

TMA4320 – Prosjekt 3 - 2021

March 25, 2021

1 Introduksjon

Formålet med dette prosjektet er å implementere noen numeriske metoder for løsning av ordinære differensialligninger og sammenligne deres egenskaper når de anvendes på et spesifikt eksempel fra mekanikk. Men aller først skal vi lære en valideringsteknikk som går ut på å teste om metoden vi har implementert har korrekt konvergensorden.

Vi velger ut et utvalg på fire numeriske metoder. Fordi vi skal integrere over veldig lange tidsintervaller, så velger vi her i hovedsak å se på eksplisitte metoder, slik at beregningskostnaden pr skritt blir ganske lite. Det er tre oppgaver som skal besvares, og i hver oppgave undersøker man en gitt egenskap ved hver av de fire metodene. Disse er *konvergensorden*, *energibevaring* og *Poincaré-kutt*. I alle tilfeller brukes samme modellproblem, nemlig Hénon-Heiles-modellen.

Når man har svart tilfredsstillende på disse undersøkelsene kan man score litt ekstra bonuspoeng ved å forsøke å undersøke litteratur for å forklare oppførselen til de ulike metodene i de to siste oppgavene. Som hovedreferanse kan man da bruke [1]

2 De numeriske metodene

Vi introduserer de 4 metodene i dette kapitlet, men la oss først gjengi ligningene i det spesifikke eksemplet vi skal anvende metodene på, nemlig Hénon-Heiles modellen. Systemet har dimensjon 4 og vi kaller variablene (q_1, q_2, p_1, p_2) . Ligningene ser slik ut:

$$\begin{aligned}q_1' &= p_1 \\q_2' &= p_2 \\p_1' &= -q_1(1 + 2q_2) \\p_2' &= -(q_2 + q_1^2 - q_2^2)\end{aligned}$$

Vi merker oss to ting med en gang: Det første er at de deriverte av q_1 og q_2 avhenger bare av p_1 og p_2 , og vice versa, de deriverte av p_1 og p_2 avhenger bare av q_1 og q_2 . En av metodene våre forutsetter en slik struktur. Det andre er at høyresidene er kvadratiske polynomer i de 4 variablene. En annen av metodene våre forutsetter dette.

Senere i denne beskrivelsen vil vi beskrive noen egenskaper ved Hamiltonske problemer mer generelt, og Hénon-Heiles modellen spesielt.

2.1 Metode A, Shampine-Bogacki, metode av orden 3

Husk metoden ved samme navn som ble brukt i Øving 6. Det var et innbygd par av metoder av orden (3,2). Her skal vi bruke konstant skritt lengde og bruker bare metoden av orden 3 som er definert for vilkårlige differensialligninger av formen $y' = F(y)$ (vi dropper muligheten for eksplisitt avhengighet av t , dvs $F(t, y)$, i dette prosjektet).

$$\begin{aligned}F_1 &= F(y_n) \\F_2 &= F(y_n + \tfrac{1}{2}hF_1) \\F_3 &= F(y_n + \tfrac{3}{4}hF_2) \\y_{n+1} &= y_n + h(\tfrac{2}{9}F_1 + \tfrac{1}{3}F_2 + \tfrac{4}{9}F_3)\end{aligned}$$

2.2 Metode B. Kuttas metode av orden 4

Denne har vi nevnt i forelesningene, den er også definert for ligninger av formen $y' = F(y)$

$$\begin{aligned}F_1 &= F(y_n) \\F_2 &= F(y_n + \tfrac{1}{2}hF_1) \\F_3 &= F(y_n + \tfrac{1}{2}hF_2) \\F_4 &= F(y_n + hF_3) \\y_{n+1} &= y_n + h(\tfrac{1}{6}F_1 + \tfrac{1}{3}F_2 + \tfrac{1}{3}F_3 + \tfrac{1}{6}F_4)\end{aligned}$$

