CHALMERS

GÖTEBORGS UNIVERSITET

Geometrisk numerisk integration av differentialekvationer

Geometric Numerical Integration of Differential Equations

Kandidatarbete inom civilingenjörsutbildningen vid Chalmers

Erik Dahllöf
Jonatan Holmquist
Jozef Zoltan
Patrik Bui

Institutionen för Matematiska vetenskaper
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2025

Geometrisk numerisk integration av differentialekvationer

Kandidatarbete i matematik inom civilingenjörsprogrammet Teknisk matematik vid Chalmers
Erik Dahllöf Jonatan Holmquist Jozef Zoltan

Kandidatarbete i matematik inom civilingenjörsprogrammet Kemiteknik med fysik vid Chalmers
Patrik Bui

Handledare: David Cohen

Institutionen för Matematiska vetenskaper
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2025

Förord

Vi vill inleda med ett stort tack till vår handledare David Cohen. David har genom hela arbetets gång stått som en stödpelare till oss. Oavsett om det var något litet som hur man korrekt översätter ett matematiskt begrepp till svenska eller någon större mer omfattande fråga har vi alltid kunnat rådfråga David. Vidare har hans kunskap inom ämnet låtit honom utmana oss och ställa rätt frågor till oss för att vi ska utvecklas och skriva ett så bra arbete som möjligt. Tack!

Denna uppsats är skriven av 4 chalmerister Erik, Jonatan, Jozef och Patrik. Under arbetets gång har en loggbok förts över vilka medlemmar som varit delaktiga i vilka delar i projektet. Ingen del kan fullständigt tillskrivas en författare eftersom alla medlemmar sett till att läsa på, förstå och renskriva alla delar. Följande uppdelning av arbetet är vem som ansvarade för att skriva första utkastet till alla delar, men som sagt är alla delar en reflektion av hela gruppen och inte en individ. För arbetsuppdelningen räknas bevis i appendix som att de är i avsnittet som satsen är i, det är alltså inte så att Jonatan har skrivit nästan alla bevis.

Bidragsrapport - vem har skrivit vad?

Kapitel	**Titel**	**Författare**
	Förord	Jozef
	Populärvetenskaplig Presentation	Patrik
	Sammandrag och Abstract	Jozef
1	**Inledning**	Patrik & Jozef
2	**ODE och numeriska metoder**	Jozef
2.1	Intro till initialvärdesproblem	Jonatan
2.2	Intro till numeriska metoder	Patrik
3	**Runge-Kutta-metoder**	Erik
3.1	Runge-Kutta-metoder	Erik
3.2	Explicita Runge-Kutta-Metoder	Erik
4	**Invarianter**	Jozef
4.1	Linjära Invarianter	Jozef
4.2	Kvadratiska Invarianter	Jozef
5	**Hamiltonska system**	Erik
6	**Average Vector Field-metoden**	Jonatan
7	**Symplektiska metoder**	Patrik
7.1	Flöde och Symplekticitet	Patrik & Jonatan
7.2	Symplektiska metoder	Patrik & Jonatan
8	**Solsystemet**	Jonatan
9	**Slutsats**	Erik
A	**Teori**	Jonatan
B	**Kod**	Jonatan

Populärvetenskaplig presentation

Tänk att du lägger 3 röda kulor och 2 blåa i en låda som du sedan sluter. Om du vid ett senare tillfälle öppnar lådan, vad ser du? Naturligtvis 3 röda kulor och 2 blåa. Denna tankegången kan appliceras på många vetenskapliga scenarion. Om du sluter en glasburk så att ingen luft kan ta sig ut kommer mängden luft i burken förbli konstant, trots att de individuella molekylerna kan reagera med varandra kemiskt. Här representerar de röda och blåa kulorna burkens olika atomer.

Atomer som reagerar i en sluten behållare är bara en av många processer som kan modelleras med differentialekvationer. Differentialekvationer är ekvationer som beskriver hur förändringen av någon storhet beror på storheten själv. Differentialekvationer kan till exempel också användas för att beskriva en bakteriekultur. Från början finns det ett antal bakterier som förökar sig. I takt med att bakterierna blir fler och fler förökar de sig snabbare och snabbare. Hastigheten för hur antalet bakterier förökar sig beror alltså på antalet bakterier.

Till skillnad från algebraiska ekvationer, där målet är att hitta ett tal, är lösningen till differentialekvationer en funktion. Trots att differentialekvationer har studerats i över 300 år är det dock endast i undantagsfall en exakt lösning kan ges. För majoriteten av differentialekvationer måste man istället luta sig mot datorer som kan approximera lösningar. När man använder dessa approximativa metoder finns det ingen garanti att konstanta mängder, som luften i den sluta glasburken, blir kvar. Om man exempelvis simulerar solsystemet med en viss dåligt vald approximativ metod, där energi inte bevaras, kan man se att vi borde ha kraschat in i solen för tusentals år sedan. I denna rapport kommer vi att undersöka hur man kan välja sin approximativa metod med det givna problemet i åtanke samt vilka konsekvenser det kan få att använda en dåligt anpassad metod, så att man, oavsett när man kollar, ser 3 röda kulor och 2 blåa.

Sammandrag

Många vetenskapliga system beskrivs av differentialekvationer. Dessa system har ofta geometriska strukturer, till exempel energibevaring i isolerade fysikaliska system eller konstant massa i kemiska reaktioner. Denna kandidatuppsats undersöker sådana strukturer hos ordinära initialvärdesproblem (IVP), med huvudsyfte att visa hur valet av numeriska integratorer kan göras utefter IVP:ets geometriska egenskaper. I arbetet introduceras och förklaras centrala begrepp inom Geometrisk Numerisk Integration (GNI), såsom invariant, hamiltonfunktion och symplekticitet. Vidare presenteras även ett urval av explicita och implicita Runge-Kuttametoder samt Average Vector Field (AVF) metoden. Dessa numeriska metoder jämförs både genom teoretiska uträkningar men även numeriskt genom simuleringar. Genom textens gång visas flertalet exempel som tydliggör de olika numeriska metodernas styrkor och svagheter. Avslutningsvis används kunskapen för att med både bra och dåliga numeriska metoder simulera planetbanorna i solsystemet.

Abstract

Many systems in the real world are modeled by differential equations. These systems often have geometrical properties - for example energy preservation in isolated physical problems and constant mass in chemical reactions. This thesis explores these structures inherent in ordinary initial value problems (IVPs) with the goal of showing how the choice of numerical methods can be made with the IVPs geometrical properties in mind. In the text, key concepts in Geometrical Numerical Integration (GNI), such as invariant, hamiltonian and symplecticity, are introduced and explained. Furthermore, a few explicit and implicit Runge–Kutta methods, as well as the Average Vector Field (AVF) method, are presented. These numerical methods are analyzed and compared through analytical calculations with numerical simulations supporting our theoretical findings. Various examples further increase understanding about the strengths and weaknesses of the numerical methods. Lastly the acquired knowledge was used to simulate the solar system with both appropriate and inappropriate numerical integrators.

Innehåll

1 Inledning

Många vardagliga och tekniska processer kan beskrivas, förutsägas och analyseras med hjälp av matematik. Newtons gravitationslagar beskriver interaktionen mellan himlakroppar [1] och Lotka–Volterras ekvationer beskriver konkurrens mellan olika djurarter [2]. Både Newtons gravitationslag och Lotka-Volterras ekvationer är fall där en funktions förändringshastighet beror på funktionsvärdet själv. Dessa typer av förhållanden ger upphov till differentialekvationer. Eftersom differentialekvationer beskriver många verkliga fenomen är det viktigt att studera dem för att få en ökad förståelse för vår omvärld.

Differentialekvationer finns på många olika former. Denna text kommer endast att behandla ordinära initialvärdesproblem. En ordinär differentialekvation (ODE) är en differentialekvation som beror på enbart en oberoende variabel och ett initialvärdesproblem (IVP) är ett problem där man utöver en känd differentialekvation också känner till ett startvärde.

Många ordinära differentialekvationer har någon typ av geometrisk egenskap. Till exempel bevaras energin hos en ideal pendel och för en isolerad kemisk reaktion kommer massan förbli konstant. Om man för en differentialekvation med en geometrisk egenskap finner en analytisk lösning, kommer den analytiska lösningen definitionsmässigt att också ha den geometriska egenskapen. Det är dock endast i undantagsfall som en analytisk lösning går att finna.

För majoriteten av ODE-problem finns ingen analytisk lösning utan istället måste numeriska metoder användas för att hitta en approximativ lösning. Eftersom att dessa numeriska metoder ger approximationer finns det ingen garanti att dessa geometriska egenskaper uppfylls. Numeriska metoder som bevarar geometriska egenskaper kallas geometriskt numeriska integratorer. Om man givet ett IVP som saknar analytisk lösning kan välja en numerisk metod så IVP:ns geometriska egenskaper bevaras blir approximationerna ofta bättre.

Arbetets syfte är att ge en tillgänglig introduktion av Geometrisk numerisk integration för studenter. Texten inleds med en kort introduktion till ODE och numeriska metoder. Därefter behandlar vi Runge–Kutta-metoder, som ger oss grunden för att skapa GNI-metoder, och invarianter, som är grunden för ämnet GNI. Arbetet går sedan in på hamiltonska system, AVF-metoden och symplektiska metoder, som är två GNI-metoder som kan användas för hamiltonska system. Därefter avslutas arbetet med ett avsnitt på solsystemet, slutsats och appendix. En sektion om AI-användning och referenser finns också. Huvudreferensen som används i detta arbetet är boken *Geometric Numerical Integration* [3]. Vi kommer anta att läsare är bekanta med linjär algebra och flervariabelanalys. Lite kunskap om ODE och numeriska metoder (mer detalj i avsnitt 2) är också förkunskaper som behövs. I appendix B behövs även förståelse för MATLAB (programmeringsspråket som har använts i detta arbetet).

2 Ordinära differentialekvationer och numeriska metoder

För att förstå arbetets innehåll krävs en grundläggande förståelse för differentialekvationer och numeriska metoder. I detta kapitel kommer vi att lägga fram baskunskaperna som krävs för att förstå resten av rapporten. Om något i denna del känns osäkert rekommenderar vi att du gör dig bekant med det innan du fortsätter.

Vi börjar med att beskriva den sortens problem som vi kommer att undersöka. Sedan presenterar vi några numeriska metoder för att lösa dessa approximativt.

2.1 Introduktion till initialvärdesproblem

Initialvärdesproblem (IVP) dyker upp i verkliga tillämpningar där man känner till något om starttillståndet. Det kan exempelvis handla om en asteroid som riskerar att kollidera med jorden där man känner till dess position och hastighet [4] eller konkurrensen mellan två djurarter där man vet de nuvarande populationerna [5].

Initialvärdesproblem går ut på att hitta en funktion x, som uppfyller en differentialekvation och

ett initialvillkor. De IVP som vi kommer att undersöka är på formen

$$\begin{cases} \dot{x}(t) = f(t, x(t)) \\ x(t_0) = x_0, \end{cases} \tag{2.1}$$

där $\dot{x}(t)$ är tidsderivatan av $x(t)$. Här är $x_0 \in U$, $t_0 \in \mathbb{R}$, $f\colon \mathbb{R} \times U \to \mathbb{R}^d$ givna och U en öppen delmängd av $\mathbb{R}^d$. Definitionsmängden för lösningen $x(t)$ kommer oftast vara hela $\mathbb{R}$ i den här rapporten, men kan vara ett mindre intervall om $x(t)$ går mot randen av U eller går mot oändligheten när t går mot någon ände av intervallet [6, sida 148]. Att f är kontinuerligt deriverbar gör att det alltid finns en unik (maximal) lösning på initialvärdesproblemet [7, sida 119]. Nu presenterar vi tre konkreta exempel på initialvärdesproblem.

Exempel 2.1. Initialvärdesproblemet

$$\begin{cases} \dot{x}(t) = \sin(t)x(t) \\ x(0) = 3 \end{cases}$$

är på samma form som i (2.1) och har lösningen $x(t) = 3\exp(1 - \cos(t))$. ■

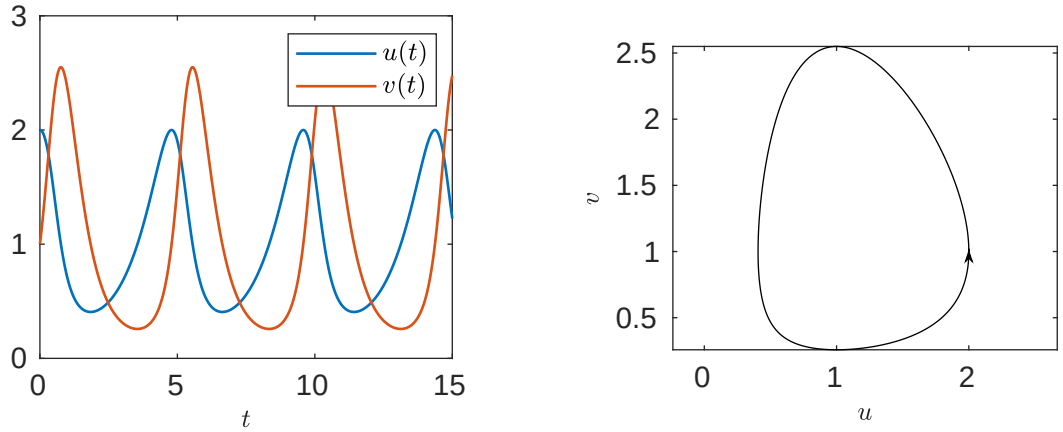
Exempel 2.2 (Lotka–Volterras ekvationer)**.** Lotka–Volterras ekvationer är två kopplade ODE:er som beskriver hur populationer av rovdjur respektive bytesdjur förändras över tid. Modellen beskrevs först av Alfred J. Lotka 1925, se [2, kapitel VIII].

Modellen innehåller flera parametrar med biologisk betydelse. För ett enkelt val av parametrar lyder Lotka–Volterras ekvationer

$$\begin{cases} \dot{u} = u(1 - v) \\ \dot{v} = 2v(u - 1), \end{cases} \tag{2.2}$$

där $u = u(t)$ är bytets population och $v = v(t)$ är rovdjurets population. I detta fallet har vi alltså två kopplade differentialekvationer, men med en enkel omskrivning kan dessa skrivas som en ensamstående differentialekvation på formen (2.1), där $x = (u, v)$ och $f(u, v) = (u(1-v), 2v(u-1))$.

Figur 2.1 visar lösningen med initialvärde $(u_0, v_0) = (2, 1)$. Lösningen i figuren är numerisk men har hög precision. Att kurvan i figur 2.1b är sluten innebär att lösningen är periodisk.



(a) Funktionsgrafer till lösningen.

(b) Banan som lösningen (u, v) tar i $\mathbb{R}^2$.

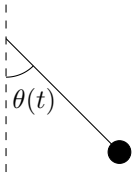
Figur 2.1: Lösning till (2.2) med initialvärden $u(0) = 2$, $v(0) = 1$ visat på två olika sätt.

