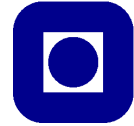


FY1008 Introduksjon til Numerisk Fysikk

Øving 3 og 4 Høst 2025

NTNU

Institutt for
fysikk

De to oppgavene i dette settet er knyttet tett til hverandre, og derfor utgjør dette settet *to* øvinger. Øving 3 er oppgave 1 i dette oppgavesettet, mens øving 4 svarer til oppgave 2.

I disse øvingene skal vi studere bevegelsen av en tennis ball. Ideen bak å gjøre simuleringer av fysikken på en datamaskin er å programmere fysikkens lover på datamaskinen, og å la datamaskinen *beregne* hva som skjer som funksjon av tiden, steg for steg inn i fremtiden. I mekanikk er det Newtons bevegelseslover som bestemmer tidsutviklingen av systemet, og vårt mål er å beregne posisjon, hastighet og akselerasjon for ballen for de ulike kreftene som måtte påvirke ballen. Vi vil senere lære hvordan man kan løse mer generelle differensial likninger numerisk, men siden du enda ikke kjenner mye til slike likninger, vil vi bruke en noe annen fremgangsmåte her som forhåpentligvis er noe lettere for deg å følge. Senere vil vi se på mer generelle numeriske metoder for å løse slike likninger.

La oss ved tiden t betegne ballens vektor posisjon som $\mathbf{r}(t)$, dens vektor hastighet som $\mathbf{v}(t)$, og dens vektor akselerasjon som $\mathbf{a}(t)$. Her brukes fete typer til å betegne vektorer. Da følger det fra definisjonen av hastighet at

$$\mathbf{v}(t) = \frac{d\mathbf{r}(t)}{dt}, \quad (1)$$

og tilsvarende for akselerasjonen

$$\mathbf{a}(t) = \frac{d\mathbf{v}(t)}{dt}. \quad (2)$$

Ved å bruke definisjonen av den deriverte vil likning (1) kunne skrives som

$$\mathbf{v}(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t} \approx \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t}. \quad (3)$$

I en data simulering vet vi allerede den nåværende posisjonen $\mathbf{r}(t)$, så ved å løse den overstående likningen med hensyn på $\mathbf{r}(t + \Delta t)$ finner man at

$$\mathbf{r}(t + \Delta t) \approx \mathbf{r}(t) + \Delta t \mathbf{v}(t). \quad (4a)$$

Dersom vi nå velger tidspunktene $t = t_n$, med $n = 1, 2, \dots$ slik at tidsintervallet $\Delta t = t_{n+1} - t_n$ er det samme for et vilkårlig positivt heltall n slik at $t_n = n\Delta t$, kan man bruke likning (4a) til å beregne fremtidige posisjoner for ballen da man kjenner nåværende posisjon og nåværende hastighet. På samme måten kan man fra likning (2) beregne fremtidig vektor hastighet fra

$$\mathbf{v}(t + \Delta t) \approx \mathbf{v}(t) + \Delta t \mathbf{a}(t), \quad (4b)$$

gitt at man kjenner nåværende hastighet og akselerasjon. Merk deg at likning (4b) også fremkommer ved å ta den tidsderiverte av likning (4a). Likningene (4) er gyldige for et vilkårlig bevegelig objekt og disse likningen representerer den såkalte *Euler algoritmen* for vårt system. Denne algoritmen er oppkalt etter den sveitsiske fysikeren og matematikeren Leonard Euler 1707–1783.

Som et alternativ, kan likningene (4) utledes ved først å bruk Newtons andre lov på ballen, $\Sigma \mathbf{F} = m\mathbf{a}$, for dernest å integrer denne likningen over et lite tidsvindu fra t til $t + \Delta t$.

I en data simulering er ideen å starte fra tilstanden til ballen ved tiden $t = t_0 = 0$ s, noe vi kaller *start betingelsene*, og bruke likningene (4) til å beregne posisjon og hastighet ved tiden $t = \Delta t$, for så å gjenta beregningene for neste tidssteg $t = 2\Delta t$, osv. Man forventer å få mer nøyaktige resultater når tidsintervallet Δt blir mindre.