2.3 Metode C. Kahans metode av orden 2

Dette er en metode som er definert for differensialligninger med kvadratiske høyresider. Den er virkelig implisitt, men kun lineær-implisitt så man slipper unna med kun å løse et lineært system i hvert skritt. Et generelt kvadratisk system kan på komponentform skrives som

$$y'_i = \sum_{j,k=1}^m a_{ijk} y_j y_k + \sum_{j=1}^m b_{ij} y_j + c_i, \quad i = 1, \dots, m,$$

for et vilkårlig valg av koeffisienter a_{ijk} , b_{ij} og c_i . Kahans metode er definert som

$$\frac{y_{i,n+1} - y_{i,n}}{h} = \sum_{j,k=1}^m a_{ijk} \frac{y_{j,n} y_{k,n+1} + y_{j,n+1} y_{k,n}}{2} + \sum_{j=1}^m b_{ij} \frac{y_{j,n} + y_{j,n+1}}{2} + c_i$$

Her betyr $y_{i,n}$ den numeriske approksimasjonen til $y_i(t_n)$. Vi må altså i hvert tidsskritt løse for $(y_{1,n+1}, \dots, y_{m,n+1})^T$, men dette blir et lineært system av ligninger (som det riktignok kan være litt jobb å sette opp). For å løse dette systemet i din implementasjon, kan du gjerne benytte deg av tilgjengelige biblioteksrutiner i numpy, f.eks. `numpy.linalg.solve`.

2.4 Metode D, Störmer-Verlet, orden 2

Denne metoden er laget for andreordens systemer av formen $q'' = f(q)$ som omskrevet til et første ordens system ($p := q'$) blir ($q' = p, p' = f(q)$) og vi ser at Hénon-Heiles systemet har nettopp en slik form. Metoden er definert som.

$$\begin{aligned}p_{n+\frac{1}{2}} &= p_n + \frac{1}{2}hf(q_n) \\ q_{n+1} &= q_n + hp_{n+\frac{1}{2}} \\ p_{n+1} &= p_{n+\frac{1}{2}} + \frac{1}{2}hf(q_{n+1})\end{aligned}$$

Merk at denne metoden trenger (med unntak av første skritt) kun å gjøre én evaluering av f i hvert skritt i en effektiv implementasjon.

3 Numerisk undersøkelse av orden

Et godt hjelpemiddel for å validere implementasjon av en numerisk metode for differensialligninger er å numerisk verifisere metodens orden. Særlig for metoder av høy orden er det en god indikasjon på kodens korrekthet dersom den har rett observert konvergensorden. Noen ganger fører gal implementasjon til at man får bare ‘søppel’, mens andre ganger kan man få noe som ser korrekt ut, men som kanskje har dårligere konvergens enn den skulle hatt. I den første oppgaven skal du sjekke orden numerisk i dine implementasjoner av de 4 metodene som ble oppgitt.

Når man generelt skal sjekke orden, kan man velge ett eller flere problemer (differensialligninger) å teste metoden på. Det bør være minst dimensjon 2 (dvs $y(t) \in \mathbb{R}^m$, $m \geq 2$), og det bør være et ikke-trivielt problem, dvs unngå f.eks. $y' = Ay + b$ der A er en matrise og b en konstant vektor. Det neste er at man trenger en referanseløsning, dvs en (nær) eksakt løsning som kan brukes til å finne feilen. Vi kan bruke Hénon-Heiles (HH) som testmodell for å sjekke orden. Noen generelle idéer for å finne referanseløsning er

1. Man kan ta et problem der eksakt løsning er kjent, slike er det mulig å konstruere
2. Enda enklere: Gjør bruk av en biblioteksrutine f.eks. `scipy.integrate.RK45` der man setter veldig strenge toleranser (`rtol`, `atol`), f.eks. `1e-12`.
3. Enklest (men litt mindre robust). Bruk din egen implementasjon med en vesentlig mindre skrittlengde, h_{ref} , enn de du tester ordenen for, f.eks. $h_{\text{ref}} = \frac{1}{10}h_{\text{min}}$.

Som anbefalt løsning foreslås å bruke (2) siden eksakt løsning ikke er tilgjengelig for Hénon-Heiles.