■

Exempel 2.3 (Ideal pendel)**.** Rörelsen hos en ideal pendel beskrivs, om man väljer tidsenhet på rätt sätt, av den ordinära differentialekvationen $\ddot{\theta}(t) = -\sin(\theta(t))$ där $\theta(t)$ är vinkeln som pendeln har mot jämviktsläget vid tidpunkten t, se [8, kapitel 2] och figur 2.2. ODE:en innehåller två

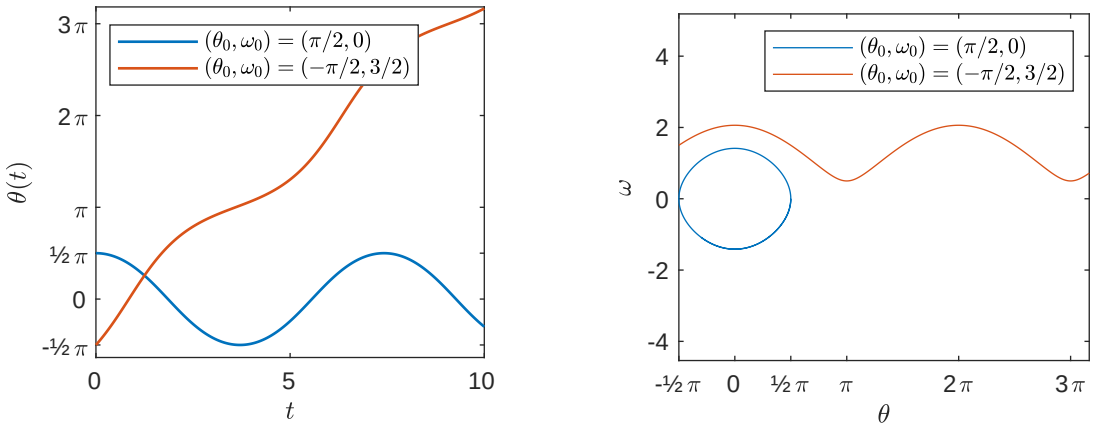
tidsderivator, så för att skriva denna ekvationen på formen (2.1), som bara innehåller en derivata, inför vi vinkelhastigheten $\omega(t) = \dot{\theta}(t)$. Vi kan nu beskriva pendelns rörelse med

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} \omega(t) \\ -\sin(\theta(t)) \end{bmatrix}. \tag{2.3}$$



Figur 2.2: Uppställningen i exempel 2.3.

Vi kan alltså omvandla högre ordningens ODE till en första ordningens ODE med en vektorvärd funktion. Två numeriska lösningar av olika initial villkor visas i figur 2.3. De beräknades på samma sätt som lösningen i exempel 2.2.



(a) Funktionsgrafer till lösningarna.

(b) Banorna som lösningarna (θ, ω) tar i $\mathbb{R}^2$.

Figur 2.3: Lösningar till (2.3) med olika initialvärden $(\theta(0), \omega(0)) = (\theta_0, \omega_0)$, visas på två olika sätt.

■

2.2 Introduktion till numeriska metoder

Till differentialekvationen i exempel 2.1 kunde vi analytiskt hitta en lösning på explicit form, men det kan vi inte göra för Lotka–Volterra modellen i exempel 2.2. Då är det istället lämpligt att använda sig av numeriska metoder för att approximera lösningen.

Generellt går numeriska metoder för initialvärdesproblem ut på att man först diskretiserar tidsintervallet för att sedan approximera lösningar vid de diskreta tidspunkterna. Om vi har ett tidsintervall $[t_0, T]$, där t_0 är starttid och T är sluttid, börjar vi med att diskretisera intervallet i $N+1$ tidspunkter, där N är antalet steg, med en konstant avstånd h, där h är ett steglängd för tid. Vårt intervall $[t_0, T]$ är nu istället en ändlig mängd punkter $\{t_0, t_1, ..., t_N\}$ där $t_n = t_{n-1} + h$ och $t_N = T$. Sedan används $f(t, x)$ och initialvärdet $x_0 = x(t_0)$, se (2.1), för att rekursivt hitta numeriska approximationer $x_n \approx x(t_n)$. I detta projekt betraktar vi bara enstegsmetoder, det vill säga numeriska metoder där varje steg x_{n+1} beräknas med enbart det föregående steget x_n. Vi kommer inte heller betrakta adaptiva metoder, alltså metoder vars steglängd kan förändras beroende på situation [9, kapitel 6.1.4].

Ett exempel på en numerisk metod är Eulers explicita stegmetod [10, sida 210]. För en differentialekvation på formen 2.1 ger den oss nästa steg genom $x_{n+1} = x_n + hf(t_n, x_n)$. Idén kommer från derivatans definition

$$\lim_{h\to 0} \frac{x(t_n + h) - x(t_n)}{h} = \dot{x}(t_n)$$

där vi använder en ändlig steglängd h istället för ett gränsvärde för att approximativt få fram en derivata

$$\frac{x(t_n + h) - x(t_n)}{h} \approx \dot{x}(t_n) = f(t_n, x(t_n)).$$

Detta ger approximationen

$$x(t_n + h) \approx x(t_n) + hf(t_n, x(t_n)).$$

För $n = 0$ får vi då

$$x(t_1) \approx x_0 + hf(t_0, x_0) := x_1.$$

Itererar vi detta får vi Eulers stegmetod

$$x_{n+1} = x_n + hf(t_n, x_n) \approx x(t_{n+1}). \tag{2.4}$$

Exempel 2.4. Vi hittar en approximativ lösning till Lotka–Volterra exempel 2.2, med samma initialvärden $u_0 = 2, v_0 = 1$ vid $t_0 = 0$, via Eulers stegmetod med steglängden $h = 0.2$. Vi börjar från x_0 och får fram $f(t_0, x_0)$ genom

$$x_0 = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \implies f(t_0, x_0) = \begin{bmatrix} u_0(1 - v_0) \\ 2v_0(u_0 - 1) \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Med Eulers stegmetod (2.4) kan vi med insättning av x_0 och $f(t_0, x_0)$ få fram x_1

$$x_1 = x_0 + hf(t_0, x_0) = \begin{bmatrix} 2 \\ 14/10 \end{bmatrix}.$$

För att fortsätta vidare med nästa steg x_2 går vi på samma sätt med x_1 istället för x_0. Vi beräknar $f(t_1, x_1)$ och sedan x_2 enligt

$$f(t_1, x_1) = \begin{bmatrix} u_1(1 - v_1) \\ 2v_1(u_1 - 1) \end{bmatrix} = \begin{bmatrix} -4/5 \\ 14/5 \end{bmatrix} \implies x_2 = x_1 + hf(t_1, x_1) = \begin{bmatrix} 46/25 \\ 49/25 \end{bmatrix}.$$

Detta kan itereras vidare för att få fram x_3, x_4 och så vidare. ■

Eulers stegmetod är en så kallad explicit metod. Nästa steg fås av förra steget med en explicit formel. Men det finns också implicita metoder för IVP-problem (2.1) där nästa steg ges av en ekvation eller ett ekvationssystem som behöver lösas för att få nästa steg. Det enklaste exemplet är Eulers implicita stegmetod [11, kapitel 10.3.6] som ges av

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1}). \tag{2.5}$$

Notera att x_{n+1} är i båda leden. I vår rapport kommer vi mestadels att undersöka implicita metoder eftersom de ibland har vissa intressanta geometriska egenskaper, vilket vi senare kommer att se. Det går ibland att ha en mycket större steglängd om man använder implicita metoder, speciellt för vissa ODE (så kallade stiffa ODE [9, kapitel 6.1.9]) som kräver små steg för att vara stabila, men implicita metoder kräver oftast mer tidskrävande beräkningar i varje steg. Det är inte heller säkert att ekvationssystemet har en lösning. Om h är tillräckligt litet kommer det finnas en unik lösning som man kan få med numerisk metod för ekvationslösning, se exempel A.1 i appendix.

Exempel 2.5. Vi gör första steget i Eulers implicita stegmetod för Lotka–Volteras ekvationer (2.2). Som i exempel 2.4 gör vi det med initialvillkor $u_0 = 2, v_0 = 1$ och med steglängd 0.2. Vi tar nästa steg genom att lösa ekvationssystemet

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{1}{5} \begin{bmatrix} u_1(1 - v_1) \\ 2v_1(u_1 - 1) \end{bmatrix}.$$

Ekvationssystemet har två lösningar $(u_1, v_1) = \left(\left(53 \pm \sqrt{569}\right)/16, \left((\mp\sqrt{569} - 3\right)/14\right)$. Den ena är ungefär $(4.80, -1.92)$ och den andra ungefär $(1.82, 1.49)$. Vi väljer den andra lösningen eftersom det är orimligt att ha en negativ population. Om man använder en numerisk lösningsmetod för att lösa ekvationssystemet, till exempel fixpunktsiterationer, så är det den lösningen man hade approximerat. ■

De två metoderna ovan använder antingen den nuvarande eller nästkommande punkten för att beräkna funktionen $f(t, x)$ och stega framåt. Ofta ligger funktionen $f(t, x)$ som ger det korrekta steg någonstans mellan de två värdena. Genom att använda $f(t, x)$ i mitten ges implicita mittpunktsmetoden, se [11, kapitel 10.3.6]

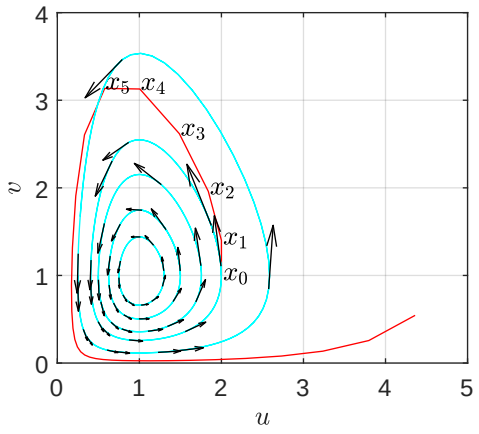
$$x_{n+1} = x_n + hf\left(t_n + \frac{h}{2}, \frac{x_n + x_{n+1}}{2}\right). \tag{2.6}$$

Istället för att räkna ut värdet $f(t, x)$ i punkten (t_n, x_n) eller (t_{n+1}, x_{n+1}) räknas den ut för punkten mitt emellan de två.

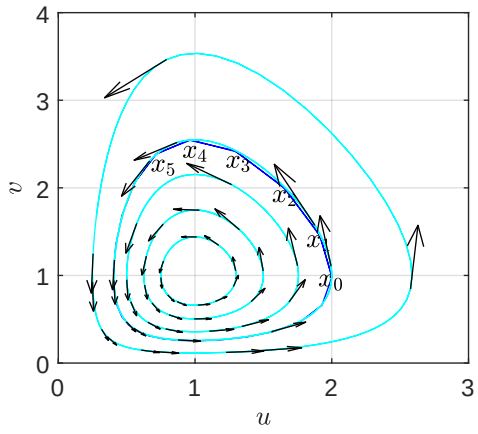
Trapetsmetoden fungerar liknande som mittpunktsmetoden, men istället för att beräkna funktionen $f(t, x)$ i mittpunkten används medelvärdet av funktionen i (t_n, x_n) och (t_{n+1}, x_{n+1}), se [11, kapitel 10.3.6]

$$x_{n+1} = x_n + h\frac{f(t_n, x_n) + f(t_{n+1}, x_{n+1})}{2}. \tag{2.7}$$

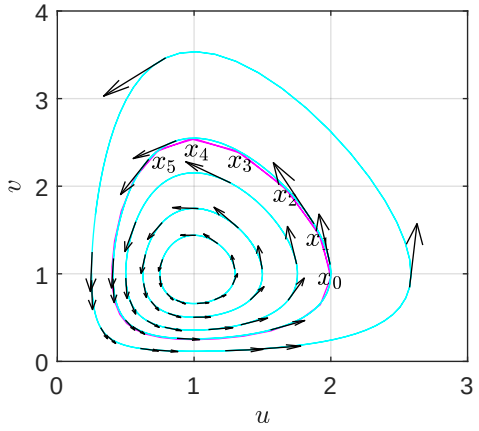
Exempel 2.6. Vi jämför Eulers explicita stegmetod med mittpunktsmetoden och trapetsmetoden för Lotka-Voltera i exempel 2.4, med samma värden, och använder de tre metoderna, se figur 2.4. De ljusblåa kurvorna är lösningsbanor för olika initialvärden u_0, approximerade med en bra nog numerisk metod att vi kan kalla lösningsbanorna exakta.



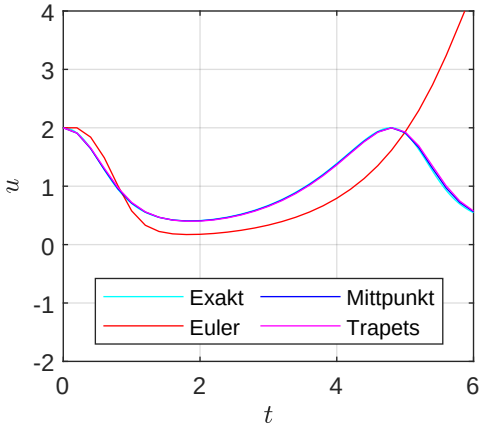
(a) Eulers stegmetod med steglängd h=0.2.



(b) Mittpunktsmetoden med steglängd h=0.2.



(c) Trapetsmetoden med steglängd h=0.2.



(d) $u-t$-graf för kurvorna i 2.4a, 2.4b och 2.4c.

Figur 2.4: Numeriska lösningar till Lotka-Volterra i exempel 2.2 genomförda i MATLAB. Notera att ”exakt” bara innebär en tillräckligt bra numeriska metod att det ungefär är exakt.

Vi ser att Eulers stegmetod ger en lösning som går längre och längre ifrån den exakta lösningen. Mittpunktsmetoden och trapetsmetoden går däremot längs den exakta lösningen. Nere till höger är en u-t-graf som visar att mittpunktsmetoden och trapetsmetoden följer de korrekta x-värdena men lite långsammare än den exakta lösningen. Även här ser vi att Eulers stegmetod ger dåligt resultat. Det går mer formellt att analysera detta med globala fel.

■

Definition 2.1 (Konvergens [12, kapitel 2])**.** En numerisk metod konvergerar till lösningskurvan $x(t)$ om det globala felet $e_N := \|x(T) - x_N\|$ går mot 0 då $h \to 0$. Metoden konvergerar av p:e ordning om $e_N = \mathcal{O}(h^p)$, där p är ett positivt heltal.

Eulers stegmetod konvergerar av 1:a ordningen, alltså $e_N = \mathcal{O}(h)$ [10, sida 218]. Mittpunktsmetoden och trapetsmetoden konvergerar istället av andra ordningen, det vill säga att $e_N = \mathcal{O}(h^2)$ [3, kapitel II.1.1]. Dessa metoder konvergerar alltså snabbare än Eulers stegmetod.

3 Runge–Kutta-metoder

Runge–Kutta-metoder är en familj numeriska metoder som används för att hitta numeriska lösningar till initialvärdesproblem som (2.1). De ovan nämnda metoderna är alla exempel på Runge–Kutta-metoder, men det går att konstruera andra av högre ordning och med önskvärda geometriska

egenskaper. Avsnittet inleder med en genomgång av generella implicita Runge–Kutta-metoder och sedan behandlas explicita Runge–Kutta-metoder.

3.1 Runge–Kutta-metoder

Alla numeriska metoder som presenterades i avsnitt 2.2 är specialfall av en mer allmän samling av numeriska metoder som kallas för Runge–Kutta-metoder (RK-metoder), se [3, kapitel II.1]. RK-metoder hittar numeriska lösningar till initialvärdesproblemet (2.1). Generellt skrivs en RK-metod på formen

$$\begin{aligned} x_{n+1} &= x_n + h\sum_{i=1}^{s} b_i k_i \quad \text{där } k_i \text{ ges av} \\ k_i &= f\left(t_n + c_i h, x_n + h\sum_{j=1}^{s} a_{ij} k_j\right). \end{aligned} \tag{3.1}$$

Man brukar alltid låta $c_i = \sum_{j=1}^{s} a_{ij}$ och för att metoden ska konvergera behövs att $\sum_{i=1}^{s} b_i = 1$.

Ofta skriver man konstanterna för en RK-metod i en så kallad Butcher-tabell som i tabell 3.1.

c_1	$a_{1,1}$	$a_{1,2}$	$\cdots$	$a_{1,s}$
c_2	$a_{2,1}$	$a_{2,2}$	$\cdots$	$a_{2,s}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
c_s	$a_{s,1}$	$a_{s,2}$	$\cdots$	$a_{s,s}$
	b_1	b_2	$\cdots$	b_s

Tabell 3.1: Butcher-tabell för Runge–Kutta-metoder.