Man skal merke seg at Euler algoritmen har egenskaper som kan gi uønskede sideeffekter i enkelte situasjoner. Metoden er kjent for å ikke bevare energien for systemet man studerer. For fysiske systemer hvor det f.eks. er mange oscillasjoner, eller i beregningene av Kepler banene i astronomi, dvs. systemer hvor man studerer dem over lang tid, kan bruk av Euler metoden simpelthen gi feilaktige resultater (da energien for systemet typisk vil øke). Heldigvis er det relativt enkelt å korrigere dette! Det kan f.eks. gjøres simpelthen ved å beholde likning (4b) for beregning av hastigheten, *men* endre likning (4a) for posisjon ved å erstatte leddet $\mathbf{v}(t)$ på høyresiden av denne likningen med $\mathbf{v}(t + \Delta t)$ som nettopp var beregnet fra likning (4b). Det resulterende likningsettet blir da

$$\mathbf{v}(t + \Delta t) \approx \mathbf{v}(t) + \Delta t \mathbf{a}(t), \quad (5a)$$

og

$$\mathbf{r}(t + \Delta t) \approx \mathbf{r}(t) + \Delta t \mathbf{v}(t + \Delta t), \quad (5b)$$

og det beskriver den *semi-implisitte Euler algoritmen* som også er kjent som *Euler-Cromer* metoden (eller symplectic Euler). Denne metoden bevarer energien i systemet langt bedre enn den originale Euler metoden. Dersom du observerer at energien ikke er bevart i dine simuleringer, for et system hvor den bør være bevart, bør du skifte til metoden i likning (5) for å redusere de numeriske problemene slik at du kan oppnå mer nøyaktige simuleringer resultater.

Oppgave 1.

Vi vil nå forsøke å løse et problem numerisk hvor man har en analytisk løsning. Dette gjøres slik at du har noe å sammenlikne svarene du oppnår mot. Får man overensstemmelse mellom de numeriske og analytiske resultatene er man trolig inne på et riktig spor, selv om dette ikke alene indikerer at koden er riktig. I motsatt fall, stemmer ikke resultatene overens, kan man konkludere at koden ikke er riktig. Når man får mer erfaring med koding, vil man fort innse at eksakt løsbare spesialtilfeller blir veldig nyttige for å kunne teste den koden man utvikler.

- a) Først vil vi anta at ballen fallen uten luftmotstand fra en høyde $h_0 = 10$ m og at den er i ro før fallbevegelsen begynner. Dette problemet kan du løse analytisk. Først skal du analytisk finne tiden det tar $t = t_1$ før ballen treffe bakken (for første gang). Hva er hastigheten til ballen når den treffer bakken?
- b) Bruk Euler algoritmen (4) til å lage en python kode for å beregne vertikal posisjonen $z(t)$ og vertikal hastigheten $v_z(t)$ for ballen som funksjon av tiden t . Lag grafer av posisjonen og hastigheten av ballen som funksjon av tiden. Du kan i første omgang bruke tidsintervallet $\Delta t = 0.1$ s, men vurder om du må endre denne verdien.
- c) Beregn den kinetiske energien E_K , den potensielle energien E_P , samt den totale energien $E_T = E_K + E_P$ for ballen som funksjon av tiden. Er energien bevart i disse simuleringene? Dersom så ikke er tilfelle bør du istedet bruke Euler-Cromer metoden og gjenta dine tidligere simuleringer (ev. alltid bruke denne metoden for å være på den sikre siden).

- d) Bestemme numerisk tiden t_1 det tar før ballen treffer bakken samt hastigheten $v_z(t_1)$. Trolig vil du finne at resultatet ikke er helt det samme som hva du overfor regnet ut analytisk. Hvorfor er dette tilfelle tror du? For å få mer riktig resultat, men ikke nødvendigvis tilstrekkelig nøyaktig, kan du redusere verdien til Δt . Prøv dette!
- e) Oppdater din python kode for å kunne beregne et mer nøyaktig numerisk estimat for t_1 og $v_z(t_1)$. Prøv å finne disse størrelsene med fire desimalers nøyaktighet. Tips: Hvor liten må Δt være for at du skal ha *fire* siffrers nøyaktighet i ditt svar?
- f) Gjenta beregningen du gjorde overfor, men nå med start betingelsene $z(0) = 0$ m og $v_z(0) = v_0$ med $v_0 = 5$ m/s. Hvor høyt når ballen før den igjen faller mot bakken? Sammenlign svaret ditt med hva du kan regne ut analytisk.
- g) Beregn også for dette tilfelle den kinetiske energien E_K , den potensielle energien E_P , samt den totale energien $E_T = E_K + E_P$ for ballen som funksjon av tiden
- h) Dersom ballen skal kunne nå akkurat høyden 25 m, hva må da start hastigheten v_0 være?