Det man nå gjør er å velge et tidsintervall $[t_0, T]$ å integrere over, for eksempel $[0, 10]$ for det valgte systemet (HH). Så tar man et utvalg av `nexps` stk skrittlengder h_i typisk logaritmisk fordelt fra 10^r til 10^s . Som et utgangspunkt, prøv $r = 0$, $s = -4$ og `nexps` = 20, men du kan prøve deg litt fram når du ser hvordan resultatene blir. Bruk `numpy.logspace` for å generere disse skrittlengdene. Beregn så en referanseløsning

y_{ref} i $t = T$ som beskrevet ovenfor. Deretter kan du for hver valgte skrittlengde h_i anvende metoden til å finne en tilnærming y_N til $y(T)$. Kanskje må du ta et siste kortere skritt hvis ikke $T - t_0$ er et heltallig multiplum av h_i (det går helt fint for ordensanalysen med ett siste kortere skritt). Nå kan du beregne den globale feilen “eksakt” ved differansen mellom approksimasjonen og referanseløsningen. Du lagrer globalfeilen for alle skrittlengdene i et numpy-array. Siden vi vet at den globale feilen $E(h) \approx C h^p$ der p er ordenen, så kan det være lurt å plotte grafen $(h_i, E(h_i))$ i et **loglog**-plot. Disse grafene skal bli tilnærmet rette linjer med stigningstall p som er det du søker. En god service overfor leseren av rapporten er å legge inn en referanselinje som viser hva som svarer til den orden metoden teoretisk sett har. For eksempel, hvis du har testet med en metode som skal ha orden 2, legg på KH^2 mot H som en stiplet linje der K justeres slik at linjen plasseres på et fornuftig sted. Hvis den er parallell med din graf fra eksperimentet, så er det lovende.

I tillegg kan det være bra å beregne estimater av ordenen som virkelige tall. Merk at for to skrittlengder h_1 og h_2 gjelder for en orden p approksimasjon

$$E(h_1) \approx Ch_1^p, \quad E(h_2) \approx Ch_2^p$$

og ved å eliminere C får vi at

$$p \approx \frac{\ln \frac{E(h_2)}{E(h_1)}}{\ln \frac{h_2}{h_1}}$$

Det kan derfor også være lurt å beregne p på denne måten for ethvert par av etterfølgende skrittlengder. Disse tallene vil da approksimere metodens orden. Når h_1, h_2 er store vil de gi unøyaktig svar fordi orden bygger på et asymptotisk premiss om at skrittlengden går mot null. Når h_1, h_2 er for små, så blir approksimasjonen av p ødelagt av at referanseløsningen ikke er helt nøyaktig, og beregningene kontamineres også av avrundingsfeil på maskinen. Du må prøve og feile litt for å finne gode verdier for r og s i algoritmen.

4 Hamiltonske systemer

I mekanikken brukes ofte to ulike rammeverk for å beskrive dynamiske systemer, det ene tilskrives Lagrange og bruker posisjon og hastighet som variable, det andre tilskrives Hamilton og bruker posisjon og impuls (q, p) som variable. q og p har samme dimensjon, d . Det betyr at Hamiltonske systemer totalt sett har like dimensjon $m = 2d$. Man introduserer en energifunksjon eller Hamiltonfunksjon $H(q, p)$ som typisk splittes opp i kinetisk energi $T(q, p)$ og potensiell energi $V(q)$ slik at $H(q, p) = T(q, p) + V(q)$. Det dynamiske systemet (difflikningene) utledes da som

$$\begin{aligned} q' &= \frac{\partial H}{\partial p}(q, p) \\ p' &= -\frac{\partial H}{\partial q}(q, p) \end{aligned} \tag{1}$$

Her tolker vi for eksempel $\frac{\partial H}{\partial p}(q, p)$ som vektoren hvis komponent i er $\frac{\partial H}{\partial p_i}(q, p)$. Ved kjerneregelen finner vi at dersom $y(t) = (q(t), p(t))$ er en løsning av (1) så vil $H(q(t), p(t)) = H(q_0, p_0)$ for alle $t \geq 0$ det vil si, energien er bevart over tid. Dette fordi

$$\begin{aligned} \frac{d}{dt}H(q(t), p(t)) &= \frac{\partial H}{\partial q}(q, p) \cdot q' + \frac{\partial H}{\partial p}(q, p) \cdot p' \\ &= \frac{\partial H}{\partial q}(q, p) \cdot \frac{\partial H}{\partial p}(q, p) + \frac{\partial H}{\partial p}(q, p) \cdot \left(-\frac{\partial H}{\partial q}(q, p)\right) = 0. \end{aligned}$$