Exempel 3.1 (Butcher-tabell)**.** Som tidigare nämndes är mittpunktsmetoden (2.6) och trapetsmetoden (2.7) båda exempel på RK-metoder. Mittpunktsmetoden har den enkla Butcher-tabellen i tabell 3.2 och trapetsmetoden har Butcher-tabellen i tabell 3.3.

$\frac{1}{2}$	$\frac{1}{2}$
	1

Tabell 3.2: Butcher-tabell för mittpunktsmetoden.

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

Tabell 3.3: Butcher-tabell för trapetsmetoden.

■

Genom att välja koefficienterna smart kan vi konstruera olika RK-metoder med olika ordning. Taylorutveckling av differensen $\|x(t_n) - x_n\|$ ger, om koefficienterna är noggrant valda, att de lägre potenserna av h tar ut varandra. Med fler delsteg kan man skapa metoder av högre ordning, se exempelvis [13, kapitel 23]. Men RK-metoder av högre ordning är inte alltid att föredra eftersom de ofta är svårare att beräkna.

Generellt är implicita Runge–Kutta-metoder med stort antal delsteg olönsamma eftersom det kräver lösning av stora ekvationssystem för varje steg. För en differentialekvation med d dimensioner där vi använder en Runge–Kutta-metod med s delsteg behöver vi i varje steg lösa ett ekvationssystem med $d \cdot s$ ekvationer. Det finns inte nödvändigtvis någon lösning alls till ett sådant ekvationssystem. Det finns dock alltid en lösning till ekvation (3.1) om h är tillräckligt litet, se [3, sida 29] och exempel A.1 i appendix A.1.

3.2 Explicita Runge–Kutta-metoder

Vi har sett att Eulers stegmetod är samtidigt explicit och en RK-metod. Explicita RK-metoder är populära eftersom de är mycket mer effektiva att beräkna men man kan ofta tappa viktiga geometriska egenskaper. Därför kommer vi mest ha användning av implicita RK-metoder under resten av texten. Om vi ställer upp Butcher-tabellen enligt tabell 3.4 får vi en grupp explicita RK-metoder med s delsteg. Vi kan se direkt att den är explicit, för att beräkna k_i behövs endast k_j, $1 \leq j < i$ (eftersom $a_{i,j} = 0$ för $j \geq i$) och k_1 kan direkt beräknas. Så om vi beräknar k_i i ordning får vi ett trivialt ekvationssystem.

0	0	0	$\cdots$	0	0
c_2	$a_{2,1}$	0	$\cdots$	0	0
c_3	$a_{3,1}$	$a_{3,2}$	$\cdots$	0	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
c_s	$a_{s,1}$	$a_{s,2}$	$\cdots$	$a_{s,s-1}$	0
	b_1	b_2	$\cdots$	b_{s-1}	b_s

Tabell 3.4: Butcher-tabell för generella explicita Runge-Kutta-metoder.

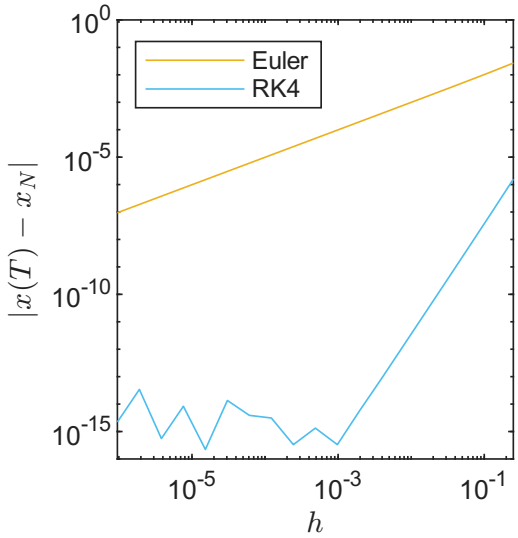
Det viktigaste för oss är att explicita metoder är ofta dåliga på att bevara geometriska egenskaper, men även om de är beräkningseffektiva kan det också vara svårt att nå upp till en hög ordning. För explicita metoder med s delsteg gäller alltid att $s \geq p$ där p är ordningen. Men om man vill uppnå högre ordning behövs mer, för $p \geq 5$ krävs $s > p$ [14, kapitel 32].

Exempel 3.2 (RK4)**.** Den kändaste Runge–Kutta-metoden är en med 4 delsteg som kallas RK4 (eller ibland bara Runge–Kutta-metoden) trots att det bara är en av många metoder med 4 delsteg. Metoden är explicit så den är väldigt beräkningseffektiv och den är av ordning 4 vilket är det högsta möjliga för en explicit RK-metod med 4 delsteg. Tabell 3.5 visar Butcher-tabellen för metoden. Figur 3.1 jämför det globala felet vid $t = 1$ för RK4 respektive Eulers stegmetod för den enkla differentialekvationen $\dot{x}(t) = -0.6x(t)$, $x(0) = 1$ på tidsintervallet $[0, 1]$. Det globala felet för RK4 minskar tydligt snabbare än för Eulers stegmetod och redan vid $h \approx 10^{-3}$ uppnår RK4 maskinprecision.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Tabell 3.5: Butcher-tabell för RK4.

■



Figur 3.1: Globalt fel för RK4 respektive Eulers stegmetod för ekvationen $\dot{x}(t) = -0.6x(t)$, $x(0) = 1$ vid $t = 1$.

4 Invarianter

I många fall när differentialekvationer tillämpas känner man till vissa egenskaper. Exempelvis vet vi att rörelsemängd bevaras i N-kroppssystem och i slutna kemiska reaktioner bevaras substansmängder av atomer [3, kapitel IV]. En storhet som är konstant längs med lösningsbanan till ett IVP kallas för en *invariant*. Invarianter är användbara eftersom de begränsar området där lösningen till differentialekvationen kommer att leva. Nedan kommer vi titta på linjära och kvadratiska invarianter till IVP på formen (2.1) och vilka Runge–Kutta-metoder som bevarar dessa.

4.1 Linjära invarianter

För att illustrera begreppet linjära invarianter tittar vi först på ett exempel om vattnets autoprotolys, där vi ser hur vissa kvantiteter förblir oförändrade även när kemiska reaktioner pågår.

Exempel 4.1 (Vattnets autoprotolys [15, kapitel 6A.4])**.** Vattnets autoprotolys i en sluten behållare beskrivs av den kemiska jämvikten $2H_2O \rightleftharpoons OH^- + H_3O^+$. Låt substansmängderna för de tre ämnena betecknas $c_1(t), c_2(t), c_3(t)$, där indexet motsvarar antalet väte (H) för ämnena och låt k_1 och k_2 vara takten som reaktionen sker åt höger respektive vänster. Betrakta $c_1(t)$, det vill säga substansmängden OH^-. Den kommer att minska proportionellt mot sig självt, $c_3(t)$ och k_2 och öka proportionellt mot k_1 och $c_2(t)^2$. Med samma resonemang för $c_2(t)$ och $c_3(t)$, se [3, kapitel IV.1], får vi differentialekvationen

$$\begin{bmatrix} \dot{c}_1(t) \\ \dot{c}_2(t) \\ \dot{c}_3(t) \end{bmatrix} = \begin{bmatrix} k_1 c_2(t)^2 - k_2 c_1(t) c_3(t) \\ -2k_1 c_2(t)^2 + 2k_2 c_1(t) c_3(t) \\ k_1 c_2(t)^2 - k_2 c_1(t) c_3(t) \end{bmatrix}. \tag{4.1}$$

Från ekvationen kan man direkt se att $\dot{c}_1(t) + \dot{c}_2(t) + \dot{c}_3(t) = 0$ vilket implicerar att $c_1(t) + c_2(t) + c_3(t)$ är konstant. Vi kan tolka detta som att den totala substansmängden i systemet inte ändras. Tillämpar vi samma tankesätt på substansmängden av H kan vi också hitta att $c_1(t)+2c_2(t)+3c_3(t)$ är konstant. Dessa två summor är exempel på så kallade linjära invarianter hos systemet (4.1). ■

Om vi tar exempel 4.1 med initialvärde $(1, 0, 0)$ får vi att den analytiska lösningen måste ligga i skärningen mellan planen $c_1 + c_2 + c_3 = 1$ och $c_1 + 2c_2 + 3c_3 = 2$. Vi vill därför att våra numeriska lösningar också ska finnas där. Exemplet 4.1 är dessutom ett exempel på linjär invariant som vi

definierar nedan.

Definition 4.1 (Linjär invariant)**.** En *linjär invariant* till ett IVP (2.1) är en invariant som kan skrivas på formen $c^T x(t)$, där c är en konstant kolonnvektor av samma dimension som $x(t)$.

Det kommer visa sig enkelt att hitta numeriska metoder som bevarar linjära invarianter, enligt sats 4.1 nedan.

Sats 4.1 (Linjära invarianters bevarande)**.** *Alla Runge–Kutta-metoder* (3.1) *bevarar linjära invarianter för ett IVP på formen* (2.1)*.*

Bevis. Bevis till sats 4.1 finns i appendix A.2. □

4.2 Kvadratiska invarianter

Vi fortsätter med kvadratiska invarianter och presenterar det i samma stil som för linjära invarianter.

Definition 4.2 (Kvadratiska invarianter)**.** *Kvadratiska invarianter* till ett IVP (2.1) är invarianter som kan skrivas på formen $x(t)^T Ax(t)$ där A är en konstant symmetrisk kvadratisk matris.

Definitionen berör endast symmetriska matriser eftersom en godtycklig matris kan delas upp i en symmetrisk och en antisymmetrisk del. För en kvadratisk matris A kan vi alltid skriva $A = A_1 + A_2$ där $A_1 = \frac{1}{2}\left(A + A^T\right)$ och $A_2 = \frac{1}{2}\left(A - A^T\right)$. Vi ser att A_1 är symmetrisk och $A_2^T = -A_2$ så den är antisymmetrisk. Om en skalär transponeras så får man tillbaka samma skalär så för en vektor x får vi $x^T A_2 x = x^T A_2^T x = -x^T A_2 x$. Alltså bidrar den antisymmetriska delen inte någonting till invarianten.

Nedan kommer ett exempel på en ODE med en kvadratisk invariant.

Exempel 4.2 (Harmonisk oscillator)**.** Kraften $F(t)$ på en massa m som hänger i en fjäder med fjäderkonstant k kan beskrivas med hjälp av Hookes lag [16, Kapitel 4], $F(t) = -kq(t)$, där $q(t)$ är massans position relativt jämviktsläget. För att beräkna positionen kan man skriva upp differentialekvationen $\ddot{q}(t) = -\frac{k}{m}q(t)$ som likt exempel 2.3 kan omvandlas till en vektorvärd differentialekvation av första ordningen. Vi inför massans hastighet $v(t) = \dot{q}(t)$ och får systemet

$$\frac{d}{dt}\begin{bmatrix} q(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ -\frac{k}{m}q(t) \end{bmatrix}. \tag{4.2}$$

Följande system har den kvadratiska invarianten $\frac{m}{2}v(t)^2 + \frac{k}{2}q(t)^2$, där

$$x(t) = \begin{bmatrix} q(t) \\ v(t) \end{bmatrix},\ A = \begin{bmatrix} k/2 & 0 \\ 0 & m/2 \end{bmatrix}.$$

Den fysikaliska tolkningen av invarianten är att systemets energi bevaras. $\frac{m}{2}v^2$ är massans kinetiska energi och $\frac{k}{2}q^2$ är den lagrade potentiella energin i fjädern. Kvadratiska invarianter dyker ofta upp i verkliga problem, bland annat i N-kroppsproblemet [3, kapitel IV]. ■

Kravet på att Runge–Kutta-metoder bevarar kvadratiska invarianter visar sig vara mer strikt än för linjära invarianter (sats 4.3). Nedan har vi först en mellanliggande sats innan sats 4.3.

Sats 4.2. *Låt $x(t)$ vara lösningen till ett IVP på formen* (2.1) *och låt $A \in \mathbb{R}^{N\times N}$ vara en symmetrisk matris. Då gäller att $x(t)^T Af(t, x(t)) = 0$ för alla t om och endast om $x(t)^T Ax(t)$ är en invariant till IVP* (2.1)*.*

Bevis. Bevis till sats 4.2 finns i appendix A.2. □

Sats 4.3 (Kvadratiska invarianters bevarande)**.** *Alla Runge-Kutta-metoder*(3.1) *som uppfyller $a_{ij}b_i + a_{ji}b_j = b_i b_j, \forall i, j = 1, 2, ..., s$ bevarar kvadratiska invarianter.*

Bevis. Bevis till sats 4.3 finns i appendix A.2. □

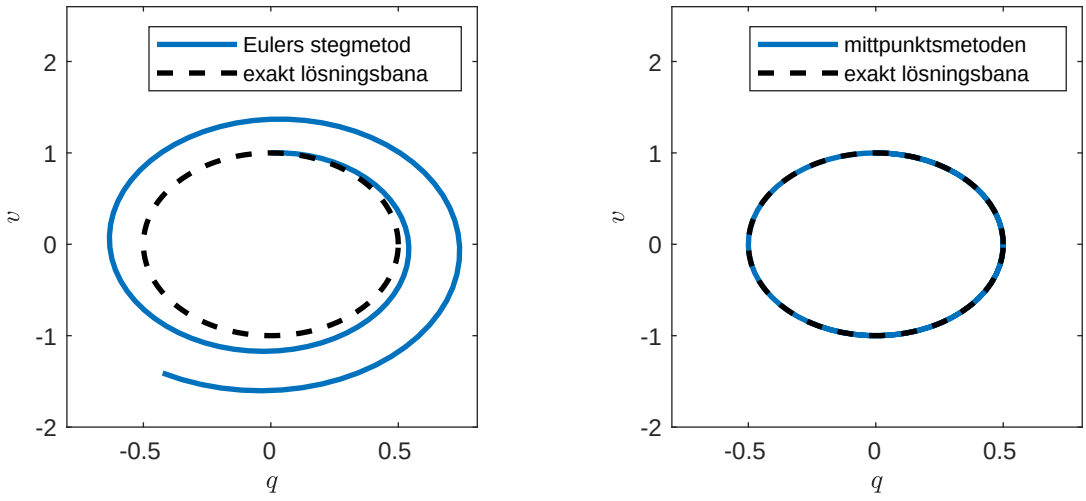
Observera att ingen explicit RK-metod kan uppfylla villkoret i Sats 4.3. För $i = j$ får vi $a_{ii}b_i + a_{ii}b_i = b_ib_i$, alltså $2a_{ii}b_i = b_i^2$. Explicita RK-metoder har nollställda diagonalelement hos a-koefficienterna så vi får $b_i = 0$ för godtyckligt i, men $\sum_{i=1}^{s} b_i = 1$ behövs för att metoden ska konvergera.

En Runge–Kutta-metod som uppfyller villkoret i sats 4.3 är Gauss–Legendre-metoden av ordning 4 [11, sida 630], vars Butcher-tabell finns i tabell 4.1.

$$\begin{array}{c|cc} \frac{1}{2} - \frac{1}{6}\sqrt{3} & \frac{1}{4} & \frac{1}{4} - \frac{1}{6}\sqrt{3} \\ \frac{1}{2} + \frac{1}{6}\sqrt{3} & \frac{1}{4} + \frac{1}{6}\sqrt{3} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Tabell 4.1: Butcher-tabell för Gauss–Legendre-metoden av ordning 4.

Exempel 4.3. Figur 4.1 visar banan för två numeriska approximationer till ODE:en för en harmonisk oscillator (4.2) med olika numeriska metoder. RK-konstanterna i mittpunktsmetodens Butcher-tabell 3.2 uppfyller kravet för Sats 4.3 och därmed stannar den numeriska lösningsbanan på invariantens nivåkurva. För Eulers explicita metod är $a = 0$, $b = 1$ vilket innebär att $b_ib_j - a_{ji}b_j - a_{ij}b_i = 1$. Alltså blir $x_{n+1}^T Ax_{n+1} = x_n^T Ax_n + h^2k^T Ak$ och invarianten bevaras inte. Detta syns i figur 4.1a där lösningsbanan rör sig utåt ifrån invariantens nivåkurva.