La oss være ærlige, det er ikke et stort poeng i å skrive et data program for å løse problemet dersom man kan løse det analytisk (utover å kontrollere at hva man beregner numerisk gjør mening). Dersom dine beregninger ikke gir svar som er nær de analytiske løsningene bør du be om hjelp før du går videre til neste oppgave.

Oppgave 2.

Vi vil nå legge til luftmotstand i vårt system. Dette betyr at man får en ekstra utfordring når man skal bestemme ballens bane og hastighet. Luftmotstanden sørger for at en ekstra kraft vil virke på ballen. For “normale” hastigheter vil denne kraften være proporsjonal med hastigheten i kvadrat. Videre vil denne kraften alltid være *motsatt* rettet av hastighetsvektoren til ballen. Om vi betegner luftmotstandskraften som $\mathbf{F}_d$ kan vi skrive

$$\mathbf{F}_d = -dv^2 \hat{\mathbf{v}}, \quad (6)$$

hvor d er en positiv konstant som vi kan kalle *drag* koeffisienten (på godt “norsk”). Denne konstanten vil avhenge av størrelsen (og formen) til ballen, samt tettheten til luften den beveger seg gjennom. Videre har vi her innført *enhets vektoren* for hastighet $\hat{\mathbf{v}} = \mathbf{v}/v$ slik at $|\hat{\mathbf{v}}| = 1$. Mer generelt vil vi bruke hat (eller “hustak”) over en vektor for å angi at vektoren er en enhets vektor og derfor har lengden en og er uten fysisk dimensjon (dvs. den er dimensjonsløs).

- a) Hva er SI-enheten for drag koeffisienten? Enheten av konstanten d skriver vi som $[d]$.
- b) Skrive ned Newtons andre lov for bevegelsen av ballen med luftmotstand når man fortsatt antar at ballen utelukkende beveger seg vertikalt langs z -aksen, dvs. bevegelsen er en-dimensjonal og i vertikal retning. Uttrykk likningen ved hjelp av forholdet d/m . Pass på at likningen er riktig uavhengig av retningen for hastighetsvektoren $\mathbf{v}$. *Hint*: bruk definisjonen av enhetsvektoren $\hat{\mathbf{v}}$. Massen til ballen vil vi betegner som m .
- c) Integrer den likningen du fant i forrige del-oppgave over et lite tidsintervall fra t til $t + \Delta t$ for å finne et uttrykk for $v_z(t + \Delta t)$, dvs. for hastigheten i z -retning ved tiden $t + \Delta t$. I din utledning, kan det være nyttig å bruke følgende tilnærming

$$\int_t^{t+\Delta t} dt' f(t') \approx \Delta t f(t), \quad (7)$$

som er god for små verdier av tidsintervallet Δt . Det kan være verdt å merke seg hvordan du regnet ut et integral i Øving 2 ved bruk av Riemann summen; her har man essensielt bare brukt *ett* rektangel i tilnærmingen.

- d) Finn et uttrykk som kan brukes til å bestemme ballens vertikal posisjon $z(t)$.
- e) Skriv et program som regner ut ballens posisjon og hastighet når ballen ved $t = 0$ s starter på bakken fra posisjon $z(0) = 0$ m og med en hastighet $v_z(0) = v_0$; dvs. samme start betingelser som tidligere.
- f) Kjør programmet ditt for følgende verdier av forholdet d/m mellom drag koeffisienten og massen til ballen (i enheter av m^{-1}): (1) 0, for å undersøke at du får det samme svaret som tidligere; (2) 0.1, og (3) 1.0. Sørg for at du oppnår fire signifikant siffer i dine beregnede resultater. Plot både hastigheten og posisjonen av ballen som funksjon av tiden slik at resultatene for de ulike verdiene av d/m vises i samme plot.
- g) Beregn den kinetiske energien E_K , den potensielle energien E_P , samt den totale energien $E_T = E_K + E_P$ for ballen som funksjon av tiden. Er energien bevart i disse simuleringene?
- h) For hver av de ulike verdiene for d/m bestem tiden t_i det tar før ballen treffer bakken ($z = 0$ m) igjen for første gang. Bestem også hvor høyt ballen når z_i i hvert av tilfellene.
- i) Nå med luftmotstand, prøv å bestemme start hastigheten $v_0^{(i)}$ ballen må ha på bakkenivå for å nå opp til en maksimal høyde på 25 m.
- j) Når du ikke kjenner resultatet analytisk, hvordan kan du da være sikker på at hva du numerisk beregner er riktig?