En annen litt mer intrikat, men kanskje enda viktigere egenskap ved løsningen til Hamiltonske systemer framkommer ved å tenke på løsningen av differensialligningen som en transformasjon av initialverdien, slik vi snakket om i forelesningene. For en (autonom) diffiligning $y' = f(y)$ med initialverdi $y(0) = y_0$ skriver vi $y(t) = \varphi_{t,f}(y_0)$, så $\varphi_{t,f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ er en mapping som tar y_0 som input og gir ut løsningen $y(t)$ ved tid t som output (tenk på t som fiksert). Denne mappingen har en Jacobi-matrise $M := D\varphi_{f,t}(y)$. Det kan vises at denne matrisen oppfyller ligningen

$$M^T J M = J \quad \text{for alle } y \text{ og } t \text{ der} \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \quad (2)$$

når $\varphi_{t,f}$ representerer løsningen av et Hamiltonsk system. Vi sier at mappinger som oppfyller denne betingelsen er *symplektiske avbildninger* og de spiller en viktig rolle når man studerer geometriske egenskaper ved Hamiltonske differensialligninger.

Eksempel. La oss illustrere med et enkelt eksempel, den harmoniske oscillatoren. I dette tilfellet er $H(q, p) = \frac{1}{2}(q^2 + p^2)$. Dette gir

$$\begin{aligned} q' &= p \\ p' &= -q \end{aligned}$$

om vi bruker formelen (1) ovenfor. Vi kan også derivere den første ligningen en gang til og får da $q'' = p' = -q$ som har generell løsning $q(t) = q_0 \cos t + p_0 \sin t$ for vilkårlige startverdier q_0 og p_0 . Dermed blir $p(t) = -q_0 \sin t + p_0 \cos t$. Du kan da sjekke at

$$H(q(t), p(t)) = \frac{1}{2}((q_0 \cos t + p_0 \sin t)^2 + (-q_0 \sin t + p_0 \cos t)^2) = \frac{1}{2}(q_0^2 + p_0^2) = H(q_0, p_0)$$

så energien er bevart. Vi har da også beskrevet mappingen $\varphi_{t,f}$ som er lineær for dette eksemplet, og kan derfor uttrykkes som en matrise

$$\varphi_{t,f}(q_0, p_0) = \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} \begin{pmatrix} q_0 \\ p_0 \end{pmatrix} = M \begin{pmatrix} q_0 \\ p_0 \end{pmatrix}$$

så matrisen M i dette uttrykket er rett og slett $D\varphi_{f,t}$. Du kan nå sjekke at

$$\begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix}^T \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

som viser at løsningsmappingen $\varphi_{t,f}$ er symplektisk. \square

Oppsummert kan man si at løsningen til et Hamiltonsk system av differensialligninger har to viktige egenskaper, den ene er at den bevarer energi (Hamiltonfunksjonen) og den andre er at den er symplektisk, dvs oppfyller (2). Som nevnt i forelesningene kan også den numeriske metoden tenkes på som en mapping $\Psi_{h,f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$, det er mappingen som tar y_n som input og gir y_{n+1} som output. Siden $\Psi_{h,f}$ er en approksimasjon til $\varphi_{h,f}$ så er det et betimelig spørsmål å stille om $\Psi_{h,f}$ også har de to nevnte egenskapene til som vi akkurat har påpekt at $\varphi_{t,f}$ har. Det generelle svaret på dette er **nei**, men det går faktisk an å konstruere numeriske metoder (dvs $\Psi_{h,f}$) som oppfyller enten den ene eller den andre betingelsen. Slike numeriske metoder kalles henholdsvis *energibevarende* og *symplektiske* metoder. En enkel måte å sjekke om, eller i hvilken grad, en metode $\Psi_{h,f}$ bevarer energi, er å studere $H(y_n)$ ($H(q_n, p_n)$) for $n = 0, 1, \dots$, for eksempel ved å plote denne som funksjon av t_n . Man kan tenke seg tre ulike forløp. Det ene er at metoden er energibevarende, og da blir plottet kun en konstant horisontal linje. Et annet mulig utfall som også er bra, er at $H(y_n)$ varierer, men holder seg i en omegn om $H(y_0)$ for alle n . Mer problematisk kan det være hvis kurven viser såkalt *drift*, det betyr at energien beveger seg stadig lengre vekk fra utgangspunktet.

5 Hénon–Heiles modellen

To astrofysikere (og matematikere) Michel Hénon og Carl Heiles studerte på 1960-tallet en modell for en stjernes bevegelse i en galakse og kom da etterhvert opp med en forenklet modell for å studere visse fenomener av interesse. Hénon–Heiles modellen er et Hamiltonsk system i 4 dimensjoner som har Hamiltonfunksjon

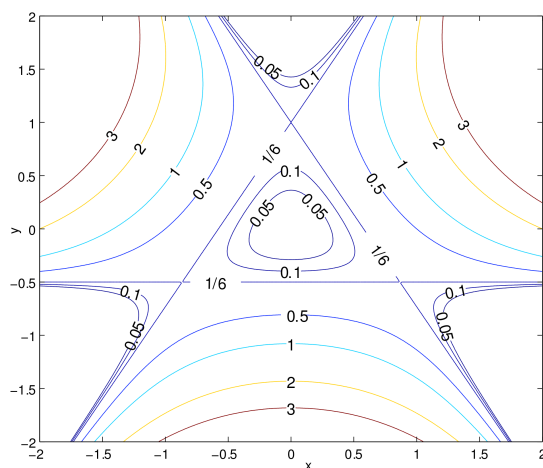
$$H(q, p) = \frac{1}{2}(p_1^2 + p_2^2) + U(q)$$

der $q \in \mathbb{R}^2$, $p \in \mathbb{R}^2$ og $U(q)$ er potensiell energi som er av formen

$$U(q) = \frac{1}{2}(q_1^2 + q_2^2) + \lambda(q_1^2 q_2 - \frac{1}{3}q_2^3).$$

For våre formål setter vi $\lambda = 1$. Det dynamiske systemet som denne Hamiltonfunksjonen representerer har fascinerende egenskaper, og løsningens oppførsel avhenger av initialverdiene som benyttes. For eksempel er løsningen kaotisk i enkelte deler av faserommet (løsningsrommet).

Nivåkurver $U(q) = c$ til potensialfunksjonen finner man på Wikipedia. Vi ser at for $c = \frac{1}{6}$ blir nivåkurven til en trekant og hjørnene er sadelpunkter for $U(q)$. Hvis initialverdien $q(0)$ ligger inne i trekanten og totalenergien $H(q_0, p_0) < \frac{1}{6}$ så vil posisjonen q holde seg inne i trekanten for all tid. Likevel gir for eksempel $H(q_0, p_0) = \frac{1}{8}$ kaotisk oppførsel, mens $H(q_0, p_0) = \frac{1}{12}$ gir en pen og glatt kurve som vi skal se nærmere på i dette prosjektet.

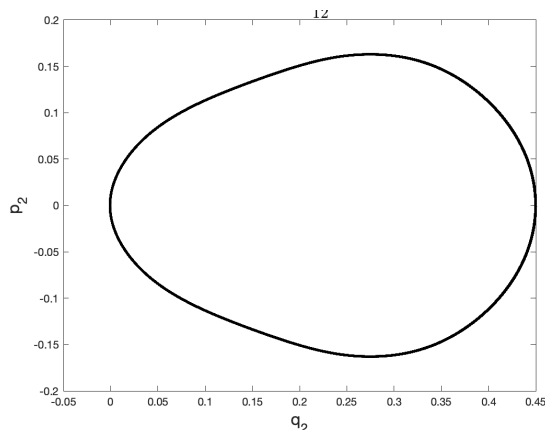


Vi minner om at diffiligningene for Hénon–Heiles blir

$$\begin{aligned} q_1' &= \frac{\partial H}{\partial p_1} = p_1 \\ q_2' &= \frac{\partial H}{\partial p_2} = p_2 \\ p_1' &= -\frac{\partial H}{\partial q_1} = -q_1(1 + 2q_2) \\ p_2' &= -\frac{\partial H}{\partial q_2} = -(q_2 + q_1^2 - q_2^2) \end{aligned}$$