(a) Eulers stegmetod i ett faspoträtt.

(b) Mittpunktsmetoden i ett faspoträtt.

Figur 4.1: Numeriska lösningar till den harmoniska oscillatorn i exempel 4.2 uträknade i MATLAB med parametrar $k = 8$, $m = 2$, steglängd 0.05 och 100 steg.

■

5 Hamiltonska system

I exempel 4.2 hittade vi genom systemets totala energi invarianten $\frac{m}{2}p(t)^2 + \frac{k}{2}q(t)^2$, där $p(t)$ var massans hastighet. Om vi istället låter $p(t)$ teckna rörelsemängden $p(t) = mv(t) = m\dot{q}(t)$ får vi ekvationssystemet

$$\begin{cases} \dot{p}(t) = -kq(t) \\ \dot{q}(t) = \frac{p(t)}{m}. \end{cases}$$

Eftersom energin fortfarande måste bevaras tecknar vi invarianten $H(p,q) = \frac{k}{2}q^2 + \frac{1}{2m}p^2$. På grund av en viss egenskap hos denna invariant visar det sig att denna ODE är ett hamiltonskt system, som vi nu definierar.

Definition 5.1 (Hamiltonska system)**.** Ett hamiltonskt system är en ODE som kan skrivas på formen

$$\dot{p}_i(t) = -\frac{\partial H}{\partial q_i}(p(t), q(t)), \quad \dot{q}_i(t) = \frac{\partial H}{\partial p_i}(p(t), q(t)), \quad \forall i = 1, \ldots, d, \tag{5.1}$$

där $H : U \to \mathbb{R}$ är kontinuerligt deriverbar och $U \subseteq \mathbb{R}^d \times \mathbb{R}^d$. Vi kallar H för systemtes hamiltonfunktion och (5.1) för Hamiltons ekvationer.

Fysikaliskt representerar hamiltonfunktionen $H(p,q)$ den totala energin i ett system. De generaliserade koordinaterna $q = (q_1, q_2, \ldots, q_d)$ beskriver systemets position exempelvis som vanliga koordinater eller vinklar. Vektorn $p = (p_1, p_2, \ldots, p_d)$ kan tolkas som en generaliserad rörelsemängd. För den harmoniska oscillatorn 4.2 vi omformulerade i början av kapitel 5 ser vi att $q(t)$ tecknar positionen relativt jämviktsläget och $p(t)$ massans rörelsemängd. En intressant observation är att systemet inte är hamiltonskt om man låter $p(t)$ teckna hastigheten istället för rörelsemängden, trots att de två endast är skilda med en faktor m.

I många fall är hamiltonfunktionen separabel, vilket betyder att den kan skrivas på formen $H(p,q) = T(p) + V(q)$. Här är $T(p)$ den totala rörelseenergin och $V(q)$ den totala lägesenergin. Som tidigare nämnt bevaras den totala energin, och därmed också hamiltonfunktionen, i ett stängt system.

Sats 5.1 (Energibevaring av hamiltonfunktionen)**.** *Hamiltonfunktionen $H(p,q)$ är konstant längs med lösningar till systemet* (5.1)*.*

Bevis. Vi kan bekräfta att hamiltonfunktionen är konstant genom att beräkna dess tidsderivata. Kedjeregeln ger

$$\frac{d}{dt}H(p(t), q(t)) = \sum_{i=1}^{k} \left(\frac{\partial H}{\partial p_i}(p(t), q(t)) \frac{dp_i}{dt}(t) + \frac{\partial H}{\partial q_i}(p(t), q(t)) \frac{dq_i}{dt}(t) \right)$$

och insättning av ekvation (5.1) ger

$$\frac{d}{dt}H(p(t), q(t)) = \sum_{i=1}^{k} \left(\frac{\partial H}{\partial p_i} \left(-\frac{\partial H}{\partial q_i} \right) + \frac{\partial H}{\partial q_i} \frac{\partial H}{\partial p_i} \right) \Bigg|_{(p,q)=(p(t),q(t))} = 0$$

□

Hamiltons ekvationer (5.1) är ett annat sätt att analysera mekaniska system än Newtons rörelselagar. Vi använder definition 5.1 och analyserar ett par mekaniska system nedan.

Exempel 5.1 (Pendel)**.** Den matematiska pendeln från exempel 2.3 är ett hamiltonskt system, med $q(t) = \theta(t)$, $p(t) = \omega(t)$. Hamiltonfunktionen är $H(p,q) = \frac{1}{2}p^2 - \cos(q)$ om referenshöjden för lägesenergin är i nivå med pendelns fästpunkt. Insättning av i $H(p,q)$ Hamiltons ekvationer (5.1) ger samma system som (2.3). Hamiltonfunktionen är separabel med $T(p) = \frac{1}{2}p^2$ och $V(q) = -\cos(q)$, $T(p)$ är rörelseenergin och $V(q)$ lägesenergin. ■

Exempel 5.2 (Keplers problem)**.** Keplers problem berör 2 kroppar som attraherar varandra med gravitationskraften [16, kapitel 4]. Låt kropparna ha rörelsemängder $p_1(t), p_2(t) \in \mathbb{R}^3$, positioner $q_1(t), q_2(t) \in \mathbb{R}^3$ och massor $m_1, m_2 > 0$. Vi har den totala rörelseenergin $\frac{1}{2m_1}\|p_1(t)\|^2 + \frac{1}{2m_2}\|p_2(t)\|^2$. Den totala lägesenergin kommer från gravitationen och den ges av $-G\frac{m_1 m_2}{||q_1(t) - q_2(t)||}$ där G är gravitationskonstanten. Så hamiltonfunktionen blir

$$H(p,q) = \frac{1}{2m_1}\|p_1\|^2 + \frac{1}{2m_2}\|p_2\|^2 - G\frac{m_1 m_2}{\|q_1 - q_2\|}.$$

I avsnitt 8 kommer vi titta på N-kroppsproblemet, en mer generell version av Keplers problem, där fler än två kroppar attraherar varandra.

■

6 Average vector field-metoden

Hittills känner vi till numeriska metoder som bevarar linjära och kvadratiska invarianter. I avsnitt 5 såg vi dock att hamiltonfunktionen kan ge upphov till betydligt mer komplicerade invarianter. Vi kommer nu presentera en metod som exakt bevarar hamiltonfunktionen för alla hamiltonska system.

Vi betraktar initialvärdesproblem (2.1) som uppfyller

$$\dot{x}(t) = f(x(t)) = M\nabla H(x(t)), \tag{6.1}$$

där $H : U \to \mathbb{R}$, $U \subseteq \mathbb{R}^d$ och M är en antisymmetrisk $d \times d$-matris, det vill säga $M^T = -M$. Att M är antisymmetrisk implicerar att $a^T M a = 0$, för alla $a \in \mathbb{R}^d$. För problem på den här formen är H en invariant, eftersom

$$\frac{d}{dt}H(x(t)) = \nabla H(x(t))^T \dot{x}(t) = \nabla H(x(t))^T M \nabla H(x(t)) = 0.$$

Alla hamiltonska system 5.1 från avsnitt 5 kan skrivas på formen (6.1) eftersom Hamiltons ekvationer (5.1) är ekvivalent med

$$\underbrace{\begin{bmatrix}\dot{p}(t)\\ \dot{q}(t)\end{bmatrix}}_{=\dot{x}(t)} = \underbrace{\begin{bmatrix}0 & -I\\ I & 0\end{bmatrix}}_{=M}\underbrace{\begin{bmatrix}\nabla_p H(p(t),q(t))\\ \nabla_q H(p(t),q(t))\end{bmatrix}}_{=\nabla H(x(t))},$$

där $x(t) = (p(t), q(t))$ och I är identitetsmatrisen av dimension d.

En metod för att hitta numeriska lösningar till IVP på formen (6.1), som bevarar mängden $H(x(t))$, är average vector field-metoden (AVF-metoden) som först beskrevs i [17, se sida 1042]. Det är en enstegsmetod där nästa steg beräknas enligt

$$k_n = \int_0^1 f(x_n + shk_n)\, ds \tag{6.2a}$$

$$x_{n+1} = x_n + hk_n. \tag{6.2b}$$

Sats 6.1. *För problem på formen* (6.1) *bevarar AVF-metoden* (6.2) H*, det vill säga* $H(x_n) = H(x_0)$ *för alla* $n = 0, \ldots, N$*.*

Bevis. Bevis till sats 6.1 finns i appendix A.2 □

Sats 6.2. *För ett IVP* (2.1) *där* f *har en kontinuerlig andraderivata, så är AVF-metoden* (6.2) *av ordning* 2*.*

Bevis. Bevis till sats 6.2 finns i appendix A.2. □

AVF är alltså en andra ordningens numerisk metod som bevarar alla hamiltonska invarianter. I kapitel 5 såg vi att pendeln hade en hamiltonfunktion som med metoderna i kapitel (4) inte gick att bevara. Med AVF-metoden kan vi nu geometriskt integrera pendelns ekvation.

Exempel 6.1 (Ideal pendel)**.** Pendelns ekvation

$$\begin{bmatrix}\dot{p}(t)\\ \dot{q}(t)\end{bmatrix} = \begin{bmatrix}-\sin(q(t))\\ p(t)\end{bmatrix}$$

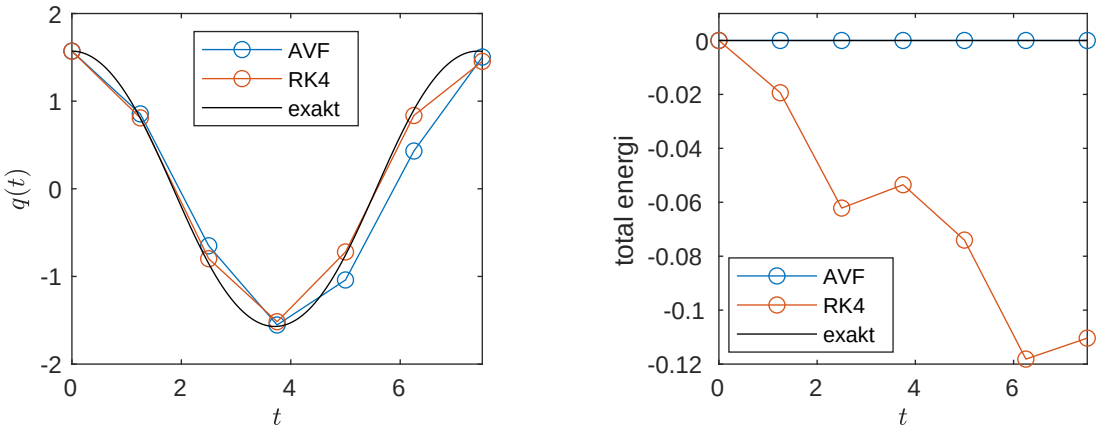
såg vi var hamiltonsk i exempel 5.1. Hamiltonfunktionen $H(p, q) = -\cos(q) + \frac{1}{2}p^2$ bevaras alltså av AVF-metoden. Vi låter steget vara (k_n, ℓ_n) så att

$$\begin{bmatrix}p_{n+1}\\ q_{n+1}\end{bmatrix} = \begin{bmatrix}p_n\\ q_n\end{bmatrix} + \begin{bmatrix}k_n\\ \ell_n\end{bmatrix}.$$

Steget ska enligt (6.2a) uppfylla

$$\begin{bmatrix} k_n \\ \ell_n \end{bmatrix} = \int_0^1 \begin{bmatrix} -\sin(q_n + sh\ell_n) \\ p_n + shk_n \end{bmatrix} ds = \begin{bmatrix} (\cos(q_n + h\ell_n) - \cos(q_n))/(h\ell_n) \\ p_n + \frac{1}{2}hk_n \end{bmatrix}.$$

I figur 6.1 jämförs AVF-metoden med Runge-Kutta 4. Vi ser att RK4 följer den exakta lösningen bättre, eftersom den har högre ordning, men att AVF-metoden bevarar invarianten H exakt. Detta bekräftar sats 6.1.



(a) Graf av vinkeln $q(t)$.

(b) Graf av den totala energin $H(p(t), q(t))$.

Figur 6.1: Grafer till numeriska lösningar av (2.3) med initialvärde $(p_0, q_0) = (0, \pi/2)$ och steglängd $h = 1.2$.

■

I exemplet ovan antas en ideal pendel utan energiförluster. I verkligheten finns det dock utomstående faktorer, såsom luftmotstånd och friktion, som gör att energin ej bevaras. För en del system vars energi $H(x(t))$ förloras genom tiden (dissipativ process) kommer de numeriska lösningarna från AVF-metoden också att förlora energi.

Exempel 6.2 (Dämpad pendel)**.** Rörelsen hos en dämpad pendel kan beskrivas med

$$\begin{bmatrix} \dot{p}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} -\sin(q(t)) - ap(t) \\ p(t) \end{bmatrix}, \tag{6.3}$$

där $a > 0$ kallas för dämpningskoefficienten. Energin $H(p, q) = -\cos(q) + \frac{1}{2}p^2$ är en så kallad lyapunovfunktion, vilket innebär att den monotont avtar längs varje lösningsbana. Detta är eftersom $\frac{d}{dt}H(p(t), q(t)) = \nabla H(p(t), q(t))^T f(p(t), q(t)) = -ap(t)^2 \leq 0$. Vi kan skriva ODE:en som

$$\begin{bmatrix} \dot{p}(t) \\ \dot{q}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 \\ 1 & -a \end{bmatrix}}_{=M} \nabla H(p(t), q(t)). \tag{6.4}$$

Matrisen M är negativt semi-definit, vilket betyder att $x^T Mx \leq 0$ för alla $x \in \mathbb{R}^2$. På samma sätt som i beviset för sats 6.1 kan man visa att H avtar längs den AVF-approximerade lösningsbanan, det vill säga att $H(x_{n+1}) \leq H(x_n)$. Mer detaljer om dissipativa processer kan hittas i [17]. ■

7 Symplektiska metoder

I avsnitt 5 introducerade vi hamiltonska system och dess invarianta hamiltonfunktion. Det är dock inte den enda geometriska egenskap hamiltonska system har. Följande kapitel kommer ägnas åt en annan geometrisk egenskap; sympekticitet.

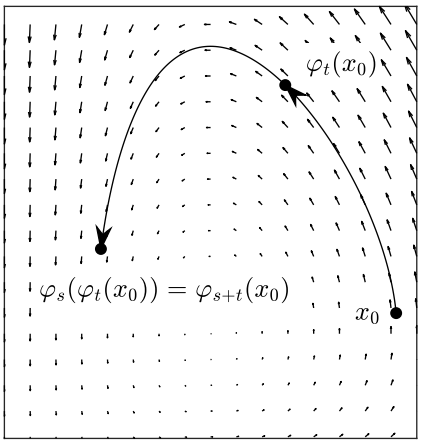
7.1 Flöde och symplekticitet

För att förstå symplekticitet och symplektiska metoder måste man först förstå flödet hos en autonom differentialekvation. Ett IVP är autonomt om det inte beror direkt på tiden, det vill säga att den kan skrivas som $\dot{x}(t) = f(x(t))$, jämför med (2.1).

