

Rapport for laboratorium i fysikk

A. Aarekol, A. Styrmoe, K. in't Veld, H. Hatlen

Institutt for fysikk, Norges teknisk-naturvitenskapelige universitet, 7491 Trondheim.

Sammendrag

Denne rapporten, skrevet i forbindelse med øvingsopplegget i TFY4125, beskriver forsøket der man triller en kule og en ring ned en bane. Forsøket besto av en numerisk del og en eksperimentell del, der numerisk og eksperimentelt resultat ble sammenlignet. Tracker ble brukt til å samle eksperimentell data og sluttfarten for kula ($1,246 \pm 0,06531$ m/s) holdt seg innenfor de numeriske beregningene, mens dataene til ringen falt utenfor. Forsøket underbygger også teorien om forskjell på en kompakt kule og en tynn ring ned en bane, der den kompaktekulen opparbeidet seg høyere kinetisk energi og dermed beveget seg raskere ned banen.

1. Introduksjon

Forsøket som beskrives i denne rapporten er en gjenstand som triller ned en bane. Laben består av to deler. En numerisk del, der numerikk blir benyttet for å beskrive bevegelsene til gjenstanden, og en eksperimentell del der målinger basert på et fysisk eksperiment ble utført. Eksperimentet ble gjennomført med først en kompakt kule, deretter en tynn ring.

Denne rapporten består av blant annet seksjoner for metode, resultater og diskusjon. Siden det først ble utført numeriske beregninger før målinger av eksperimentet, så har vi valgt å dele overnevnte seksjoner i to deler, en for numerisk del og en for eksperimentell del.

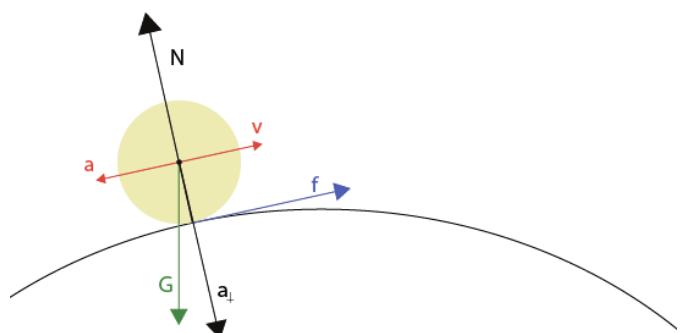
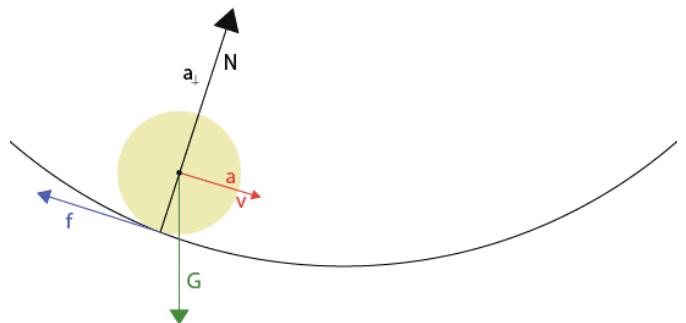
Mot slutten av denne rapporten diskuteres sammenhengen mellom resultatene, usikkerheten i eksperimentet og forskjellene på å trille en kule og en ring ned en bane.

2. Teori

Figur 1 viser en illustrasjon av et objekt som ruller i en krum bane, både når krumningen er konveks og konkav. Illustrert i figuren er også alle krefter som virker på objektet, samt vektorer for hastighet og akselerasjon. Denne seksjonen viser hvordan en kan utlede formler for størrelsene i figuren som senere vil bli brukt i numeriske utregningene. Teorien presentert i [1] danner mye av grunnlaget som det som presenteres her.

2.1. Hastighet

Hastigheten til objektet er parallell med retningen til banen. Ved å bruke prinsippene for energibevaring (se ligning 1) kan en utlede et uttrykk for v som funksjon av høyden, h . Merk at den kinetiske energien er sammensatt av både translasjonsenergien og rotasjonsenergien. Høyre side av ligning 1 beskriver mekanisk energi i toppen av banen (bare potensiell energi tilknyttet tyngdefeltet) og venstre



Figur 1: Krefter og andre fysiske størrelser som virker på et rullende objekt på krumt underlag, for to uavhengige situasjoner.

side beskriver mekanisk energi i bunn. Sammen beskriver begge sidene balansen mellom energitypene på objektets ferd gjennom banen. Her er tyngekraften $G = Mg$, der M er massen til objektet og g er tyngdens akselerasjon. Dette gjøres også videre i rapporten.

$$\frac{1}{2}Mvh^2 + \frac{1}{2}Mvh^2C = Mgh \quad (1)$$

C har med massetetheten til objektet å gjøre. Fra at $h = y_0 - y$ får man følgende uttrykk for v , denne gang som funksjon av y .

$$v(y) = \sqrt{\frac{2g(y_0 - y)}{1 + c}}$$

I dette eksperimentet uttrykkes ligningene som funksjon av plassering langs x-aksen (horisontalt) istedenfor t eller y , slik som rett over. Siden banefunksjonen er kjent kan man finne ut av begge størrelsene. Dermed får man følgende formel (ligning 2).

$$v(x) = \sqrt{\frac{2g(y_0 - y(x))}{1 + c}} \quad (2)$$

2.2. Normalkraft

For å regne ut normalkraften brukes Newtons 2. lov uttrykt ved sentripetalakselerasjon, a_{\perp} . Sentripetalaksesasjonen er gitt ved $a_{\perp} = v^2 k$, der k er krumningen på banen. Krumningen er gitt ved

$$k = \frac{y''}{[1 + (y')^2]^{3/2}} \quad (3)$$

som gir oss

$$a_{\perp} = \frac{2g(y_0 - y(x))}{1 + c} \cdot \frac{y''}{[1 + (y')^2]^{3/2}} \quad (4)$$