Poincaré-seksjoner. Franskmannen Poincaré var en matematiker som blant annet gjorde store oppdagelser innenfor teorien for dynamiske systemer. Vi skal se nærmere på en av hans oppfinnelser. Utgangspunktet er at løsningen av diffiligningen er en kurve i \mathbb{R}^m . Så ser man på et underrom av lavere dimensjon, det kunne for eksempel være et to-dimensjonalt plan i \mathbb{R}^3 og så registrerer man hver gang kurven krysser dette planet. Hvis løsningskurven underrommet (transversalt) gang på gang, så vil man kunne se på mengden av alle slike krysningspunkter og det vil fortelle oss noe om dynamikken i problemet.

Vi bruker Hénon-Heiles modellen som eksempel her. Der er $m = 4$ og vi lar underrommet være mengden av alle punkter av formen $(0, q_2, p_1, p_2)$. Fra en gitt initialverdi registrerer vi når løsningsens q_1 -komponent skifter fortegn fra en retning der $p_1 > 0$. Vi er interessert i hvordan de ulike numeriske metodene våre takler dette problemet. Figuren viser omtrent hvordan Poincaré-kuttet skal bli når $H_0 = \frac{1}{12}$, projisert på (q_2, p_2) -planet.



Numerisk prosedyre.

Startverdi Vi skal velge startverdi slik at $H(q_0, p_0) = \frac{1}{12}$. Dette gjøres ved å først definere $q_{0,1} = 0$ (q_0 sin første komponent) slik at vi starter i vår Poincaré-seksjon. Så lar vi $q_{0,2} = 0.45$ og $p_{0,2} = 0$. Til slutt beregner vi $p_{1,0}$ fra kriteriet $H(q_0, p_0) = \frac{1}{12}$ der vi insisterer at $p_{0,1} > 0$. Du kan også teste ut $H_0 = \frac{1}{10}$ og la deg overraske.

Integrasjon. Vi bruker nå vår numeriske metode og integrerer framover i tid med konstant skrittlengde, mens vi hele tiden sjekker om $p_1 > 0$ og om q_1 skifter fortegn fra ett skritt til det neste. Hvis dette skjer i skritt n så kan vi bruke lineær interpolasjon for å bestemme mer nøyaktige verdier av (q_2, p_1, p_2) der $q_1 = 0$. Lagre verdiene (q_2, p_2) og integrer igjen inntil du finner et nytt krysningspunkt. Tilsammen kan du generere så mange krysninger som din PC er i stand til innen rimelig kjøretid. Det er viktig å huske på at når du har passert et krysningspunkt og skal integrere videre, så gjør du dette med approksimasjonen som kommer fra metoden, og ikke den interpolerte verdien.

Plotting. Så plotter vi til slutt alle disse punktene (q_2, p_2) i et punkt-plott, det vil si hvert krysningspunkt (q_2, p_2) plottes som et punkt. For den eksakte løsningen (med denne verdien av $H(q_0, p_0)$) forventer vi at alle disse punktene ligger på en glatt lukket kurve. Det interessante nå er å studere hvilke av metodene som klarer å reprodusere en slik glatt pen kurve over lang tid.

6 Oppgaver

Dere skal gjøre tre oppgaver som her spesifiseres. Alle oppgavene skal bruke Hénon-Heiles modellen som test case.

Oppgave 1. Verifikasjon av konvergensorden for globalfeil i numeriske metoder. Her skal et utvalg av numeriske metoder implementeres, og konvergensorden skal estimeres og framstilles for alle metodene. Man må inkludere både $\log\log$ -plot av globalfeil vs skrittlengde og vise beregnede tall for ordensestimat.

- Bogacki-Shampine (orden 3)
- Kuttas metode (orden 4)
- Kahans metode (orden 2)
- Størmer-Verlet-metoden (orden 2)

Oppgave 2. Det skal undersøkes hvor godt metodene fra Oppgave 1 er i stand til å bevare energi i Hamiltonske systemer. Her kan man potensielt utforske så lange tidshorisonter som overhodet mulig. Det er derfor viktig at implementasjonen er effektiv.