Definition 7.1 (Flöde [3, kapitel I.1.1])**.** För en autonom differentialekvation $\dot{x}(t) = f(x(t))$ är flödet $\varphi_t(x_0) = x(t)$ där $x(t)$ är lösningen med initialvärde $x(0) = x_0$. Det vill säga flödet uppfyller

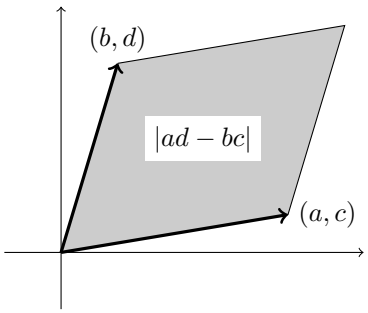
$$\begin{cases} \dot{\varphi}_t(x_0) = f(\varphi_t(x_0)) \\ \varphi_0(x_0) = x_0. \end{cases}$$

Med flödet ser vi lösningen till en autonom ODE som en funktion av initialvärdet x_0. Figur 7.1 visar hur flödet avbildar två punkter x_0 och $\varphi_t(x_0)$. Figuren illustrerar även en intuitiv egenskap hos flödet, nämligen att $\varphi_{s+t}(x_0) = \varphi_s(\varphi_t(x_0))$. Det gäller att $\varphi_0(x_0) = x_0$ och det följer att $\varphi_t'(x_0) = I$ för alla x_0 (φ_t' betyder jacobimatrisen av φ_t, inte φ:s partiella t-derivata).



Figur 7.1: Flödet av en differentialekvation i planet. De små pilarna visar ODE:ens högerled $f(x)$.

Arean av ett parallellogram som spänns upp av två vektorer $u = (a, c)$, $v = (b, d) \in \mathbb{R}^2$ som i figur 7.2 är $|\det[u \quad v]| = |ad - bc|$. Det visar sig lämpligt att skriva determinanten som $\det[u \quad v] = u^T J v$ där $J = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$.



Figur 7.2: Parallellogram som spänns upp av två vektorer.

Om en 2×2 matris A uppfyller $A^T J A = J$ (vilket är ekvivalent med $\det A = 1$) så har parallellogram som spänns up av Au och Av samma area som det som spänns upp av u och v eftersom då är $(Au)^T J A v = u^T A^T J A v = u^T J v$. Detta leder oss till idén med symplekticitet; att arean

bevaras under en avbildning A. Symplekticitet kan utökas till deriverbara avbildningar genom att undersöka hur arean förändras för små parallellogram vars vektorer går mot nollvektorn före och efter avbildningen. Om jacobimatrisen $f'(x)$ för en deriverbar avbildning $f : \mathbb{R}^2 \to \mathbb{R}^2$ uppfyller $(f'(x))^T J f'(x) = J$ för alla x så bevarar f area. Det vill säga att om $S \subseteq \mathbb{R}^2$ så är arean av S lika med arean av $f(S) = \{f(x) : x \in S\}$.

Vi kommer nu utöka begreppet symplekticitet till $2d$ dimensioner. Alla matriser i det här avsnittet kommer vara $d \times d$ eller $2d \times 2d$ matriser. I följande definition överger vi den geometriska tolkningen av symplekticitet som areabevaring av parallellogram. Det finns en geometrisk tolkning av symplekticitet i högre dimensioner via projektioner av flerdimensionella volymer till summor av tvådimensionella areor [3, kapitel VI.2], men det ligger utanför detta arbetet.

Definition 7.2 (Symplekticitet [3, kapitel VI.2])**.** Låt $U \subseteq \mathbb{R}^{2d}$. En deriverbar avbildning $f : U \to \mathbb{R}^{2d}$ är symplektisk om

$$f'(x)^T J f'(x) = J$$

för alla $x \in U$ där $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$ och I är en $d \times d$ identitetsmatris.

Symplekticitet relateras till hamiltonska system enligt följande sats, som bevisades av Poincaré år 1899.

Sats 7.1 ([3, kapitel VI.2])**.** *Flödet φ_t för ett hamiltonskt system 5.1 är symplektiskt.*

Bevis. Bevis till sats 7.1 finns i appendix A.2 □

Det visar sig också vara så att alla ODE:er vars flöde är symplektiskt är (lokalt) ett hamiltonsk system [3, kapitel VI.2]. Att flödet för en ODE är symplektiskt är alltså ekvivalent med att ODE:en är hamiltonsk.

7.2 Symplektiska metoder

En numerisk metod för hamiltonska initialvärdesproblem är symplektisk om de approximativa punkterna från den numeriska lösningen (p_n, q_n) är symplektiska, när de ses som en funktion av initialvärdet (p_0, q_0). Det vill säga att för alla n är

$$\left(\frac{\partial(p_n, q_n)}{\partial(p_0, q_0)}\right)^T J \frac{\partial(p_n, q_n)}{\partial(p_0, q_0)} = J. \tag{7.1}$$

där vi nu använder $\frac{\partial(p_n, q_n)}{\partial(p_0, q_0)}$ som en notation för jacobimatrisen av (p_n, q_n) med avseende på p_0 och q_0. För en enstegsmetod (se avsnitt 2.2) är det ekvivalent med att varje steg $(p_n, q_n) \mapsto (p_{n+1}, q_{n+1})$ är symplektiskt.

7.2.1 Symplektiska Euler-metoden

Det enklaste exemplet på en sådan metod är symplektiska Euler-metoden som är av ordning 1 och har, för ett hamiltonskt system 5.1 formen

$$\begin{aligned} p_{n+1} &= p_n - h\frac{\partial H}{\partial q}(p_{n+1}, q_n) \\ q_{n+1} &= q_n + h\frac{\partial H}{\partial p}(p_{n+1}, q_n). \end{aligned} \tag{7.2}$$

Sats 7.2 ([3, kapitel VI.3] eller [12, kapitel 15.3])**.** *Symplektiska Euler-metoden* (7.2) *är symplektisk.*

Bevis. Bevis till sats 7.2 finns i appendix A.2. □

7.2.2 Symplektiska Runge–Kutta-metoder

Vi kommer bevisa att Runge–Kutta-metoder som bevarar kvadratiska invarianter (avsnitt 4.2) också bevarar symplekticitet, det vill säga att dessa Runge–Kutta-metoder också är symplektiska metoder. Vi kommer använda ett lemma och introducerar uttökningen av en ODE med dess variansekvation.

För ett IVP

$$\begin{cases} \dot{x}(t) = f(x(t)) \\ x(0) = x_0 \end{cases} \tag{7.3}$$

låter vi $\Psi(t) = \varphi_t'(x_0)$. Om vi deriverar ODE:en med avseende på x_0 får vi att $\dot{\Psi}(t) = f'(x(t))\Psi(t)$ vilket kallas för ODE:ens variationsekvation. Dessutom är $\Psi(0) = I$. Vi kan uttöka (7.3) med variansekvationen och få IVP:et

$$\begin{cases} \dot{x}(t) = f(x(t)) \\ \dot{\Psi}(t) = f'(x(t))\Psi(t) \\ x(0) = x_0; \Psi(0) = I. \end{cases} \tag{7.4}$$

Lemma 7.3 (se sida 191 i [3])**.** *Vi föreställer oss att vi löser* (7.3) *med en Runge–Kutta-metod och får approximativa punkter* $x_0, x_1, \ldots, x_N$*. Vi löser också* (7.4) *med samma Runge–Kutta-metod och samma steglängd och får approximativa punkter* $x_0, x_1, \ldots, x_N, \Psi_0, \Psi_1, \ldots, \Psi_N$ *(den numeriska lösningen för* x *kommer vara lika). Då kommer* $\frac{\partial}{\partial x_0} x_n = \Psi_n$ *för alla* n*.*

Bevis. Bevis till lemma 7.3 finns i appendix A.2. □

Vi har nu allt som behövs för att bevisa följande sats.
Sats 7.4 ([3, kapitel VI.4.1])**.** *Alla Runge–Kutta-metoder* (3.1) *som bevarar kvadratiska invarianter är symplektiska.*

Bevis. Att flödet φ_t av en ODE är symplektiskt innebär att $\Psi(t)^T J \Psi(t) = J$. Alltså är den kvadratiska funktionen $(x, \Psi) \mapsto \Psi^T J \Psi$ en invariant för det utökade IVP:et (7.4), den är matrisvärd men kan ses som $2d \cdot 2d$ skalärvärda kvadratiska invarianter. Om man då beräknar x_n, Ψ_n med en Runge–Kutta-metod som bevarar kvadratiska invarianter så blir

$$\Psi_n^T J \Psi_n = \Psi_0^T J \Psi_0 = I^T J I = J.$$

Enligt lemma 7.3, innebär detta att $(\frac{\partial}{\partial x_0} x_n)^T J \frac{\partial}{\partial x_0} x_n = J$ vilket är definitionen av en symplektisk metod. □

8 Solsystemet

Vi kommer nu att visa upp metoder som vi har sett i tidigare avsnitt genom att simulera de yttre planeterna i solsystemet samt Pluto. Geometriska numeriska metoder har använts för att simulera solsystemet ända sedan den, för ämnet, viktiga artikeln "Symplectic Maps for the N-body Problem" av Wisdom och Holman 1991 [18], där de använde symplektiska metoder. Symplektiska metoder har efter det använts bland annat för att simulera Solen, Jupiter och Saturnus rörelse under 25000 år i [19] och alla solsystemets åtta planeters rörelse under 50000 år i [20].

Vi kommer presentera N-kropps problemet generellt. Låt K vara antalet kroppar (N används redan som antal tidssteg för numeriska metoder), G vara gravitationskonstanten och p_i, q_i, m_i vare den i:te kroppens rörelsemängd, position respektive massa. Vi kan beskriva systemet som ett hamiltonskt system med hamiltonfunktionen

$$H(p, q) = \frac{1}{2} \sum_{i=1}^{K} \frac{\|p_i\|^2}{m_i} - \sum_{1 \le i < j \le K} \frac{G m_i m_j}{\|q_j - q_i\|},$$

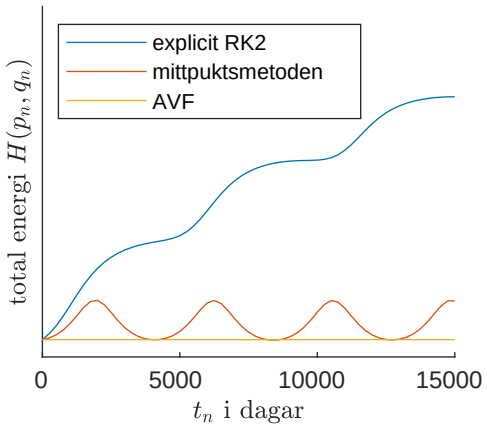
se till exempel [3, sida 13] eller [20]. Insättning av detta i Hamiltons ekvationer (5.1) ger

$$\dot{p}_i(t) = \sum_{\substack{j=1 \\ j\neq i}}^{K} \frac{Gm_i m_j (q_j(t) - q_i(t))}{\|q_j(t) - q_i(t)\|^3}$$
$$\dot{q}_i(t) = \frac{p_i(t)}{m_i}.$$

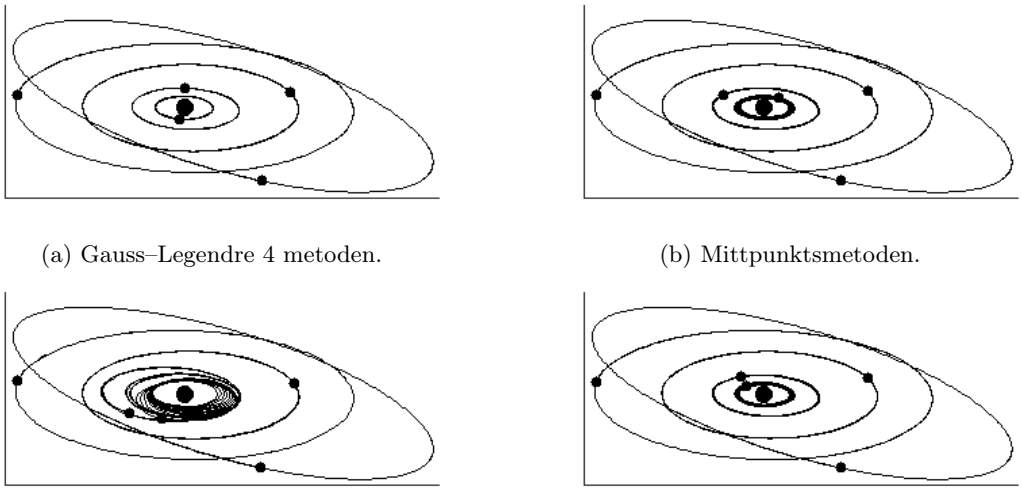
Vi har simulerat rörelsen hos Jupiter, Saturnus, Uranus, Neptunus, Pluto och Solen, det vill säga $K = 6$. Vi har tagit all data från "Geometric Numerical Integration" skriven av Harrier, Lubich och Wanner se [3, sida 13 och 14]. Planeternas initiella rörelsemängd och position är som det var 00:00 den 5:e September 1994.

Vi har använt tre metoder med ordning 2: en explicit Runge–Kutta-metod där nästa steg ges av $x_{n+1} = x_n + \frac{1}{2}h(f(t_n, x_n) + f(t_{n+1}, x_n + hf(t_n, x_n)))$, mittpunktsmetoden (2.6) och AVF (6.2). Vi har också använt Gauss–Legendre-metoden med ordning 4, vars Butcher-tablå är i tabell 4.1. Mittpunktsmetoden och Gauss–Legendre-metoden är symplektiska enligt sats 4.3 och sats 7.4.

Resultatet av simuleringarna visas i figur 8.1, som är solsystemets hamiltonfunktion som funktion av tid, och i figur 8.2, som visar planeternas omloppsbana beroende på numerisk metod. I figur 8.1 ser vi att AVF-metoden exakt bevarar energin av systemet. Detta ger en stabil omloppsbana för planeterna i figur 8.2d. Mittpunktsmetoden bevarar inte energin i figur 8.1; energin oscillerar istället mellan det korrekta värdet och ett för högt värde. Mittpunktsmetoden är däremot symplektiskt och vi ser i figur 8.2b att planeternas omloppsbana fortfarande är stabila. Gauss–Legendre-metoden är också symplektisk och är dessutom av ordning 4 vilket ger ännu stabilare omloppsbanor i figur 8.2a. För explicita RK2 ökar däremot hamiltonfunktionen H vilket vi ser i figur 8.1 och vi ser i figur 8.2c att planeterna närmast solen åker iväg från solen.



Figur 8.1: $H - t$-diagram där H är hamiltonfunktionen, dvs energin av systemet. Steglängden $h = 250$ dagar.



(a) Gauss–Legendre 4 metoden.

(b) Mittpunktsmetoden.

(c) explicita Runge–Kutta 2 metod.

(d) AVF-metoden.

Figur 8.2: Planeternas omloppsbanor i ett solsystem för fyra numeriska metoder simuleras i 100000 dagar med steglängden $h = 250$ dagar.

9 Slutsats

I detta arbete har vi studerat geometriska egenskaper hos ordinära initialvärdesproblem och hur dessa kan bevaras vid numerisk approximation. Vårt fokus har legat i att visa hur strukturer som linjära och kvadratiska invarienter, samt hamiltonfunktioner och symplekticitet kan bevaras med genomtänkta val av numeriska metoder.

Vårt resultat ger flertalet alternativ till numeriska metoder och visar på när de kan användas. För initialvärdesproblem med linjära och kvadratiska invarianter finns det många val av numeriska metoder som bevarar invarianterna. Dock såg vi att det i många fysikaliska problem framkommer komplicerade invarianter som kräver mer avancerade metoder för att bevaras. Till dessa introducerades hamiltonska system och hamiltonfunktioner tillsammans med AVF-metoden och symplektiska metoder.

Det huvudsakliga budskapat vi tar med oss från arbetet är vikten av att studera problemet framför en istället för att bara slänga godtycklig numerisk metod mot problemet. Att använda Eulers explicita metod för att simulera solsystemet ger orimliga resultat. Samtidigt är metoder som AVF och Gauss-Legendre mycket mer komplexa än Eulers metoder och även om de ger bättre resultat för mer komplicerade problem behövs nödvändigtvis inte dessa metoder för enklare problem som exempelvis vattnets autoprotolys.

