



EUROPASKOLAN STRÄNGNÄS

GYMNASIEARBETE

Dubbelpendeln och kaosteori

*Oscar Landberg
KFSCI23b*

Handledare:
Erik Waltersson

Reviderat första utkast inlämnad: 2026-02-18

Abstract

The double pendulum has long been considered a simple, yet well-defined chaotic system, which this report has intended to study by using numerical simulation. Therefore, Lagrangian mechanics were used to derive the equations of motions, which were numerically solved and simulated by using Eulers method and the Runge-Kutta method of the 4th order (the RK4 method). The double pendulum was then analyzed for a wide variety of starting angles, and it was shown that the RK4 method gave much more accurate simulations compared to Eulers method. Moreover, the simulations demonstrated the general chaotic nature of the double pendulum, which gave the conclusion that the level of chaos tended to increase as we increased the starting energy in the system.

Innehåll

Inledning	1
1.1 Introduktion och bakgrund	1
1.2 Syfte och frågeställning	2
Teoretisk bakgrund	3
2.1 Lagrange-mekanik	3
2.1.1 Principen om minsta verkan	3
2.2 Dubbelpendeln	4
2.2.1 Härledning av dubbelpendels rörelseekvationer	4
2.3 Teori inför numerisk simulering	7
2.3.1 Eulers stegmetod	7
2.3.2 Runge-Kuttametoden (RK4-metoden)	8
Metod/Numerisk simulering	10
3.1 Behandling av rörelseekvationerna för simulering	10
3.2 Numerisk simulering	11
Resultat	13
Diskussion	20
5.1 Slutsats	22
Litteraturförteckning	23

Appendix	I
A - Härledning av Euler-Lagrange ekvationen	I
B - Kod för simulering av dubbelpendeln	II

Inledning

1.1 Introduktion och bakgrund

Pendeln är något som alltid har fascinerat mänskligheten genom tiderna. Redan under det första århundradet lyckades de gamla kineserna att utveckla en seismograf med hjälp av pendeln, vars funktion var att aktivera ett säkerhetssystem vid jordbävningar [1]. Inte minst används också pendlar än idag; Mora-klockors tidshållning bygger på en svängande pendel, medan den klassiska metronomens tickande styrs av en inverterad variant. Det är därmed tydligt hur pendeln är relevant än idag.

Det är därför inte förvånande att pendeln länge har varit en central del av fysikundervisningen, inte minst på gymnasiet. De flesta före detta naturvetenskapliga gymnasieelever känner säkert igen att majoriteten av de enkla pendlarna kan beskrivas som en harmonisk svängningsrörelse, och att formeln för en pendels svängningstid är $T = 2\pi\sqrt{\frac{l}{g}}$.

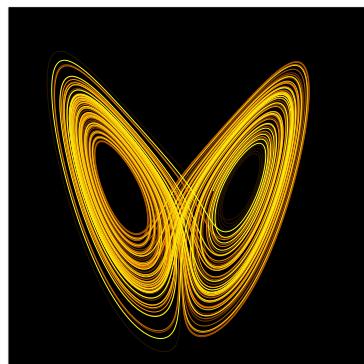
Tyvärr så ingår det ingen större fördjupning för pendlar inom gymnasiestudierna¹, och mycket av det som lärs ut om pendlar gäller bara om begynnelsevinkeln θ är relativt liten. Det är bara när detta villkor är uppfyllt som pendlar kan beskrivas som en harmonisk svängningsrörelse, inte minst gäller detta även för svängningstidsformeln ovan.

På så vis kan en till synes ”enkel” pendel bli förhållandevise komplicerad, långt utanför gymnasiefysikens gränser. Det finns dock flera olika sätt att vidareutveckla problemet, bland annat går det att skapa en så kallad *dubbelpendel* genom att koppla två enkla pendlar ihop. Mycket forskning har redan skett på dubbelpendeln, och idag används dubbelpendeln ofta som klassiskt exempel inom *kaosteorin* [2].

Kaosteori är nämligen det tvärvetenskapliga området inom fysiken och matematiken som studerar deterministiska system vars långsiktiga beteende blir praktiskt oförutsägbara (CITE). En av de tidiga pionjärerna inom kaosteori var metrologen Edward Lorenz, som år 1961 försökte simulera väderprognoser tillsammans med Ellen Fetter och Margaret Hamilton. Det var här som Lorenz och hans kollegor upptäckte att minimala skillnader i begynnelsevillkoren för simuleringen orsakade

¹Det går även att argumentera för motsatsen, det kanske är bättre att lämna det åt universitetsstudenter...

drastiskt olika väderprognoser. Lorenz, som blev väldigt fascinerat av atmosfärens kaotiska system, bestämde sig för att förenkla hans modeller till ett system av tre linjära ordinära differentialekvationer. Lösningen till detta system är vad som grafiskt brukas kallas ”Lorenz-atraktorn”, se figur 1.1. Lorenz lyckades visa att denna atraktor kommer alltid att divergera och att lösningarna till ekvationssystemet aldrig kommer att återupprepa sig. Dessutom, det som Lorenz blev mest känd för, var självaste formen av en fjäril i atraktorn. Detta, tillsammans med titeln på ett av hans tal² från 1972, är ursprunget till det populärvetenskapliga uttrycket ”fjärilseffekten” – att fladdrandet av en fjärils vinge i Brasilien kan orsaka en tromb i Texas³.



Figur 1.1: Exempel på en Lorenz-atraktor. Notera utseendet av en fjäril. Bildkälla: *Lorenz system simulation*. Bild av Wikimol, 2005, Wikimedia Commons.

Även om en dubbelpendlar kanske inte orsakar tromber i Texas, gäller principen om fjärilseffekten och kaos fortfarande: dubbelpendlar är extremt känsliga för sina begynnelsevillkor. Många experiment har gjorts på dubbelpendlar, och ofta betraktas dubbelpendeln som en introduktion till kaotiska system. Däremot går det att analysera dubbelpendeln vidare, och frågan kvarstår om går det att hitta något mönster i dess kaotiska natur.

1.2 Syfte och frågeställning

Syftet med detta arbete är därmed att undersöka dubbelpendelns kaotiska egenskaper. Genom att härleda dubbelpendelns rörelseekvationer med Lagrange-mekanik, kommer vi numeriskt simulera en dubbelpendel genom Eulers-stegmetod och Runge-Kuttametoden av fjärde ordningen, och sedan dra analyser utifrån dess egenskaper. Därmed har gymnasiearbetet gått ut på att besvara frågeställningarna: ’Hur och vad är den bästa metoden för att simulera en dubbelpendel?’, samt ’Vilka slutsatser går det att dra utifrån dubbelpendelns kaotiska egenskaper?’.

²Ursprungstitel på engelska ’Predictability: Does the Flap of a Butterfly’s Wings in Brazil Set Off a Tornado in Texas?’. Talet hölls inför det 139:de årsmötet av American Association for the Advancement of Science (AAAS) [2].

³För vidare läsning om Lorenz attraktorn, se [2, 3].

Teoretisk bakgrund

2.1 Lagrange-mekanik

För att lösa mer komplexa problem, som exempelvis en beskrivning dubbelpendeln, är det oftare lättare att använda Lagrange-mekanik för att beskriva dess rörelse. Även om det är teoretiskt sätt möjligt att beskriva en dubbelpendels rörelse med klassisk newtonsk mekanik, kan det fort bli mödosamt och onödigt krångligt. Lagrange-mekanik är egentligen bara en annan matematisk metod för att beskriva omvärlden, vilket i en viss typ av problem, blir betydligt lättare att lösa.

2.1.1 Principen om minsta verkan

Lagrange-mekanik bygger sin grund på *principen om minsta verkan*, eller ibland även kallad *principen om stationär verkan* [4]. Denna princip säger att ett objekt kommer alltid att sträva efter att färdas den väg som minimerar den fysikaliska *verkan* [4, s.2]. Givet att vi har ett objekt Q som rör sig längs $x(t)$, där Q har startpunkten $x(t_1)$ och slutpunkten $x(t_2)$, samt att $T(t)$ och $V(t)$ är objektets rörelse respektive kinetiska energi, definieras verkan S inom tidsintervallet $t_1 \leq t \leq t_2$ som [4, s.10]:

$$S = \int_{t_1}^{t_2} (T(t) - V(t)) dt. \quad (2.1)$$

Denna storhet har enheten [Js] och har dimensionerna (Energi \times Tid) [5, s.221]. Själva differensen $T(t) - V(t)$ visar sig vara så relevant inom fysiken, att den har fått namnet 'the Lagrangian' och brukar betecknas \mathcal{L} [5, s.218], det vill säga:

$$\mathcal{L} = T - V \quad (2.2)$$

Principen om minsta verkan säger att fysikens lagar alltid kommer att sträva efter att förminka ekvation (2.1)¹. Detta är så fysikaliskt grundläggande, att nästan all fysik kan härledas ur detta grundläggande antagande [4].²

¹Egentligen, rent matematiskt, strävar verkan S att hitta ett stationärt värde av S , det vill säga en lokal extrem- eller terrasspunkt till grafen av S . Detta är anledningen varför principen även kallas för *principen om stationär verkan* [5, s.222].

²Men varför vill naturen alltid förminka S ? Problemet med att besvara denna fråga är att svaret innefattar kvantmekanik, vilket ligger utanför denna rapportsyfte/omfattning. För en enklare förklaring om varför principen om minsta verkan, se [5, s. 225].

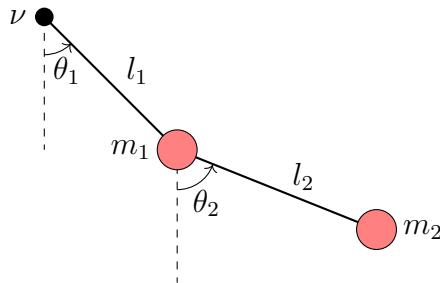
För att bestämma ett objekts väg som minimerar verkan S används *Euler-Lagrange ekvationen* [5, s. 222], se ekvation (2.3). I ekvation (2.3) är $i = 1, 2, 3, \dots$, och normalt sätt betecknar de olika koordinataxlar, eller i detta fall olika vinklar. Härledning av (2.3) ges i appendix A.

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0, \quad (2.3)$$

2.2 Dubbelpendeln

Om två vanliga enkla pendlar är sammankopplade sägs de bilda en dubbelpendel, se figur 2.1. Den första pendeln består av en punktformig massa m_1 som är kopplad till en masslös pinne med längden l_1 , vilket sitter i en friktionslös rotationspunkt ν . Den andra pendeln består också av en punktformig massa m_2 som är kopplad till en masslös pinne med längden l_2 . Denna andra pendeln har sin friktionslösa rotationspunkt tillkopplad i m_1 . Den enda kraften som verkar på respektive massa är tyngdkraften och alla annan friktion från t.ex luftmotstånd försummas.

Låt vinklarna θ_1, θ_2 vara vinklarna mellan den lodräta linjen från rotationspunkten och den faktiska positionen av pendeln, se figur 2.1. Vinklarna θ_1 och θ_2 är inte begränsade mellan $-\pi \leq \theta_{1,2} \leq \pi$, dvs de kan rotera fritt hur många varv som helst runt rotationspunkterna. Det antas även här att varje pendel inte kan kollidera med varandra.



Figur 2.1: Schematisk figur över en matematisk dubbelpendel.

2.2.1 Härledning av dubbelpendels rörelseekvationer

Härledningen utgår från figur 2.1, där dubbelpendeln ritats in i ett koordinatsystem där origo utgår från vrädpunkten ν . De punktformiga massorna m_1 och m_2 har koordinaterna (x_1, y_1) respektive (x_2, y_2) .

Vi kan beskriva punkterna x_1 , x_2 , y_1 och y_2 genom trigonometriska samband enligt:

$$x_1 = l_1 \sin \theta_1 \quad (2.4)$$

$$y_1 = -l_1 \cos \theta_1 \quad (2.5)$$

$$x_2 = l_1 \sin \theta_1 + l_2 \sin \theta_2 \quad (2.6)$$

$$y_2 = -l_1 \cos \theta_1 - l_2 \cos \theta_2. \quad (2.7)$$

Eftersom tidsderivatan av position är hastighet, kan vi få hastigheterna \dot{x}_1 , \dot{y}_1 , \dot{x}_2 och \dot{y}_2 enligt:

$$\dot{x}_1 = \dot{\theta}_1 l_1 \cos \theta_1 \quad (2.8)$$

$$\dot{y}_1 = \dot{\theta}_1 l_1 \sin \theta_1 \quad (2.9)$$

$$\dot{x}_2 = \dot{\theta}_1 l_1 \cos \theta_1 + \dot{\theta}_2 l_2 \cos \theta_2 \quad (2.10)$$

$$\dot{y}_2 = \dot{\theta}_1 l_1 \sin \theta_1 + \dot{\theta}_2 l_2 \sin \theta_2. \quad (2.11)$$

Vi kan därmed definiera dubbelpendelns potentiella energi V som summan av den potentiella energin för respektive massa. Detta ger:

$$V = m_1 g y_1 + m_2 g y_2$$

$$(2.5), (2.7) \Rightarrow V = -(m_1 + m_2) g l_1 \cos \theta_1 - m_2 g l_2 \cos \theta_2. \quad (2.12)$$

Vi kan också bestämma dubbelpendelns kinetiska energi T enligt:

$$T = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2$$

$$= \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2)$$

$$(2.8) - (2.11) \Rightarrow = \frac{1}{2} m_1 \left(\dot{\theta}_1^2 l_1^2 \cos^2 \theta_1 + \dot{\theta}_1^2 l_1^2 \sin^2 \theta_1 \right) + \frac{1}{2} m_2 \left(\dot{\theta}_1^2 l_1^2 \cos^2 \theta_1 \right.$$

$$+ 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_2 \cos \theta_1 \cos \theta_2 + \dot{\theta}_2^2 l_2^2 \cos^2 \theta_2 \dot{\theta}_1^2 l_1^2 \sin^2 \theta_1$$

$$\left. + 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_2 \sin \theta_1 \sin \theta_2 + \dot{\theta}_2^2 l_2^2 \sin^2 \theta_2 \right)$$

$$\Rightarrow = \frac{1}{2} m_1 \dot{\theta}_1^2 l_1^2 (\cos^2 \theta_1 + \sin^2 \theta_1) + \frac{1}{2} m_2 \left(\dot{\theta}_1^2 l_1^2 (\sin^2 \theta_1 + \cos^2 \theta_1) \right.$$

$$\left. + \dot{\theta}_2^2 l_2^2 (\sin^2 \theta_2 + \cos^2 \theta_2) 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_2 (\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2) \right).$$

Eftersom $\sin^2 \theta + \cos^2 \theta = 1$ och $\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 = \cos(\theta_1 - \theta_2)$ (trigonometriska ettan respektive subtraktionsformeln för cosinus) kan vi förenkla uttrycket till:

$$T = \frac{1}{2} m_1 \dot{\theta}_1^2 l_1^2 + \frac{1}{2} m_2 \left(\dot{\theta}_1^2 l_1^2 + \dot{\theta}_2^2 l_2^2 + 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_2 \cos(\theta_1 - \theta_2) \right). \quad (2.13)$$

Nu när vi har uttryckt V och T som funktioner av θ_1 och θ_2 , kan vi beräkna \mathcal{L} . Enligt (2.2), (2.12) och (2.13) får vi att:

$$\begin{aligned}\mathcal{L} &= T - V \\ &= \frac{1}{2}m_1\dot{\theta}_1^2l_1^2 + \frac{1}{2}m_2\left(\dot{\theta}_1^2l_1^2 + \dot{\theta}_2^2l_2^2 + 2\dot{\theta}_1\dot{\theta}_2l_1l_2\cos(\theta_1 - \theta_2)\right) \\ &\quad - (-(m_1 + m_2)gl_1\cos\theta_1 - m_2gl_2\cos\theta_2) \\ \Rightarrow \mathcal{L} &= \frac{1}{2}m_1\dot{\theta}_1^2l_1^2 + \frac{1}{2}m_2\left(\dot{\theta}_1^2l_1^2 + \dot{\theta}_2^2l_2^2 + 2\dot{\theta}_1\dot{\theta}_2l_1l_2\cos(\theta_1 - \theta_2)\right) \\ &\quad + (m_1 + m_2)gl_1\cos\theta_1 + m_2gl_2\cos\theta_2.\end{aligned}\tag{2.14}$$

Tillämpning av Euler-Lagrange ekvationen

Vi ska nu använda ekvation (2.14) i Euler-Lagrange ekvationen (2.3) för att lösa ut ekvationen som beskriver vinklarna θ_1 och θ_2 i dubbelpendeln. Vi beräknar för fallet för $q_i = \theta_1$.

Deriveringsregler för att bestämma $\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}$ och $\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right)$ ger:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1} &= m_1l_1^2\dot{\theta}_1 + m_2l_1^2\dot{\theta}_1 + m_2l_1l_2\dot{\theta}_2\cos(\theta_1 - \theta_2) \\ \Rightarrow \frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right) &= m_1l_1^2\ddot{\theta}_1 + m_2l_1^2\ddot{\theta}_1 + m_2l_1l_2\ddot{\theta}_2\cos(\theta_1 - \theta_2) \\ &\quad - m_2l_1l_2\dot{\theta}_2\sin(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2) \\ \Rightarrow \frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right) &= (m_1 + m_2)l_1^2\ddot{\theta}_1 + m_2l_1l_2\ddot{\theta}_2\cos(\theta_1 - \theta_2) \\ &\quad - m_2l_1l_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + m_2l_1l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2).\end{aligned}\tag{2.15}$$

Deriveringsregler för att bestämma $\frac{\partial\mathcal{L}}{\partial\theta_1}$ ger:

$$\frac{\partial\mathcal{L}}{\partial\theta_1} = -m_2l_1l_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) - (m_1 + m_2)gl_1\sin\theta_1.\tag{2.16}$$

Insättning av (2.15) och (2.16) i (2.3) ger:

$$\begin{aligned}0 &= (m_1 + m_2)l_1^2\ddot{\theta}_1 + m_2l_1l_2\ddot{\theta}_2\cos(\theta_1 - \theta_2) - m_2l_1l_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) \\ &\quad + m_2l_1l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2) + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + (m_1 + m_2)gl_1\sin\theta_1 \\ \Leftrightarrow 0 &= (m_1 + m_2)l_1\ddot{\theta}_1 + m_2l_2\ddot{\theta}_2\cos(\theta_1 - \theta_2) + m_2l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2) \\ &\quad + (m_1 + m_2)g\sin\theta_1.\end{aligned}\tag{2.17}$$

Därmed är (2.17) ekvationen som beskriver vinkeln θ_1 i dubbelpendeln. Nu ska Euler-Lagrange ekvationen (2.3) lösas fast för $q_i = \theta_2$.

Deriveringsregler för att bestämma $\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2}$ och $\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} \right)$ ger:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} &= m_2 l_2^2 \dot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1 \cos(\theta_1 - \theta_2) \\ \Rightarrow \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} \right) &= m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \left(\ddot{\theta}_1 \cos(\theta_1 - \theta_2) - \dot{\theta}_1 \sin(\theta_1 - \theta_2) (\dot{\theta}_1 - \dot{\theta}_2) \right) \\ \Rightarrow \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_2} \right) &= m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \\ &\quad + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2). \end{aligned} \quad (2.18)$$

Deriveringsregler för att bestämma $\frac{\partial \mathcal{L}}{\partial \theta_2}$ ger:

$$\frac{\partial \mathcal{L}}{\partial \theta_2} = m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - m_2 g l_2 \sin \theta_2. \quad (2.19)$$

Insättning av (2.18) och (2.19) i (2.3) ger:

$$\begin{aligned} 0 &= m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \\ &\quad + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_2 g l_2 \sin \theta_2 \\ \Leftrightarrow 0 &= m_2 l_2 \ddot{\theta}_2 + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 g \sin \theta_2. \end{aligned} \quad (2.20)$$

Detta innebär att (2.20) är den ekvation som beskriver vinkeln θ_2 i dubbelpendeln. Därmed har vi nu funnit våra två ekvationer som enligt Euler-Lagrange ekvationen (2.3) beskriver dubbelpendelns rörelse, se ekvationssystemet (2.21):

$$\begin{cases} 0 &= (m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ &\quad + (m_1 + m_2) g \sin \theta_1. \\ 0 &= m_2 l_2 \ddot{\theta}_2 + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 g \sin \theta_2. \end{cases} \quad (2.21)$$

2.3 Teori inför numerisk simulerings

Differentialekvationerna i (2.21) är icke-linjära och av andra ordningen, vilket gör de omöjligt att lösa rent analytiskt. Därmed måste ekvationerna lösas numeriskt. Detta arbete berör två typer av numeriska lösningsmetoder för differentialekvationer: *Eulers stegmetod* och *Runge-Kuttametoden* av fjärde ordningen.

2.3.1 Eulers stegmetod

Eulers stegmetod är den enklaste explicita metoden för att bestämma lösningar till differentialekvationer med begynnelsevillkor. Metoden grundar sig i att dela in en differentialekvationen i flera tidsintervall h . Eftersom Eulers stegmetod är en metod av första ordningen, innebär det att felet relativt den faktiska lösningen vid varje tidsteg är proportionell tidssteget h [6].

Definition av Eulers stegmetod

Givet en differentialekvation på formen $y' = f(t, y)$ med det kända begynnelsevillkoret $y(t_0)$ och det tidssteget $h > 0$, definieras nästa värde y_{n+1} enligt:

$$y_{n+1} = y_n + f(t_n, y_n) \cdot h \quad (2.22)$$

$$t_{n+1} = t_n + h \quad (2.23)$$

Alternativt, om vi har ett system av ordinära differentialekvationer av grad n , kan vi definiera en tillståndsvektor $\mathbf{Y} = [y_1, y_2, y_3, \dots, y_n]$ som beskriver ett systems rörelse, där begynnelsevillkoret \mathbf{Y}_0 även är känt. Låt även $\mathbf{f} = [f_1, f_2, \dots, f_n]$ vara en vektor med funktioner som beskriver systemets rörelseförändring, det vill säga $\mathbf{Y}' = \mathbf{f}(t, \mathbf{Y})$. Då kan Eulers stegmetod definieras enligt:

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + \mathbf{f}(t_n, \mathbf{Y}_n) \cdot h \quad (2.24)$$

$$t_{n+1} = t_n + h \quad (2.25)$$

2.3.2 Runge-Kuttametoden (RK4-metoden)

Runge-Kuttametoden av fjärde ordningen³ är annan vanlig metod för att lösa differentialekvationer, dock till bättre precision än exempelvis Eulers stegmetod. Metoden bygger på att approximera lösningen stegvis med ett litet tidssteg h , men istället för att bara beräkna nuvarande lutningen (det som sker i Eulers stegmetod), så beräknas flera mellanliggande lutningar, där ett viktad medelvärde istället dras. konsekvensen blir att felmarginalen blir betydligt mindre hos RK4-metoden än hos Eulers stegmetod. Till skillnad från Eulers stegmetod, där felet minskar proportionellt mot steglängden h , så minskar felet mot steglängden upphöjt till 4 [6].

Definitionen av RK4-metoden

Givet en differentialekvation på formen $y' = f(t, y)$ med det kända begynnelsevillkoret $y_0(t_0)$ och tidssteget $h > 0$, definieras nästa y_{n+1} enligt:

$$y_{n+1} = y_n + \frac{h}{6} (k_{1n} + 2k_{2n} + 2k_{3n} + k_{4n}) \quad (2.26)$$

$$t_{n+1} = t_n + h, \quad (2.27)$$

³Egentligen är Runge-Kuttametoderna en familj av metoder, då den innefattar ett flertal olika metoder, bland annat Eulers stegmetod. Däremot brukas RK4-metoden menas med benämningen "Runge-Kuttametoden", vilket är den som användes här. För vidare förklaring, se [7].

där k_{1n} , k_{2n} , k_{3n} och k_{4n} definieras som:

$$k_{1n} = f(t, y_n) \quad (2.28)$$

$$k_{2n} = f\left(t + \frac{h}{2}, y_n + \frac{h}{2}k_{1n}\right) \quad (2.29)$$

$$k_{3n} = f\left(t + \frac{h}{2}, y_n + \frac{h}{2}k_{2n}\right) \quad (2.30)$$

$$k_{4n} = f(t + h, y_n + hk_{3n}). \quad (2.31)$$

Alternativt, om vi har ett system av ordniära differentialekvationer av grad n , kan vi definiera en tillstånds vektor $\mathbf{Y} = [y_1, y_2, y_3, \dots, y_n]$ som beskriver ett systems rörelse, där begynnelsevillkoret \mathbf{Y}_0 även är känt. Låt även $\mathbf{f} = [f_1, f_2, \dots, f_n]$ vara en vektor med funktioner som beskriver systemets rörelseförändring, det vill säga $\mathbf{Y}' = \mathbf{f}(t, \mathbf{Y})$. Då kan RK4-metoden definieras enligt:

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + \frac{h}{6} (\mathbf{K}_{1n} + 2\mathbf{K}_{2n} + 2\mathbf{K}_{3n} + \mathbf{K}_{4n}) \quad (2.32)$$

$$t_{n+1} = t_n + h, \quad (2.33)$$

där \mathbf{K}_{1n} , \mathbf{K}_{2n} , \mathbf{K}_{3n} och \mathbf{K}_{4n} definieras som:

$$\mathbf{K}_{1n} = \mathbf{f}(t, \mathbf{Y}_n) \quad (2.34)$$

$$\mathbf{K}_{2n} = \mathbf{f}\left(t + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_{1n}\right) \quad (2.35)$$

$$\mathbf{K}_{3n} = \mathbf{f}\left(t + \frac{h}{2}, \mathbf{Y}_n + \frac{h}{2}\mathbf{K}_{2n}\right) \quad (2.36)$$

$$\mathbf{K}_{4n} = \mathbf{f}(t + h, \mathbf{Y}_n + h\mathbf{K}_{3n}). \quad (2.37)$$

Numerisk simulerings och metod

3.1 Behandling av rörelseekvationerna för simulerings

Som visades i sektion 2.2.1, ekvation (2.21), har rörelseekvationerna för dubbelpendeln härletts med hjälp av Lagrange-mekanik. Dessa är:

$$\begin{cases} 0 = (m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ \quad + (m_1 + m_2) g \sin \theta_1. \\ 0 = m_2 l_2 \ddot{\theta}_2 + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + m_2 g \sin \theta_2. \end{cases} \quad (3.1)$$

För att ändå kunna använda (3.1) för numerisk simulerings, måste de skrivas om till ordinära differentialekvationer på formen $y' = f(t, y)$. Därför genomfördes substitutionen $\dot{\theta}_1 = \omega_1$ och $\dot{\theta}_2 = \omega_2$, vilket gav:

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\theta}_2 = \omega_2 \\ 0 = (m_1 + m_2) l_1 \dot{\omega}_1 + m_2 l_2 \dot{\omega}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \omega_2^2 \sin(\theta_1 - \theta_2) \\ \quad + (m_1 + m_2) g \sin \theta_1 \\ 0 = m_2 l_2 \dot{\omega}_2 + m_2 l_1 \dot{\omega}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \omega_1^2 \sin(\theta_1 - \theta_2) + m_2 g \sin \theta_2. \end{cases} \quad (3.2)$$

Däremot är fortfarande tredje och fjärde ekvationerna i (3.2) sammankopplade (på engelska, 'coupled'). För att kunna använda (3.2) med numeriska simulerings gjordes följande substitution enligt (3.3).

$$\begin{cases} \Delta\theta = \theta_1 - \theta_2 \\ \alpha = (m_1 + m_2) l_1 \\ \beta = m_2 l_2 \cos(\Delta\theta) \\ \gamma = m_2 l_1 \cos(\Delta\theta) \\ \delta = m_2 l_2 \\ \epsilon = -m_2 l_2 \omega_2^2 \sin(\Delta\theta) - (m_1 + m_2) g \sin \theta_1 \\ \zeta = m_2 l_2 \omega_1^2 \sin(\Delta\theta) - m_2 g \sin \theta_2 \end{cases} \quad (3.3)$$

Det går nu skriva om ekvationssystemet (3.2) med hjälp av substitutionerna i (3.3) enligt:

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\theta}_2 = \omega_2 \\ 0 = \alpha\dot{\omega}_1 + \beta\dot{\omega}_2 - \epsilon \\ 0 = \delta\dot{\omega}_2 + \gamma\dot{\omega}_1 - \zeta. \end{cases} \quad (3.4)$$

Därmed kan nu tredje och fjärde ekvationerna i (3.4) skrivas om som ett system av ekvationer i matrisform enligt:

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} \epsilon \\ \zeta \end{pmatrix}^1 \quad (3.5)$$

Multiplikation med inversen av $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$ i (3.5) ger:

$$\begin{aligned} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}^{-1} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} &= \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}^{-1} \begin{pmatrix} \epsilon \\ \zeta \end{pmatrix} \\ \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} &= \frac{1}{\alpha\delta - \beta\gamma} \begin{pmatrix} \delta & -\beta \\ -\gamma & \alpha \end{pmatrix} \begin{pmatrix} \epsilon \\ \zeta \end{pmatrix} \\ \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} &= \frac{1}{\alpha\delta - \beta\gamma} \begin{pmatrix} \delta\epsilon - \beta\zeta \\ -\gamma\epsilon + \alpha\zeta \end{pmatrix} \\ \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} &= \begin{pmatrix} \frac{\delta\epsilon - \beta\zeta}{\alpha\delta - \beta\gamma} \\ \frac{\alpha\zeta - \gamma\epsilon}{\alpha\delta - \beta\gamma} \end{pmatrix} \end{aligned} \quad (3.6)$$

Det går därmed att avläsa från (3.6) nya, icke-sammankopplade uttryck för $\dot{\omega}_1$ och $\dot{\omega}_2$, vilket ger ett ekvationssystemet (3.7) som beskriver dubbelpendelns rörelse i ordinära, icke-sammankopplade differentialekvationer.

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\omega}_1 = \frac{\delta\epsilon - \beta\zeta}{\alpha\delta - \beta\gamma} \\ \dot{\theta}_2 = \omega_2 \\ \dot{\omega}_2 = \frac{\alpha\zeta - \gamma\epsilon}{\alpha\delta - \beta\gamma} \end{cases} \quad (3.7)$$

3.2 Numerisk simulerings

För att kunna simulera dubbelpendeln definierades en tillståndsvektor $\mathbf{X} = [\theta_1, \omega_1, \theta_2, \omega_2]$, vilket därmed ger att $\dot{\mathbf{X}} = [\omega_1, \dot{\omega}_1, \omega_2, \dot{\omega}_2]$. Ett program skrevs sedan i Python med hjälp av dessa tillståndsvektorer, först med Eulers-stegmetod, sedan med Runge-Kuttametoden, för att simulera en dubbelpendel. Dessa jämfördes enligt begynnelsevinklarna i tabell 3.1 att demonstrera effektiviteten av respektive metod.

¹Detta eftersom matrismultiplikation ger att $\alpha\dot{\omega}_1 + \beta\dot{\omega}_2 = \epsilon$ och $\gamma\dot{\omega}_1 + \delta\dot{\omega}_2 = \zeta$, vilket är de ursprungliga ekvationerna.

Sedan skrevs kod där två pendlar simulerades med RK4-metoden, där den andra pendelns hade 0.00001% större begynnelsevinklar relativt den första pendeln. Detta genomfördes för begynnelsevinklarna $\theta_1 = 2.5$ och $\theta_2 = 2.5$ [rad] under 10 sekunder.

Därefter plottades vinklarna θ_1 och θ_2 parametriskt mot varandra enligt startvinklarna i tabell 3.2. Dessutom plottades vinklarna tillsammans mot varandra som funktion av tiden enligt samma värden i tabell 3.2.

Sedan simulerades en pendel med startvillkoren $\theta_1 = 2.9$ och $\theta_2 = -3.1$ [rad] under 180 s. Vinklarna θ_1 och θ_2 plottades parametriskt mot varandra, samt plottades de som funktion av tiden. Därefter simulerades en annan pendel på exakt liknande sätt, men med begynnelsevinklarna $\theta_1 = 0.7$ och $\theta_2 = 0.2$ [rad].

Till sist simulerades två pendlar med begynnelsevinklarna $\theta_1 = 2.899$, $\theta_2 = 1.913$ och $\theta_1 = 2.226$, $\theta_2 = -2.114$ [rad] under 30 sekunder. Vinklarna för respektive pendel plottades även parametriskt och som funktion av tiden.

För alla pendlar antogs att pendlarna släpptes från vila, det vill säga att ω_1 och $\omega_2 = 0$ vid $t = 0$ s. Dessutom antogs att båda massorna $m_1, m_2 = 1$ kg, längderna $l_1, l_2 = 1$ m, samt att $g = 9.82$ m/s². Steglängden h för alla simuleringar var $h = 0.00005$ s. All simuleringskod som skrevs sammanställdes i appendix B.

θ_1	θ_2	'Energinivå'
0.5	0.5	Låg
2.5	2.5	Hög

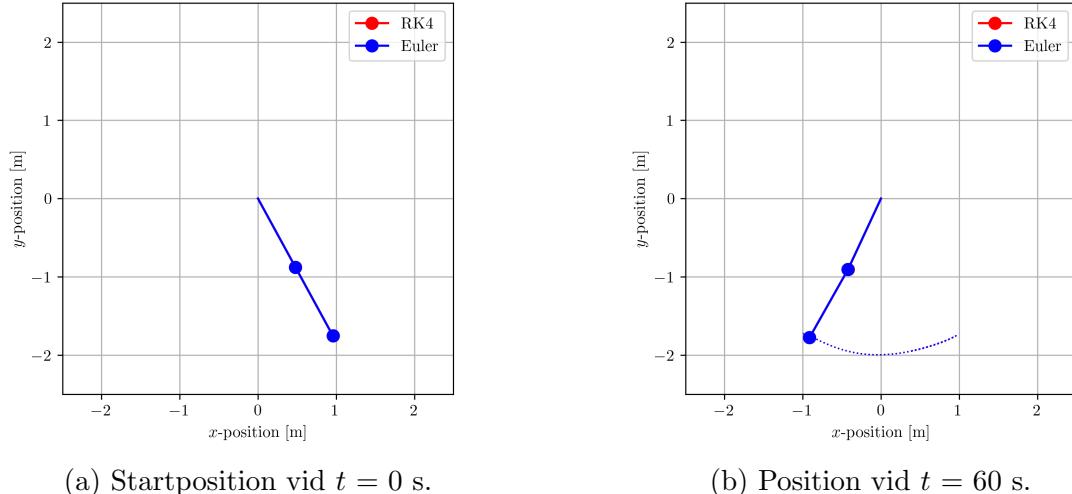
Tabell 3.1: Tre begynnelsevinklar för θ_1 och θ_2 , där θ_1 och θ_2 motsvarar vinklarna i figur 2.1. Vinklarna har även angetts med respektive 'energinivå', det vill säga pendelns ungefärliga mängd potentiella energi som de hade vid startläget.

θ_1	θ_2
0.5	0.5
2.5	2.5
-1.5	1.0
0.42	-1.337
0.5	0.3
-0.5	0.3

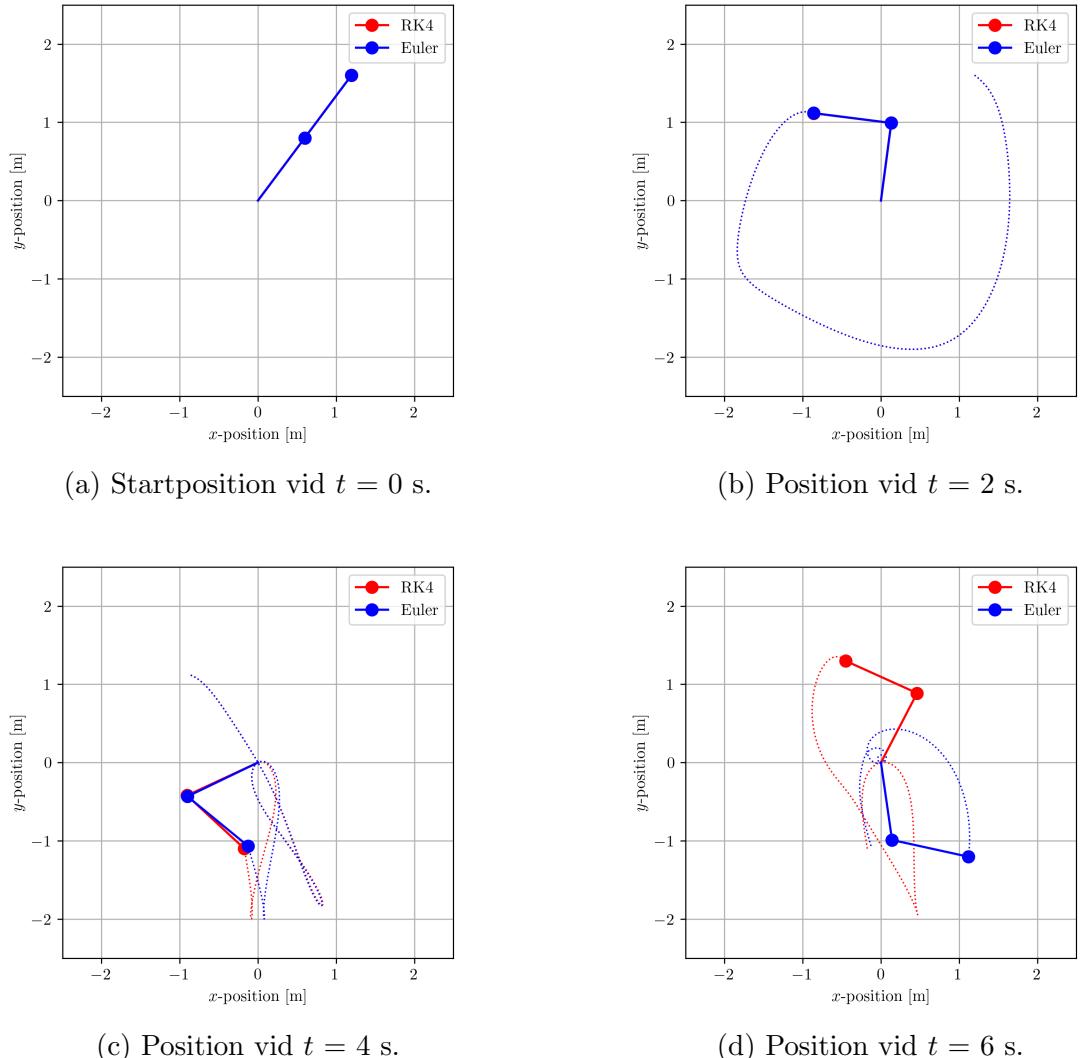
Tabell 3.2: Flera olika begynnelsevinklar för θ_1 och θ_2 , där θ_1 och θ_2 motsvarar vinklarna i figur 2.1.

Resultat

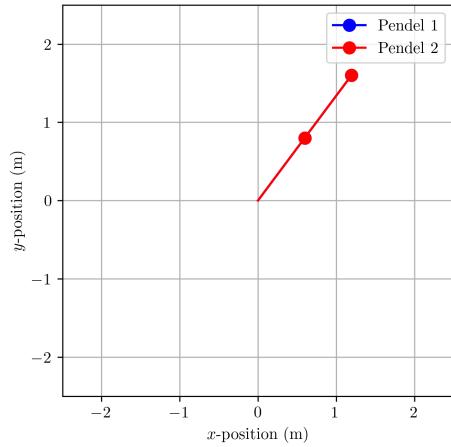
Resultatet av simuleringarna som beskrevs i sektion 3.2 redovisades i figurerna nedan. Första simuleringen, där Eulers-stegmetod och Runge-Kuttametoden jämfördes, redovisades i figur 4.1 och 4.2. Simuleringen där två pendlar simulerades med RK4-metoden, där den andra pendeln hade 0.00001% större begynnelsevinklar, redovisades i figur 4.3. Resultatet från de parametriska plottarna av θ_1 och θ_2 , samt plottarna av vinklarna θ_1 och θ_2 som funktion av tiden enligt tabell 3.2, redovisades i figur 4.4 respektive figur 4.5. Simuleringen av pendlarna som simulerades i 180 s sammanställdes i figur 4.6 och 4.7. Till sist sammanställdes resultatet från simuleringen med begynnelsevinklarna $\theta_1 = 2.899$, $\theta_2 = 1.913$ och $\theta_1 = 2.226$, $\theta_2 = -2.114$ [rad] i figur 4.8.



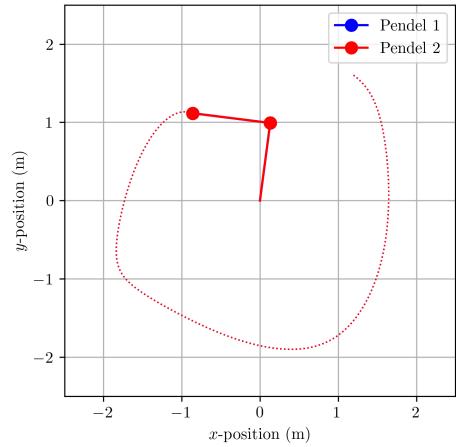
Figur 4.1: Jämförelse mellan Eulers stegmetod och RK4-metoden, den blåa pendeln representerar Eulers stegmetod och den röda pendeln representerar RK4-metoden. Båda pendlarna hade startvinklarna $\theta_1 = 0.5$ rad och $\theta_2 = 0.5$ rad, enligt tabell 3.1. Eftersom metoderna aldrig divergerar märkbart går det inte att se den röda pendeln. Steglängden för båda metoderna var 0.00005 s. Streckade linjerna motsvarar färdvägen av pendeln de senaste två sekunderna



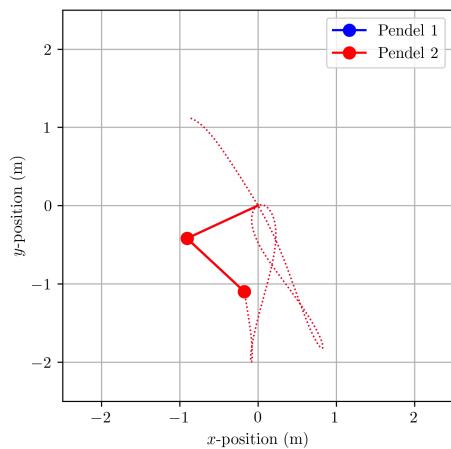
Figur 4.2: Jämförelse mellan Eulers stegmetod och RK4-metoden, där den blåa pendeln representerar Eulers stegmetod och den röda pendeln representerar RK4-metoden. Båda pendlarna hade startvinkelarna $\theta_1 = 2.5$ rad och $\theta_2 = 2.5$ rad enligt tabell 3.1. Vid simuleringen hade båda metoderna steglängden 0.00005 s. De streckade linjerna representerar färdvägen för pendeln under de senaste 2 sekunderna.



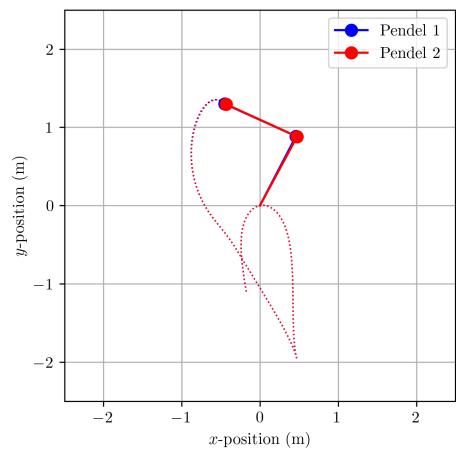
(a) Startposition vid $t = 0$ s.



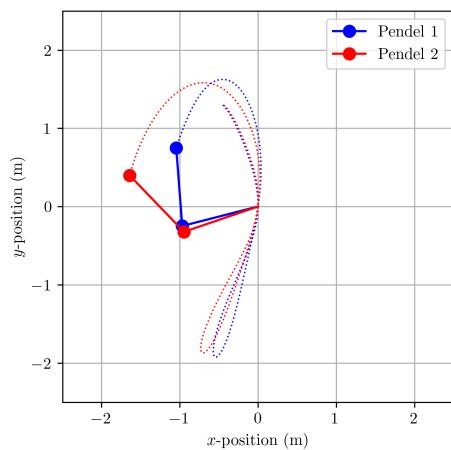
(b) Position vid $t = 2$ s.



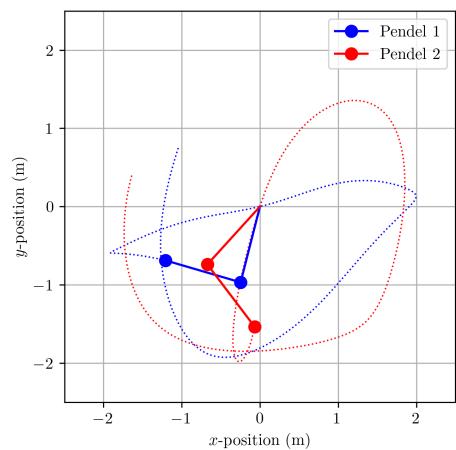
(c) Position vid $t = 4$ s.



(d) Position vid $t = 6$ s.

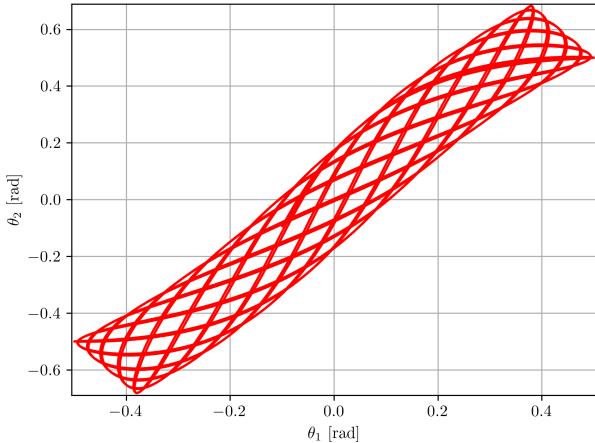


(e) Position vid $t = 8$ s.

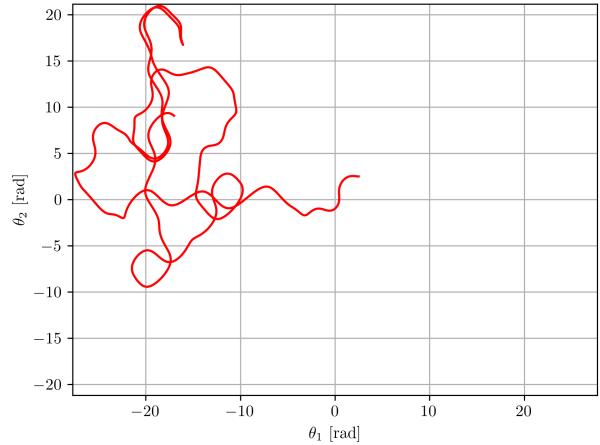


(f) Position vid $t = 10$ s.

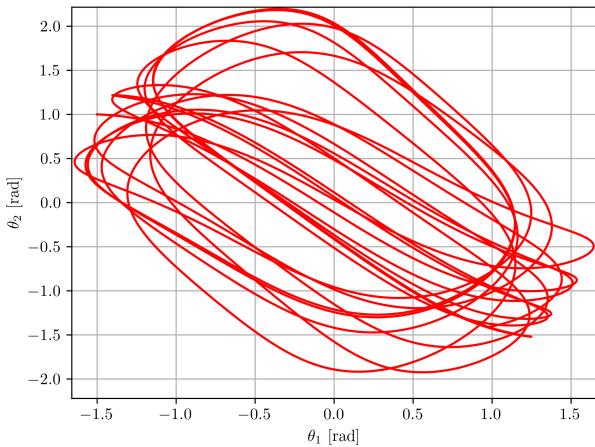
Figur 4.3: Simulering av två pendlar med RK4-metoden. För den första pendeln (blåa) var begynnelsevinklarna $\theta_1 = 2.5$ rad och $\dot{\theta}_1 = 0$, medan för den andra pendeln (röda) var begynnelsevinklarna 0.00001% större relativt den första pendeln. De streckade linjerna representerar färdvägen för pendlarna under de senaste 2 sekunderna.



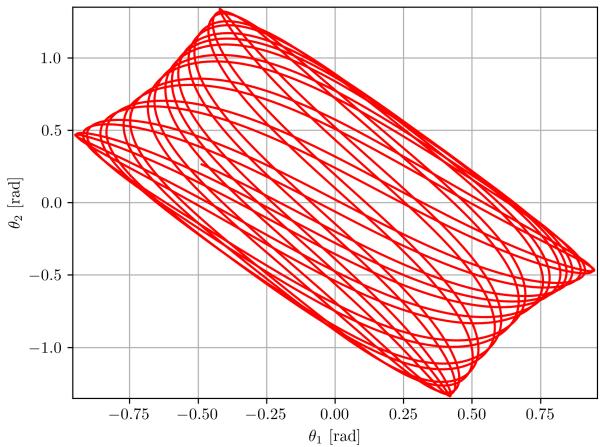
(a) Startvinklar: $\theta_1 = 0.5$ och $\theta_2 = 0.5$ [rad].



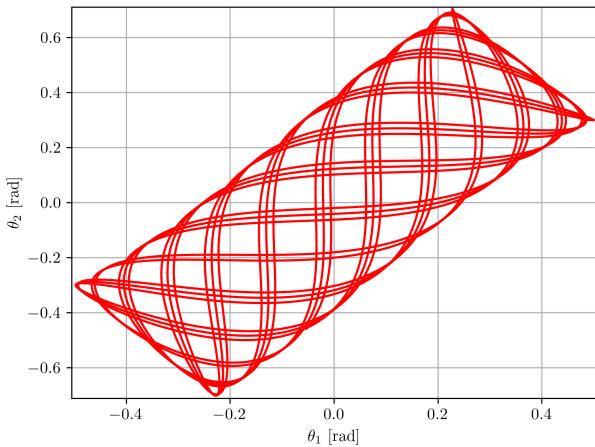
(b) Startvinklar: $\theta_1 = 2.5$ och $\theta_2 = 2.5$ [rad].



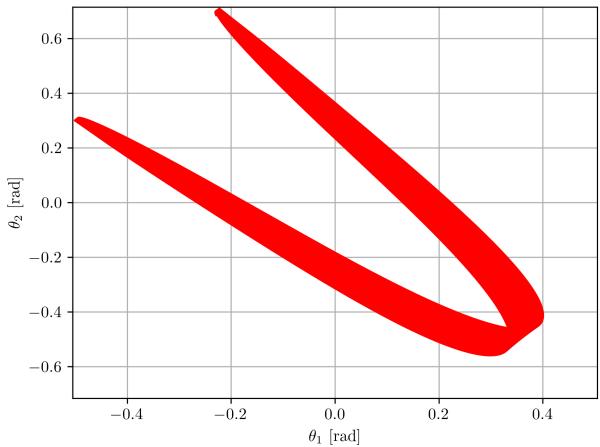
(c) Startvinklar: $\theta_1 = -1.5$ och $\theta_2 = 1.0$ [rad].



(d) Startvinklar: $\theta_1 = 0.42$ och $\theta_2 = -1.337$ [rad].

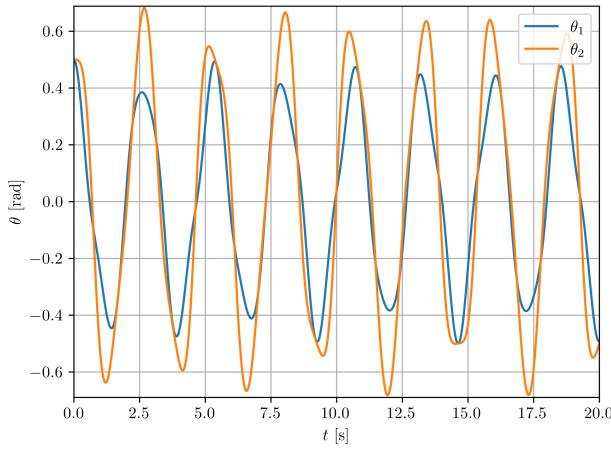


(e) Startvinklar: $\theta_1 = 0.5$ och $\theta_2 = 0.3$ [rad].

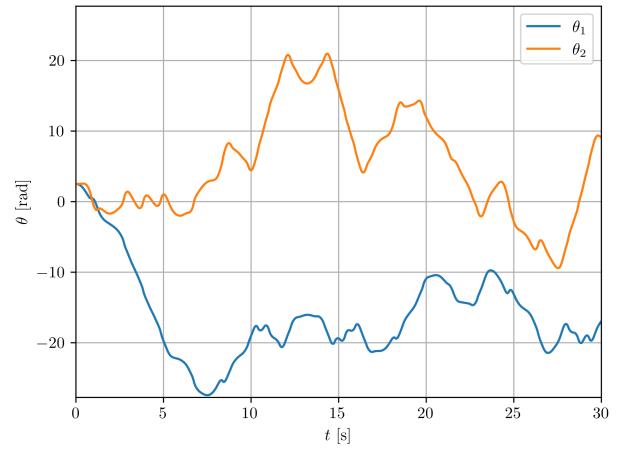


(f) Startvinklar: $\theta_1 = -0.5$ och $\theta_2 = 0.3$ [rad].

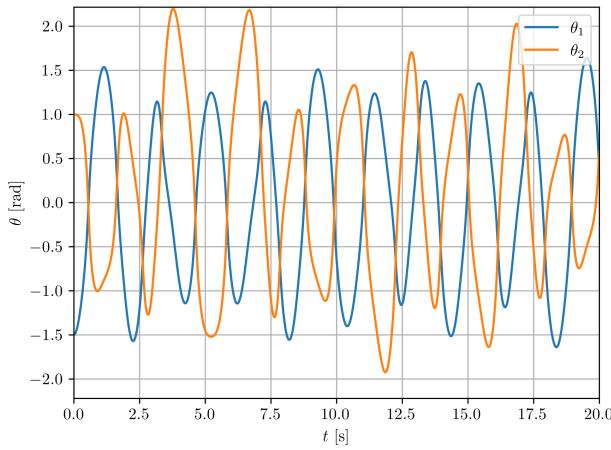
Figur 4.4: Parametrisk plot av vinklarna θ_1 och θ_2 som funktion av tiden för dubbelpendlar som simulerades 30 s med RK4-metoden. Startvinklarna för θ_1 och θ_2 anges i respektive bildtext.



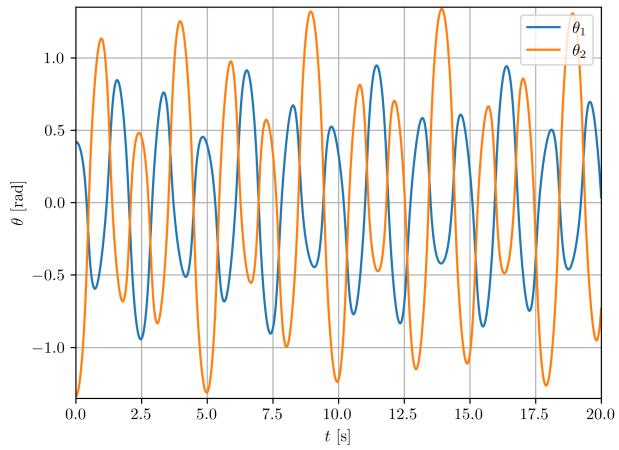
(a) Startvinklar: $\theta_1 = 0.5$ och $\theta_2 = 0.5$ [rad]. Simuleringstid: 20s.



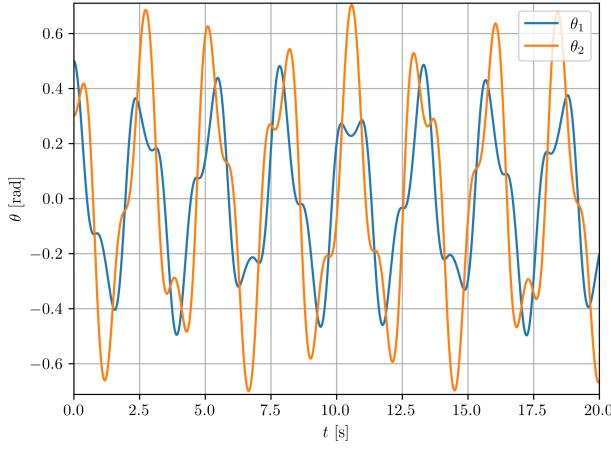
(b) Startvinklar: $\theta_1 = 2.5$ och $\theta_2 = 2.5$ [rad]. Simuleringstid: 30s.



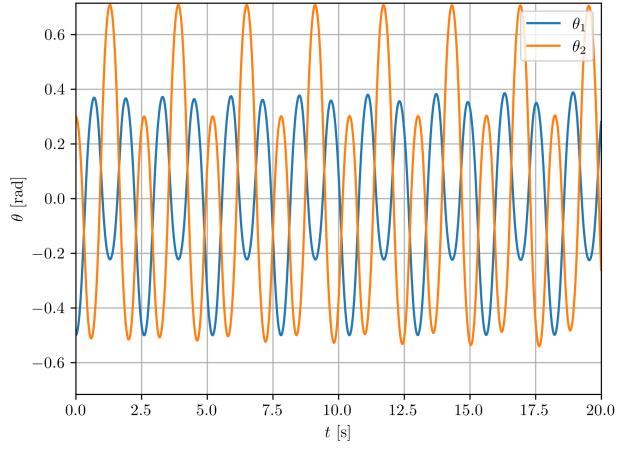
(c) Startvinklar: $\theta_1 = -1.5$ och $\theta_2 = 1.0$ [rad]. Simuleringstid: 20s.



(d) Startvinklar: $\theta_1 = 0.42$ och $\theta_2 = -1.337$ [rad]. Simuleringstid: 20s.

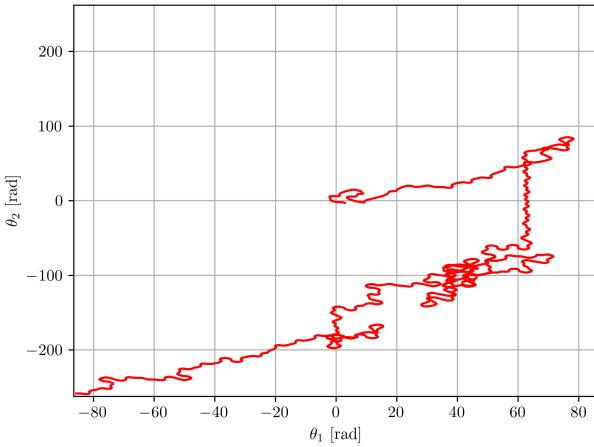


(e) Startvinklar: $\theta_1 = 0.5$ och $\theta_2 = 0.3$ [rad]. Simuleringstid: 20s.

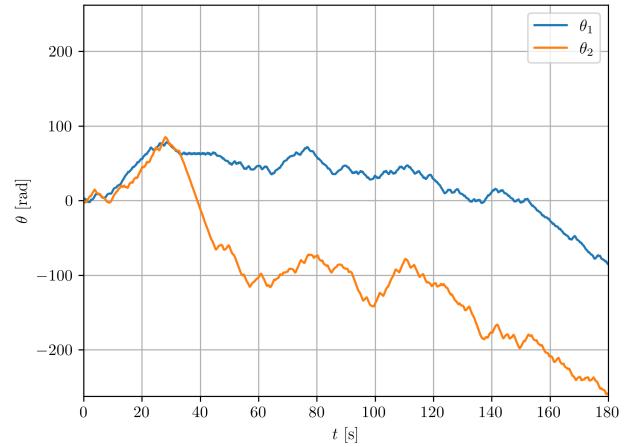


(f) Startvinklar: $\theta_1 = -0.5$ och $\theta_2 = 0.3$ [rad]. Simuleringstid: 20s.

Figur 4.5: Plot av vinklarna θ_1 och θ_2 som funktion av tiden. Alla dubbelpendlar simulerades i 20s, förutom i figur 4.5b, där den simulerades i 30 s. Simuleringen genomfördes med RK4-metoden.

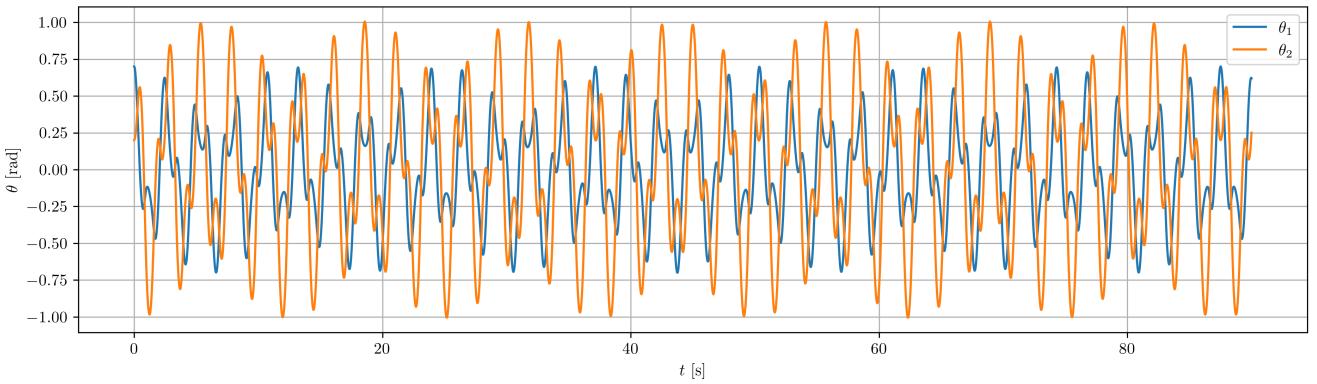


(a) Parametrisk plot för θ_1 och θ_2 under 180 s.
Startvillkor: $\theta_1 = 2.9$, $\theta_2 = -3.1$ [rad].

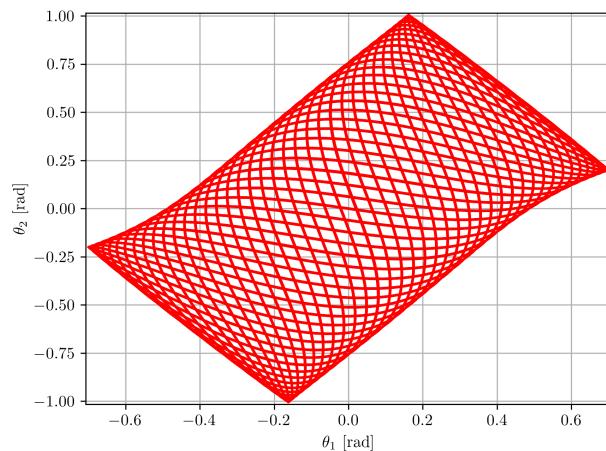


(b) Plot av θ_1 och θ_2 som funktion av tiden, totalt i 180 s. Startvillkor: $\theta_1 = 2.9$, $\theta_2 = -3.1$ [rad].

Figur 4.6: Plot av dubbelpendel med startvillkoren $\theta_1 = 2.9$ och $\theta_2 = -3.1$ [rad]. Simuleringen pågick under 180 s, och genomfördes med RK4-metoden med en steglängd på 0.00005 s.

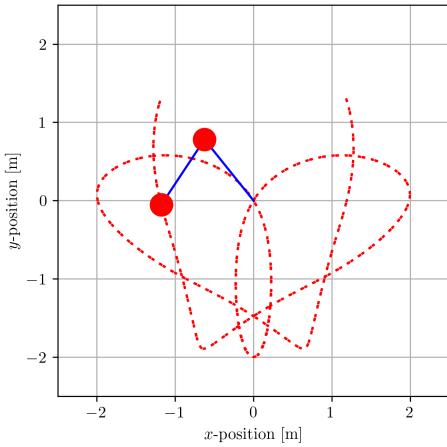


(a) Plot av θ_1 och θ_2 som funktion av tiden under 90 s.

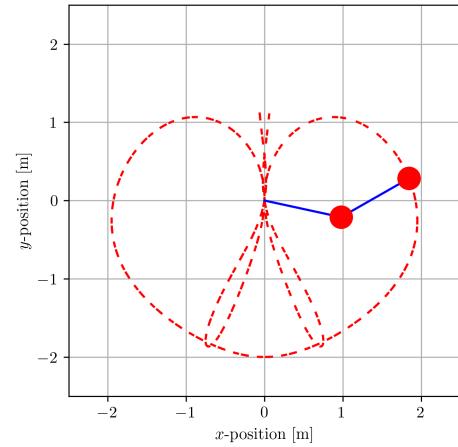


(b) Parametrisk plot av θ_1 och θ_2 under 180 s.

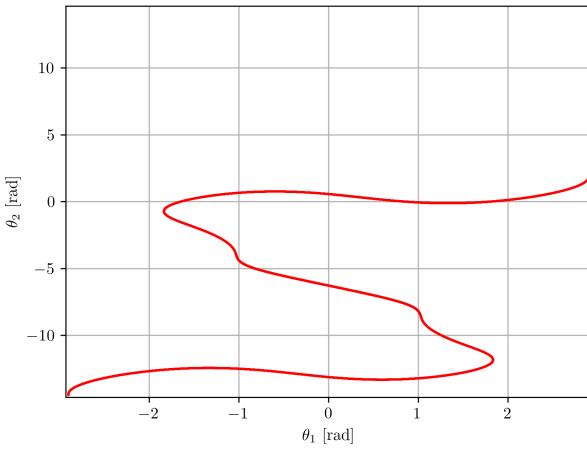
Figur 4.7: Plot av dubbelpendel med startvillkoren $\theta_1 = 0.7$ och $\theta_2 = 0.2$ [rad]. Simuleringen genomfördes i 180 s, men för figur a plottades bara 90 s för att göra kurvan mer läsbar.



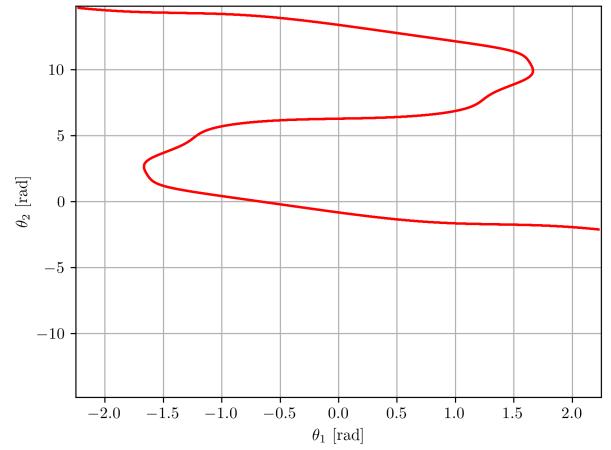
(a) Färdväg för dubbelpendeln. Startvillkor: $\theta_1 = 2.899$, $\theta_2 = 1.913$.



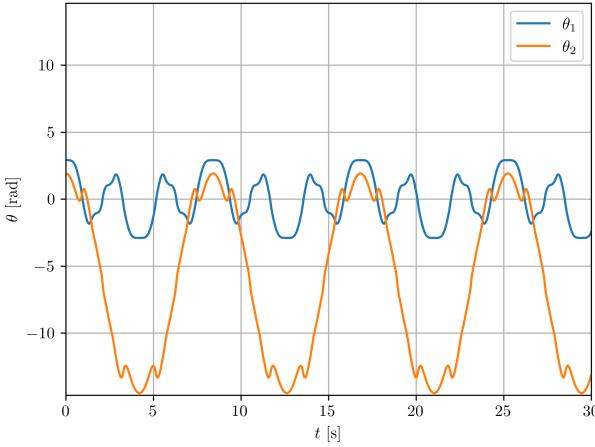
(b) Färdväg för dubbelpendel. Startvillkor: $\theta_1 = 2.226$, $\theta_2 = -2.114$.



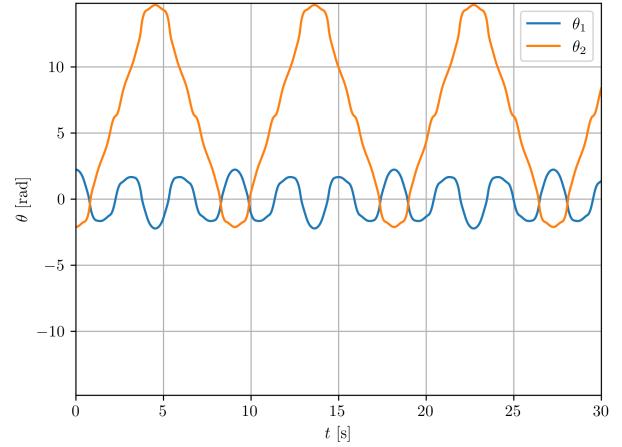
(c) Parametriskt plot under 30 s av θ_1 och θ_2 . Startvillkor: $\theta_1 = 2.899$, $\theta_2 = 1.913$.



(d) Parametriskt plot under 30 s av θ_1 och θ_2 . Startvillkor: $\theta_1 = 2.226$, $\theta_2 = -2.114$.



(e) Plot av vinklarna θ_1 (blå) och θ_2 (orange) som funktion av tiden. Startvillkor: $\theta_1 = 2.899$, $\theta_2 = 1.913$.



(f) Plot av vinklarna θ_1 (blå) och θ_2 (orange) som funktion av tiden. Startvillkor: $\theta_1 = 2.226$, $\theta_2 = -2.114$.

Figur 4.8: Simulering av pendlar med begynnelsevillkoren $\theta_1 = 2.899$ och $\theta_2 = 1.913$ [rad] (figur 4.8a, 4.8c och 4.8e), samt $\theta_1 = 2.226$ och $\theta_2 = -2.114$ [rad] (figur 4.8b, 4.8d och 4.8f). Figur 4.8a och 4.8b visar den streckade linjen den färdväg som pendeln tagit. Figur 4.8c och 4.8d visar en parametriskt plot för pendlarna. Figur 4.8e och 4.8f visar vinklarna θ_1 och θ_2 som funktion av tiden. Alla simuleringar genomfördes med RK4-metoden.

Diskussion

Genom resultatet från simuleringarna av dubbelpendeln går det att besvara frågeställningarna: '*Hur och vad är den bästa metoden för att simulera en dubbelpendel?*', samt '*Vilka slutsatser går det att dra utifrån dubbelpendelns kaotiska egenskaper?*'.

Till en början genomfördes simuleringar för att undersöka vilken numerisk metod, Eulers stegmetod eller RK4-metoden, var lämpligast för att simulera en dubbelpendel. Detta resultat redovisades i figur 4.1 och 4.2, där båda simuleringarna använde en steglängd på $h = 0.00005$ s. I figur 4.1, där begynnelsevinklarna var relativt små ($\theta_1 = 0.5$, $\theta_2 = 0.5$ [rad]), började pendlarna att aldrig märkbart divergera, även efter 60 sekunder. Detta går att förklara med att pendlarna aldrig började utveckla kaotiska inslag, utan att pendeln fortsatte att svänga periodiskt i sitt låga energitillstånd. Till skillnad från dubbelpendlarna i figur 4.2, där begynnelsevinklarna var relativt stora ($\theta_1 = 2.5$, $\theta_2 = 2.5$ [rad]), så divergerade pendlarna mycket fort. Redan i figur 4.2c, när $t = 4$ s, började pendlarna divergera märkbart, och i figur 4.2d, vid $t = 6$ s, visade pendlarna tydligt helt annorlunda banor.

Detta visar uppenbart hur känslig dubbelpendeln är till dess begynnelsevillkor, men även vikten av noggrannheten av dess simulering. Även om simuleringarna genomfördes med en relativt liten steglängd ($h = 0.00005$ s), så växer felet hos Eulers stegmetod så markant förhållandvis till RK4-metoden att Eulers-metoden blir praktiskt sätt oanvändbar. Endast vid låga energinivåer för pendeln, det vill säga när pendelns begynnelsevinklar är små, kan Eulers stegmetod vara användbar. Därmed utfördes resterande delar av alla simuleringar endast med RK4-metoden.

Orsaken till varför RK4-metoden fungerar så mycket bättre i detta fall är hur felet minskar relativt steglängden h . För Eulers stegmetod avtar felet proportionellt mot steglängden h , men för RK4-metoden avtar felet proportionellt mot steglängden h^4 [**runge·kutta·methods·2026**]. Med andra ord, om vi halverar steglängden h , skulle felet för Eulers stegmetod halveras, medan för RK4-metoden skulle det minska med en faktor 16. Därmed blir det tydligt hur Eulers stegmetod fort divergerar relativt RK4-metoden.

Figur 4.3 visar ett annat exempel på hur känsliga dubbelpendlar är för begynnelsevillkor. I simuleringen sattes den andra pendelns begynnelsevillkor till 0.00001% större än den första pendeln (som hade startvillkoret $\theta_1 = 2.5$ och $\theta_2 = 2.5$ [rad]). Redan i figur 4.3d ($t = 6$ s) blir skillnaderna synliga mellan pendlarna, och sedan vid figur 4.3e och 4.3f ($t = 8$ respektive $t = 10$ s) har pendlarnas bana helt divergerat.

För att ge ett annat perspektiv på dubbelpendelns kaos plottades vinklarna θ_1 och θ_2 parametriskt mot varandra i figur 4.4, och sedan plottades vinklarna som funktion av tiden i figur 4.5. Dessa plotter visar att att dubbelpendelns rörelse tenderar att följa unika mönster vid lägre energinivåer/begynnelsevinklar. I figurerna 4.4a, 4.4d, 4.4e och 4.4f visas olika mönster som bildats av dubbelpendelns simulerings. Alla dessa pendlar vid lägre energinivåer/startvinklar visar former av olika periodiska svängningsrörelser och kommer aldrig att börja visa kaotiska beteenden. Varför de aldrig kommer bli kaotiska kan visas med hjälp av ett enkelt energiresonemang: Eftersom energin i systemet alltid kommer att bevaras (och att ingen energi tillförs), samt att pendlarnas lägesenergi vid startpositionen är så låg, kommer pendelns massor aldrig kunna rotera ett helt varv runt sin vridpunkt och därmed börja utveckla kaotiska beteenden. Att pendlarna följer en slags periodisk svängningsrörelse bekräftas även i figurerna 4.5a, 4.5d, 4.5e och 4.5f, där kurvorna tydligt visar sinusformade mönster. Att pendlar med små begynnelsevillkor aldrig kommer att divergera under längre tid demonstreras även i figur 4.7, där en pendel simulerades under 180 s.

Simuleringar genomfördes även för pendlar med medelstora energinivåer/begynnelsevinklar. I figur 4.4c visas en simulering med begynnelsevillkoret $\theta_1 = -1.5$ och $\theta_2 = 1.0$ [rad], där det syns att det parametriska mönstret som bildas är betydligt mindre väldefinierat än de med lägre energinivåer/begynnelsevinklar, men att θ_1 och θ_2 fortfarande stannar generellt inom samma område. Att pendeln inte följer ett lika väldefinierat mönster visas även av de mer oregelbundna sinuskurvorna i figur 4.5c.

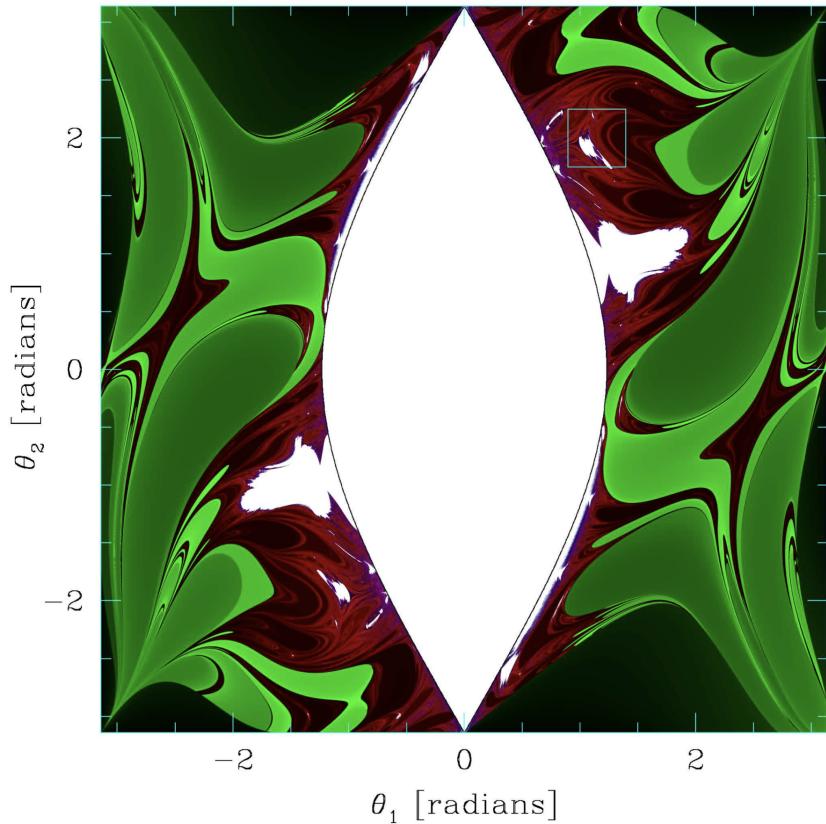
Vid höga energinivåer/begynnelsevinklar visar simuleringarna att vinklarna θ_1 och θ_2 inte har något mönster alls. I figur 4.4b visas en pendel som hade startvinklarna $\theta_1 = 2.5$ och $\theta_2 = 2.5$ [rad]. Vinklarna börjar direkt bli till synes slumpartade, och kurvan i figur 4.5b visar en liknande divergering. Samma händer vid ännu längre simuleringar, som i figur 4.6a och 4.6b där pendlarna simulerades under 180 sekunder.

Generellt sett har simuleringarna visat att pendlarna tenderar att likna periodiska, sinusliknande mönster vid låga energinivåer/begynnelsevinklar, och kaotiska mönster vid höga energinivåer/begynnelsevinklar. Rimligtvis går det att ställa sig frågan: 'Var går gränsen till när en dubbelpendel blir kaotisk?'.

Ett sätt att besvara detta är att plotta för vilka startvinklar dubbelpendeln gör så att dubbelpendeln vrider ett helt varv. Detta har bland annat Jeremy S. Heyl gjort [8], se figur 5.1. Där visas det vita som startpositioner där dubbelpendeln aldrig vänder ett helt fullt varv. Det som Heyl har beräknat ger även samma bild som tidigare: att vid högre energitillstånd/startvinklar tenderar pendlarna att bli mer kaotiska. Däremot visar hans graf att det även finns några energirika begynnelsepositioner som ger stabila pendlar.

Därför demonstreras även pendlarna i figur 4.8 med startvinklarna $\theta_1 = 2.899$, $\theta_2 = 1.913$ samt $\theta_1 = 2.226$, $\theta_2 = -2.114$ [rad]. Figurerna visar att även fast pendlarna var relativt energirika, lyckas pendlarna ändå behålla stabila mönster. Detta bekräftas även i figurerna 4.8c, 4.8d, 4.8e och 4.8f, där de parametriska

mönsterna som bildas både återupprepar sig, men även är periodiska.



Figur 5.1: Fraktal över vilka begynnelsevinklar som ger att en dubbelpendel välter över, och därmed blir kaotisk. Färgkodningen i grafen visar hur fort pendeln välter över, där grön är relativt snabbt, och rött är relativt långsamt. För mer info och bildkälla, se [8].

5.1 Slutsats

Som svar till frågeställningarna visar resultaten att RK4-metoden är en betydligt effektivare metod för att simulera en dubbelpendeln. Dessutom visar resultaten att dubbelpendeln tenderar att visa mer kaotiska utslag vid större begynnelsevinklar, och mer periodiska mönster vid lägre begynnelsevinklar. Detta bekräftas även av andra källor, bland annat av Jeremy S. Heyls artikel [8] och fraktal över vilka dubbelpendlar som är stabila respektive kaotiska. Dessutom påvisades mönster (främst vid låga energinivåer/begynnelsevinklar), både när vinklarna θ_1 och θ_2 plottades parametriskt mot varandra, men även när de plottades som funktion av tiden.

Litteraturförteckning

- [1] W. Scott Morton och Charlton M. Lewis. *China: Its History and Culture*. New York: McGraw-Hill, Inc., 2005, s. 70.
- [2] The Editors of Encyclopaedia Britannica. *Chaos Theory*. Hämtad 2026-01-18. Encyclopædia Britannica, Inc. 19 dec. 2025. URL: <https://www.britannica.com/science/chaos-theory>.
- [3] Edward N. Lorenz. “Deterministic Nonperiodic Flow”. I: *Journal of the Atmospheric Sciences* (1963). URL: <https://cdanfort.w3.uvm.edu/research/lorenz-1963.pdf>.
- [4] N. S. Manton. *The Principle of Least Action in Dynamics*. <https://www.damtp.cam.ac.uk/user/nsm10/PrincLeaAc.pdf>. Hämtad: 2025-01-21. Cambridge, UK, 25 april 2013.
- [5] David Morin. *Introduction to Classical Mechanics: With Problems and Solutions*. Cambridge University Press, 2007.
- [6] Okänd författare. *6.2 Runge–Kutta Methods*. Hämtad: 2026-02-13. MAT290 Pages. n.d. URL: https://mat290.pages.dev/6.2?utm_source=chatgpt.com.
- [7] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. 2. utg. Chichester, UK: John Wiley & Sons Ltd, 2008. ISBN: 978-0-470-72335-7.
- [8] Jeremy S. Heyl. *The Double Pendulum Fractal*. Technical Report. Accessed: 2026-02-13. Department of Physics och Astronomy, University of British Columbia, 11 aug. 2008. URL: <https://www.famaf.unc.edu.ar/~vmarconi/fiscomp/Double.pdf>.

Appendix

A - Härledning av Euler-Lagrange ekvationen (2.3)

Denna härledningen är huvudsakligen baserat på [5, s. 222-223]. Givet att funktionen $x_0(t)$ beskriver den väg av ett objekt, med startpunkterna $x(t_1) = x_1$ och $x(t_2) = x_2$, som ger ett stationärt¹ värde av objektets verkan S , då gäller att:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0. \quad (1)$$

Bevis: Eftersom $x_0(t)$ är den funktion som ger ett stationärt värde av S , så kommer en funktion som är väldigt nära $x_0(t)$ ger i princip exakt samma verkan. Betrakta därmed funktionen:

$$x_a(t) = x_0(t) + a\beta(t), \quad (2)$$

där a är ett tal och $\beta(t)$ uppfyller $\beta(t_1) = \beta(t_2) = 0$ så att ändpunkterna för $x_a(t)$ och $x_0(t)$ är detsamma.

¹Kan alltså vara ett lokalt minimum, maximum eller terrasspunkt av S . Däremot är detta oftast ett minimum, därfor det oftast benämns som *principen om minsta verkan*.

B - Kod för simulerings av dubbelpendeln

Bifogat i Appendix B finns grundkoderna för RK4-metoden och Eulers stegmetod. Alla simuleringar som genomfördes använde någon form av dessa koder. För övrig kod som generera de resterande graferna/simuleringarna, se <https://github.com/uskung/gya/tree/main/code>.

Kod för RK4-metoden

Grundkod för Runge-Kuttametoden. Koden genererar en figur över pendeln och dess färdväg under hela simuleringens gång. Exakta denna kod användes för att simulera pendlarna i figur 4.8a och 4.8b.

```
1 ##### This code generates a still image of a double pendulum and its
2     corresponding path of the second mass.
3 # The double pendulum is simulated with the RK4 method.
4
5
6
7 #####
8 # Initial conditions
9 m_1 = 1 # mass 1[kg]
10 m_2 = 1 # mass 2[kg]
11 l_1 = 1 # length of rod 1 [m]
12 l_2 = 1 # length of rod 2 [m]
13 g = 9.82 # gravitational acceleration [m/s^2]
14
15 ome1 = 0 # angular velocity of the1 [rad/s]
16 ome2 = 0 # angular velocity of the2 [rad/s]
17 #####
18
19 #####
20 # User input for starting angles
21 the1 = float(eval(input("Ange startvinkel för theta_1: "))) # angle 1
22     [rad]
22 the2 = float(eval(input("Ange startvinkel för theta_2: "))) # angle 2
23     [rad]
24
25 # User input for simulation time
26 t_tot= float(input("Hur många sekunder vill du simulera pendeln? "))
27 #####
28 #####
29 # Conditions for simulation
30 h = 0.00005 # time step [s]
31 t0 = 0 # initial time [s]
32 #####
33
34 #####
```

```

35 # Define lists that will be used in simulation to store data
36 the1_list = [the1] # list with the1 angles
37 the2_list = [the2] # list with the2 angles
38 x1pos = [] # list with x1 positions
39 y1pos = [] # list with y1 positions
40 x2pos = [] # list with x2 positions
41 y2pos = [] # list with y2 positions
42 ######
43
44 # Define state vector, which describes the pendulums state
45 state = np.array([the1, ome1, the2, ome2])
46
47 #### Define function that calculates next domega1 and domega2, returns
48 # an array which is the time derivative of the 'state' array
49 def derivative(state):
50     #### calculates coefficients for RK4-method
51     the1, ome1, the2, ome2 = state
52     dtheta = the1 - the2
53     alpha = (m_1 + m_2)*l_1
54     beta = m_2*l_2*np.cos(dtheta)
55     gamma = m_2*l_1*np.cos(dtheta)
56     delta = m_2*l_2
57     epsilon = -1*m_2*l_2 * ome2**2 * np.sin(dtheta) - (m_1 + m_2)*g*np.
58     sin(the1)
59     zeta = m_2*l_2 * ome1**2 * np.sin(dtheta) - m_2*g*np.sin(the2)
60
61     domega1 = (delta*epsilon - beta*zeta)/(alpha*delta - beta*gamma) # #
62     # calculates new domega1
63     domega2 = (alpha*zeta - gamma*epsilon)/(alpha*delta - beta*gamma) # #
64     # calculates new domega2
65
66     return np.array([ome1, domega1, ome2, domega2]) ## returns array
67     which is the time derivative of the 'state' array
68
69 while t0 < t_tot + h:
70     ## calculates coordinates of mass 1
71     x1 = l_1*np.sin(state[0])
72     x1pos.append(x1)
73     y1 = -1 * l_1 * np.cos(state[0])
74     y1pos.append(y1)
75
76     ## calculates coordinates of mass 2
77     x2 = l_1*np.sin(state[0]) + l_2*np.sin(state[2])
78     x2pos.append(x2)
79     y2 = -1 * l_1*np.cos(state[0]) - l_2*np.cos(state[2])
80     y2pos.append(y2)
81
82     ## calculates K1 through K4 according to RK4 method
83     K1 = derivative(state)
84     K2 = derivative(state + h/2 * K1)
85     K3 = derivative(state + h/2 * K2)
86     K4 = derivative(state + h * K3)

87
88     ## define next state
89     state = state + h/6 * (K1 + 2*K2 + 2*K3 + K4)

```

```

85     the1_list.append(state[0])
86     the2_list.append(state[2])
87
88     ## for each iteration, the time is increased by h
89     t0 += h
90
91
92 ######
93 # Makes font to Metafont (same as in LaTeX)
94 plt.rc('font', size = 11, family='serif')
95 plt.rc('text', usetex=True)
96 plt.rc('font', serif='Computer Modern')
97 #####
98
99 #####
100 # Plots axis, axis-labels and grid
101 ax = plt.gca()
102 ax.set_xlim([-2.5,2.5])
103 ax.set_ylim([-2.5,2.5])
104 ax.set_aspect('equal', adjustable='box')
105
106 plt.xlabel('$x$-position [m]')
107 plt.ylabel('$y$-position [m]')
108
109
110 plt.grid()
111 #####
112
113 #####
114 # Plots the pendulum
115 plt.plot([0,x1pos[-1]], [0,y1pos[-1]], color='blue') # plots rod 1
116 plt.plot([x1pos[-1], x2pos[-1]], [y1pos[-1], y2pos[-1]], color='blue') # plots rod 2
117 plt.plot(x2pos, y2pos, color='red', ls=':', label='Fardvag av massa 2') # plots path of mass 2
118 plt.plot(x1pos[-1], y1pos[-1], 'o', markersize=15, color='red') ## plots mass 1
119 plt.plot(x2pos[-1], y2pos[-1], 'o', markersize=15, color='red') ## plots mass 2
120 #####
121
122 # Saves png to directory. Currently commented away to make it easier to run independently
123 #plt.savefig(f'runge-kutta_plots/runge-kutta_plot_the1={the1}_the2={the2}_at_{t_tot}s.png', dpi=300)
124
125 # Plots figure
126 plt.show()

```

Listing 1: Grundkod för RK4-metoden

Kod för Eulers stegmetod

Grundkod för Eulers stegmetod. Koden genererar en figur över pendeln och dess färdväg under hela simuleringens gång.

```
1 ##### This code generates a still image of a double pendulum and its
2     corresponding path of the second mass.
3 # The double pendulum is simulated with Euler's method.
4
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 #####
9 # Initial conditions
10 m_1 = 1 # mass 1[kg]
11 m_2 = 1 # mass 2[kg]
12 l_1 = 1 # length of rod 1 [m]
13 l_2 = 1 # length of rod 2 [m]
14 g = 9.82 # gravitational acceleration [m/s^2]
15
16 ome1 = 0 # angular velocity of the1 [rad/s]
17 ome2 = 0 # angular velocity of the2 [rad/s]
18 #####
19 #####
20 # User input for starting angles
21 the1 = float(eval(input("Ange startvinkeln for theta_1: "))) # angle 1
22     [rad]
23 the2 = float(eval(input("Ange startvinkeln for theta_2: "))) # angle 2
24     [rad]
25
26 # User input for simulation time
27 t_tot= float(input("Hur många sekunder vill du simulera pendeln? "))
28 #####
29 #####
30 # Conditions for simulation
31 h = 0.00005 # time step [s]
32 t0 = 0 # initial time [s]
33 #####
34 #####
35 # Define lists that will be used in simulation to store data
36 the1_list = [the1] # list with the1 angles
37 the2_list = [the2] # list with the2 angles
38 x1pos = [] # list with x1 positions
39 y1pos = [] # list with y1 positions
40 x2pos = [] # list with x2 positions
41 y2pos = [] # list with y2 positions
42 #####
43
44 while t0 < t_tot + h:
45     ## calculates coordinates of mass 1
46     x1 = l_1*np.sin(the1)
```

```

47 x1pos.append(x1)
48 y1 = -1 * l_1 * np.cos(the1)
49 y1pos.append(y1)
50
51 ## calculates coordinates of mass 2
52 x2 = l_1*np.sin(the1) + l_2*np.sin(the2)
53 x2pos.append(x2)
54 y2 = -1 * l_1*np.cos(the1) - l_2*np.cos(the2)
55 y2pos.append(y2)
56
57 #calculate coefficients for Euler's method
58 dtheta = the1 - the2
59 alpha = (m_1 + m_2)*l_1
60 beta = m_2*l_2*np.cos(dtheta)
61 gamma = m_2*l_1*np.cos(dtheta)
62 delta = m_2*l_2
63 epsilon = -1*m_2*l_2 * ome2**2 * np.sin(dtheta) - (m_1 + m_2)*g*np.
   sin(the1)
64 zeta = m_2*l_2 * ome1**2 * np.sin(dtheta) - m_2*g*np.sin(the2)
65
66 domega1 = (delta*epsilon - beta*zeta)/(alpha*delta - beta*gamma) #
   calculates new domega1
67 domega2 = (alpha*zeta - gamma*epsilon)/(alpha*delta - beta*gamma) #
   calculates new domega2
68
69 ome1 = ome1 + h*domega1 # calculates new omega1
70 ome2 = ome2 + h*domega2 # calculates new omega2
71
72 the1 = the1 + h*ome1 # calculates new theta1
73 the1_list.append(the1)
74 the2 = the2 + h*ome2 # calculates new theta2
75 the2_list.append(the2)
76
77 ## for each iteration, the time is increased by h
78 t0 += h
79
80 ######
81 # Makes font to Metafont (same as in LaTeX)
82 plt.rc('font', size = 11, family='serif')
83 plt.rc('text', usetex=True)
84 plt.rc('font', serif='Computer Modern')
85 #####
86
87 #####
88 # Plots axis, axis-labels and grid
89 ax = plt.gca()
90 ax.set_xlim([-2.5,2.5])
91 ax.set_ylim([-2.5,2.5])
92 ax.set_aspect('equal', adjustable='box')
93
94 plt.xlabel('$x$-position [m]')
95 plt.ylabel('$y$-position [m]')
96
97
98 plt.grid()

```

```

99 #####
100 #####
101 #####
102 # Plots the pendulum
103 plt.plot([0,x1pos[-1]], [0,y1pos[-1]], color='blue') # plots rod 1
104 plt.plot([x1pos[-1], x2pos[-1]], [y1pos[-1], y2pos[-1]], color='blue')
    # plots rod 2
105 plt.plot(x2pos, y2pos, color='red', ls=':', label='Fardvag av massa 2')
    # plots path of mass 2
106 plt.plot(x1pos[-1], y1pos[-1], 'o', markersize=15, color='red') ##
    plots mass 1
107 plt.plot(x2pos[-1], y2pos[-1], 'o', markersize=15, color='red') ##
    plots mass 2
108 #####
109 #####
110 # Saves png to directory. Currently commented away to make it easier to
    run independently
111 #plt.savefig(f'euler_plots/euler_plot_the1={the1}_the2={the2}_at_{t_tot}\
}s.png', dpi=300)
112
113 # Plots figure
114 plt.show()

```

Listing 2: Grundkod för Eulers stegemetod