Man skal forsøke å trekke konklusjoner omkring hvilke metoder som er best både som funksjon av skrittlengde og hvor langt man integrerer.

Det gis kreditt dersom man klarer å lese seg opp fra litteratur hva grunnen er til at enkelte metoder gjør det bedre enn andre i ulike sammenhenger.

Oppgave 3. Poincaré-seksjoner for Hénon-Heiles problemet. Beregn og plott Poincaré-kutt for de oppgitte metodene. Her må dere gjøre noen avveininger når det gjelder beregningstid. For $H_0 = \frac{1}{12}$ er tidsintervallet mellom to krysningspunkt typisk omtrent $6 - 7$. Man kan kjøre tester med skrittlengder i området $10^{-4} - 10^{-1}$ og integrere fra $t_0 = 0$ og helt opp til $T > 10^6$ for de største skrittlengdene, eller hvor lenge man har tid til å vente. Det viktigste svaret på denne oppgaven er at man har klart å lage gode Poincaré-grafer som beskrevet i notatet. Men man bør også kommentere på resultatene, og gjerne studere litteratur som kan forklare resultatene man finner. Vi kan røpe så mye som at for $H_0 = \frac{1}{12}$ skal Poincaré-grafen bli en lukket kurve.

Hint om litteratur. Boka [1] i referanselista er kanskje den beste kilden som fins på dette området. Det kan nok være vanskelig å forstå detaljene, men ikke umulig å plukke opp noen interessante fakta og resultater derfra. Blant annet finner dere der eksempler på Poincaré-plott for Hénon-Heiles problemet på side 15. Boka skal dere kunne laste ned gratis om dere er på NTNU-nettet (f.eks. via VPN).

<https://link.springer.com/content/pdf/10.1007%2F3-540-30666-8.pdf>

7 Vurderingskriterier og krav til rapport

Innleveringskrav

Gruppestørrelse: 1–3 personer

Frist: Rapporten skal leveres innen torsdag 22. april, kl 23:59 i Inspira.

Format: Rapporten skal fortrinnsvis innleveres som en Jupyter Notebook (ipynb), men det tillates at tekst og matematikk leveres som en pdf-fil som enten er generert fra et tekstbehandlingsystem (f.eks. L^AT_EX) eller scannet/fotografert (pen og ryddig) håndskrift. Det er et krav at koden er kjørbart for den som retter, men også at alle resultater, figurer og plott skal kunne leses uten nødvendigvis å måtte kjøre koden først. Merk at alt må lastes opp i Inspira. Hvis man for eksempel kun innleverer en lenke til en skytjeneste der prosjektet er lagret så vil det ikke bli godkjent som innlevert.

Kriterier som vektlegges Det vil alltid være et helhetsinntrykk som legges til grunn i vurderingen, men her kommer noen indikasjoner på hva som blir mest vektlagt

- **Programkoden** bør være ryddig og lesbar. Legg gjerne inn kommentarer, og lag en letleselig struktur. Det er viktig å etterstrebe en rimelig god effektivitet. For

eksempel gis det trekk hvis man beregner eller lagrer unødvendig informasjon, hvis man for eksempel beregner samme uttrykk flere ganger, eller legger kodelinjer inni ei løkke når de kunne vært trukket utenfor osv. Se ellers “Pinned post” på Piazza med overskrift *Koden min kjører treigt! Skal den være så treig?*

- Vi er mest interessert i **egenskaper og konklusjoner** knyttet til de numeriske metodene og de fenomenene vi undersøker. Alle plott, figurer, tabeller etc som inkluderes bør kommenteres, og kan gjerne analyseres og forklares.
- I utgangspunktet er dette prosjektet som oppgave ganske veldefinert. Men vi åpner for at det kan gis noen ekstra poeng dersom man kan **trenge inn i teorien** og forstå litt om hvorfor resultatene blir som de blir. Særlig gjelder dette de to siste oppgavene. Her kan man i ganske stor grad basere seg på referansen [1].

References

- [1] E. Hairer, Ch. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2010. Structure-preserving algorithms for ordinary differential equations, Reprint of the second (2006) edition.