Newton 2. lov gir følgende ligning:

$$N - Mg \cos \beta = Ma_{\perp},$$

der β er helningsvinkelen til banen. Dette gir følgende uttrykk for normalkraften:

$$N = M(a_{\perp} + g \cos \beta), \quad (5)$$

der a_{\perp} er gitt ved ligning 4. Helningsvinkelen, β kan regnes ut på følgende måte:

$$\beta = \arctan \frac{dy}{dx} \quad (6)$$

2.3. Akselerasjon

Akselerasjonen til objektet, a , kan regnes ut ved hjelp av Newtons 2. lov for forflytningen til objektet og for rotasjonen om massesenteret (CM).

Ligningen vil bli den samme uavhengig om objektet ruller i oppoverbakke eller nedoverbakke, i henhold til Newtons 2. lov. Bruker man et koordinatsystem parallelt med fartsgrenningen til objektet kan følgende ligning utledes:

$$-Mg \sin \beta + f = Ma \quad (7)$$

Newton 2. lov for rotasjon om CM er gitt ved

$$fR = -I \frac{d\omega}{dt}$$

Vinkelhastigheten ω kan uttrykkes som $\omega = v/R$. Bruker man dette og at $I = cMR^2$, kan følgende ligning for friksjonskraft, f , utledes.

$$f = -cM \frac{dv}{dt} = cMa \quad (8)$$

Ved å sette ligning 8 som friksjonskraften i ligning 7, får følgende uttrykk for akselerasjonen.

$$a = -\frac{g \sin \beta}{1 + c} \quad (9)$$

2.4. Friksjonskraft

Til slutt utledes friksjonskraften ved å bruke ligningene over (ligning 8 og 9).

$$f = \frac{cM \sin \beta}{1 + c} \quad (10)$$

2.5. Gjøre posisjon og fart til funksjon av tid

Uttrykkene som ble opparbeidet i seksjonen over kan brukes for å finne vertikal posisjon og fart som funksjon av horisontal posisjon. Under presenteres formler for å uttrykke det samme som funksjon av tid.

Isolerer først fart i x -retning.

$$v_x(x) = v \cos(\beta) \quad (11)$$

Videre, for å være litt mer presise, regnes ut gjennomsnittsfart mellom hver av punktene. I den anvendte utregningen i python brukes $\Delta x = 1$ mm.

$$\bar{v}_x(x) = \frac{v_x(x + \Delta x) + v_x(x)}{2}$$

En formel for endring i tid i gitt en økning i x kan uttrykkes på følgende måte.

$$\Delta t(x) = \frac{\Delta x}{\bar{v}(x)}$$

Tilslutt gjenstår å akkumulere sammen fra start for å få den totale tiden for å komme seg til x .

$$t(x) = \int_0^x \Delta t(\chi) d\chi \quad (12)$$

Videre kan man ta inversen av $t(x)$ som gir $x(t)$. Siden dette kan løses i python, trenger man bare å bytte om på aksene.

Uttrykk for $v(t)$ finner man ved å derivere $x(t)$.

2.6. Usikkerhet

Hver gang en tar målinger av noe i den virkelige verden vil man måtte regne med en viss usikkerhet. En vil ha usikkerheter knyttet til systematiske feil, som oppstår når man benytter en bestemt metode, og en vil oppleve det man kaller tilfeldige feil. En kan se nærmere på tilfeldige feil knyttet til ulike målinger ved å benytte stokastisk analyse og statistikk, da man regner med at disse tilfeldige feilene følger en sannsynlighetsfordeling.

Et gjennomsnitt av flere målinger med tilfeldige feil vil gi et estimat for den mest sannsynlige verdien for det som måles. Gjennomsnittet, \bar{x} , for N målinger der x_i er målingen til i 'ende måling er gjengitt i ligning 13.

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (13)$$

Da kan man regne ut et standardavvik, som vil gi avviket hver enkelt måling har til den mest sannsynlige verdien, nemlig gjennomsnittet. Dette kan man også se på som usikkerheten til hver enkelt måling[2]. Er denne verdien lav kan en si at målingene er presise. Standardavviket, δx , er gjengitt i ligning 14.

$$\delta x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (14)$$

Hvis en ønsker å finne usikkerheten til selve gjennomsnittet, kan standardfeilen finnes gjengitt i ligning 15.

$$\delta \bar{x} = \frac{\delta x}{\sqrt{N}} \quad (15)$$

Det er vanlig å oppgi en usikker verdi, som for eksempel et gjennomsnitt, på denne måten: $\bar{x} \pm \delta \bar{x}$. Det er viktig å ta høyde for usikkerhet ved alle verdier som er direkte målt da denne usikkerheten vil forplante seg i de senere analysene.

3. Metode

3.1. Numerisk metode

I den numeriske delen av laben var ønsket å blant annet regne ut objektets posisjon og fart til et gitt tidspunkt langs banen. Det ble brukt en ren implementasjon av formlene oppgitt i teorideelen, men hovedsakelig *cube-spline*-biblioteket til *scipy*, for å definere banen til forsøket numerisk, og ligning 2 ble brukt for å beregne fart. Videre ble fart i x-retning funnet som funksjon av tid. Denne data-en kunne dermed sammenlignes med den rå posisjonsdataen som senere ble skaffet med Tracker.

Alle genererte grafer og data er gjengitt i seksjon 4. Skriptet er gjengitt i appendiks A. I tabell 1 gjengis hvilke metoder eller formler som ble brukt i skriptet for å finne de ulike størrelsene numerisk.

Tabell 1: Hvordan numeriske størrelser ble funnet.

Numeriske størrelser	Formel eller metode brukt
Banens krumming over x	3
Banens helningsvinDEL over x	6
Hastighet over x	2
Normalkraft over x	5, 6 og 4
Forholdet f/N over x	10, 6 og normalkraft over x
x over t	Invers av 12
v over t	Derivert x over t

3.2. Eksperimentell metode

I neste del i laben, som ble gjennomført 25. februar, ble selve forsøket gjennomført og det ble samlet inn data til sammenligning med numerisk del.

3.2.1. Oppsett

Oppsettet til eksperimentet ble satt opp i tråd med feste punktene i seksjon 4.1. Fotostativet ble plassert med et mobilkamera foran oppsettet, slik at hele oppsettet var synlig i bildet. I tillegg ble to målestokker teipet til oppsettet for referanse. Se bilde 2 for oppsettet.



Figur 2: Bilde av oppsettet.

Vekten som ble brukt for å veie objektene hadde en oppført usikkerhet på 0,1 gram. Dimensjonene ble målt med en usikkerhet på 0,5 millimeter. Målingene i tabell 2 ble gjort to ganger uavhengig av hverandre.

Tabell 2: Fysiske størrelser i eksperimentet.

Gjenstand	Vekt	Dimensjoner
Kule	30,0 g	Diameter: 2,2 cm
Ring	13,2 g	Diameter: 5,0 cm (ytre) og 4,3 cm (indre) Høyde: 1,0 cm liggende

3.2.2. Gjennomføring

10 forsøk med en kule og 10 forsøk med en ring (figur 3) ble gjennomført. For å få til litt variasjon i fremgangsmåten, ble to personer valgt til å slippe gjenstandene halvparten av gangene hver.



Figur 3: Grunnet problemer med å spore ringen i Tracker, ble det valgt å teipe på et punkt slik at den ble lettere å spore.

3.2.3. Hente inn data

Filmene ble analysert i Tracker. Tracker genererte datasett for hvert forsøk (2 gjenstander ganger 10 forsøk) som senere ble analysert med et python-skript. Skriptet, som gav gjennomsnitt, standardavvik og standardfeil til dataene, er gjengitt i appendiks B og resultatene fra skriptet er gjengitt i seksjon 4.

4. Resultater

4.1. Valg av bane

Skriptet i `cubicspline.py`¹ ble brukt til å generere ulike baner. Banen som er avbildet i figur 4 ble valgt.

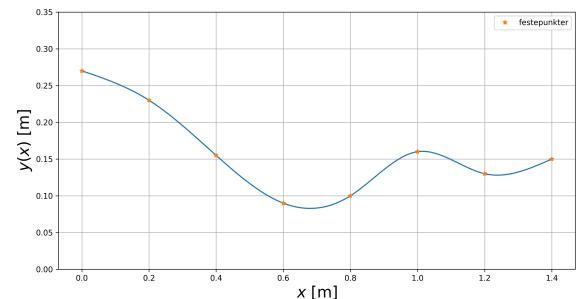
Festepunktene (i meter) som ble valgt til slutt² var: [0,262, 0,194, 0,219, 0,142, 0,090, 0,131, 0,093, 0,146].

4.2. Numeriske resultater

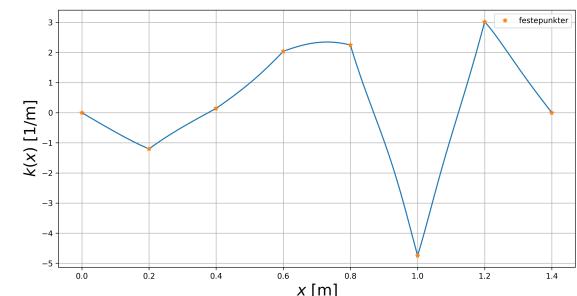
De numeriske beregningene ble regnet ut ved hjelp av et python-skript. Skriptet ble brukt for å generere grafene og dataene. En finner hele skriptet som vedlegg i appendiks A.

4.2.1. Beregninger av banen

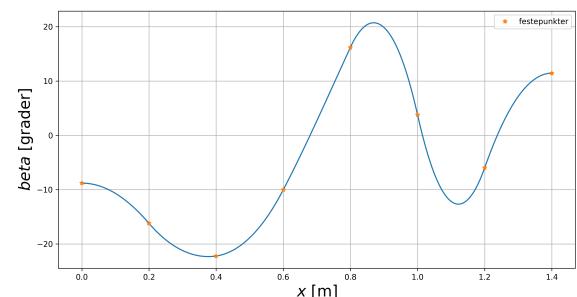
Figurene 5 og 6 beskriver banen i dette forsøket. Viser også til figur 4.



Figur 4: Grafen til banen. Festepunkter i vertikal retning som funksjon av posisjon i horisontal retning.



Figur 5: Banens krumming som funksjon av posisjon i horisontal retning.

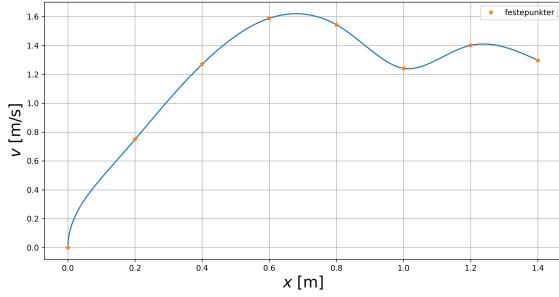


Figur 6: Banens helningsvinkel som funksjon av posisjon i horisontal retning.

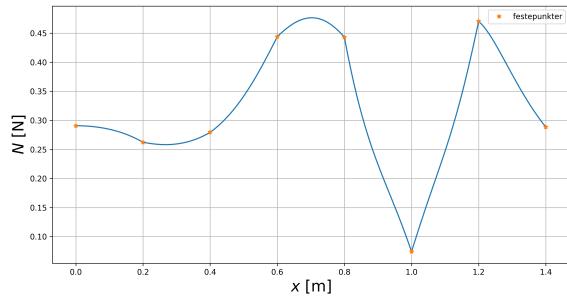
¹Dette skriptet ble utlevert av emneansvarlige.
²Da normalkraften ble regnet ut som funksjon av posisjon i horisontal retning, $N(x)$, i den numeriske delen, fant en ut at gjenstanden ville forlate planet en kort stund, dvs. at $N(x) \leq 0$ for noen verdier av x . Løsningen ble å justere et punkt, slik at N akkurat ble større enn null i det aktuelle punktet.

4.2.2. Kulens bevegelse

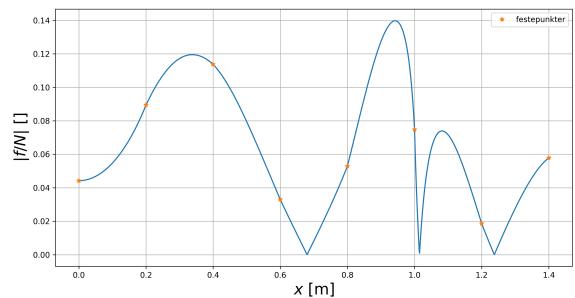
Se figurene 7 til 11 for kulens numeriske resultater.



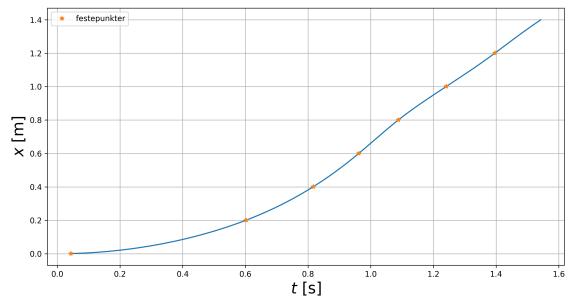
Figur 7: Kulens hastighet over posisjon i horisontal retning. Kulen oppnår høyestehastighet når $x = 0,68$ m, hvor den har en hastighet på 1,62 m/s. Fra figur 4 kan vi også se at dette er banens laveste punkt.



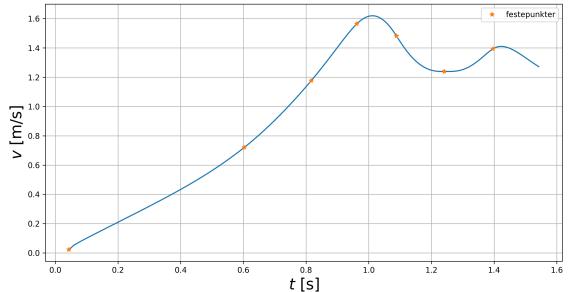
Figur 8: Normalkraft som funksjon av posisjon i horisontal retning forkulen. En kan se at normalkraften alltid er over 0 N, så kuleen forlater aldri overflatene.



Figur 9: Forholdet mellom friksjonskraft og normalkraft som funksjon av posisjon i horisontal retning forkulen.



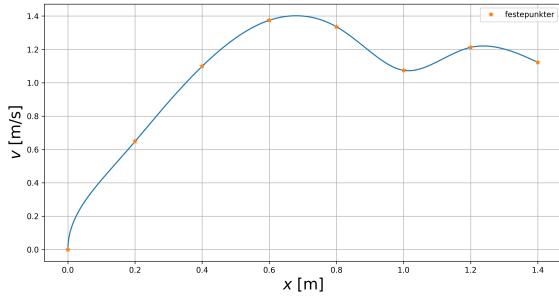
Figur 10: Horisontal posisjon som funksjon av tid forkulen. En kan se at hele reisen tar litt over 1,54 s.



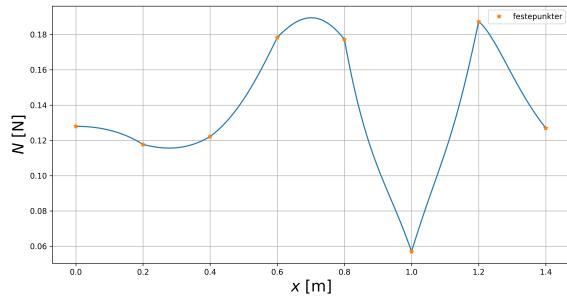
Figur 11: Hastighet som funksjon av tid forkulen. En kan se at kuleen når sin topphastighet etter litt over 1 s.

4.2.3. Ringens bevegelse

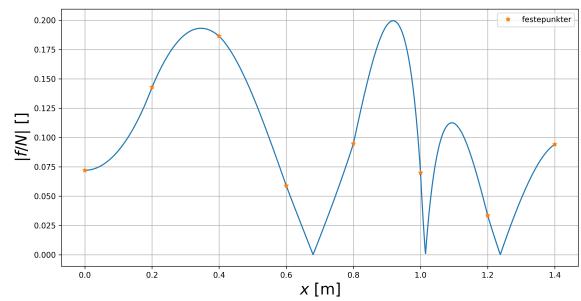
Se figurene 12 til 16 for ringens numeriske resultater.



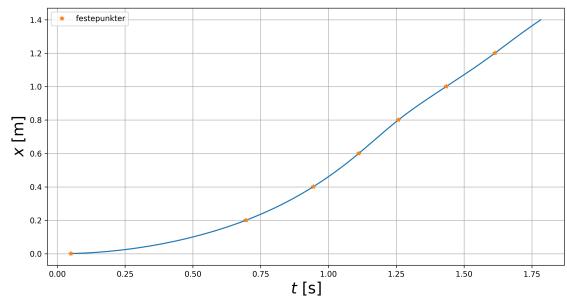
Figur 12: Ringens hastighet over posisjon i horisontal retning. Ringen oppnår sin høyeste hastighet når $x = 0,68$ m, samme som for kulen vår. Den oppnår en maksimal hastighet på 1,4 m/s, noe tregere enn kulen.



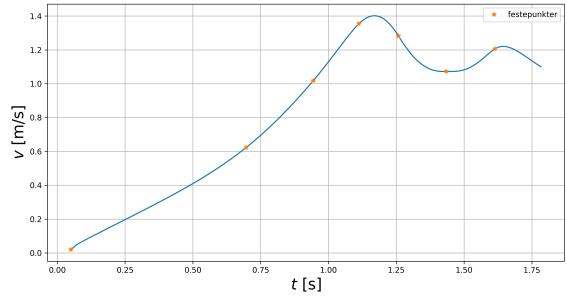
Figur 13: Normalkraft som funksjon av posisjon i horisontal retning for ringen. Samme som for kulen, så er normalkraften alltid over 0 N, så den forlater aldri overflaten.



Figur 14: Forholdet mellom friksjonskraft og normalkraft som funksjon av posisjon i horisontal retning for ringen.



Figur 15: Horisontal posisjon som funksjon av tid for ringen. En kan se at hele reisen tar litt over 1,78 s.



Figur 16: Hastighet som funksjon av tid for ringen.

4.2.4. Fart ved endepunkt for kule og ring

Kule	$1,27 \pm 0,1$	m/s
Ring	$1,1 \pm 0,1005$	m/s

4.3. Målte resultater

Måledataene ble matet inn i et skript (se appendiks B) som gav dataene gjengitt i tabell 3.

Tabell 3: Analyse av hastighetsdataene ved slutt-punktet samlet inn etter fysisk eksperiment. Enheterne er i m/s. Gj.snitt. = Gjennomsnittshastighet. SA = Standardavvik. SE = Standardfeil.

	Kule	Ring
Gj.snitt.	1,246	0,8090
SA	0,2065	0,1596
SE	0,06531	0,05046

5. Diskusjon

5.1. Diskusjon av numeriske resultater

I posisjon $x = 1$ m har vi et betydelig oppdrag i normal-kraft både for kulen og for ringen. Dette er noe som gir veldig mening da vi på dette punktet er på et toppunkt rett etter at gjenstanden høyeste hastighet har blitt oppnådd (i banens lavest punkt som er i $x = 0,7$ m). Det gjenstanden opplever i $x = 1$ m er nærmest "et hopp". Det er dog ikke et hopp, siden vi ser at gjenstanden alltid har kontakt med banen ved at N aldri er negativ eller 0.

5.1.1. Usikkerhet til numeriske resultater

Vi har usikkerhet knyttet til målingen av våre fysiske størrelser i tabell 2. Vekten gir et bidrag på usikkerheten med 0,1 g. Siden det å måle størrelser til kuler er vanskelig, kan vi regne med en betydelig usikkerhet knyttet til bruk av vårt måleutstyr (også da målebåndet i seg selv introduserer en usikkerhet). Vi anslår usikkerheten tilknyttet måling av dimensjoner til å være 0,5 mm.

Dette gjør at vi har en usikkerhet som forplanter seg i våre numeriske beregninger. Når vi i seksjon 4 presenterer en sluttfart vil den ha en usikkerhet knyttet til seg grunnet tyngdeakselerasjonen vi bruker og dimensjonene som vi mäter.

Hvis vi ser på kulen har vi bare usikkerhet knyttet til g , og siden den er under rottegnet i uttrykket for $v(x)$, har vi en usikkerhet på 0,1. Her har vi gått utifra at $g = 9,81 \pm 0,01$ m/s², da g ved Oslo-området er 9,819 m/s²[3]. Vi anslår at g i Oslo-området er tilnærmet lik g i Trondheim.

Hvis vi ser på ringen har vi ikke bare usikkerhet fra g , men også fra en variabel som er avhengig av en usikker storrelse. c -konstanten for ringen er $c = (1 + r^2/R^2)/2$, der r er indre radius og R er ytre radius (begge med usikkerhet på 0,5 mm). Dette betyr at ved å benytte utrykket i likning 2, får vi en usikkerhet på 0,1005 m/s og vi benytter derfor 2 gjeldende siffer (bestemt av måling av dimensjoner i

tabell 2). Vi ser for oss at typen usikkerhet i målingen av gjenstandenes dimensjoner er en statisk feil da vi brukte samme måleutstyr og gjorde flere målinger.

5.2. Diskusjon av eksperimentelle resultater

Da vi satte opp oppsettet sørget vi for etter beste evne, med vater, at den ikke var skjev. Alle festepunktene ble målt to ganger for å forsikre oss om at de ble målt riktig.

Vi ser av tabell 3 atkulen har høyere gjennomsnittsfart enn ringen. Dette kan forklares med ligning 2 for fart og dens utledning i seksjon 2. Farten avhenger av den kinetiske energien som vi har opparbeidet oss gjennom banen. Den er sammensatt av translasjonsenergi og rotasjonsenergi. Massen til gjenstandene er viktig i begge energiformene og gjør derfor at den kompaktekulen med klart mest masse opparbeider seg mest kinetisk energi. I ringens tilfelle gis massen litt mer betydning enn kulens hvis vi ser på rotasjonsenergi grunnet c -konstanten, men dette er altså ikke nok.

5.2.1. Usikkerhet til eksperimentelle resultater

Vi har flere kilder til avvik og usikkerhet når vi utfører selve eksperimentet. Vi kan velge å slippe gjenstanden ulikt fra start (altså gi gjenstanden en liten dytt helt i starten) og startposisjonen som vi slipper gjenstanden fra kan være litt ulik.

Det introduseres også en usikkerhet knyttet til vår filming av eksperimentet. Vår opplosning og valg av bilder pr sekund (vi hadde 30 bilder per sekund og rundt 1,1 piksler per millimeter) er altså med på å gi en usikkerhet. Det vil også ligge en usikkerhet i selve oppsettet, da vi ikke kan være helt sikre på at banen er målt riktig. Likevel anser vi denne usikkerheten som neglisjerbar, da vi gjorde to uavhengige målinger av banen.

Hvis vi skal tallfeste usikkerheten og avvikene som nevnt over kan vi bruke mange av elementene fra skriptet i appendiks B. Vi regner oss frem til verdiene for standardavvik, som er et vanlig mål på usikkerheten i hver enkelt måling (se seksjon 2.6), gjengitt i tabell 4. Denne tabellen sier noe om avvikene knyttet til vår slipping av gjenstandene.

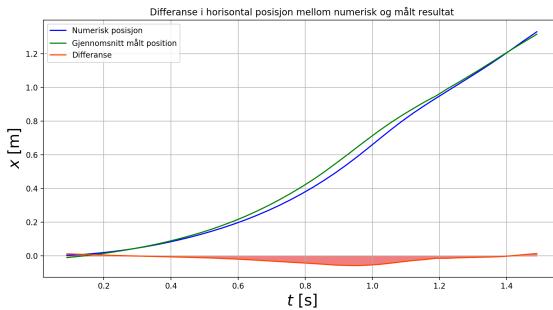
Tabell 4: Standardavvik for flere verdier ved $t = 0$ i eksperimentell del.

	Kule	Ring
Posisjon [m]	0,02756	0,007221
Fart [m/s]	0,06905	0,03294

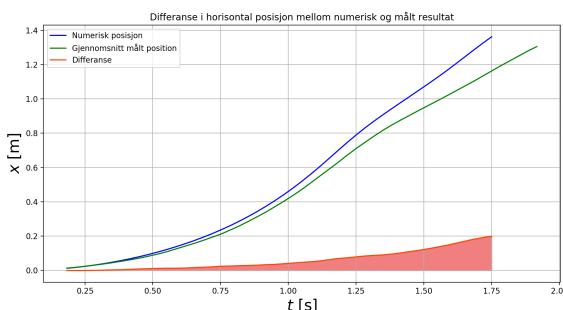
Gjennomsnittsfarten gjengitt i tabell 3 er oppgitt med standardavvik og standardfeil. Vi ser at den er ganske lik for kule og ring, noe som kan tyde på at vi har vært ganske konsistente med avvikene våre.

5.3. Sammenligning av numerisk analyse og gjennomføring

I figurene 17 og 18 sammenligner vi våre resultater fra numerisk analyse og faktisk gjennomføring av forsøk. Differansen er også ført inn i plottene. Som vi kan se er det ganske god overensstemmelse mellom dataene. Kulen avviker minst fra nummerikken, mens den tynne ringen får større og større differanse jo lenger ut i banen vi går. Noe av grunnen til dette kan være slingringen som den tynne og lette ringen utsettes for som ikke den kompaktekulen opplever.



Figur 17: Horisontal posisjon forkulen. Numerisk og målt.



Figur 18: Horisontal posisjon for ringen. Numerisk og målt.

Vi ser at kulens numeriske hastighet i endepunktet ($1,27 \pm 0,1$ m/s) er innenfor vårt konfidensintervall for kulens målte hastighet i endepunktet. Tabell 3 gir et konfidensintervall på $1,246 \pm 0,06531$. Her har vi da sett på gjennomsnittet av målingene og standardfeilen. Likeså er målt gjennomsnitt innenfor usikkerheten vår tilknyttet vår numeriske beregning.

Ringens numeriske hastighet i endepunktet regnet vi til å være $1,1 \pm 0,1005$ m/s, og vi ser at denne verdien ikke er innenfor vårt målte konfidensintervall ($0,8090 \pm 0,8090$). Grunnen til dette antar vi er, som tidligere nevnt, at ringen slingret i banen. Dette var ikke noe vi klarte å ta høyde for i vår numeriske analyse, men som kom til syne i den virkelige verden.

Vi kan i ettertid se at vi har en usikkerhet i banens faktiske form med tanke på overensstemmelse mellom festepunktene brukti numerisk analyse og festepunktene brukti

virkeligheten. Dette har vi ikke tatt hensyn til i beregningen av usikkerhet.

6. Konklusjon

Den numeriske farten for kula ved endepunktet, u , ble regnet ut med følgende formel

$$u = \bar{v} \pm SE$$

der $u = 1,27 \pm 0,1$ m/s, $\bar{v} = 1,246$ (målt gjennomsnittsfart ved endepunkt) og $SE = 0,06531$ (standardfeil). Med andre ord holder u seg innenfor konfidensintervallet. Mye av grunnen til dette er at det er en viss usikkerhet tilknyttet beregningene som ble gjort, noe som gjør at ting oppfører seg slik som forventet.

Det ble også observert noen forskjeller på å slippe ned en kompakt kule og å slippe ned en tynn ring. For det første stemmer observasjonene og de numeriske beregningene overens med teorien. Man kan blant annet se at kulen triller raskere enn ringen grunnet dens større massetetthet. For det andre ble det observert at ringen avviker mer fra nummerikken enn kulen. Dette er sannsynligvis grunnet større påvirkning fra luftmotstand og slingring, som gjør seg mer gjeldende med et mindre massivt objekt, og siden det er vanskeligere å ha kontroll på de påvirkningskraftene i numeriske beregninger.

Referanser

- [1] NTNU, Institutt for fysikk: *TFY41xx Fysikk: Labprosjekt Vår 2020*. <http://home.phys.ntnu.no/brukdef/undervisning/fyslab/files/labligninger.pdf>, (sjekket 11.02.2020).
- [2] M. B. Lilledahl, V. Risinggård: *Målinger og usikkerhet*. <http://home.phys.ntnu.no/brukdef/undervisning/fyslab/files/usikkerhet.pdf>, (sjekket 16.03.2020).
- [3] Wikipedia: *Tyngdeakselerasjon*. <https://no.wikipedia.org/wiki/Tyngdeakselerasjon>, (sjekket 17.03.2020).

Appendiks

A. Numeriske beregninger og plot

Skriptet under ble brukt til å beregne numeriske verdier for første del av laben og til å generere plottene våre.

```
#!/bin/env python3

import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import CubicSpline
from sys import argv

headless = False
saveFile = True
ball = False

for arg in argv[1]:
```

```

if "==" in arg:
    key, value = arg.split("=", 2)
    if "headless".startswith(key.lower()):
        headless = value[0].lower() in ["t", "1", "y"]
    if "savefile".startswith(key.lower()):
        saveFile = value[0].lower() in ["t", "1", "y"]
    if "ball".startswith(key.lower()):
        ball = value[0].lower() in ["t", "1", "y"]

print(f"[*] headless = {headless}")
print(f"[*] saveFile = {saveFile}")
print(f"[*] ball = {ball}")

# -----
# Innstillinger:

# -----
# Banepunkter:
# Bane fra PDF
y_feste = np.asarray([0.270, 0.230, 0.155, 0.090,
                     0.100, 0.160, 0.130, 0.150])
# Vår bane
# y_feste = np.asarray([0.262, 0.194, 0.219,
#                      0.142, 0.075, 0.131, 0.093, 0.146])
# Vår bane endre litt slik at ballen ikke letter
# y_feste = np.asarray([0.262, 0.194, 0.219,
#                      0.142, 0.090, 0.131, 0.093, 0.146])
# -----


# -----
# Type objekt:
# c = 1/2           # kompakt skive
if ball:
    c = 2/5          # kompakt kule
else:
    r = 0.043 / 2    # vår målte indre radius
    R = 0.050 / 2    # vår målte ytre radius
    c = (1 + r**2 / R**2)/2 # ring, r = indre
                           # radius, R = ytre radius
# -----


# -----
# Masse:
# Vekt i oppgave PDF
# M = 0.100
if ball:
    # Vår målte vekt av kule
    M = 0.0300
else:
    # Vår målte vekt av ring
    M = 0.0132
# -----


# -----
# Data:
# Tyngdekraft:
g = 9.81

# Punkter
xmin = 0.000
xmax = 1.401
dx = 0.001
x = np.arange(xmin, xmax, dx)  # verdier [0.001,
                               0.002, 0.003, ..., 1.397, 1.398, 1.399,
                               1.400]

# Horizontal avstand mellom festepunktene er 200
# mm
h = 0.200

```

```

x_feste = np.asarray([0, h, 2*h, 3*h, 4*h, 5*h,
                     6*h, 7*h])

# Generer bane fra punkter
cs = CubicSpline(x_feste, y_feste, bc_type='natural')
# bane
y = cs(x)
# bane derivert
dy = cs(x, 1)
# bane dobbelt derivert
d2y = cs(x, 2)
# -----
# Utregninger

# beta = banens helningsvinkel
beta = np.arctan(dy)
beta_deg = np.degrees(beta)  # beta i grader/
                             degrees

# k = banens kruming
k = d2y / (1 + (dy**2))**(3 / 2)

# v = hastighet i hvert punkt
v = np.sqrt((2 * g * (y[0] - y)) / (1 + c))  # 2
                                                 gjeldende siffer

# a = sentripedalakselerasjon
a = v**2 * k

# N = normalkraft
N = M * (g * np.cos(beta) + a)

# f = statisk friksjonskraft
f = (c * M * g * np.sin(beta)) / (1 + c)
fN = np.abs(f / N)

# v_x = hastighet i x retning
v_x = v * np.cos(beta)

# v_x_mean = 0.5 * (v_x[n-1] + v_x[n]) = parvis
# gjennomsnitt mellom alle v_x punkter
v_x_mean = (v_x[1:] + v_x[:-1]) / 2

# t_delta = tid objektet har brukt mellom hvert
# punkt
t_delta = dx / v_x_mean
# bytt ut inf med 0
t_delta[t_delta == np.inf] = 0

# t_x = kummulativ sum = tid objektet brukte til
# hvert punkt
t_x = np.cumsum(t_delta)

# -----
# Plotting:

filename = None

def fig(title, y, y_unit, x='x', x_unit='m'):
    global filename
    filename = title
    plt.figure(title, figsize=(12, 6))
    # plt.title(title)
    plt.xlabel(f"${x} [{x_unit}]", fontsize=20)
    plt.ylabel(f"${y} [{y_unit}]", fontsize=20)
    plt.grid()

def plot(x, y, show=True, *args, **kw):

```

```

plt.plot(x, y, *args, **kw)
plt.plot(x[::200], y[::200], '*', label='festepunkter')
plt.legend()
if show:
    if not headless:
        plt.show()
    elif saveFile:
        global filename
        filename = filename.lower().replace(" ", "-")
path = f'fig/{filename}.png'
plt.savefig(path, dpi=300)
print(f'Saved file {path}')


orig_round = round

def round(num, digits=0):
    if digits < 0:
        num *= 10**(-digits)
    num = orig_round(num, digits)
    if digits < 0:
        num /= 10**(-digits)
    if digits <= 0:
        num = int(num)
    return num


# - - -
# Plot banens form

print(f'Start høyde, y[0] = {y[0] * 1000}mm')
print(f'Laveste festepunkt, y[{np.argmin(y_feste)}] = {np.min(y_feste) * 1000}mm')
print(f'Laveste punkt, y[{np.argmin(y)}]mm = {round(np.min(y) * 1000)}mm')

fig('Banens form', 'y(x)', 'm')
plt.ylim(0, 0.350)
plot(x, y)

# - - -
# Plot banens helningsvinkel
print()

beta_max = round(np.max(np.abs(np.rad2deg(beta))), 1)
print(f'Banens absolute helningsvinkel <= {beta_max} grader')

fig('Banens helningsvinkel', 'beta', 'grader')
plot(x, np.rad2deg(beta))

# - - -
# Plot banens krumning
print()

k_min = round(np.min(k))
k_max = round(np.max(k))
kr_min = round(np.min(np.abs(k[200:-200:200])), 3) * 100 # TODO: Noe virke file her muligens ...?
print(f'Banens krumning ligger mellom {k_min} og {k_max} per meter, slik at minste krumningsradius er {kr_min / 10}cm')

fig('Banens krumning', 'k(x)', '1/m')
plot(x, k)

```

```

# - - -
# Plot fartsgrafen
print()

print(f'Maksimal hastighet oppnås ved x = {np.argmax(v) / 1000}m')
print(f'Maksimal fart = {round(np.max(v), 2)} m/s')
print(f'Laveste punkt, y[{np.argmin(y)}mm] = {round(np.min(y) * 1000)}mm')

fig(f'{"Ballens" if ball else "Ringens"} fartsgraf', 'v', 'm/s')
plot(x, v)

# - - -
# Plot normalkraften
print()

print(f'Laveste normalkraft: {round(np.min(N), 2)} N')
print(f'Største normalkraft: {round(np.max(N), 2)} N')
fig(f'Normalkraft for {"ballen" if ball else "ringen"}', 'N', 'N')
plot(x, N)

# - - -
# Plot forholdet mellom friksjonskraft og normalkraft
print()

print(f'Forholdet mellom friksjonskraften f og normalkraften N overstiger ikke verdien {round(np.max(fN), 2)}')
fig(f'Forholdet mellom friksjonskraft og normalkraft for {"ballen" if ball else "ringen"}', '|f/N|', '')
plot(x, fN)

# - - -
# Horizontal posisjon som funksjon av tid
print()

print(f'Hele reisen tok ca {round(t_x[-1], 2)} sekunder')

fig(f'Horizontal posisjon som funksjon av tid for {"ballen" if ball else "ringen"}', 'x', 'm', 't', 's')
plot(t_x, x[1:])

# - - -
# Plot hastigheten som funksjon av tid
print()

bunnpunkt_n = np.argmin(y)
bunnpunkt_y = round(y[bunnpunkt_n] * 1000)
print(f'Bunnpunktet (x, y) = ({bunnpunkt_n}mm, {bunnpunkt_y}mm) nås etter {round(t_x[bunnpunkt_n], 2)} sekunder')

fig(f'Hastighet som funksjon av tid for {"ballen" if ball else "ringen"}', 'v', 'm/s', 't', 's')
plot(t_x, v_x_mean)

if saveFile:
    if ball:
        path = "analyse/ball.numeric.csv"

```

```

else:
    path = "analyse/ring.numeric.csv"
with open(path, 'w') as f:
    for t, tx, ty in zip(t_x, x, y):
        f.write(f'{t},{tx},{ty}\n')
    print(f'Done writing {path}')

```

B. Analysering av målte data

Skriptet under ble brukt til å analysere målte data.

```

import numpy as np
import sys

"""
- 'ball' eller 'ring' skal gis som argument!!!
- Argument nummer to er ant forsøk gjort
"""

program = sys.argv[1]
observations = int(sys.argv[2])

# Samle data

data = []

for observation in range(observations):
    filename = "målinger/" + program + "/" + str(
        observation + 1) + ".csv"
    file_object = open(filename, "r")
    lines = file_object.readlines()
    siste = list(map(float, lines[-1].strip().split(",")))
    nestsistre = list(map(float, lines[-2].strip().split(",")))
    speedx = (siste[1] - nestsistre[1]) / (siste[0] - nestsistre[0])
    speedy = (siste[2] - nestsistre[2]) / (siste[0] - nestsistre[0])
    speed = np.sqrt(speedx**2 + speedy**2)
    data.append(speed)

# Regne data

gjennomsnitt = sum(data) / observations

SEE = 0
for i in range(observations):
    SEE += (data[i] - gjennomsnitt)**2

standardavvik = np.sqrt(1 / (observations - 1) * SEE)

standardfeil = standardavvik / np.sqrt(
    observations)

# Skrive data

print("Gjennomsnitt:", gjennomsnitt)
print("Standardavvik:", standardavvik)
print("Standardfeil:", standardfeil)

```