Även om uppsatsen ger mycket nyttig information inom GNI innefattar den naturligtvis långt ifrån allt. Den enda energi-bevarande metoden som vi har presenterat är AVF-metoden som har ordning 2. Det går att generalisera den för att uppnå högre ordning, se till exempel [21] eller [22]. Det går också att ta fram metoder som liknar AVF, fast som fungerar på mer generella ODE:er, där matrisen i (6.1) inte är konstant [23]. Vi har diskuterat symplektiska Runge–Kutta-metoder, som i princip kan ha hur hög ordning som helst, men vi har inte diskuterat kring hur man hittar parametrarna i Butcher-tabellen för att få en viss ordning. För det finns det flera metoder, se till exempel [13, kapitel 3] eller [3, kapitel III]. Uppsatsen tar heller inte upp projiceringsmetoder som efter varje steg projicerar tillbaka approximationen ifall den strävat iväg från invarianten [3, kapitel IV.4]. Det går också att använda geometrisk numerisk integration för andra problem, till exempel partiella differentialekvationer [24]. För den intresserade läsaren som vill läsa på mer rekommenderar vi starkt boken *Geometrical Numerical Integration* [3].

Referenser

[1] R. MacKay och J. Meiss, *Hamiltonian Dynamical Systems: A REPRINT SELECTION*. CRC Press, 2020, ISBN: 9781000112085.

[2] A. J. Lotka, *Elements of mathematical biology. (tidigare publicerad med titeln Elements of Physical Biology)*. Dover Publications, Inc., New York, 1958, s. xxx+465.

[3] E. Hairer, C. Lubich och G. Wanner, *Geometric Numerical Integration* (Springer Series in Computational Mathematics). Springer, Heidelberg, 2010, vol. 31, s. xviii+644, ISBN: 978-3-642-05157-9.

[4] R. Armellin, P. Di Lizia och M. Berz, "Asteroid close encounters characterization using differential algebra: The case of Apophis", *Celestial Mechanics and Dynamical Astronomy*, årg. 107, s. 451–470, 2011. DOI: 10.1007/s10569-010-9283-5.

[5] M. W. Hirsch, "Systems of differential equations which are competitive or cooperative: III. Competing species", *Nonlinearity*, årg. 1, nr 1, s. 51, febr. 1988. DOI: 10.1088/0951-7715/1/1/003.

[6] C. Chicone, *Ordinary differential equations with applications* (Texts in Applied Mathematics), Third. Springer, Cham, 2024, vol. 34, s. xxii+729, ISBN: 978-3-031-51651-1. DOI: 10.1007/978-3-031-51652-8.

[7] H. Logemann och E. P. Ryan, *Ordinary differential equations* (Springer Undergraduate Mathematics Series). Springer, London, 2014, s. xiv+333, ISBN: 978-1-4471-6397-8. DOI: 10.1007/978-1-4471-6398-5.

[8] L. Fraenkel, D. Gottfridsson och U. Jonasson, *Impuls. Fysik. 2*, 1. utg. Malmö: Gleerups Utbildning AB, 2012, ISBN: 9789140677082.

[9] R. Anders, A. Bengt, O. Louise och A. Ronnie, *Mathematical Modeling in Chemical Engineering*, first. Cambridge University Press, 2014, ISBN: 9781107049697.

[10] G. Stoyan och A. Baran, *Elementary numerical mathematics for programmers and engineers* (Compact Textbooks in Mathematics), Hungarian. Birkhäuser/Springer, Cham, 2016, s. ix+220, ISBN: 978-3-319-44659-2. DOI: 10.1007/978-3-319-44660-8.

[11] W. Gander, M. J. Gander och F. Kwok, *Scientific computing* (Texts in Computational Science and Engineering). Springer, Cham, 2014, vol. 11, s. xviii+905, ISBN: 978-3-319-04324-1. DOI: 10.1007/978-3-319-04325-8. URL: https://doi.org/10.1007/978-3-319-04325-8.

[12] D. F. Griffiths och D. J. Higham, *Numerical Methods for Ordinary Differential Equations* (Springer Undergraduate Mathematics Series). Springer, London, 2010, ISBN: 978-0-85729-147-9.

[13] J. C. Butcher, *Numerical methods for ordinary differential equations*, Second. John Wiley & Sons, Ltd., Chichester, 2008, s. xx+463, ISBN: 978-0-470-72335-7. DOI: 10.1002/9780470753767.

[14] J. C. Butcher, *Numerical methods for ordinary differential equations*, Third. John Wiley & Sons, Ltd., Chichester, 2016, s. xxiii+513, ISBN: 978-1-119-12150-3. DOI: 10.1002/9781119121534.

[15] L. Laverman, P. Atkins och L. Jones, *Chemical Principles: The Quest for Insight*, 7. utg. Macmillan Higher Education, 2016, s. 828, ISBN: 9781319154196.

[16] L. Fraenkel, D. Gottfridsson och U. Jonasson, *Impuls Fysik 1*, 1. utg. Malmö: Gleerups Utbildning AB, 2011, ISBN: 9789140674159.

[17] R. I. McLachlan, G. R. W. Quispel och N. Robidoux, "Geometric integration using discrete gradients", *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, årg. 357, nr 1754, s. 1021–1045, 1999, ISSN: 1364-503X. DOI: 10.1098/rsta.1999.0363.

[18] J. Wisdom och M. Holman, "Symplectic maps for the N-body problem", *Astronomical Journal*, årg. 102, s. 1528–1538, 1991. DOI: 10.1086/115978.

[19] J. Laskar och P. Robutel, "High order symplectic integrators for perturbed Hamiltonian systems", *Celestial Mech. Dynam. Astronom.*, årg. 80, nr 1, s. 39–62, 2001, ISSN: 0923-2958. DOI: 10.1023/A:1012098603882.

[20] S. Blanes, F. Casas, A. Farrés, J. Laskar, J. Makazaga och A. Murua, "New families of symplectic splitting methods for numerical integration in dynamical astronomy", *Appl. Numer. Math.*, årg. 68, s. 58–72, 2013, ISSN: 0168-9274. DOI: 10.1016/j.apnum.2013.01.003.

[21] G. R. W. Quispel och D. I. McLaren, "A new class of energy-preserving numerical integration methods", *J. Phys. A*, årg. 41, nr 4, s. 045206, 7, 2008, ISSN: 1751-8113. DOI: 10.1088/1751-8113/41/4/045206.

[22] E. Hairer, "Energy-preserving variant of collocation methods", *JNAIAM. J. Numer. Anal. Ind. Appl. Math.*, årg. 5, nr 1-2, s. 73–84, 2010, ISSN: 1790-8140.

[23] D. Cohen och E. Hairer, "Linear energy-preserving integrators for Poisson systems", *BIT*, årg. 51, nr 1, s. 91–101, 2011, ISSN: 0006-3835. DOI: 10.1007/s10543-011-0310-z.

[24] R. J. LeVeque, *Numerical methods for conservation laws* (Lectures in Mathematics ETH Zürich), Second. Birkhäuser Verlag, Basel, 1992, s. x+214, ISBN: 3-7643-2723-5. DOI: 10.1007/978-3-0348-8629-1.

[25] N. L. Carothers, *Real analysis*. Cambridge University Press, Cambridge, 2000, s. xiv+401, ISBN: 0-521-49756-6. DOI: 10.1017/CBO9780511814228.

[26] G. R. W. Quispel och D. I. McLaren, "A new class of energy-preserving numerical integration methods", *J. Phys. A*, årg. 41, nr 4, s. 045206, 7, 2008, ISSN: 1751-8113. DOI: 10.1088/1751-8113/41/4/045206.

10 AI-användande

I detta arbete har AI använts. Vissa av oss använt ChatGPT för att få förslag på svenska översättningar av matematiska ord som förekommer på engelska, grammatik och LaTeX-koder.

A Appendix 1 – teori

A.1 Fixpunktsiterationer

Det är ofta vi vill lösa ekvationer på formen $x = F(x)$. I en del fall kan vi göra det approximativt genom att börja med något startvärde x_0 och göra iterationer $x_{n+1} = F(x_n)$. Om följden $(x_n)_{n=0}^{\infty}$ har ett gränsvärde x, och F är kontinuerlig måste $\lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} F(x_n) \implies x = F(x)$. För att garantera att gränsvärdet x existerar behöver vi ett krav på F.

Definition A.1 (kontraktionsavbildning, mer generellt finns i [25, sida 98])**.** En funktion $F : X \to Y$, där $X, Y \subseteq \mathbb{R}^d$, är en kontraktionsavbildning om $\|F(x) - F(y)\| \leq \theta\|x - y\|$ för något $\theta \in [0, 1)$ och alla $x, y \in X$.

Vi kan nu formulera resultatet.

Sats A.1 (Banachs fixpunktssats, mer generellt finns i [25, sida 98])**.** *Låt $D \subseteq \mathbb{R}^d$ vara sluten och icketom och $F : D \to D$ vara en kontraktionsavbilning. Då finns det exakt en lösning $x \in D$ till $x = F(x)$ som är gränsvärdet av följden $(x_n)_{n=0}^{\infty}$ där $x_{n+1} = F(x_n)$ och x_0 är en godtycklig punkt i D.*

Vi bevisar inte satsen här. Det är viktigt att $F : D \to D$, det vill säga att $F(D) \subseteq D$. Till exempel har $x = 1+x^2$ ingen reell lösning trots att $F(x) = 1+x^2$ är en kontraktionsavbilning på $[-1/3, 1/3]$.

Nu kommer ett ganska tekniskt exempel på användning av fixpunktsiterationer som har betydelse för vårt arbete.

Exempel A.1. Vi kommer visa att vi kan lösa ekvationen i Eulers implicita stegmetod (2.5), för ODE:en $\dot{x}(t) = f(x(t))$, där $f : U \to \mathbb{R}^d$ är kontinuerligt deriverbar och $U \subseteq \mathbb{R}^d$ är öppen. Vi behöver lösa ekvationen $k = f(x_n + hk) = F(k)$, vi kommer använda sats A.1 för att visa att denna ekvationen alltid har en lösning om h är tillräckligt litet.

Om f är kontinuerligt deriverbar i en omgivning av x_n, så är f Lipschitz-kontinuerlig i en boll $\overline{B_\varepsilon(x_n)} = \{x \in \mathbb{R}^d : \|x - x_n\| \leq \varepsilon\}$. Låt Lipschitz-konstanten vara L. Välj $h > 0$ så att

$$Lh < 1,$$
$$Lh(\|f(x_n)\| + \varepsilon) \leq \varepsilon \tag{A.1}$$
$$h(\|f(x_n)\| + \varepsilon) \leq \varepsilon \implies \forall k \in \overline{B_\varepsilon(f(x_n))} : x_n + hk \in \overline{B_\varepsilon(x_n)} \tag{A.2}$$

Då är $F(\overline{B_\varepsilon(f(x_n))}) \subseteq \overline{B_\varepsilon(f(x_n))}$ ty om $k \in \overline{B_\varepsilon(f(x_n))}$ så gäller $\|F(k) - f(x_n)\| = \|f(x_n + hk) - f(x_n)\| \leq_{(A.2)} Lh\|k\| \leq Lh(\|f(x_n)\| + \varepsilon) \leq_{(A.1)} \varepsilon$. Funktionen F är en kontraktionsavbildning på $\overline{B_\varepsilon(f(x_n))}$ eftersom $\|F(k) - F(\ell)\| \leq_{(A.2)} Lh\|k - \ell\|$. Alltså finns enligt sats A.1 unik k så att $k = F(k)$ i $\overline{B_\varepsilon(f(x_n))}$.

Ett liknande argument kan användas för generella implicita Runge–Kutta-metoder för att visa att det alltid går att ta ett steg, och det ger en enkel metod för att lösa ekvationerna numeriskt. Notera att steglängden h väljs beroende på x_n, så man kan behöva ta mindre och mindre steg. Om $f : \mathbb{R}^d \to \mathbb{R}^d$ och är globalt Lipschitz-kontinuerlig så kan man visa att det går att välja ett h som fungerar oavsett x_n. ■

A.2 Bevis

Bevis till sats 4.1:

Sats 4.1 (Linjära invarianters bevarande)**.** *Alla Runge–Kutta-metoder* (3.1) *bevarar linjära invarianter för ett IVP på formen* (2.1)*.*

Bevis. Antag att vi har ett IVP på formen (2.1) med en linjär invariant $c^T x(t) = C$ för någon konstant C. Vi vill visa att invarianten kvarstår för den numeriskt uppskattade lösningen given av en godtycklig Runge–Kutta-metod (3.1), alltså $c^T x_{n+1} = c^T x_n$ för $n \in \{0, 1, ..., N - 1\}$.

Vi vet att $c^T x(t) = C$ vilket ger oss att $c^T f(t, x(t)) = c^T \dot{x}(t) = \frac{d}{dt}(c^T x(t)) = \frac{d}{dt}(C) = 0$ för godtyckligt t och $x(t)$. Enligt (3.1) är

$$c^T x_{n+1} = c^T x_n + c^T h \sum_{i=1}^{s} b_i k_i = c^T x_n + h \sum_{i=1}^{s} b_i c^T k_i.$$

Eftersom $k_i = f\left(t_n + c_i h, x_n + h \sum_{j=1}^{s} a_{ij} k_j\right)$ kan vi se att den sista termen blir 0 eftersom

$$c^T k_i = c^T f\left(t_n + c_i h, x_n + h \sum_{j=1}^{s} a_{ij} k_j\right) = c^T f(\tau_n, \xi_n) = 0,$$

där $\tau_n = t_n + c_i h$ och $\xi_n = x_n + h \sum_{j=1}^{s} a_{ij} k_j$, vilket ger oss $c^T x_{n+1} = c^T x_n$. □

Bevis till sats 4.2:
Sats 4.2. *Låt $x(t)$ vara lösningen till ett IVP på formen (2.1) och låt $A \in \mathbb{R}^{N \times N}$ vara en symmetrisk matris. Då gäller att $x(t)^T A f(t, x(t)) = 0$ för alla t om och endast om $x(t)^T A x(t)$ är en invariant till IVP (2.1).*

Bevis. Först vill vi visa att $x(t)^T A f(t, x(t)) = 0$ om $x(t)^T A x(t)$ är en invariant. Antag ett ODE problem på formen (2.1) med en kvadratisk invariant $x(t)^T A x(t)$ där A är kvadratisk och symmetrisk.

Eftersom invarianten är konstant är tidsderivatan av den 0. Vi använder produktregeln och får att

$$0 = \frac{d}{dt}\left(x(t)^T A x(t)\right) = \dot{x}(t)^T A x(t) + x(t)^T A \dot{x}(t).$$

Eftersom A är symmetrisk ser vi att $(x(t)^T A \dot{x}(t))^T = \dot{x}(t)^T A x(t)$. Detta låter oss skriva om uttrycket enligt

$$\dot{x}(t)^T A x(t) + x(t)^T A \dot{x}(t) = 2x(t)^T A \dot{x}(t),$$

vilket ger oss att

$$0 = 2x(t)^T A \dot{x}(t) = 2x(t)^T A f(t, x(t)) \implies x(t)^T A f(t, x(t)) = 0.$$

Nu vill vi visa omvändningen. Alltså att $x(t)^T A f(t, x(t)) = 0$ implicerar att $x(t)^T A x(t)$ är en invariant. Antag att vi har ett IVP på formen (2.1) och att det för en kvadratisk symmetrisk matris A gäller att $x(t)^T A f(t, x) = 0$. Vi använder samma ekvivalenser som ovan men går åt andra hållet och får att

$$0 = 2x(t)^T A f(t, x(t)) = 2x(t)^T A \dot{x}(t) = \dot{x}(t)^T A x(t) + x(t)^T A \dot{x}(t) = \frac{d}{dt}\left(x(t)^T A x(t)\right).$$

Eftersom derivatan av $(x(t)^T A x(t))$ är noll är $x(t)^T A x(t)$ konstant och därmed en invariant. Implikationen åt båda riktningar är bevisad och så är därmed även ekvivalensen. □

Bevis till sats 4.3:
Sats 4.3 (Kvadratiska invarianters bevarande)**.** *Alla Runge-Kutta-metoder*(3.1) *som uppfyller $a_{ij} b_i + a_{ji} b_j = b_i b_j, \forall i, j = 1, 2, ..., s$ bevarar kvadratiska invarianter.*

Bevis. Antag att vi har ett IVP på formen (2.1) med en kvadratisk invariant $x^T(t) A x(t)$ där A är kvadratisk och symmetrisk. Likt sats 4.1 skriver vi om x_{n+1} enligt (3.1):

$$x_{n+1}^T A x_{n+1} = (x_n + h \sum_{i=1}^{s} b_i k_i)^T A (x_n + h \sum_{j=1}^{s} b_j k_j) =$$

$$= x_n^T A x_n + x_n^T A h \sum_{j=1}^{s} b_j k_j + h \sum_{i=1}^{s} b_i k_i^T A x_n + h^2 \sum_{i,j=1}^{s} b_i k_i^T A b_j k_j. \tag{A.3}$$

Om vi kan visa att det tre sista termerna summerar till noll under villkoren ser vi att $x_{n+1}^T A x_{n+1} = x_n^T A x_n$, alltså att kvadratiska invarianter bevaras. För att visa detta börjar vi med att behandla de två mittentermerna.

$$x_n^T A k_i = (x_n^T + h\sum_{j=1}^{s} a_{ij} k_j^T) A f(t_n + c_i h, x_n + h\sum_{j=1}^{s} a_{ij} k_j) - h\sum_{j=1}^{s} a_{ij} k_j^T A k_i$$

Enligt sats 4.2 är $\xi^T A f(\tau, \xi) = 0$ för alla ξ och τ. Genom att välja $\xi = x_n + h\sum_{j=1}^{s} a_{ij} k_j$ och $\tau = t_n + c_i h$ får vi

$$(x_n^T + h\sum_{j=1}^{s} a_{ij} k_j^T) A f(t_n + c_i h, h\sum_{j=1}^{s} a_{ij} k_j) = \xi^T A f(\tau, \xi) = 0$$

vilket lämnar oss med

$$x_n^T A k_i = -h\sum_{j=1}^{s} a_{ij} k_j^T A k_i \implies x_n^T A h\sum_{j=1}^{s} b_j k_j = -h^2 \sum_{i,j=1}^{s} a_{ij} k_i^T A b_j k_j.$$

Eftersom att A är symmetrisk är $k_i^T A x_n = x_n^T A k_i$. Detta låter oss göra samma sak med den andra mittentermen i huvuduttrycket (A.3)

$$h\sum_{i=1}^{s} b_i k_i^T A x_n = h\sum_{i=1}^{s} b_i x_n^T A k_i = -h^2 \sum_{i,j=1}^{s} a_{ji} k_i^T A b_i k_j.$$

Vi tar med oss dessa omskrivningar och lägger in dem i huvuduttrycket (A.3). Då får vi

$$x_{n+1}^T A x_{n+1} = x_n^T A x_n + h^2 \sum_{i,j=1}^{s} k_i^T A k_j (b_i b_j - a_{ji} b_j - a_{ij} b_i).$$

Men eftersom att $a_{ij} b_i + a_{ji} b_j = b_i b_j \implies b_i b_j - a_{ji} b_j - a_{ij} b_i = 0$ är hela summan 0 och kvar blir endast $x_{n+1}^T A x_{n+1} = x_n^T A x_n$. Alltså bevaras invarianten av RK-metoden. □

Bevis till sats 6.1:
Sats 6.1. *För problem på formen* (6.1) *bevarar AVF-metoden* (6.2) H*, det vill säga* $H(x_n) = H(x_0)$ *för alla* $n = 0, \ldots, N$.

Bevis. Det här beviset finns i [26]. Det räcker att visa att $H(x_{n+1}) = H(x_n)$ för alla steg n. Eftersom $\dot{x}(t) = M\nabla H(x(t))$ kan vi skriva (6.2a) som

$$k_n = M \int_0^1 \nabla H(x_n + shk_n) ds.$$

Vi multiplicerar båda leden från vänster med $(\int_0^1 \nabla H(x_n + shk_n)\, ds)^T$ och ser att högerledet, eftersom M är antisymmetrisk, blir

$$\left(\int_0^1 \nabla H(x_n + shk_n)\, ds\right)^T M \int_0^1 \nabla H(x_n + shk_n)\, ds = 0.$$

Efter att vi använt att skalärprodukten $(a, b) \mapsto a^T b$ är kommutativ blir vänsterledet

$$k_n^T \int_0^1 \nabla H(x_n + shk_n)\, ds.$$

Om vi nu förlänger vänsterledet med h och flyttar in hk_n^T innanför integralen kan vi använda kedjereglen omvänt och få

$$\frac{h}{h} k_n^T \int_0^1 \nabla H(x_n + shk_n)\, ds = \frac{1}{h} \int_0^1 hk_n^T \nabla H(x_n + shk_n)\, ds = \frac{1}{h} \int_0^1 \frac{d}{ds} H(x_n + shk_n)\, ds.$$

Enligt analysens huvudsats blir integralen $H(x_n + hk_n) - H(x_n)$. Ställer vi nu vänsterledet mot högerledet ser vi att $\frac{1}{h}(H(x_n + hk_n) - H(x_n)) = 0 \implies H(x_{n+1}) = H(x_n)$.

□

Bevis till sats 6.2:
Sats 6.2. *För ett IVP* (2.1) *där f har en kontinuerlig andraderivata, så är AVF-metoden* (6.2) *av ordning* 2.

Vi börjar med att definiera lokalt trunkeringsfel för en numerisk metod innan beviset.

Definition A.2 (LTE)**.** [12, kapitel 9.2] Lokalt trunkeringsfel T_{n+1} (LTE) är definierad enligt

$$T_{n+1} := x(t_{n+1}) - x_{n+1}$$

under antagandet att $x_n = x(t_n)$.

Bevis. Vi visar att det lokala trunkeringsfelet är $\mathcal{O}(h^3)$. Vi taylorutvecklar integranden i (6.2a),

$$\begin{aligned} k_n &= \int_0^1 \left(f(x_n) + f'(x_n)shk_n + \mathcal{O}(h^2)\right) ds, \\ &= \int_0^1 \left(f(x_n) + f'(x_n)shf(x_n) + \mathcal{O}(h^2)\right) ds, \\ &= f(x_n) + \frac{h}{2} f'(x_n)f(x_n) + \mathcal{O}(h^2). \end{aligned}$$

Alltså är $x_{n+1} = x_n + hk_n = x_n + hf(x_n) + \frac{1}{2}h^2 f'(x_n)f(x_n) + \mathcal{O}(h^3)$. Vi taylorutvecklar $x(t_{n+1})$ runt t_n

$$\begin{aligned} x(t_n + h) &= x(t_n) + h\dot{x}(t_n) + \frac{1}{2}h^2\ddot{x}(t_n) + \mathcal{O}(h^3) \\ &= x(t_n) + hf(x(t_n)) + \frac{1}{2}h^2 f'(x(t_n))f(x(t_n)) + \mathcal{O}(h^3). \end{aligned}$$

Givet att $x_n = x(t_n)$ så är $x_{n+1} - x(t_{n+1}) = \mathcal{O}(h^3)$, vilket också innebär att globala felet är $\mathcal{O}(h^2)$. □

Bevis till sats 7.1:
Sats 7.1 ([3, kapitel VI.2])**.** *Flödet φ_t för ett hamiltonskt system 5.1 är symplektiskt.*

När vi bevisar satsen använder vi likheterna $J^{-1} = J^T = -J$ som är enkla att verifiera.

Bevis. Om vi låter $x(t) = (p(t), q(t))$ kan vi skriva alla hamiltonska system som $\dot{x}(t) = J^T\nabla H(x(t))$. Eftersom $\varphi_0'(x_0)^T J\varphi_0'(x_0) = I^T JI = J$ räcker det att visa att $\frac{d}{dt}(\varphi_t'(x_0)^T J\varphi_t'(x_0)) = 0$ för alla t and x_0. Vi börjar med

$$\dot{\varphi}_t'(x_0) = \frac{\partial}{\partial x_0}(\dot{\varphi}_t(x_0)) = \frac{\partial}{\partial x_0}(J^T\nabla H(\varphi_t(x_0))) = J^T\nabla^2 H(\varphi_t(x_0))\varphi_t'(x_0),$$

där $\nabla^2 H$ är hessematrisen till H, vilket är en symmetrisk matris. Alltså är

$$\begin{aligned} &\frac{d}{dt}\left(\varphi_t'(x_0)^T J\varphi_t'(x_0)\right) = \dot{\varphi}_t'(x_0)^T J\varphi_t'(x_0) + \varphi_t'(x_0)^T J\dot{\varphi}_t'(x_0) \\ &= \left(J^T\nabla^2 H(\varphi_t(x_0))\varphi_t'(x_0)\right)^T J\varphi_t'(x_0) + \varphi_t'(x_0)^T JJ^T\nabla^2 H(\varphi_t(x_0))\varphi_t'(x_0) = \\ &= \varphi_t'(x_0)^T \underbrace{\nabla^2 H(\varphi_t(x_0))^T}_{=\nabla^2 H(\varphi_t(x_0))} \underbrace{JJ}_{=-I} \varphi_t'(x_0) + \varphi_t'(x_0)^T \underbrace{JJ^T}_{=I} \nabla^2 H(\varphi_t(x_0))\varphi_t'(x_0) = \\ &= -\varphi_t'(x_0)^T\nabla^2 H(\varphi_t(x_0))\varphi_t'(x_0) + \varphi_t'(x_0)^T\nabla^2 H(\varphi_t(x_0))\varphi_t'(x_0) = 0. \end{aligned}$$

Med det följer, enligt argumentet i början, att $\varphi_t'(x_0)^T J\varphi_t'(x_0)$ är konstant, dvs

$$\varphi_0'(x_0)^T J\varphi_0'(x_0) = J.$$

□

Bevis till sats 7.2:
Sats 7.2 ([3, kapitel VI.3] eller [12, kapitel 15.3])**.** *Symplektiska Euler-metoden* (7.2) *är symplektisk.*

Ekvationen för symplektiska Eulers metod (7.2) är

$$p_{n+1} = p_n - h\frac{\partial H}{\partial q}(p_{n+1}, q_n)$$
$$q_{n+1} = q_n + h\frac{\partial H}{\partial p}(p_{n+1}, q_n).$$

Bevis. Vi behöver visa att kravet för att vara en symplektiskt metod (7.1) gäller, vilket kräver Jacobi-matrisen. Först beräknas partiella derivatorna av p_{n+1}, q_{n+1} med avseende på p_n, q_n, vilket ger

$$\frac{\partial p_{n+1}}{\partial p_n} = I - hH_{pq}(p_{n+1}, q_n)\frac{\partial p_{n+1}}{\partial p_n}, \qquad \frac{\partial p_{n+1}}{\partial q_n} = -hH_{qq}(p_{n+1}, q_n) - hH_{pq}(p_{n+1}, q_n)\frac{\partial p_{n+1}}{\partial q_n},$$
$$\frac{\partial q_{n+1}}{\partial p_n} = hH_{pp}\frac{\partial p_{n+1}}{\partial p_n}, \qquad \frac{\partial p_{n+1}}{\partial q_n} = I + hH_{qp}(p_{n+1}, q_n) + hH_{pp}(p_{n+1}, q_n)\frac{\partial p_{n+1}}{\partial q_n},$$

där H_{pq} är en matris där varje element är partiella derivator $\frac{\partial^2 H}{\partial p_j \partial q_i}$ för rad i och kolonn j. Dessa kan skrivas om på matrisform med hjälp av Jacobi-matrisen som sedan kan lösas ut

$$\begin{bmatrix} I + hH_{pq}(p_{n+1}, q_n) & 0 \\ -hH_{pp}(p_{n+1}, q_n) & I \end{bmatrix} \frac{\partial(p_{n+1}, q_{n+1})}{\partial(p_n, q_n)} = \begin{bmatrix} I & -hH_{qq}(p_{n+1}, q_n) \\ 0 & I + hH_{qp}(p_{n+1}, q_n) \end{bmatrix},$$
$$\implies \frac{\partial(p_{n+1}, q_{n+1})}{\partial(p_n, q_n)} = \begin{bmatrix} I + hH_{pq}(p_{n+1}, q_n) & 0 \\ -hH_{pp}(p_{n+1}, q_n) & I \end{bmatrix}^{-1} \begin{bmatrix} I & -hH_{qq}(p_{n+1}, q_n) \\ 0 & I + hH_{qp}(p_{n+1}, q_n) \end{bmatrix}.$$

Vi beräknar nu först inversmatrisen och därefter Jacobanen och dess transponat, mha några satser för blockmatriser. Definiera matrisen $A = I + H_{pq}$. För inversmatrisen får vi

$$\begin{bmatrix} I + hH_{pq}(p_{n+1}, q_n) & 0 \\ -hH_{pp}(p_{n+1}, q_n) & I \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ hH_{pp}A^{-1} & I \end{bmatrix}$$

vilket kan verifieras genom att matrismultiplicera inversmatrisen och dess vanliga matris och se att identitetsmatrisen erhålls. Med en matrismultiplikation blir Jacobi-matrisen och dess transponat

$$\frac{\partial(p_{n+1}, q_{n+1})}{\partial(p_n, q_n)} = \begin{bmatrix} A^{-1} & -hA^{-1}H_{qq} \\ -hH_{pp}A^{-1} & -h^2H_{pp}A^{-1}H_{qq} + (I + hH_{qp}) \end{bmatrix} \tag{A.4}$$
$$\implies \left(\frac{\partial(p_{n+1}, q_{n+1})}{\partial(p_n, q_n)}\right)^T = \begin{bmatrix} A^{-T} & hA^{-T}H_{pp}^T \\ -hH_{qq}^TA^{-T} & -h^2H_{qq}^TA^{-T}H_{pp}^T + A \end{bmatrix} \tag{A.5}$$

där $H_{qp}^T = H_{pq}$ har använts. Insättning av ekvation (A.4) och (A.5) till vänsterledet av ekvation (7.1) ger slutligen högerledet efter matrismultiplicering och förenkling, det vill säga

$$\left(\frac{\partial(p_{n+1}, q_{n+1})}{\partial(p_n, q_n)}\right)^T J\frac{\partial(p_{n+1}, q_{n+1})}{\partial(p_n, q_n)} = J.$$

□

Bevis till lemma 7.3:
Lemma 7.3 (se sida 191 i [3])**.** *Vi föreställer oss att vi löser* (7.3) *med en Runge–Kutta-metod och får approximativa punkter* $x_0, x_1, \ldots, x_N$*. Vi löser också* (7.4) *med samma Runge–Kutta-metod och samma steglängd och får approximativa punkter* $x_0, x_1, \ldots, x_N, \Psi_0, \Psi_1, \ldots, \Psi_N$ *(den numeriska lösningen för* x *kommer vara lika). Då kommer* $\frac{\partial}{\partial x_0}x_n = \Psi_n$ *för alla* n*.*

Det autonoma IVP:et (7.3) som nämns i lemmat är

$$\begin{cases} \dot{x}(t) = f(x(t)) \\ x(0) = x_0, \end{cases}$$

och den utökade IVP:et (7.4) är

$$\begin{cases} \dot{x}(t) = f(x(t)) \\ \dot{\Psi}(t) = f'(x(t))\Psi(t) \\ x(0) = x_0; \Psi(0) = I. \end{cases}$$

Bevis. Om en Runge–Kutta-metod används på det utökade systemet (7.4), får vi

$$\begin{aligned} (x_{n+1}, \Psi_{n+1}) &= (x_n, \Psi_n) + h \sum_{i=1}^{s} b_i (k_i, \ell_i) \\ (k_i, \ell_i) &= (f(X_i), f'(X_i)(\Psi_n + \textstyle\sum_{j=1}^{s} a_{ij} \ell_j)) \end{aligned}$$

där $X_i = (x_n + h \sum_{j=1}^{s} a_{ij} k_j)$. Ekvationerna som påverkar x_n är samma som dem man får av att använda Runge–Kutta-metoden på (7.3), alltså är de numeriska lösningarna av x samma för (7.3) och (7.4). Vi ska visa att $\frac{\partial}{\partial x_0} x_n = \Psi_n$ för alla n. Vi utför ett induktionsbevis. Vi har basfallet eftersom $\frac{\partial}{\partial x_0} x_0 = I = \Psi_0$. Antag att $\frac{\partial}{\partial x_0} x_n = \Psi_n$ för något n. För nästa steg deriverar vi ekvationen för k_i med x_0 och får att

$$\begin{aligned} \frac{\partial k_i}{\partial x_0} &= f'(X_i) \frac{\partial X_i}{\partial x_0} = f'(X_i) \frac{\partial}{\partial x_0} \left(x_n + h \sum_{j=1}^{s} a_{ij} k_j \right), \\ &= f'(X_i) \left(\frac{\partial x_n}{\partial x_0} + h \sum_{j=1}^{s} a_{ij} \frac{\partial k_j}{\partial x_0} \right), \\ &= f'(X_i) \left(\Psi_n + h \sum_{j=1}^{s} a_{ij} \frac{\partial k_j}{\partial x_0} \right). \end{aligned}$$

Vi konstaterar att $\dfrac{\partial k_i}{\partial x_0} = \ell_i$ eftersom de är lösningen till samma ekvation. Alltså är

$$\frac{\partial x_{n+1}}{\partial x_0} = \frac{\partial x_n}{\partial x_0} + h \sum_{i=1}^{s} b_i \frac{\partial k_i}{\partial x_0} = \Psi_n + h \sum_{i=1}^{s} b_i \ell_i = \Psi_{n+1}.$$

□

B Appendix 2 – kod

I detta avsnitt presenteras några MATLAB-koder som har använts. Koder för numeriska metoder presenteras först och sedan presenteras koder för solsystemet.

B.1 Allmäna metoder

Här är MATLAB-kod för några allmänna numeriska metoder. De ger approximativa lösningar till problem på formen (2.1). För alla dessa MATLAB-funktioner ges initialvärde `x0` som en kolonnvektor och högerledet `f` ges som en funktion som tar en skalär och en kolonnvektor och ger en kolonnvektor. Den approximativa lösningen `x` ges som en $d \times (N+1)$ matris där N är antalet tidssteg.

En explicit Runge-Kutta-metod med ordning 2 där nästa steg ges av $x_{n+1} = x_n + \frac{1}{2}h[f(t_n, x_n) + f(t_{n+1}, x_n + hf(t_n, x_n))]$.

```
function [t, x] = RK2(f, t0, x0, h, N)
% Beräkna numerisk lösning till ODE med en Runge-Kutta-metod
% av ordning 2
% In: ODE:ens högerled f, initialtid t0, initialvärde x0,
steglängd h och
%     antal tidssteg N.
% Ut: tidsgitter t och approximativ lösning x
    t = t0 + h*(0:N);
    x = zeros(length(x0), N+1);
    x(:, 1) = x0;
    for n = 1:N
        k1 = f(t(n), x(:,n));
        k2 = f(t(n)+h, x(:,n) + h*k1);
        x(:, n+1) = x(:,n) + h*0.5*(k1+k2);
    end
end
```

Mittpunktsmetoden är en implicit metod som beskrivs i avsnitt 2.2. Nästa steg ges av $x_{n+1} = x_n + f(t_n + h/2, (x_n + x_{n+1})/2)$.

```
function [t,x] = mittpunkt(f,t0, x0, h, N)
% Beräkna numerisk lösning till ODE med mittpunktsmetoden
% In: ODE:ens högerled f, initialtid t0, initialvärde x0,
%     steglängd h och antal tidssteg N.
% Ut: tidsgitter t och approximativ lösning x
    t = t0 + h*(0:N);
    x = zeros(length(x0), N+1);
    x(:,1) = x0;
    for n = 1:N
        k = f(t(n),x(:,n));
        for iter = 1:20 %fixpunktsiterationer
            k = f(t(n)+h/2,x(:,n)+h*k/2);
        end
        x(:,n+1) = x(:,n) + h*k;
    end
end
```

Gauss-Legendre-metoden med ordning 4 är en implicit Runge–Kutta-metod med Butcher-Tablå i Tabell 4.1

```
function [t, x] = GL4(f, t0, x0, h, N)
% Beräkna numerisk lösning till ODE med Gauss-Legendre-metoden
% av ordning 4
```

```
% In: ODE:ens högerled f, initialtid t0, initialvärde x0,
%      steglängd h och antal tidssteg N.
% Ut: tidsgitter t och approximativ lösning x
    a12 = 1/4 - sqrt(3)/6;
    a21 = 1/4 + sqrt(3)/6;
    c1 = 1/4 + a12; c2 = a21 + 1/4;

    t = t0 + h*(0:N);
    x = zeros(length(x0), N+1);
    x(:,1) = x0;

    for n=1:N
        k1 = f(t(n),x(:,n));
        k2 = f(t(n),x(:,n));
        for iter = 1:20 % fixpunktsiterationer
            k1ny = f(t(n) + h*c1, x(:,n) + h*( k1/4  + a12*k2));
            k2ny = f(t(n) + h*c2, x(:,n) + h*(a21*k1 +  k2/4 ));
            k1 = k1ny; k2 = k2ny;
        end
        x(:,n+1) = x(:,n) + h*0.5*(k1+k2);
    end
end
```

B.2 Solsystemet

Nu kommer kod för att implementera föregående metoder och AVF-metoden för att simulera solsystemet som beskrivs i kapitel 8. Som innan är K antalet kroppar, p_i, q_i är rörelsemängd respektive position för kropp i, $H(p,q)$ är den totala energin i systemet och N är antalet tidssteg för en numerisk lösning. Vi låter p^n, q^n vara de numeriska approximationerna av $p(t_n), q(t_n)$, vi skriver inte "p_n, q_n" eftersom indexering redan används för att specificera kropp. Det naturliga sättet att räkna på den här ODE:en är att låta p^n, q^n vara $3 \times K$ matriser. I vår Matlab-kod låter vi rörelsemängderna `p` och positionerna `q` vara $3 \times K \times (N+1)$ fält (arrays).

Funktion för att beräkna rörelsemängdens derivata $\dot{p} = -\nabla_q H(p,q)$ som ges av

$$\dot{p}_i = \sum_{\substack{j=1 \\ j \neq i}}^{K} \frac{G m_i m_j (q_j - q_i)}{\|q_j - q_j\|^3}.$$

```
function Dp = solsystempderivata(q, m, G)
% Beräkna dp/dt = -dH/dq för solsystemet
% In: postitioner q, massor m och gravitationskonstanten G
% Ut: rörelsemängderna p:s derivata
    K = length(m);
    Dp = zeros(3, K);
    for i = 1:K
        j = 1:K ~= i;
        qdiff = q(:, j) - q(:, i);
        Dp(:,i) = G*m(i)*sum(m(j).*qdiff./vecnorm(qdiff).^3, 2);
    end
end
```

Högerledet för solsystemet på formen (2.1). Detta är för kunna använda bland annat funktionerna som är i B.1 tidigare i appendix.

```
function Dx = solsystemfaelt(x, m, G)
```

```
% Beräkna högerledet f(x) för solsystemet skrivet på formen
% dx/dt(t) = f(x(t)).
% In: punkt x, massor m och gravitationskonstant G
% Ut: högerledet f(x)
    K = length(m);
    p = reshape(x(1     : 3*K), 3, K);
    q = reshape(x(3*K+1 : end), 3, K);
    Dx = [reshape(solsystempderivata(q,m,G), 3*K, 1)
          reshape(p./m, 3*K, 1)                      ];
end
```

Beräkning av hamiltonfunktionen $H(p, q)$.

```
function H = solsystemHamiltonian(p,q,m,G)
% In: rörelsemängder p, positioner q, massor m, gravitationskonstanten G.
%     p och q ges som 3*K*(N+1) arrayer, där K är antalet kroppar och N
%     antalet tidssteg
% Ut: Hamiltonfunktionen (totala energin) i alla tider som radvektor
    K = length(m);   % Antal kroppar
    H = 0.5*sum(dot(p,p)./m, 2); % Total rörelseenergi
    for i = 2:K      % Total Lägesenergi
        j = 1:i-1;
        H = H - G*m(i)*sum(m(j)./vecnorm(q(:,j,:)-q(:,i,:)), 2);
    end
    H = reshape(H, 1, []); % Gå från 1 * 1 * N till 1 * N
end
```

Huvudfil med kod för att skapa figur 8.1 och figur 8.2.

```
% gravitationskonstanten
G  = ...
% kropparnas massor
m  = ...
% initialpositioner
q0 = ...
% initiella rörelsemängder
p0 = ...
% antal kroppar
K  = length(m);

% f och x0 om man skriver problemet på formen
%       dx/dt(t) = f(x(t))
%          x(t0) = x0
% med x = (p,q).
f = @(t, x) solsystemfaelt(x, m, G);
x0 = [reshape(p0, 3*K, 1); reshape(q0, 3*K, 1)];

%% Energi-bild
h = 250;
dagar = 15000;
N = dagar/h;
clf
hold on

[t, x] = RK2(f, 0, x0, h, N);
p = reshape(x(1     : 3*K, :), 3, K, N+1);
q = reshape(x(3*K+1 : end, :), 3, K, N+1);
```

```
H = solsystemHamiltonian(p,q,m,G);
plot(t,H)

[t, x] = mittpunkt(f, 0, x0, h, N);
p = reshape(x(1     : 3*K, :), 3, K, N+1);
q = reshape(x(3*K+1 : end, :), 3, K, N+1);
H = solsystemHamiltonian(p,q,m,G);
plot(t,H)

[t,p,q] = solsystemAVF(0,p0,q0,m,G,h,N);
H = solsystemHamiltonian(p,q,m,G);
plot(t,H)

hold off
xlabel('$t_n$ i dagar', 'Interpreter','latex')
ylabel('total energi $H(p_n,q_n)$', 'Interpreter','latex')
legend('explicit RK2', 'mittpuktsmetoden', 'AVF', 'Location','northwest')

%% rita solsystemet
h = 250;
dagar = 100000;
N = dagar/h;

[t, x] = GL4(f, 0, x0, h, N); %man kan byta metod här
p = reshape(x(    1 : 3*K, :), 3, K, N+1);
q = reshape(x(3*K+1 : end, :), 3, K, N+1);
%[t,p,q] = solsystemAVF(0,p0,q0,m,G,h,N);

qrel = q - q(:,1,:); % Position relativ till solen
clf
hold on
scatter3(0, 0, 0, 40, 'filled', 'k')
for i = 2 : K
    % Matlab vill inte plotta när det är 1 * 1 * (N+1),
    % så vi gör om till radvektorer.
    x = reshape(qrel(1, i, :), 1, []);
    y = reshape(qrel(2, i, :), 1, []);
    z = reshape(qrel(3, i, :), 1, []);
    plot3(x, y, z, 'k');
    scatter3(x(end), y(end), z(end), 15, 'filled', 'k')
end
axis equal
hold off
xticks([]); yticks([]); zticks([])
view(0,0) % Vi vill se solsystemet från en häftig vinkel.
```

B.2.1 AVF för solsystemet

För att implementera AVF-metoden, som ges av (6.2), måste vi räkna ut några integraler. Vi låter k_i, ℓ_i vara steg så att

$$p_i^{n+1} = p_i^n + hk_i,$$
$$q_i^{n+1} = q_i^n + h\ell_i.$$

Om vi informellt låter $q = q^n, p = p^n$, och stoppar in formler för $\dot{p}_i, \dot{q}_i$ i (6.2a) så gäller

$$\ell_i = \int_0^1 \nabla_{p_i} H(p + shk, q + sh\ell)\, ds = \int_0^1 \frac{1}{m_i}(p_i + shk_i)\, ds = \frac{1}{m_i}\left(p_i + \frac{h}{2}k_i\right),$$

och

$$k_i = -\int_0^1 \nabla_{q_i} H(p + shk, q + sh\ell)\, ds = \int_0^1 \sum_{\substack{j=1 \\ j\neq i}}^{K} \frac{Gm_i m_j (q_j - q_i + sh(\ell_j - \ell_i))}{\|q_j - q_i + sh(\ell_j - \ell_i)\|^3}\, ds,$$

$$= \sum_{\substack{j=1 \\ j\neq i}}^{K} Gm_i m_j \left(\int_0^1 \frac{(q_j - q_i)ds}{\|q_j - q_i + sh(\ell_j - \ell_i)\|^3} + \int_0^1 \frac{sh(\ell_j - \ell_i)ds}{\|q_j - q_i + sh(\ell_j - \ell_i)\|^3} \right).$$

Vi kan beräkna integralerna med formlerna

$$a := \frac{1}{h}\left(\frac{1}{\|x + hy\|} - \frac{1}{\|x\|}\right),$$
$$\int_0^1 \frac{ds}{\|x + shy\|^3} = \frac{1}{\|x\|^2\|y\|^2 - (x \cdot y)^2}\left(\frac{\|y\|^2}{\|x + hy\|} + (x \cdot y)a\right),$$
$$\int_0^1 \frac{sh\, ds}{\|x + shy\|^3} = -\frac{1}{\|y\|^2}\left((x \cdot y)\int_0^1 \frac{ds}{\|x + shy\|^3} + a\right),$$

för $x, y \in \mathbb{R}^3, h > 0$.

Funktion för att beräkna integral $\int_0^1 \frac{x + shy}{\|x + shy\|^3}\, ds$

```
function result = solsystemAVFintegral(x, y, h)
% In: matriser x, y som ses som en samling kolonnvektorer och skalär h
% Ut: Om x och y är kolonnvektorer: integralen av s |-> (x+shy)/||x+shy||^3
%     över intervallet [0,1]. Om x och y är matriser görs detta kolonnvis
%     och resultatet är en radvektor.
    a = (1./vecnorm(x+h*y) - 1./vecnorm(x))/h;
    I1 = (dot(y,y)./vecnorm(x+h*y)+dot(x,y).*a) ...
                ./(dot(x,x).*dot(y,y)-dot(x,y).^2);
    I2 = -(dot(x,y).*I1 + a)./dot(y,y);
    result = I1.*x + I2.*y;
end
```

AVF-metoden för solsystemet: Vi löser ekvationen (6.2a) med fixpunktsiterationer.

```
function [t,p,q] = solsystemAVF(t0, p0, q0, m, G, h, N)
% Beräknar approximativ lösning för solsystemet med AVF-metoden
% In: initialtid t0, initiella rörelsemängder q0, initialpositioner q0,
%     massor m, gravitationskonstanten G, steglängd h och antal tidssteg N
% Ut: tidsgitter t, approximativa rörelsemängder p och
%     approximativa positioner q
    t = t0 + h*(0:N);
    K = length(m);
    p = zeros(3, K, N+1); p(:, :, 1) = p0;
    q = zeros(3, K, N+1); q(:, :, 1) = q0;

    for n = 1:N
        pn = p(:,:,n);
        qn = q(:,:,n);

        %start k och l
```

```
        l = pn./m;
        k = solsystempderivata(qn,m,G);

        %fixpunkts-iterationer
        kny = zeros(3,K);
        for iter = 1:20
            for i = 1:K
                j = 1:K ~= i;
                qdiff = qn(:, j) - qn(:, i);
                ldiff = l(:, j) - l(:, i);
                I = solsystemAVFintegral(qdiff,ldiff,h);
                kny(:,i) = G*m(i)*sum(m(j).*I,2);
            end
            lny = (pn+0.5*h*k)./m;
            l = lny;
            k = kny;
        end

        %steget
        p(:,:,n+1) = pn + h*k;
        q(:,:,n+1) = qn + h*l;
    end
end
```