

Elias Odden Slettevoll

# Programmering i fysikkundervisning

Et verktøy for læring eller læring av et verktøy?

Masteroppgave i Lektorutdanning i realfag

Veileder: Trine Høyberg Andersen

Medveileder: Magnus Strøm Kahrs

Juni 2023



Elias Odden Slettevoll

# Programmering i fysikkundervisning

Et verktøy for læring eller læring av et verktøy?

Masteroppgave i Lektorutdanning i realfag  
Veileder: Trine Højberg Andersen  
Medveileder: Magnus Strøm Kahrs  
Juni 2023

Norges teknisk-naturvitenskapelige universitet  
Fakultet for naturvitenskap  
Institutt for fysikk



Kunnskap for en bedre verden





# Sammendrag

Den nye læreplanen setter et økt fokus på bruken av numeriske metoder, digitale beregningsverktøy og derunder programmering for grunnopplæringen i det norske skolesystemet. Anno våren 2023 er programmering fortsatt ikke en selvfølge i fysikkundervisningen, selv om viktigheten av digital beregningskompetanse har vært økende i samfunnet i lang tid. Det finnes ulike grunner til dette, blant annet at noen elever synes programmering er utfordrende, samt at noen lærere ikke føler de har tilstrekkelig kontroll på programmering selv for å innlemme det i sin undervisning. Denne oppgaven skal belyse hvordan programmering kan benyttes som et verktøy for læring i fysikk, i motsetning til å bruke fysikktimer til å utelukkende lære programmering, gjennom å sette søkelys på elevenes erfaringer angående slik undervisning.

Problemstillingen for denne oppgaven er: *«Hvordan kan programmering som verktøy brukes i undervisning i fysikk 1 på en måte som oppleves relevant for elevene?»*.

Masterstudien kan betraktes som en kvalitativ intervensjonsstudie med fleksibelt design. Med utgangspunkt i relevant teori og et selvutviklet undervisningsopplegg, intervjues fire elever for å få innsikt i hvordan de selv erfarer å bruke programmering i fysikkundervisning i forbindelse med undervisningsopplegget. I tillegg til intervjuene gjennomføres det en spørreundersøkelse blant hele klassen som gjennomførte undervisningsopplegget, for å avdekke generelle funn som kan nærmere bekreftes og utdypes i intervjuene.

Teoretiske perspektiver anser digitale beregningsverktøy som et godt verktøy til undervisning i flere fag som kan bidra med relevante kunnskaper og egenskaper. For fysikkfaget kan digitale beregningsverktøy være relevant i arbeid med abstrakte fenomener og fysiske konsepter. Under digitale beregningsverktøy, kan programmering innlemmes som en naturlig del av fysikkfaget og mane frem egenskaper som blant annet algoritmisk og kritisk tenkning, idéutvikling og argumentasjon.

Resultatene peker mot at programmering kan nyttes som et verktøy for læring i fysikkundervisning, da undervisningsopplegget gir et utbytte for elevene innen flere kompetanser og egenskaper som er tiltenkt faget. Elever synes programmering er utfordrende, men viktig å lære seg. Selv om teorien tilsier at opplegget er relevant for elever i fysikk 1, er det ikke alle elever som mener at fysikken som læres i faget, krever digitale beregningsverktøy. Til tross for dette, ser elever relevansen av digital beregningskompetanse i fysikkdisiplinen generelt og i samfunnet.



# Abstract

The new Norwegian curriculum places increased emphasis on the use of numerical methods, digital computational tools and programming in K-12 education. As of spring 2023, programming is still not an obvious part of physics education despite the growing importance of digital literacy in society for a long time. There are various reasons for this, including some students finding programming challenging as well as some teachers not feeling sufficiently confident in their own programming skills to incorporate it into their teaching. This study aims to shed light on how programming can be used as a learning tool in physics, rather than using physics classes to exclusively learn programming, by focusing on students' experiences regarding such teaching.

The research question for this study is: *"How can programming be used as a tool in teaching Physics 1 in a way that is perceived as relevant by students?"*. The master's thesis can be considered as a qualitative intervention study with a flexible design. Based on relevant theory and a self-developed teaching program, four students are interviewed with the purpose of gaining insight into their own experiences of using programming in physics education within the context of the teaching program. In addition to the interviews, a survey is conducted among the entire class that participated in the teaching program to uncover general findings that can be further confirmed and elaborated on in the interviews.

From theoretical perspectives, digital computational tools are seen as valuable tools for teaching in multiple subjects that can contribute with relevant knowledge and skills. For the subject of physics, digital computational tools can be particularly relevant in decomposing abstract phenomena and physical concepts. Programming can be integrated as a natural part of physics education, promoting qualities such as critical thinking, idea development and argumentation.

The results indicate that programming can be used as a learning tool in physics education, as the teaching program provides benefits for students in several competencies and qualities intended for the subject. Students find programming challenging but important to learn. Although the theory suggests that the program is relevant for Physics 1 students, not all students believe that the physics taught in the subject requires digital computational tools. Despite this, students perceive relevance to the discipline of physics in general, as well as the importance of digital literacy in society.



# Forord

Denne masteroppgaven leveres som avslutningen på mine fem år som lektorstudent i fysikk og matematikk ved NTNU. Studiet har gitt meg innsikt og kunnskap om fysikk, matematikk, pedagogikk og lærerprofesjonen. Når jeg startet utdanningsløpet hadde jeg en visjon om hvilken lærer jeg ønsket å bli. Med inspirasjon fra min egen fysikklærer på Volda videregående skole og gjennom flere praksisperioder, har denne visjonen selvsagt blitt endret en del, men kjernevisjonen og ønsket om å være en engasjert lærer som gjør undervisningen relevant og givende for elever, forblir. Jeg starter nå yrkeslivet som lærer med et ønske om å få bruke alt det jeg har lært, tilegne meg mer kunnskap, i tillegg til å få lov til å være med å forme fremtidens samfunnsborgere og kanskje fysikere og matematikere.

En masteroppgave skrives ikke av seg selv, denne er ikke noe unntak. Jeg ønsker å takke mine veiledere, Trine Høyberg Andersen og Magnus Strøm Kahrs, som har stilt opp ved enhver anledning. De ukentlige veiledningstimene har motivert meg både i forkant av timene og i etterkant. De har svart på alle mine henvendelser, gjerne sent på kvelden og tidlig på morgenen. Masterarbeidet har vært en krevende periode, men med gode støttespillere og veiledere har jeg følt meg godt ivaretatt.

En stor takk rettes mot faglærerne og elevene som har vært med på prosjektet. Humøret, engasjementet og samarbeidsviljen har gitt meg inspirasjon til masterarbeidet og læreryrket. Det har vært stor stas å få lov til å teste ut et undervisningsopplegg som har krevd mye tid, tankekraft og lidenskap.

Videre ønsker jeg å takke medelever og venner for en fin studietid. Takk til Konrad Jervell og Anders A. Tidemann for faglige diskusjoner og god sparring i løpet av fem år. Takk til Svein Ove H. Fagerheim, Jørgen Salvesen, Viljar Jodaa og Martin Slettum for sosiale lag og konspiratoriske tankeeksperiment innen fysikken og de fleste andre disipliner. Jeg ønsker også å takke min familie og min kjæreste, som alltid stiller opp som både korrekturlesere og mental støtte. Jeg er veldig glad i dere.

Til slutt ønsker jeg å takke alle flotte lærere der ute. De som ser hele eleven og klarer å tilpasse undervisningen på en måte som engasjerer. Jeg håper denne oppgaven kan være et bidrag til fysikklærere og skoler, samt gi en innsikt i hvordan elevene kan oppfatte deler av fysikkundervisningen.



# Innhold

|  |      |
|--|------|
| Figurer.....   | XIII |
| Tabeller .....   | XIV  |
| 1 Introduksjon.....  | 1    |
| 1.1 Bakgrunn .....   | 1    |
| 1.2 Problemstilling og forskningsspørsmål.....                 | 2    |
| 1.3 Oppgavens struktur .....                                   | 3    |
| 2 Teori.....   | 5    |
| 2.1 Hva skal elever lære i fysikk 1? .....                     | 5    |
| 2.2 Programmering i undervisningssammenheng .....              | 7    |
| 2.3 Algoritmisk tenkning.....                                  | 9    |
| 2.4 Digital beregningskompetanse .....                         | 13   |
| 2.5 Læreplanverkets tolkning av den algoritmiske tenkeren..... | 16   |
| 2.6 Nasjonal implementering av LK20 .....                      | 18   |
| 3 Casebeskrivelse .....  | 19   |
| 3.1 Generelt om undervisningsopplegget.....                    | 20   |
| 3.2 Didaktiske valg.....                                       | 21   |
| 3.2.1 Samarbeid.....   | 23   |
| Sosiokulturell læringsteori .....                              | 24   |
| 3.2.2 Veiledning .....   | 25   |
| 3.3 Bruken av taksonomi for algoritmisk tenkning .....         | 25   |
| 3.3.1 Datahåndtering .....                                     | 27   |
| 3.3.2 Modellering og simulering .....                          | 30   |
| 3.3.3 Digital problemløsning.....                              | 31   |
| 3.3.4 Systemtenkning .....                                     | 33   |
| 4 Metode.....  | 35   |
| 4.1 Innsamling av data .....                                   | 36   |

|       |  |    |
|-------|--|----|
| 4.1.1 | Utvalg .....   | 38 |
| 4.2   | Analyse av datamaterialet .....  | 40 |
| 4.3   | Etiske betraktninger.....  | 43 |
| 4.4   | Troverdighet .....   | 44 |
| 5     | Resultater .....   | 49 |
| 5.1   | Elevenes arbeid med problemløsning .....   | 49 |
| 5.1.1 | Elevenes møte med numeriske utfordringer.....  | 49 |
|       | Nye problemer kommer til syne i senere oppgaver.....                                       | 52 |
|       | Elevenes utfordringer rundt representasjoner .....   | 53 |
|       | Når fysikk, matematikk og programmering møtes.....   | 54 |
| 5.1.2 | Elevenes strategi for problemløsning .....   | 56 |
|       | Bruke programmet til å ta en fysisk slutning .....   | 56 |
|       | Koden er ikke begrenset til en deloppgave .....  | 57 |
| 5.2   | Elevenes betraktninger om opplegget generelt .....   | 58 |
| 5.2.1 | Elevsamarbeid .....  | 58 |
|       | Samarbeid rundt en datamaskin .....  | 59 |
|       | Samarbeid fører til likere kode.....   | 60 |
| 5.2.2 | Veiledning utenfra gir ikke hele svar .....  | 61 |
| 5.2.3 | Opplevd læringsutbytte .....   | 63 |
| 5.2.4 | Opplevd relevans.....  | 64 |
| 6     | Diskusjon.....   | 67 |
| 6.1   | Elevs perspektiver på programmering i fysikk 1 .....                                       | 67 |
| 6.2   | Fysikk, matematikk og programmering i samme kodelinje.....                                 | 69 |
| 6.3   | Algoritmisk tenkning i praksis .....   | 70 |
| 6.4   | Programmering i fysikkundervisning: Et verktøy for læring eller læring av et verktøy?..... | 71 |
| 6.5   | Forbedringsmuligheter rundt undervisningsopplegget.....                                    | 72 |
| 6.6   | Metodisk refleksjon.....   | 74 |



|     |  |     |
|-----|--|-----|
| 6.7 | Implikasjoner.....   | 77  |
| 7   | Konklusjon.....  | 79  |
|     | Referanseliste .....   | 81  |
|     | Vedlegg .....  | 85  |
|     | Vedlegg A: Undervisningsopplegg - Elevversjon .....                | 86  |
|     | Vedlegg B: Undervisningsopplegg - Løsningsforslag .....            | 89  |
|     | Vedlegg C: Powerpoint-presentasjon .....                           | 98  |
|     | Vedlegg D: Resultat fra spørreundersøkelse.....                    | 101 |
|     | Vedlegg E: Utfyllende intervjuguide med teoretisk forankring ..... | 110 |
|     | Vedlegg F: Informasjonsskriv godkjent av Sikt .....                | 113 |

## Figurer

|  |    |
|--|----|
| Figur 1: En oversikt over hvordan forskningsspørsmålene informerer hverandre, først fra F1 til F3 gjennom F2, så tilbake fra F3 til F2 og F1 igjen. ....                                       | 4  |
| Figur 2: Teoretisk plassering av digital beregningskompetanse i fysikk (Odden et al., 2019, s. 3).....   | 14 |
| Figur 3: Illustrasjon av den algoritmiske tenkeren (Utdanningsdirektoratet, 2019). ....  | 17 |
| Figur 4: Introduksjonsteksten til undervisningsopplegget. ....   | 22 |
| Figur 5: En oversikt over hvilke deler av programmeringen i et program brukeren har tilgang til i ulike tilnærminger (Orban et al., 2018, s. 832).....   | 23 |
| Figur 6: Oppgave 3a (elevversjon).....   | 28 |
| Figur 7: Oppgave 3a, linje 16-19 (løsningsforslag). ....   | 28 |
| Figur 8: Akselerasjonsgraf på oppgave 3b (løsningsforslag). ....   | 29 |
| Figur 9: Fartsgraf og tilleggsspørsmål. Oppgave 3b (løsningsforslag).....  | 30 |
| Figur 10: Oppgave 1a og 1b (elevversjon).....  | 31 |
| Figur 11: Oppgave 3b - Posisjonsgraf (løsningsforslag).....  | 33 |
| Figur 12: Det tematiske nettverket utviklet fra intervjuene. ....  | 41 |
| Figur 13: En oversikt over hvordan hovedtemaene besvarer forskningsspørsmålene. ....   | 42 |
| Figur 14: En visualisering som viser en akselerasjonsgraf der den første verdien har motsatt fortegn i forhold til de andre verdiene. Grafen gjør da et hopp fra positiv til negativ verdi.... | 53 |

# Tabeller

|  |    |
|--|----|
| Tabell 1: Figuren er en oversikt over Weintrop et al. (2016, s. 135) sin taksonomi for «computational thinking». Oversatt av Nordby (2019, s. 28) og videreutviklet av meg. .... | 10 |
| Tabell 2: Odden et al. sitt utvidede rammeverk for de tre pilarer (2019, s. 8). Egen oversettelse. ....  | 15 |
| Tabell 3: En oversikt over hvilke praksiser fra taksonomien for algoritmisk tenkning (fra Weintrop et al., 2016, s. 135) som blir brukt i oppgaven (markert i grønn). ....       | 27 |
| Tabell 4: En oversikt med bakgrunnsinformasjon for hver informant til intervjuene. ....  | 39 |
| Tabell 5: En gjennomgang av hva som kjennetegner en god tematisk analyse (Braun & Clarke, 2006, s. 96). ....   | 43 |
| Tabell 6: En oversikt over hvor mange koder hvert av hovedtemaene bestod av. ....  | 43 |
| Tabell 7: En oversikt over hvilke oppgaver elevene gjennomførte. ....  | 49 |
| Tabell 8: En oversikt over om timen føltes relevant for elevene. ....  | 65 |

# 1 Introduksjon

## 1.1 Bakgrunn

I 2020 trådte en ny læreplan, LK20, i kraft i det norske skoleverket (Kunnskapsdepartementet, 2017). Dette innebar at digitale beregningsverktøy, herunder programmering skulle benyttes i større grad enn tidligere. Grunnlaget for endringen lå i hvordan samfunnet endret seg i en digital retning med stadig større krav til individer om å kunne håndtere digitale verktøy. Denne endringen har skjedd gradvis, og forskere som Andrea A. diSessa (2000) og Jeannette M. Wing (2006) konstaterte for flere år siden programmeringens rolle i samfunnet og behovet for en økt kompetanse innen digitale beregningsverktøy i befolkningen. I læreplanen blir «digitale ferdigheter» nevnt som grunnleggende ferdigheter for fysikkelever, på lik linje med å kunne å regne, lese, skrive og snakke om fysikk. Utdanningsdirektoratet (2021) skriver dette om digitale ferdigheter i fysikk:

*«Digitale ferdigheter i fysikk innebærer å bruke digitale ressurser til å registrere, bearbeide, analysere, modellere og presentere data. Det innebærer også å innhente relevant informasjon for å studere fysikkfaglige fenomener og problemstillinger. Videre innebærer det å bruke programmering og dynamiske verktøy til å utforske fysiske problemstillinger.»*

Selv om den overordnede delen av læreplanen skulle implementeres fra skoleåret 2020/2021 (Utdanningsdirektoratet, 2023), har implementeringen skjedd mer gradvis, sporadisk og lokalt enn det som først var planlagt, mye grunnet COVID-19 (Høvik, 2022). Det er også i stor grad opp til hver respektive faglærer å tolke hvordan momenter fra læreplanen innlemmes konkret i undervisningen (Gjengset, 2022). Dette kan medføre at elementer som digital beregningskompetanse ikke inkluderes i like stor grad som samfunnet rundt preges av den samme kompetansen. Dette strider mot formålet med opplæringen, da Kunnskapsdepartementet (2017) skriver at «elevene og lærlingene skal utvikle kunnskap, kompetanse og holdninger for å kunne mestre livene sine og for å kunne delta i arbeid og felleskap i samfunnet». Satt på spissen, er det fare for at ekskludering av digital beregningskompetanse vil medføre at elever ikke opparbeider seg tilstrekkelige kunnskaper

som kreves i arbeidslivet og samfunnet, og dermed ikke er like godt forberedt til et liv etter skolen.

Min motivasjon for masterarbeidet slekter til en praksisperiode fra 2022, der jeg spurte elever som studerte fysikk 1 om hva de syntes om programmering generelt. Flere av elevene jeg spurte syntes ikke programmering var givende, og de gangene de hadde programmert i matematikk- eller fysikktimene hadde fokuset blitt programmeringsteknisk. Dette gjorde meg oppmerksom på hvor krevende det kan være for både lærere og elever å sette seg inn i endringene som kommer med en ny læreplan. Jeg bestemte meg for å utvikle et undervisningsopplegg som benyttet programmering i fysikkundervisning, men med det formål å gi elevene et størst mulig utbytte innen fysikk, ikke bare programmeringsteknisk. Grunnlaget og motivasjonen for masterarbeidet er nettopp tidligere erfaringer fra praksisperioder og ønsket om å lære mer om hvordan programmering kan benyttes på en hensiktsmessig måte i fysikkundervisningen.

Det finnes tidligere norske studier innen bruken av programmering i fysikkundervisning på videregående skoler, blant annet Jonathan B. Waters sitt masterprosjekt som omhandler dybdeløring i fysikk ved bruk av programmering (2020). I likhet med undertegnede, er stien delvis tråkket opp av Sigurd Nordby sitt masterarbeid som omhandler programmering og algoritmisk tenkning i fysikk, med en grundig litteraturstudie og tre eksempler på undervisningsopplegg. Dette arbeidet ble presentert allerede før den nye læreplanen trådte i kraft (Nordby, 2019). I nyere tid har blant annet Fridtjof Gjengset gjennomført intervju med lærere om hvordan de bruker programmering i sin undervisning for å se hvordan deler av den nye læreplanen brukes i undervisningen i den norske skolen (Gjengset, 2022). Min masterstudie kan sees på som et konkret forsøk på å gjennomføre et undervisningsopplegg som benytter seg av programmering, men med fokus på at elevene skal få ny innsikt og kunnskap i fysikk, i tillegg til elevenes egne erfaringer rundt hvordan slik undervisning bør foregå. Undervisningsopplegget ble også gjennomført av et årskull som på mange måter befinner seg mellom to læreplaner, da de fullførte grunnskolen med den tidligere læreplanen (Utdanningsdirektoratet, 2006), og kun har fått et og et halvt skoleår med undervisning i den nye læreplanen (Utdanningsdirektoratet, 2023).

## 1.2 Problemstilling og forskningsspørsmål

Med motivasjon i fysikkelevenes erfaringer angående programmering, et ønske om å lære mer om feltet og tidligere teori ble problemstillingen med forskningsspørsmål etablert:

P: *Hvordan kan programmering som verktøy brukes i undervisning i fysikk 1 på en måte som oppleves relevant for elevene?*

F1: *Hvorfor skal elever programmere i fysikk 1?*

F2: *Hvordan utvikle et undervisningsopplegg som bruker programmering som verktøy i fysikk 1?*

F3: *Hvilke erfaringer hadde elevene tilknyttet undervisningsopplegget?*

### 1.3 Oppgavens struktur

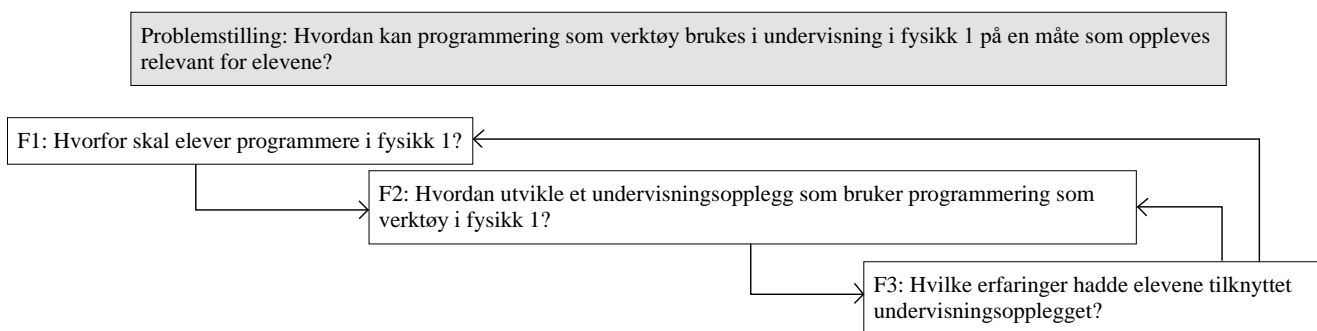
Forskningsspørsmål 1 vil undersøke hva kompetanse i fysikk 1 innebærer, programmeringens tiltenkte rolle i fysikkundervisningen og hvilke deler av denne kompetansen programmering komplimenterer. Weintrop et al. (2016) sin *taksonomi for algoritmisk tenkning (computational thinking)* er en sentral teori i masterarbeidet sammen med teorier som bygger under viktigheten av programmering og *digital beregningskompetanse (digital literacy)* (diSessa, 2000; Odden et al., 2019) i samfunnet, i fysikken og i undervisningssammenheng. Forskningsspørsmål 1 medfører ikke en strukturert litteraturstudie, men danner et teoretisk grunnlag og en tilstrekkelig oversikt til å besvare forskningsspørsmålet og til utvikling av undervisningsopplegget.

Den neste delen av masteroppgaven vil ta for seg hvordan et konkret undervisningsopplegg kan se ut. Forskningsspørsmål 2 vil bruke taksonomien for algoritmisk tenkning fra forskningsspørsmål 1 som en rettleiende mal for hvilke egenskaper hos elevene undervisningsopplegget bør stimulere til, samt generell didaktisk teori sett fra et sosiokulturelt perspektiv. Undervisningsopplegget er utviklet i henhold til kompetansemålet for fysikk 1 som sier at «eleven skal kunne bruke numeriske metoder og programmering til å modellere og utforske bevegelser i situasjoner der akselerasjonen ikke er konstant» (Utdanningsdirektoratet, 2021). Undervisningsopplegget som ble utviklet, ble tatt med ut i en norsk videregående skole og utført hos en fysikk 1 klasse bestående av 22 elever.

Et viktig perspektiv i masterarbeidets begynnelse var nettopp elevenes perspektiv. Derfor ble det gjennomført en overordnet spørreundersøkelse blant elevene som deltok i undervisningsopplegget, i tillegg til tre semistrukturerte intervjuer. For å sørge for at undervisningsopplegget ble utført på tiltenkt måte, var jeg selv tilstede som klasseleder, noe som gav meg muligheten til å gjøre observasjoner som kan støtte opp under funnene gjort i analysearbeidet av intervjuet og spørreundersøkelsen. Forskningsspørsmål 3 har som formål å

undersøke hvilke erfaringer elevene hadde med undervisningsopplegget som ble gjennomført. Med begrepet erfaringer menes elevenes *oppfatninger* og *utfordringer* tilknyttet undervisningsopplegget.

De tre forskningsspørsmålene skal i et samarbeid kunne gi et helhetlig bilde på hvordan et undervisningsopplegg anno 2023 kan bruke programmering som et konstruktivt verktøy for læring i fysikk, og ikke bare som læring av et verktøy i fysikktimene. Som Figur 1 viser, har forskningsspørsmål 1 informert forskningsspørsmål 2 med teoretiske perspektiver. Undervisningsopplegget som er en del av forskningsspørsmål 2, ble utprøvd i praksis der elevene ble intervjuet i etterkant. I tillegg til å besvare forskningsspørsmål 3, dannes det elevperspektiver som gir nye innspill til forskningsspørsmål 2 og forskningsspørsmål 1. Masterstudien gir et viktig bidrag til forskningsfeltet med perspektiver fra målgruppen selv (fysikkelevne) rundt et undervisningsopplegg utviklet fra teori som omhandler målgruppen.



Figur 1: En oversikt over hvordan forskningsspørsmålene informerer hverandre, først fra F1 til F3 gjennom F2, så tilbake fra F3 til F2 og F1 igjen.

Masterstudien kan ses på som en intervensjonsstudie med et fleksibelt design med semistrukturerte intervju som primær metode for datainnsamling. I arbeidet med kvalitative studier der jeg som forsker har stor påvirkningskraft, har for eksempel metodisk triangulering og forskningstriangulering vært viktig. Dette for å sikre at jeg med min bakgrunn og ønske om å danne elevperspektiver som fremmer programmerings rolle i undervisning av fysikk 1 ikke påvirker resultatene i analysearbeidet. Spørreundersøkelsen som ble utført har gitt en større oversikt over elevenes oppfatninger og utfordringer, som videre rettet intervjuene. Intervjuene kunne da fokusere mer på interessante funn fra spørreundersøkelsen. Transkripsjonene av intervjuene har blitt gjennomlest av veiledere, noe som bidro til en styrket analyse. I tillegg har resultatene fra intervjuene blitt satt opp mot resultater fra spørreundersøkelsen og egne observasjoner fra undervisningsøkten med samme formål som forskertrianguleringen.

## 2 Teori

Dette kapittelet presenterer relevant teori som legger grunnlaget for å svare på forskningsspørsmål 1 om hvorfor elever skal programmere i fysikk 1. Kapittelet innledes med direktiver fra Utdanningsdirektoratet om hvilke kunnskaper som er relevant for fysikk 1 og programmerings rolle i grunnutdanningen. Etter dette vil teori om programmerings rolle som verktøy i fysikkundervisningen presenteres. Deler av teorien og skriv fra Utdanningsdirektoratet er brukt i utviklingen av undervisningsopplegget, hvilket gjennomgås i kapittel 3. I kapittel 3 presenteres det også ytterligere teori som ligger til grunn for didaktiske valg.

Det er verdt å nevne at kapittelet ikke er en systematisk litteraturstudie. Masterstudien tar utgangspunkt i teorien om *computational thinking* (Weintrop et al., 2016; Wing, 2006) og *computational literacy* (diSessa, 2000; Odden et al., 2019), som videre suppleres av andre relevante perspektiver. Det teoretiske fundamentet er ikke heldekkende, hvilket medfører en fare for at konstruktive bidrag til besvarelsen ikke blir tatt i betraktning. Til tross for dette, mener jeg at de teoriene som inkluderes og presenteres, gir et forsvarlig grunnlag for å besvare forskningsspørsmålene.

### 2.1 Hva skal elever lære i fysikk 1?

Den overordnede delen av læreplanen (Kunnskapsdepartementet, 2017) omfatter all grunnutdanning fra barneskole frem til og med videregående skole. Utdanningsdirektoratet anbefaler å bruke den overordnede delen av læreplanen aktivt i undervisningshverdagen til blant annet planlegging av undervisning. Læreplanen er delt inn i «Opplæringens verdigrunnlag», «Prinsipper for læring, utvikling og danning» og «Prinsipper for skolens praksis» (Kunnskapsdepartementet, 2017). Under opplæringens verdigrunnlag skriver Utdanningsdirektoratet blant annet om «Kritisk tenkning og etisk bevissthet», hvorav skolen skal bidra til at elevene utvikler en vitenskapelig og kritisk tenkning i møte med utfordringer og fenomener. For å oppnå ny kunnskap må elevene få argumentere for og imot sine ideer. «Skaperglede, engasjement og utforskertrang» ligger også under opplæringens verdigrunnlag. Undervisningen skal mane frem denne skapergleden og utforskertrangen, da skolen skal legge til rette for at elever får omsette ideer til handling (Kunnskapsdepartementet, 2017). Under

«Prinsipper for læring, utvikling og danning» og videre «Grunnleggende ferdigheter» skriver Utdanningsdirektoratet (2017) at elevene i tillegg til å beherske lesing, skriving, regning og muntlige ferdigheter, skal inneha digitale ferdigheter. Selv om ferdighetene gjelder alle fag vil de vektlegges ulikt, avhengig av hvilket fag som undervises.

Fysikkundervisningen er i likhet med alle andre fag ansvarlig for å realisere verdigrunnet for opplæringen. Læreplanen for fysikk (Utdanningsdirektoratet, 2021) forklarer at elevene skal utvikle en vitenskapelig og kritisk tenkemåte, der påstander skal kunne utfordres og gi rom til undring. Fysikkfagene skal lære elevene om hvordan teknologisk utvikling påvirker både individet og samfunnet, i tillegg til at elevene skal se verdien i samarbeid og idéutveksling. Undervisningen skal også bidra til at elevene får utfolde skaperglede, nysgjerrighet og engasjement gjennom fysikkfagets utforskende og eksperimentelle natur. Digitale ferdigheter er relevant i fysikkfagene, da elevene skal kunne bruke digitale hjelpemiddel til å registrere, analysere, modellere og presentere data. En del av dette er å innhente relevant informasjon for videre å studere fenomener og problemstillinger i fysikken. Nye fenomen og problemstillinger skal ifølge læreplanen utforskes ved bruk av blant annet programmering. I tillegg til at digitale ferdigheter er en grunnleggende ferdighet som fysikkelever skal opparbeide seg, er programmering spesifikt nevnt under «Praksiser og tenkemåter» som et kjerneelement i fysikk (Utdanningsdirektoratet, 2021). I fysikk 1 er det likevel kun ett kompetansemål som eksplisitt nevner numeriske metoder og programmering: «Eleven skal kunne bruke numeriske metoder og programmering til å modellere og utforske bevegelse i situasjoner der akselerasjonen ikke er konstant». I fysikk 2 er dette kompetansemålet utvidet til at «Eleven skal kunne bruke numeriske metoder og programmering til å utforske og modellere fysiske fenomener» (Utdanningsdirektoratet, 2021).

Kritisk tenkning, skaperglede, utforskertrang, digitale ferdigheter og derunder programmering, programmering som praksis, i tillegg til kompetansemålet som konkret nevner numeriske metoder og programmering, ligger alle til grunn i utviklingen av undervisningsopplegget som blir presentert i denne masterstudien. I dette delkapittelet er programmeringens tiltenkte rolle i fysikkundervisningen, samt sentrale kunnskaper elever i fysikkfagene skal tilegne seg gjennomgått, men hva ligger til grunn for dette? De neste delkapitlene vil belyse hvordan programmering kan bidra til å knytte kunnskapen som elevene skal opparbeide seg.



## 2.2 Programmering i undervisningssammenheng

Selv om Utdanningsdirektoratet gir programmeringen en rolle i fysikkfaget og generelt i den norske undervisningen, finner Gjengset (2022) i sin masterstudie ut at hvordan programmering undervises og tolkes, i stor grad varierer mellom enkelte fysikklærere. Gjengsets resultater tilsier at lærerne anser elevenes manglende programmeringskompetanse som en utfordring, i tillegg til lite erfaring i undervisning av programmering som en kompliserende faktor. Så hvordan kan programmering inngå i undervisningen?

I boken Fysikkdidaktikk (Angell et al., 2019) gjennomgås det blant annet hvordan fysikkundervisning kan foregå og hvilke tiltak som kan gjøres for å skape en engasjerende og lærerik undervisning for elevene. Programmering nevnes som et verktøy som er mer utbredt på universitetsnivå, men fortsatt kan gi fysikkundervisningen på videregående skole positive bidrag. Programmering kan brukes som et *utforskende* verktøy i situasjoner der en analytisk tilnærming er mer tungvint, i tillegg til at beregningene kan gi elevene rom for eksperimentering av parametere med letthet. Den utforskende undervisningen gjenkjennes som en elevstyrt læringsform, med fokus på utviklingen av interesse, utforskende kompetanse og en kritisk og spørrende holdning blant elevene (Andersen et al., 2018). Fysikkfagets eksperimentelle og forskningsbaserte natur, gjør muligheten for utforskende undervisning mer relevant enn i enkelte andre fag. Dette vises igjen i læreplanen i fysikk, der ordet «utforske» er nevnt i fem av kompetansemålene (Utdanningsdirektoratet, 2021). Bruken av programmering i fysikkundervisning kan også begrunnes som en sentral ferdighet på lik linje med bruk av andre måle- og beregningsverktøy. Programmering kan gi innsikt i hvordan forskning i fysikk foregår, og kan i tillegg gi elevene en økt motivasjon og mestring, da andre ferdigheter kreves i tillegg til tradisjonelle fysikkfaglige ferdigheter (Angell et al., 2019, s. 256-257).

I masterarbeidet til Waters (2020) undersøker han hvordan programmering kan bidra til dybdelæring i fysikkfaget. Gjennom et selvutviklet undervisningsopplegg som benytter programmeringsspråket Python konkluderer han med at programmering gav elevene mulighet til å se sammenhenger mellom ulike størrelse i formler, og hvordan de påvirket bevegelse. Læringsutbytte ble følgende både programmeringsteknisk og fysikkfaglig i form av konsepter og innsikt i hvordan forskning foregår. Waters argumenterer også for at om elevene skal få et dypere læringsutbytte i fysikkfaget, må det programmeringstekniske ferdighetsnivået reduseres nok til at elevene kan fokusere på fysikken.

Allerede før innføringen av LK20 utviklet Nordby (2019) i sin masterstudie tre undervisningsopplegg for videregående skole, i tillegg til en litteraturstudie som forsøkte å forutsi ulike utfordringer og muligheter med programmering i fysikkundervisningen. Nordby konkluderer med at programmering kan bidra til fysikkfaget med numeriske beregninger, simulering, visualisering og produktutvikling. Nordby sine resultater tilsier at det programmeringstekniske ferdighetsnivået må tilpasses elevene, men at de fort tar til seg kunnskap om programmering og kan utforske videre med mer kompliserte programmeringstekniske beregninger. Nordbys undervisningsopplegg omhandler Euler-metoden for numerisk integrasjon i et inhomogent felt og utviklingen av en «Fotonmator» som lar en bruker taste inn en bølgelengde for å avgjøre fargen og fotonenergien som bølgelengden representerer. Masterstudien viser blant annet at programmering kan brukes i flere fysikkfaglige tema på videregående nivå enn det kompetansemålene konkret nevner i den nye læreplanen for fysikk 1 (Utdanningsdirektoratet, 2021).

Læreboken elevene i den aktuelle klassen i denne masterstudien bruker, inkluderer programmering med programmeringsspråket Python (Fossum et al., 2021). Boken har underveis, som en integrert del av kapitlene, oppgaver som eksemplifiserer en nyttig bruk av programmering. Denne integreringen av programmering står i stil med Odden et al. (2019) sine argumenter for å inkludere digitale beregningsverktøy som en del av fysikkundervisningen, i motsetning til å undervise programmering som en egen vitenskap med kunnskaper og ferdigheter tilsidesatt den tradisjonelle fysikken.

Sørby og Angell (2012) gjør i sin studie en analyse av arbeidet med programmering i fysikk blant universitetsstudenter. Av studien fremkommer det at elevene arbeider i såkalte *moduser*, da henholdsvis i *matematikkmodus*, *fysikkmodus* og *programmeringsmodus*. Matematikkmodus gjenkjennes ved at elevene diskuterer matematiske sammenhenger mellom for eksempel formler og variabler. Fysikkmodus gjenkjennes ved at elevene bruker enten fysikkfaglige konsepter eller hverdagsforestillinger i sitt arbeid, og programmeringsmodus gjenkjennes ved at elevene arbeider med syntaks (programmeringsspråkets oppbygning og logikk) tilknyttet koden. De tre modusene ble tatt i betraktning i utviklingen av intervjuguiden for denne studien (se Vedlegg E), da Sørby og Angell (2012) ser funnene i lys av sosiokulturell læringsteori.

## 2.3 Algoritmisk tenkning

Seymour Papert introduserte i 1980 begrepet *computational thinking*. Papert brukte programmering i undervisningssammenhenger og ønsket å mane frem et spesifikt tankesett hos barn ved å la dem interagere med digitale verktøy (1980). Jeanette M. Wing oppsummerte i 2006 hva begrepet omfatter. Wing mente at *computational thinking* er ment å være relevant for alle og at barn også burde opparbeide seg de evnene som *computational thinking* innebærer. Problemløsning, utarbeide og designe systemer og forstå menneskelig atferd gjennom å bruke grunnleggende konsepter fra den digitale vitenskapen, er egenskaper begrepet rommer (Wing, 2006). På norsk oversettes *computational thinking* til *algoritmisk tenkning*, til tross for at det norske begrepet er nærere språklig beslektet til *algorithmic thinking*, et engelsk begrep som er smalere og kan regnes inn under *computational thinking* (Doleck et al., 2017; Labusch et al., 2019). Definisjonen av den norske oversettelsen algoritmisk tenkning (*computational thinking*) blir utbrodert og konkretisert av Weintrop et al. (2016).

I 2016 kom Weintrop et al. med en taksonomi for algoritmisk tenking. Flere definisjoner på begrepet «*computational thinking*» (les algoritmisk tenking) hadde blitt presentert tidligere, men Weintrop et al. rettet sin definisjon mer mot det amerikanske læreverket og high school-klasserommet som stadig økte sitt fokus og læring mot den digitale vitenskapen. Taksonomien som ble utviklet tar utgangspunkt i de viktigste egenskapene og karakteristikkene som trengs for å stimulere elevenes behov, men også for å reflektere den matematiske og vitenskapelige industriens innhold og ferdigheter. Gjennom flere metodiske steg kommer Weintrop et al. frem til en taksonomi som representerer hva begrepet algoritmisk tenking innebærer i undervisningssammenheng (se Tabell 1).

| <b>Datahåndtering</b> | <b>Modellering og simulering</b>                       | <b>Digital problemløsning</b>               | <b>Systemtenkning</b>                       |
|-----------------------|--|---|---|
| Samle data            | Bruke digitale modeller for å forstå et begrep         | Forberede problemer til å løse dem digitalt | Undersøke et komplekst system som en helhet |
| Generere data         | Bruke digitale modeller til å finne og teste løsninger | Programmere                                 | Forstå sammenhenger innad i et system       |
| Behandle data         | Vurdere digitale modeller                              | Velge effektive beregningsverktøy           | Tenke i nivåer                              |
| Analysere data        | Utforme digitale modeller                              | Vurdere effektive beregningsverktøy         | Kommunisere informasjon om et system        |
| