

Til studieadministrasjonen
Fysisk institutt, UiO

Rapport fra alternativ ukeoppgave/oblig for FAM-studenter i INF1100

Bakgrunn

FAM-studenter har gitt uttrykk for at de ønsker mer kontakt med fysikk-faget i første semester. De aksepterer og forstår at studieopplegget er som det er, fordi det gir en mer effektiv læring av fysikk at man har vært gjennom nok matematikk og informatikk på forhånd. Problemet er at FAM studenter praktisk talt ikke har hatt **noen** kontakt med fysikk og Fysisk institutt det første semesteret, og det ønsker de en endring i.

Tiltak

Fysisk institutt har høsten 2013 en plan om at FAM-studentene skal gjennomføre en ukeoppgave-oblig i INF1100 og en oblig i MAT-INF1100 i fysikkbygget, hvor temaet er knyttet opp til FAMs fagområder. Vi har søkt MatNat fakultetet og fått økonomisk støtte for at vi kan tilby studentene pizza og brus for at også sosiale sider av studiet kan foregå i fysikkbygget sammen med noen studenter som er kommet lengre i studiet enn dem selv.

Forberedelser

I uke 39 ble FAM-studenter i INF1100 tilbudt en alternativ ukeoppgave/oblig i fysikkbygget. Oppgaven hadde en praktisk del, hvor studentene repeterte Galileis eksperiment i 1672 fra det skjeve tårn i Pisa, men nå ved å slippe gjenstander fra ulike høyder i trappeoppgangene i fysikkbyggets vest- og østfløy. Gjenstandene vi valgte var Carl Angell's muffinsformer, hvor vi kan variere massen (og dermed gravitasjonskraften), men likevel beholde en friksjonskraft som er temmelig uavhengig av masse. Studentene gikk så over til å lage et dataprogram for å plote resultatene og å sammenligne med en fallbevegelse hvor det var konstant akselerasjon. Oppgaven er vedlagt.

Oppgaven ble utarbeidet og testet av undertegnede. Oppgaven ble justert ut fra Hans Petter Langtangens planer for undervisning forut for uke 39, og hans planer for INF1100's oblig for den aktuelle uken. Utkast til oppgave ble presentert for Langtangen, og han ga positiv tilbakemelding og forslag til ekstra punkter for at volumet skulle bli passe.

Opplegget og oppgavene ble drøftet med gruppelærerne FAM-studentene har i INF1100, og med supergruppelærer som tar seg av utarbeiding av Devilry-rutiner for innlevering og tilbakemelding på obliger.

Totalt sett ble tiltaket møtt med stor velvilje fra Langtangen og gruppelærerne. To av tre gruppelærere stilte opp da "deres" studenter gjennomførte den alternative oppgaven i fysikkbygget og hjalp til med veiledning i programmering helt gratis (fikk dog pizza og brus, selvfølgelig).

Det hører med til historien at vi valgte å legge den alternative oppgaven til fysikkbygget på kveldstid, fra kl 17 til kl 20. Dette valget er



begrunnet ved at vi derved unngikk kollisjoner med undervisning disse FAM-studentene måtte ha, og vi hadde mer ro i fysikkbygget ved å legge øvelsen til etter kl 17. Dataklasserommet FV329 var også ledig de tre ukedagene mandag - onsdag i den aktuelle tiden. Det var også lettere å få hjelp fra videregående studenter på denne tiden.

Fra Fysisk institutt har Grete Stavik sendt ut e-mail til FAM-studentene i første semester og gitt informasjon om tilbudet. Espen laget et påmeldingsskjema som studentene brukte for å melde seg på (nødvendig for å tilpasse bestillinger av pizza og brus). Kamilla på ekspedisjonskontoret har stått for selve bestillingene. Alle har vist stor velvilje og positivitet.

En viktig faktor i opplegget var at de ferske FAM-studentene skulle møte studenter som var kommet litt lenger i studiet. Jeg spurte Fysisk fagutvalg om de kunne stille med ca to studenter hver av de tre kveldene. Disse studentene skulle hjelpe til ved den praktiske oppgaven, hjelpe til med servering av pizza og brus, og evt også hjelpe til litt med programmeringen om det ble aktuelt. Anne-Marthe Hovda, leder for Fysisk fagutvalg, var positiv og fikk to studenter til å stille hver av de tre kveldene. De gjorde en flott innsats!

Vi trengte også å ha ekspertise på Python-programmering, og det var på ingen måte gitt at de vanlige INF-gruppelærerne ville stille gratis på kveldstid. Dessuten er det visse forskjeller mellom Python-installasjoner ved IFI og ved FI, slik at jeg ønsket å ha ekspertise lokalt. Selv programmerer jeg sjeldent i Python, og kunne derfor ikke gi noe detaljert veiledning til de ferske studentene. Jeg forela problemet for Morten Hjorth Jensen, og han mente at noen av hans master og PhD-studenter antakelig kunne hjelpe til. Via de studentene som hjalp til ved prosjektoppgaven i FYS2130 i vår (Milad, Sveinsson, Dragly) ble det sendt ut en generell oppfordring til CompPhys-studentene om å delta. I alt fem studenter stilte, og var til uvurderlig hjelp.

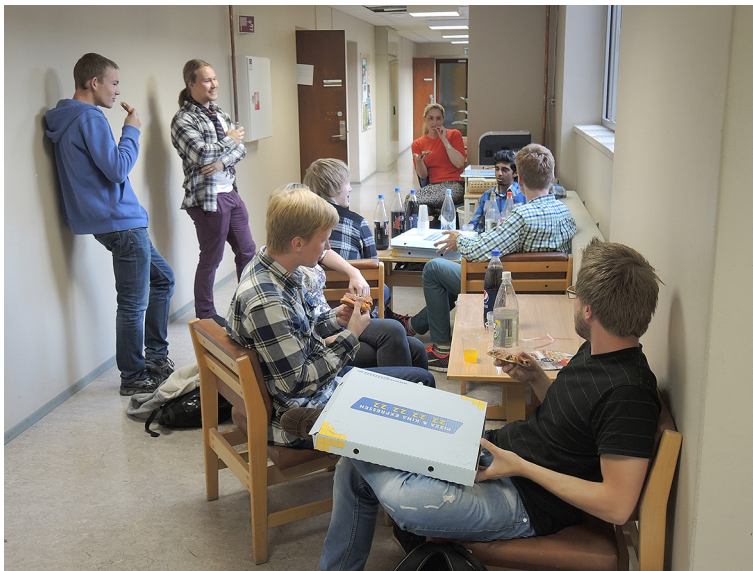
Gjennomføringen

Studentene møtte ”ved pendelen” kl 17. I praksis kom noen for sent, slik at vi ikke startet opp før ca kl 1710. Jeg ønsket vel møtt og holdt en meget kort omtale av oppgaven ut fra Galileis erfaringer hvor jeg blant annet pekte på at resultatet de kom fram til fra den eksperimentelle oppgaven ikke kunne forstås fullt ut etter dagens opplegg. Det trengs numeriske løsningsmetoder for å få en grundigere forståelse for det vi observerer, men at det kanskje kunne bli en annen øvelse senere i studiet.

Studentene ble så delt i grupper på tre, og disse fikk utlevert oppgaveteksten (papirversjon, elektronisk versjon var gitt dem nesten en uke i forveien elektronisk). Hver gruppe fikk også låne en stoppeklokke (utlånt fra Dag Kristian Dysthe for anledningen) og et sett med muffinsformer. Gruppene ble så fordelt på trappeoppgangene i øst- og vest-karr, og studentrepresentantene fra Fysisk fagutvalg fulgte med til hvert sitt sted.

Gjennomføringen av falløvelsen gikk stort sett meget bra. Det var litt mange som skulle slippe muffinsformer ”samtidig” fra ulike etasjer, slik at det iblant ble litt rot i rekkefølgen, men det var ikke et stort problem. Litt verre var det at trafikken rundt trappeoppgangen i fysikkbyggets





vestfløy iblant var så stor at det førte til uheldige luftstrømninger slik at muffinsformene ikke falt helt rett, men kom borti gelendere eller lamper på sin vei ned til første etasje. Det førte til at noen studenter måtte gjenta noen av slippene flere ganger.

Jeg hadde stipulert at den praktiske oppgaven kunne gjennomføres på ca 20 min dersom alt gikk uten venting, men at 30 min var mer realistisk. I praksis brukte gruppene mellom 30 og 45 minutter.

I løpet av denne tiden ble pizza og brus levert (kl 1730) og gjort klar på bordene i gangen utenfor dataklasserommet FV3329. CompPhys-studentene hjalp til her. Etter at gruppene ble ferdig med den praktiske oppgaven, ble de sendt til FV329 hvor de spiste og koste seg med prat internt blant de ferske FAM-studentene og sammen med fagutvalg- og CompPhys-studenter. Spisingen varte minst en halvtime for mange av dem.

Deretter ble det programmering, hvor vi forsøkte å ”trenge oss litt på” for å demme opp for en forventet effekt at terskelen til å be om hjelp kanskje var høyere enn normalt siden settingen var annerledes. Med få unntak syntes dette å fungere bra.

Vi hadde gitt beskjed om at kl 20 stengte vi rommet, og det gikk fint de to første kveldene. Siste kvelden var det fem studenter som gjerne ville sitte litt lenger, og jeg tillot det mens jeg ryddet opp alt etter oss (pizzaesker til returpapir i kjeller i Kjemibygget, tilbakelevering av stoppeklokker, frakte bort flasker som var brukt m.m.). Ca kl 2030 fikk jeg omsider låst døra.

Erfaringer

Jeg synes alt i alt at opplegget har fungert bra, og jeg har fått mange positive tilbakemeldinger fra INF1100-staben, Fysisk fagutvalg-studentene, CompPhys-studentene og ikke minst fra de ferske FAM-studentene selv. Det er uvanlig ved vårt institutt med så mye positivitet.

Alle mest har jeg satt pris på at Eli O Hole, vår leder for utdanning, også har vist stor interesse for prosjektet og spurt om hvordan det har gått og hvilke erfaringer vi har gjort oss. Det er ikke vanlig i vår undervisningspraksis og kan indikere at prosessen med avprivatisering av undervisning allerede er i gang, noe jeg i så fall ønsker velkomment!

Det er likevel forbedringspotensialer for opplegget. Jeg fikk navnene fra Fysisk fagutvalg så sent at jeg ikke rakk å gi disse studentene et eksempel på hvordan den eksperimentelle oppgaven samt programmeringen kunne løses. Det hadde vært bedre om vi kunne hatt en 30 min gjennomgang med disse studentene før de skulle gjøre jobben sin.

Ved INF1100 har de nå to valgfrie ”løp” mhp obliger. De fleste studentene tar ”løp 1” som innebærer ukentlige innleveringer hvor et visst antall må være godkjent for å få gå opp til eksamen. Det var disse studentene vi hadde laget et opplegg for. Noen studenter tar ”løp 2” som innebærer om lag tre større innleveringer i stedet for de ukentlige. Vårt opplegg passet ikke for dem.

Vi visste ikke på forhånd hvor mange som fulgte de ulike løpene, men jeg har siden fått høre at det er bare i størrelsesorden en håndfull studenter som tar løp 2. Vår satsing synes derfor ok.



På den annen side deltok bare 34 av de ferske FAM-studentene. Det er et tall vi ikke er helt fornøyd med. Vi skulle gjerne hatt med flere. Ville det vært bedre å forsøke å gjennomføre dette på dagtid i stedet for kveldstid? Det kan hende, men det ville antakelig bli så mange andre typer problemer med et slikt opplegg at jeg ser ikke helt syn på å forsøke å administrere en slik forandring.

Annonsering ble gjennomført både ved e-mail og ved at Langtangen fortalte om opplegget på en forelesning. Det kan at vi ikke nådde alle på denne måten, siden ikke alle leser e-mail regelmessig og ikke alle går på forelesning. Det kan hende vi neste høst bør fortelle om dette opplegget allerede i fadderuken for å forberede studentene bedre på opplegget.

Det kan også hende at opplegget vårt hadde en for høy terskel for noen. Vi forsøkte å holde oss på en linje som ikke skulle virke skremmende, men noen studenter vil antakelig likevel ikke våge å delta på et slikt opplegg.

En kamel jeg måtte svelge var at obliken ikke kunne lede opp til en rapport hvor fysiske beskrivelser og argumenter kunne brukes som i en vanlig labrapport. Det som leveres i INF1100 er Python-programmer, nothing more. Jeg forsøkte å smugle inn at studentene måtte skrive til skjerm noen få kommentarer om hva de observerte m.m. Dette er en kunstig måte å gjøre ting på, men vi må tilpasse oss det opplegget som finnes i et kurs (laget ut fra andre formål enn fysikk), så slik måtte det bare bli. Vi får forhåpentligvis drøftet fysikk med disse studentene senere!

Et praktisk problem var knyttet til Python generelt. Python har en kjerne og i tillegg en rekke ekstra programbibliotek. Tema for programmeringen i vår uke var plotting. Uheldigvis har IFI og FI installert ulike plottebibliotek. Det førte til litt unødvendige problemer. Dette er en ulempe med Python generelt, og som har ført til at jeg i FYS2130 har lagt opp til at det er Matlab som er førstevalg, og at studentene må ta eget ansvar for biblioteker osv dersom de velger å programmere i Python.

Uansett, totalt sett mener jeg erfaringene er så gode at tiltaket bør gjentas neste høst. Det er imidlertid essensielt å time opplegget i forhold til undervisningen i INF1100 og man må være ute i tilstrekkelig god tid både med INF-gruppelærere, Fysisk fagutvalg og CompPhys-studenter. Selv om velviljen var stor, tar det tid å få alle detaljer på plass.

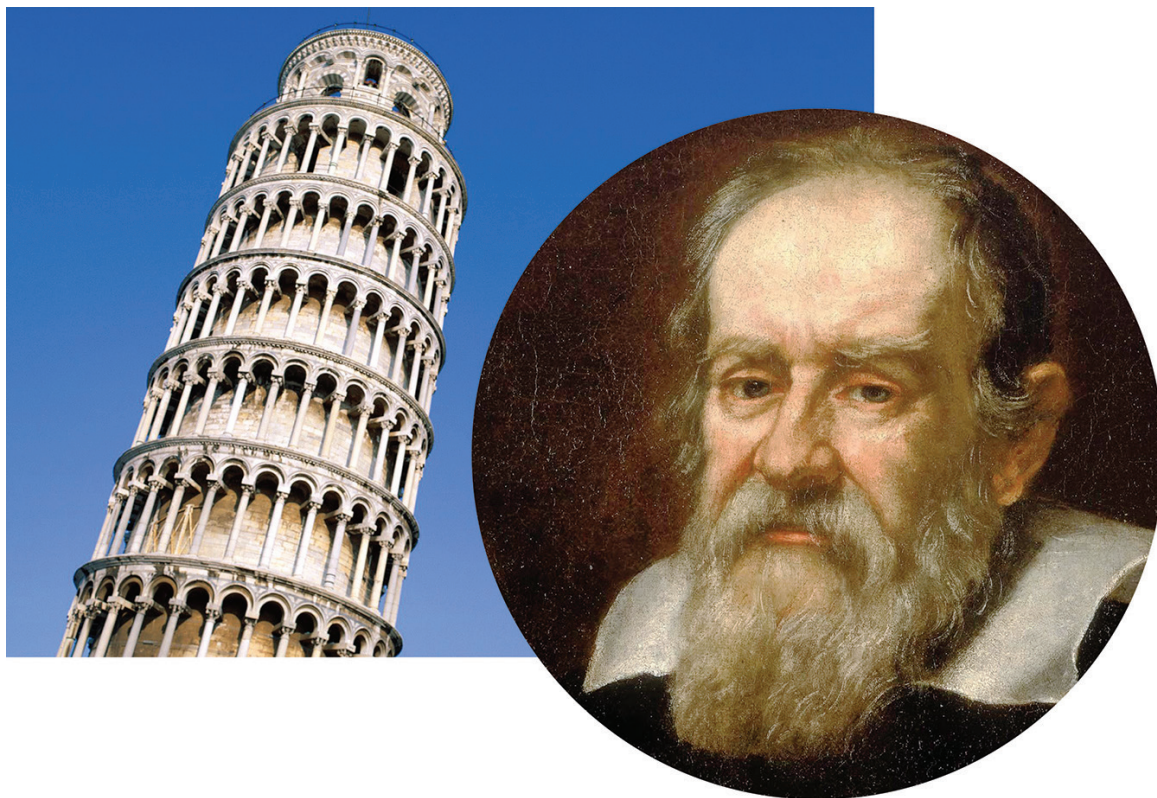
Jeg har nå avtalt med Knut Mørken at jeg skal lage et utkast til en oblig som kan tilbys FAM-studenter i MAT-INF-kurset senere i høst. Det blir en ny utfordring, og jeg håper på like stor imøtekommenhet og hjelp fra impliserte parter også da!

Jeg ønsker å si en hjertelig takk til alle som har bidratt til arrangementet i uke 39!

Arnt Inge Vistnes

Arnt Inge Vistnes

Vedlegg: Invitasjonsskriv, oppgavetekst, et Python-program.



Vil du utfordre Galilei?

En alternativ oblig/ukeoppgave i uke 39 for FAM-studenter som tar INF1100 høsten 2013

Du kan droppe de vanlige gruppetimene i INF1100 uke 39 og i stedet møte opp i Fysikkbyggets vestibyle ("ved pendelen") kl 1700 mandag 23 september dersom du er i gruppe 4, tirsdag 24. september dersom du er i gruppe 3, eller onsdag 25. september om du er gruppe 2. Det er mulig å bytte til en annen kveld enn den du er oppsatt på dersom det er eneste mulighet for at du kan komme. Ledelsen for INF1100 har 19. sept. bestemt at man kan levere inn de ordinære obligatoriske oppgavene i INF1100 denne uka (5.2, 5.3, 5.12 og 5.17) i tillegg til de alternative oppgavene, og få poeng for begge deler.

Vi starter kl 1700 med en kort praktisk oppgave relatert til Galilei, etterfulgt av programmering på et dataklasserom i Fysikkbygget. Gruppelærer og/eller en fysikkstudent som hjelpelærer vil kunne veilede ved programmeringen. Vi gir veiledning til ca kl 20. Innlevering av en oblig i denne alternative oppgaven vil telle likt med oppgaven som de øvrige studentene gjør den aktuelle uken.

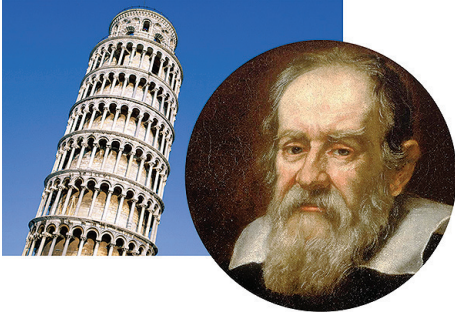
Det blir servert gratis brus og pizza. På grunn av maten må vite hvor mange som kommer. Påmelding senest fredag 20. september på <https://nettskjema.uio.no/answer/56288.html>.

Dersom du vil forberede deg til hva vi skal gjøre, kan du lese første boksen på websiden: <http://hyperphysics.phy-astr.gsu.edu/hbase/airfri2.html>

Forberedelser til programmeringsbiten: Den samme som om du skulle tatt den ordinære INF1100-oppgaven den angitte uka. (Bruk av lister og plotting.)

Oppgavetekst sender vi ut i første del av uke 38, og blir også fra 17. september tilgjengelig på <http://folk.uio.no/arntvi/ggoblig.pdf>.

Har du spørsmål om opplegget, kontakt Arnt Inge Vistnes, a.i.vistnes@fys.uio.no, 93 45 11 91.



Vil du utfordre Galilei?

En alternativ oblig/ukeoppgave i uke 39 for FAM-studenter som tar INF1100 høsten 2013

Når og hvor?

Fremmøte kl 1700 i Fysikkbyggets vestibyle den dagen du har meldt deg på. Ta med en liten notis-blokk eller andre skrivesaker. Studentene vil bli delt opp i grupper på tre eller fire studenter som jobber sammen, i alle fall under den praktiske oppgaven. Hver gruppe får utdelt materiell som trengs for den praktiske oppgaven (stoppeklokk og gjenstander) og blir fordelt mellom østfløy og vestfløy i bygningen. Arbeidet med obligen totalt forventes å ta to til tre timer.

Praktisk oppgave, kort fortalt

Den praktiske oppgaven vil gå ut på å slippe bestemte gjenstander fra trappeoppgangen i 2., 3. og 4. etasje og måle hvor lang tid fallet tar inntil gjenstanden når gulvet i første etasje. Vi kan variere massen på gjenstanden uten at luftmotstanden endrer seg nevneverdig. Vi opererer med tre ulike masser. Det skal foretas fire målinger per etasje og per masse, totalt 36 målinger. Ved litt effektiv organisering innen gruppen, bør dette kunne gjennomføres på ca 20 minutter.

Programmeringen, kort fortalt

Det skal lages et program hvor man skriver inn alle måledataene på en slik måte at vi lett kan lage middelerverdier for de fire målingene som foretas ved samme masse og høyde. Middelerverdiene beregnes, og dataene grupperes som falltid som funksjon av høyde i tre ulike serier (som svarer til ulike masser). Verdien null falltid ved null høyde legges til slik at hver av de tre seriene inneholder fire tall (svarende til falltider ved ulik fallhøyde).

Resultatene plottes i et falltid versus høyde-diagram. Det blir tre kurver i samme diagram. Disse svarer til ulik masse.

Resultatet må kommenteres. Er resultatet i tråd med Galileis observasjoner?

Det skal så lages et program som beregner falltid for ulike fallhøyder dersom det ikke var friksjon.

Resultatene skal plottes.

Sist skal du beregne friksjonskoeffisienten for den formen vår gjenstand har når den faller i luft.

Forberedelser

Det er en fordel (men ikke nødvendig) å repetere "normal" beskrivelse av fall av et legeme i jordens gravitasjonsfelt, og hva som kommer i tillegg ved luftmotstand. For sistnevnte finner du info på

<http://hyperphysics.phy-astr.gsu.edu/hbase/airfri2.html>

For programmeringen er det lurt å lese om lister s 55-58 i læreboka, og om plotting s 198-204.

Vi skal ha det litt gøy også!

Det blir servert gratis brus og pizza, og det legges opp til at folk kan skvadre med hverandre både under den praktiske delen og under programmeringen.

Har du spørsmål om opplegget, kontakt Arnt Inge Vistnes, a.i.vistnes@fys.uio.no, 93 45 11 91.

Mer detaljert om arbeidet

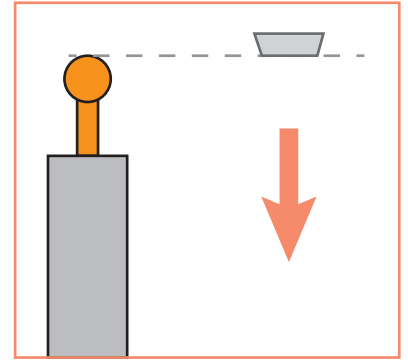
Praktisk oppgave

Gjenstandene skal slippes med sin underkant omtrent i høyde med toppen av gelenderet på det stedet gelenderet i trappeoppgangen er horisontalt.

Horizontal avstand mellom gelender og gjenstanden bør være om lag 30 cm for at gjenstanden ikke skal kollidere verken med lampen eller gelender i etasjene under.

Start stoppeklokka idet gjenstanden slippes. Mål tiden til gjenstanden når gulvet i første etasje.

Det skal gjennomføres fire identiske slipp for hver masse for hver etasje. Resultatene kan f.eks. føres inn i vedlagte skjema.



Høydeforskjellene mellom topp av gelender og gulvet i første etasje for de tre etasjene er:

For østfløyen: 5.11, 9.48, og 13.86 meter.

For vestfløyen: 5.11, 9.45, og 13.87 meter.

Programmeringen, første del

1) Måledataene må inn i programmet. Det kan gjøres på ulikt vis. Vi anbefaler at du i dette tilfellet skriver måledataene inn i selve programmet (du vil lære bedre metoder senere i INF1100). Forsøk å skrive dem inn (bruk f.eks. lister) slik at dataene blir oversiktlige og organisert slik at du lett kan beregne middelerverdien av de fire målingene ved hver forsøksbetingelse. Det kan være en avgjørt fordel å lage en egen funksjon `av()` som beregner middelerverdien av en liste med tall for å få programmet forøvrig mer oversiktlig.

2) Beregn de ni middelerverdiene og legg dem inn i nye lister hvor hver liste svarer til falltidene ved ulike fallhøyder for en bestemt masse.

3) Legg til fallhøyden 0 (som svarer til falltid 0) ved hjelp av insert-kommandoen for de tre listene.

4) Lag en liste som gir fallhøydene, ordnet i samme rekkefølge som falltidene.

5) Plot falltider som funksjon av fallhøyder: Tre ulike kurver (svarer til ulike masse) i samme diagram. Få med takst langs aksene, "legend" og tittel på plottet.

6) Bestem gjennomsnittshastigheten for bevegelsene for de tre ulike massene. Bestem selv hvordan du vil gjøre dette. Resultatet skrives ut til skjermen.

7) Synes bevegelsen å være i tråd med (husk: friksjonskraft = gravitasjonskraften ved terminal hastighet)

- a) en friksjonskraft som er proporsjonal med hastigheten, og som leder til en terminal hastighet proporsjonal med massen, eller
- b) en friksjonskraft som er proporsjonal med kvadratet av hastigheten, og som leder til en terminal hastighet proporsjonal med kvadratroten av massen?

(Henviser til f.eks. <http://hyperphysics.phy-astr.gsu.edu/hbase/airfri2.html>)

Gi en kommentar.

MERK: La programmet skrive til skjerm de kommentarene du vil gi. Det er naturlig å kommentere både linjenes form i plottet og hvordan fallhastigheten varierer med relativ masse. En siste kommentar hvor du sammenligner dine resultater med Galileis, kan også være morsom å ha med.

Dette programmet gis navnet `muffinsEksp.py`.

Fortsetter neste side.

Programmeringen, andre del

1) Lag et program som beregner falltiden for ulike fallhøyder dersom det ikke var noe luftmotstand. Vi vet fra løsning av Newtons lover at fallhøyden h for ulike falltider t er gitt ved

$$h = g t^2 / 2$$

La fallhøyden variere jevnt mellom 0 og 14 m, totalt 100 verdier, og beregn tilsvarende falltid.

2) Plott resultatet, og gi en egnet tekst langs aksene og en egnet tittel på plottet.

3) La programmet skrive til skjerm en kort kommentar om minst to markante forskjeller mellom disse beregningene og resultatet vi fikk fra det praktiske forsøket.

4) Dette programmet gis navnet **fallUtenFriksjon.py**.

Programmeringen, tredje del

1) Luftmotstanden vi har i vårt praktiske eksperiment fører til en terminal hastighet v_{term} gitt ved uttrykket:

$$v_{\text{term}} = \sqrt{\frac{2mg}{C_p A}}$$

Her er

m massen til gjenstanden som faller (0.32 g i vårt tilfelle med relativ masse 1),

g er tyngdens akselerasjon (9.81 m/s²),

C er en friksjonsfaktor som avhenger av gjenstandens form (se figuren nedenfor),










ρ er luftens massetetthet (1.2041 kg/m³ ved 20 grader C),

A er tverrsnittet til den fallende gjenstanden på tvers av fallretningen (diameter er 6.5 cm).

Bruk resultatene fra falleksperimentet dere gjennomførte og analyserte, for å bestemme friksjonsfaktoren C ("drag coefficient" på engelsk). Beregningen skal gjøres i et enkelt python-program, og resultatet skal skrives ut på skjerm sammen med en kort egnet tekst.

Programmet gis navnet **dragCoefficient.py**.

LYKKE TIL!

Shape	Drag Coefficient
Sphere → 	0.47
Halfsphere → 	0.42
Cone → 	0.50
Cube → 	1.05
Angled Cube → 	0.80
Long Cylinder → 	0.82
Short Cylinder → 	1.15
Streamlined Body → 	0.04
Streamlined Halfbody → 	0.09

Measured Drag Coefficients

Falltider ved én etasjes fallhøyde

	Relativ masse 1	Relativ masse 2	Relativ masse 4
1. forsøk			
2. forsøk			
3. forsøk			
4. forsøk			

Falltider ved to etasjers fallhøyde

	Relativ masse 1	Relativ masse 2	Relativ masse 4
1. forsøk			
2. forsøk			
3. forsøk			
4. forsøk			

Falltider ved tre etasjers fallhøyde

	Relativ masse 1	Relativ masse 2	Relativ masse 4
1. forsøk			
2. forsøk			
3. forsøk			
4. forsøk			

Målingene ble foretatt i østfløyen / vestfløyen.
(Stryk det som ikke passer)

Pythonprogram for oppgaven

Koden er skrevet av Yang Tran, en av gruppelærerne i INF1100

```
import matplotlib.pyplot as plt
import math
import numpy
```

```
"""
```

The program is for plotting data from fall-experiments with cupcake cups. The code is rewritten from Arnt Inge Vistnes' code in Matlab suitable for execution in Python.

```
"""
```

```
# (1)
```

```
def avg(numbers):
```

```
    n = float(len(numbers))
```

```
    return sum(numbers)/n
```

```
z1 = [0.0, 5.11, 9.48, 13.86]    # east wing
```

```
z2 = [0.0, 5.11, 9.45, 13.87]    # west wing
```

```
# (2)
```

```
# fall time from 1.floor
```

```
t11 = avg([3.62, 3.83, 3.41, 3.60])
```

```
t12 = avg([2.19, 1.98, 2.40, 2.17])
```

```
t14 = avg([1.88, 2.02, 1.74, 1.93])
```

```
# fall time from 2.floor
```

```
t21 = avg([6.63, 6.69, 6.48, 6.85])
```

```
t22 = avg([4.31, 3.97, 4.47, 5.00])
```

```
t24 = avg([3.43, 2.84, 3.53, 3.78])
```

```
# fall time from 3.floor
```

```
t31 = avg([10.0, 9.21, 10.80, 10.18])
```

```
t32 = avg([5.79, 5.55, 6.02, 5.84])
```

```
t34 = avg([4.53, 4.96, 5.00, 4.81])
```

```
# (3)
```

```
T1 = [0, t11, t21, t31]
```

```
T2 = [0, t12, t22, t32]
```

```
T4 = [0, t14, t24, t34]
```

```
# (5)
```

```
plt.figure()
```

```
plt.plot(z1, T1, '-r')
```

```
plt.plot(z1, T2, '-b')
```

```
plt.plot(z1, T4, '-g')
```

```
plt.xlabel('Drop [m]')
```

```
plt.ylabel('Fall time [s]')
```

```
plt.legend(['Rel. mass 1', 'Rel.mass 2', 'Rel. mass 4'])
```

```
plt.title('Fall time as function of fall height for three relative masses')
```

```

# comments about the result of the plot
print "\nThe plot shows three fairly straight lines which"
print "indicates that the velocity is constant for all heights."
print "The cupcake cups achieve terminal velocity pretty fast after"
print "initialization. The terminal velocity differ with the mass.\n"

# (6)
v1 = avg([z1[i]/T1[i] for i in range(1,4)])
v2 = avg([z1[i]/T2[i] for i in range(1,4)])
v4 = avg([z1[i]/T4[i] for i in range(1,4)])
print 'Average speed for relativ mass 1 is %.2f m/s' % v1
print 'Average speed for relativ mass 2 is %.2f m/s' % v2
print 'Average speed for relativ mass 4 is %.2f m/s\n' % v4

print "The results show that the terminal velocity is proportional"
print "with the square root of relative mass. And they are NOT"
print "valid with Galilei's observations in 1589.\n"

# quick indication if the terminal velocity is proportional with sqrt(m)
mplt.figure()
mplt.plot(z1, [t/math.sqrt(2) for t in T1], '-k')
mplt.plot(z1, [t/2 for t in T1], '-c')
mplt.xlabel('time [s]')
mplt.ylabel('height [m]')
mplt.title('terminal velocity prop. sqrt(m)')
mplt.legend(['T1/sqrt(2)', 'T1/2'])
mplt.show()

```

```

import math
import cupcake_exp

rho = 1.2041
r = (6.5/2)*0.01
g = 9.81
m = 0.32*0.001
C = (2.0*m*g)/(cupcake_exp.v1**2*rho*math.pi*r**2)
print 'Factor of friction C is %.2f' % C

```

```

import matplotlib.pyplot as mplt
import numpy

N = 100
g = 9.81
h = numpy.linspace(0, 14, N)
t = numpy.sqrt(2*h/float(g))
mplt.figure()
mplt.plot(h, t, '-b')
mplt.xlabel('Drop height [m]')
mplt.ylabel('Fall time [s]')
mplt.title('Fall time vs fall height no air resistance')
mplt.show()

```