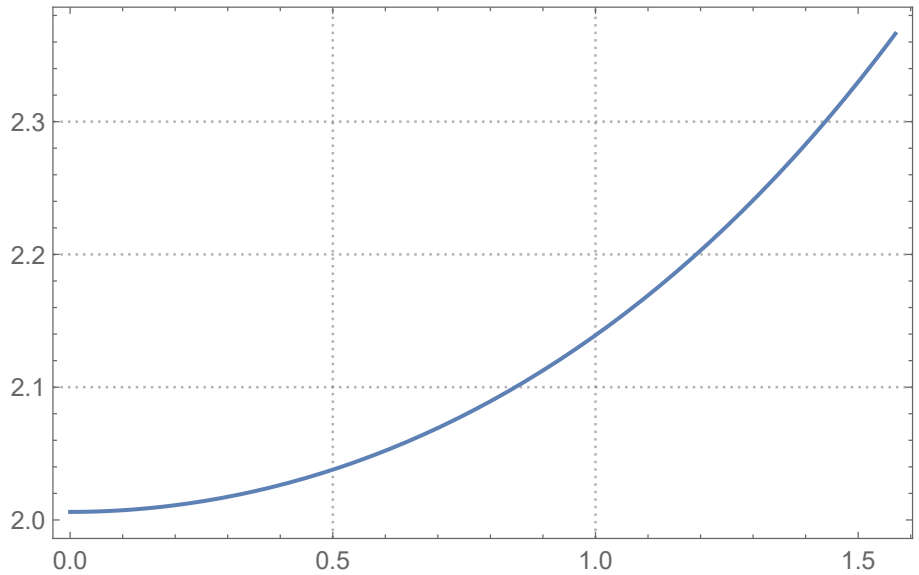
Kde za n volíme, do jakého řádu chceme aproximovat. Například pro $n = 10$ dostáváme

$$T \approx 2\pi \sqrt{\frac{l}{g}} \left(1 + \frac{1}{16} \theta_0^2 + \frac{11}{3072} \theta_0^4 + \frac{173}{737280} \theta_0^6 + \frac{22931}{1321205760} \theta_0^8 + \frac{1319183}{951268147200} \theta_0^{10} \right)$$

Do inetgrálu také můžeme dosadit konkrétní hodnoty a dostat dosti přesnou hodnotu periody v závislosti na čase. Výsledkem jsou diskrétní hodnoty periody, viz. tabulka (1), které jsme následně proložily polynomem (metoda nejmenších čtverců), viz. graf (1).

θ_0	T
0.001	2.00607
0.101	2.00735
0.201	2.01114
0.301	2.01749
0.401	2.02642
0.501	2.038
0.601	2.05231
0.701	2.06947
0.801	2.08959
0.901	2.11284
1.001	2.13938
1.101	2.16945
1.201	2.20327
1.301	2.24115
1.401	2.28339
1.501	2.33039

Tabulka 1.



Graf 1.

Navíc z fyzikální představy víme, že pro $\theta_0 = \pi$ se kyvadlo nahází ve stabilní poloze vratké, ve které se bylo mělo nacházet s periodou oscilace rovnu nekonečno za předpokladu, že na kyvadlo nepůsobí žádné vnější síly, což je systém popsáný právě (1). Dalo by se tedy předpokládat, že pro $\theta_0 \rightarrow \pi$ se dá očekávat, že $T \rightarrow \infty$.

4. NUMERICKÝ VÝPOČET

Jak již bylo řečeno, je obtížné najít explicitní řešení rovnice (1) ve tvaru elementárních funkcí. Programy, jako například *Wolfram Mathematica*, který zde budeme využívat, ovšem umožňují rychlé nalezení numerického řešení a to hned pomocí několika numerických metod s nastavitelnými parametry.

Stěžejní příkaz pro výpočet numerického řešení naší diferenciální rovnice (1) v softwaru *Wolfram Mathematica* bude:

```
NDSolve[{\[Theta]''[t] + (g/l)*Sin[\[Theta][t]] == 0, \[Theta][0] == \[Theta]_0, \[Theta]'[0] == 0}, \[Theta], {t,0,n}]
```

Příčemž po celý dokument budeme ve výpočtech uvažovat délku kyvadla $l = 1$, $m = 1$ a gravitační zrychlení $g = 9.81$.

Hlavním cílem bude příkazem *NDSolve* numericky určit periodu kmitů systému popsáného rovnicí (1), zjistit numerické řešení rovnice (1), prozkoumat její zachovávání/nezachovávání celkové mechanické energie během děje a všechny tyto body

porovnat s reálným chováním matematického kyvadla. Výpočty budeme provádět postupně pomocí: explicitní Eulerovy metody, Rungeovy-Kuttovy metody a Symplektické Rungeovy-Kuttovy metody

4.1. Explicitní Eulerova metoda. Explicitní Eulerova metoda je jedna z jednodušších, ale ukážeme si, že také jedna z nejméně spolehlivých.

4.1.1. Princip explicitní Eulerovy metody. Mějme diferenciální rovnici $\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0$ s počátečními podmínkami v čase $t_0 = 0$: $\theta(t_0) = \theta_0$, $\frac{d\theta}{dt}(t_0) = 0$. Pak řešení v čase $t_i = i\Delta t$, $i = 1, \dots, n$ (zde ve vysvětlení principu předpokládáme pro zjednodušení rovnoměrné dělení a tedy Δt je časový krok metody) získáme ze soustavy rovnic:

$$\theta_{i+1} = \theta_i + v_i \Delta t \quad (16)$$

$$v_{i+1} = v_i - \frac{g}{l} \sin \theta_i \Delta t \quad (17)$$

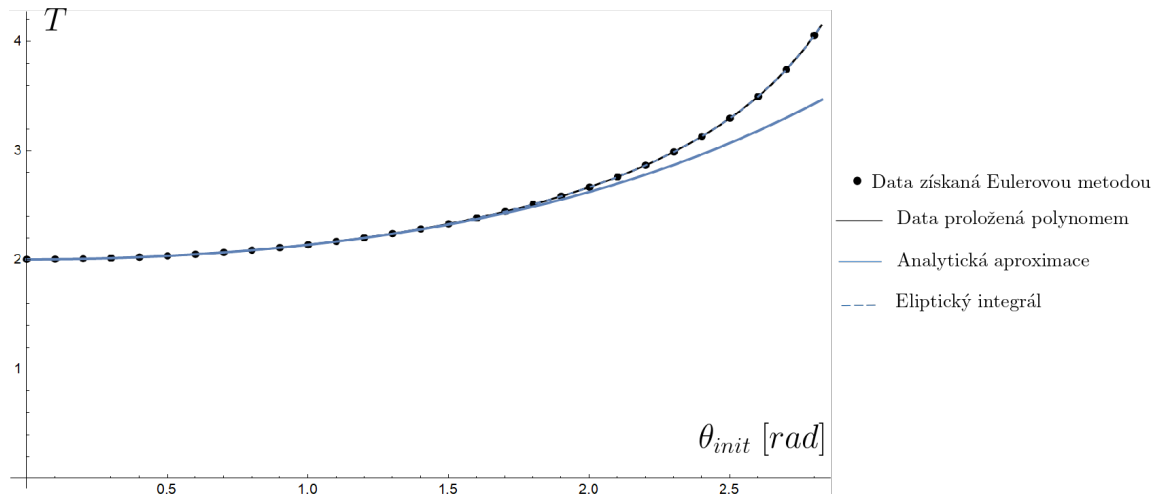
, kde $v_i = \frac{d\theta_i}{dt}$, $\theta_i = \theta(t_i)$, $\theta_{i+1} = \theta(t_{i+1})$, $v_i = v(t_i)$, $v_{i+1} = v(t_{i+1})$

4.1.2. Aproximace periody. Nejdříve zkoumejme periodu T , kterou získáme jako 4krát dobu, za kterou se dostane kyvadlo z úhlu θ_0 (počáteční výchylky) do úhlu $\theta = 0$.

O takovýto výstup můžeme požádat *Mathematicu* ve tvaru:

```
NDSolve[{θ'[t] + g/l*Sin[θ[t]] == 0, θ[0] == θ_0, θ'[0] == 0,
WhenEvent[θ[t] == 0, {Print[t*4], "StopIntegration"}]},
θ, {t, 0, n}, Method -> "ExplicitEuler"]
```

Důležitou vlastností matematického kyvadla je závislost periody T na počáteční výchylce θ_0 , tuto závislost ukazuje Graf 2.



Graf 2. Závislost T na θ_0 pro různé metody výpočtu

Tečky, které jsme proložili polynomem, znázorňují hodnotu periody získané pomocí Eulerovy metody. Tyto data jsme porovnali s analytickou aproximací pomocí rovnice (15) do 4. řádu a s výpočtem periody přes eliptický integrál, který by měl dávat přesné hodnoty. V tomto případě se zdá být explicitní Eulerova metoda velmi přesná. To je dáno tím, že numerické řešení pomocí explicitní Eulerovy metody se s přesným řešením rozbíhá až pro vyšší časy. Zatímco výsledky analytické aproximace jsou pro větší počáteční výchylky už i v malých časech poměrně nedostatečné.

4.1.3. Řešení rovnice $\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0$. Na grafu 3 je znázorněno řešení pohybu kyvadla vychýleného z počáteční výchylky $\theta_0 = 2\pi/3$. Modré řešení je výpočet pomocí Eulerovy metody:

```
euler = NDSolve[{θ'[t] + g/l*Sin[θ[t]] == 0, θ[0] == θ_0, θ'[0] == 0,
θ, {t, 0, n}, Method -> "ExplicitEuler"]
```

zároveň je zde oranžově vykresleno i řešení pomocí příkazu *NDSolve*, kde ale *Mathematica* vybere takovou numerickou metodu, aby bylo řešení co nejpřesnější a pro nás tedy teď bude dostatečně dobrou aproximací přesných kmitů kyvadla v závislosti na čase.

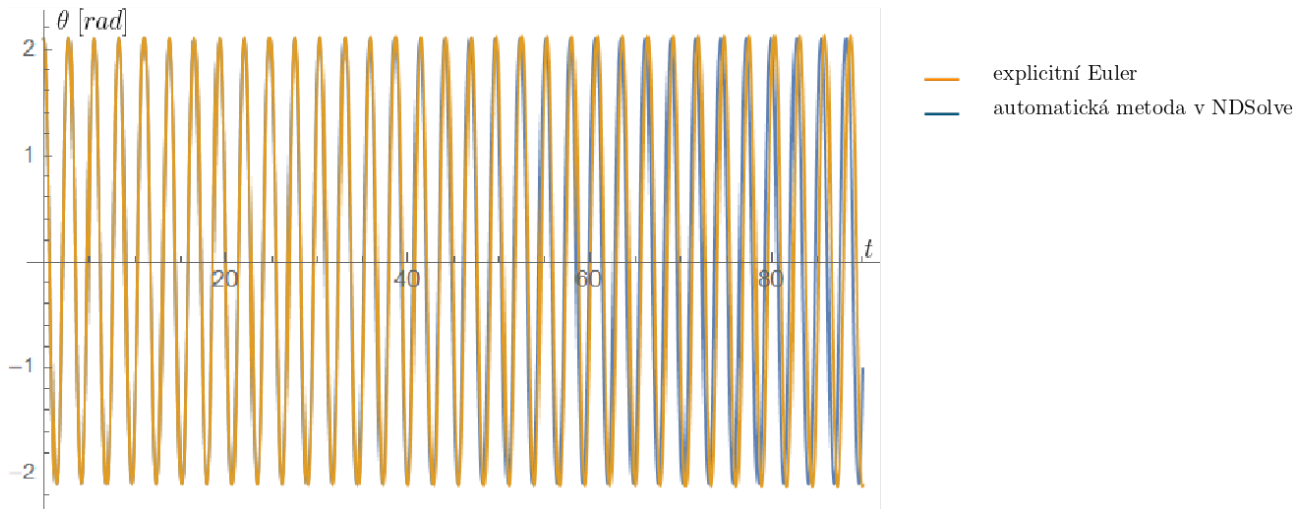
Příkaz kterým toho dosáhneme zní:

```
auto = NDSolve[{θ'[t] + g/l*Sin[θ[t]] == 0, θ[0] == θ_0, θ'[0] == 0,
θ, {t, 0, n}]
```

Abychom tyto metody mohli porovnat, dáme je do jednoho grafu příkazem:

```
Plot[Evaluate[{θ[t] /. euler, θ[t] /. auto}], {t, 0, 100}, PlotRange -> All]
```

Z grafu je patrné, že Eulerova metoda pro zvyšující se t začíná předbíhat druhé řešení a zároveň se u této metody zvyšuje amplituda. Důvodem je v tomto případě neschopnost Eulerovy metody zachovávat celkovou mechanickou energii systému, což si ukážeme ještě lépe za chvíli.



Graf 3. explicitní Euler vs automatická metoda v NDSolve

4.1.4. *Energie systému.* První integrál rovnice (1) je ve tvaru

$$\frac{1}{2} \left(\frac{d\theta}{dt} \right)^2 - \frac{g}{l} \cos \theta = E. \quad (18)$$

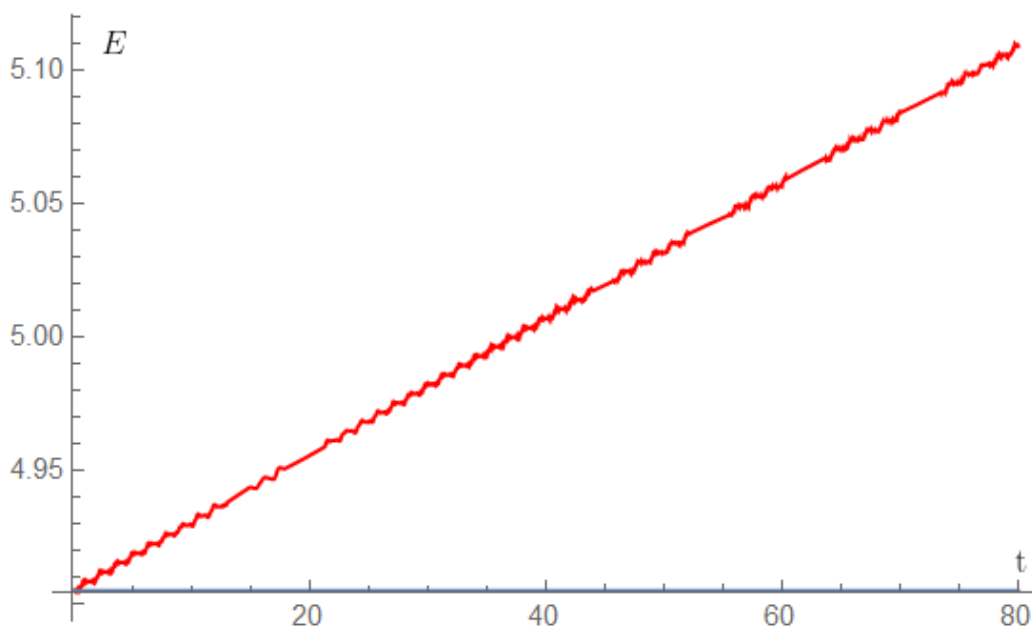
Což říká, že podél řešení je první integrál roven konstantě E , neboli součtu kinetické a potenciální energie, tedy dohromady celkové mechanické energii systému.

Pro explicitní Eulerovu metodu tato rovnost ovšem neplatí. Volme například $\theta_0 = 2\pi/3$. Pro tuto počáteční výchylku by se měla zachovávat podél celého řešení celková energie $E = 4.905$. Eulerova metoda ale konstantní hodnotu E nezachovává, naopak s rostoucím časem energii čím dál víc zvyšuje.

Graf se dá vykreslit serií příkazů:

```
p1 = Plot[Evaluate[{((\[Theta]')^2)/2 - (g/l) Cos[\[Theta][t]]} /. v], {t,0,80},
PlotStyle -> Red, PlotRange -> All];
p2 = Plot[Evaluate[-g/l *Cos[\[Theta]_0]], {t,0,80}, PlotRange -> All];
Show[p1, p2, PlotRange -> All]
```

, kde za v dosadíme získané numerické řešení pomocí Eulerovy metody.



Graf 4. Energie v závislosti na čase pro Eulerovu metodu.

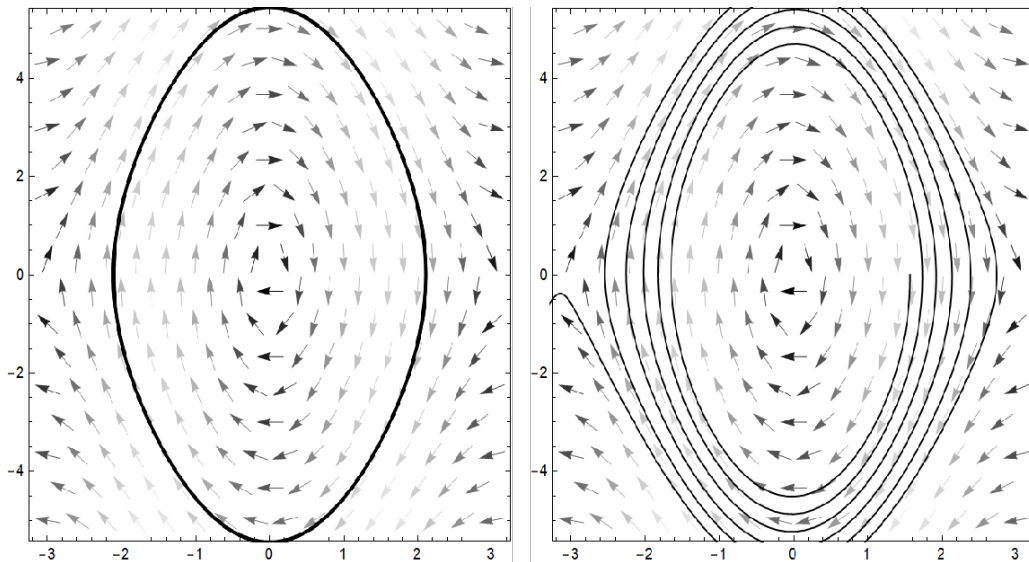
Jak moc Eulerova metoda nezachovává celkovou energii systému lze vidět z trajektorie ve fázovém prostoru.

Na levém obrázku je výpočet s počáteční výchylkou $\theta_0 = \pi/2$, kde *Mathematica* přikážeme počítat Eulerovou metodou, ale nenastavíme žádné další upřesňující details. *Mathematica* tedy sice bude počítat s obecně málo přesnou Eulerovou metodou, ale automaticky si nastaví takové parametry, aby byl výpočet co nejpřesnější.

Během každé periody (což odpovídá jednomu oběhu ve fázovém prostoru) se energie pomalu zvyšuje a křivka se tak v průběhu oběhu zvětšuje. Tím pádem se ale v bodě $\{\pi/2, 0\}$ křivka neuzavře a postupně po několika periodách vykreslí spirálu a protože se pro tento systém energie v průběhu děje zvyšuje, bude spirála pokračovat směrem ven (výchylka kmitů se bude zvyšovat).

Jelikož ale toto řešení není až tak nedostatečně nepřesné, vidíme místo spirály zhušťující se křivku.

Na pravém obrázku *Mathematica* přikážeme, aby Eulerovu metodu počítala s počátečním krokem $1/99$ (za metodu v příkazu *NDSolve* přidáme: *StartingStepSize* $\rightarrow 1/99$). Tím se zvýší nepřesnost výpočtu a systém začne nabírat energii zřetelně rychleji.



Nalevo: **Obrázek 2.** automaticky zabudovaná Eulerova metoda

Napravo: **Obrázek 3.** Eulerova metoda s *StartingStepSize* $\rightarrow 1/99$

4.2. Rungeova-Kuttova metoda. Rungeova-Kuttova metoda je jednokroková metoda pro numerické řešení obyčejných diferenciálních rovnic. Pro rovnici ve tvaru $y' = f(t, y)$ s okrajovou podmínkou $y(t_0) = y_0$, vhodně zvolený krok $h > 0$ a $n = 0, 1, 2, 3, \dots$, je $n + 1$ krok metody definován následovně.

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$t_{n+1} = t_n + h$$

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

4.2.1. *Aproximace periody.* V následujícím jsme použily program *Mathematica* k nalezení numerické aproximace periody kyvadla v závislosti na počáteční podmínce. V první části kódu jsme si zadefinovaly funkci:

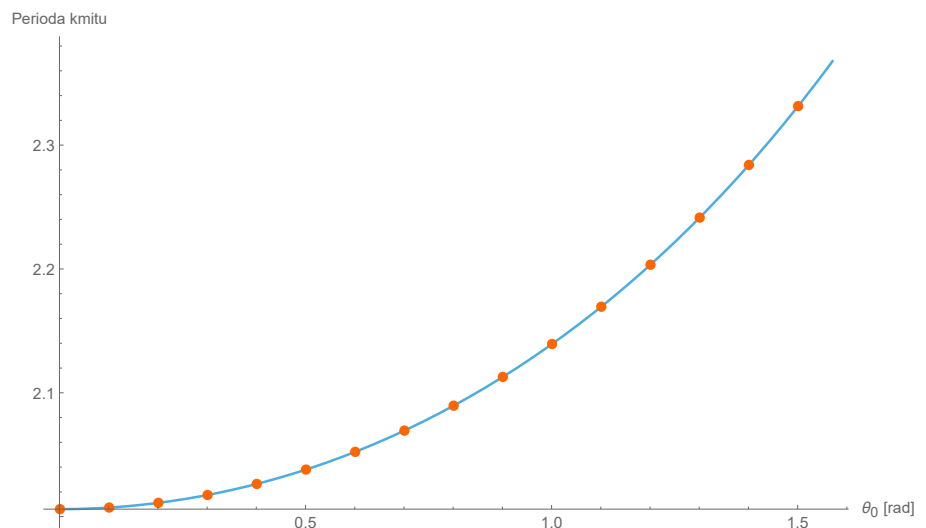
```
Period[\[Theta]_0_] := (time := 0;
NDSolve[{\[Theta]''[t] + g/l Sin[\[Theta][t]] == 0, \[Theta][0] == \[Theta]_0, \[Theta]'[0] == 0,
WhenEvent[\[Theta][t] == 0, {time = t, "StopIntegration"}]}, \[Theta], {t, 0, 3 },
Method -> "ExplicitRungeKutta"];
4*time)
```

Která na vstupu přijímá počáteční výchylku kyvadla a obratem nám vrací periodu kmitu. Pro naše potřeby jsme uvažovali tyto hodnoty konstant: $g = 9.81, l = 1$.

Poté si stačilo nechat příkazem `Table[]` s použitím naší funkce vytvořit tabulku hodnot (2). Následně jsme hodnoty metodou nejmenšího čtverce proložili polynomem a ten, společně s diskretními hodnotami, vykreslili do grafu (5).

θ_0	T
0.001	2.00607
0.101	2.00735
0.201	2.01114
0.301	2.01749
0.401	2.02642
0.501	2.038
0.601	2.05231
0.701	2.06947
0.801	2.0896
0.901	2.11285
1.001	2.13942
1.101	2.16953
1.201	2.20344
1.301	2.24148
1.401	2.28401
1.501	2.33149

Tabulka 2.



Graf 5.

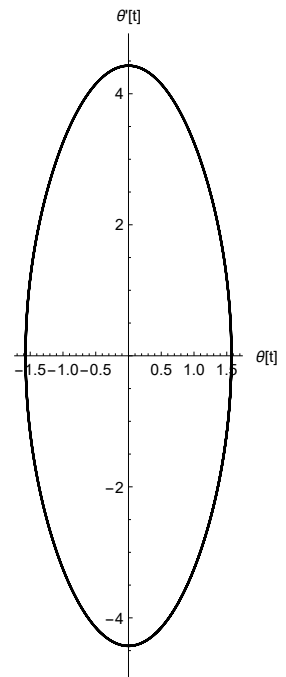
4.2.2. *Energie systému.* V této sekci zkoumáme jak dobře explicitní Rungeova-Kuttova metoda zachovává celkovou energii systému. Konkrétně se budeme zabývat počáteční podmínkou: $\theta_0 = \pi/2$. První z vizualizací bude tzv. fázový diagram, což je parametrický graf vývoje celkové energie v čase. Na grafech tohoto typu lze velmi přímočaře identifikovat co se s energií děje. Pokud graf vypadá jako spirála, která se v čase vyvíjí směrem dovnitř/ven tak celková energie systému klesá/stoupá, ale pokud se jedná o uzavřenou křivku, tak je celková energie konstantní.

Pro vykreslení diagramu bylo potřeba si rovnici (2) převést na systém rovnic prvního řádu a následně ho numericky vyřešit pro funkce $x[t]$ a $y[t]$. Kde funkce x reprezentuje funkci $\theta[t]$ a funkce y její první derivaci (rychlost) $\theta'[t]$. Příslušný kód v Mathematice vypadá následovně:

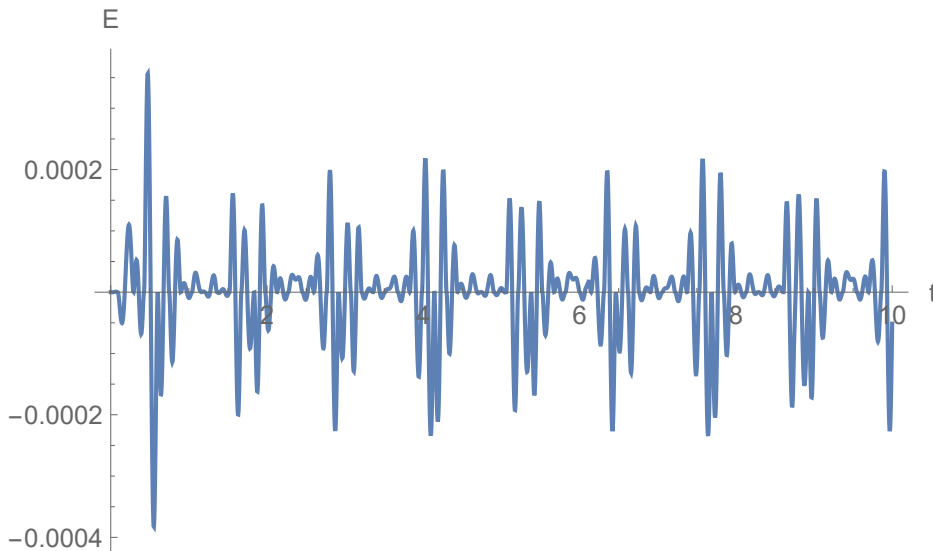
```
system := NDSolve[{x'[t] == y[t], y'[t] == -g/l Sin[x[t]], x[0] == Pi/2,
y[0] == 0}, {x[t], y[t]}, {t, 0, 10}, Method -> "ExplicitRungeKutta"]
```

Plod naší snahy můžete nahlédnout na obrázku (4), kde nám vyšel fázový diagram uzavřený. Což znamená, že na první pohled je Rungeova-Kuttova metoda dobrá v zachovávání celkové energie.

K tomu jak moc dobrá v tomto metoda je, nám pomůže druhá vizualizace, kterou bude graf hodnoty celkové energie v čase. Celkovou energii získáme sečtením energie kinetické a potenciální, což v našem případě konkrétně vypadá následovně: $E = \frac{1}{2}(\theta')^2 - \frac{g}{l}\cos\theta$. Takže stačí Rungeovo-Kuttovou metodou získat funkci $\theta[t]$, tu dosadit do výrazu pro $E[t]$ a funkci energie si nechat vykreslit v čase. Pro náš případ je výsledkem graf (6), z kterého jde vidět, že celková energie je skoro konstantní až na výkyvy v řádu ± 0.0004 .



Obrázek 4. Fázový diagram.



Graf 6. Celková energie v čase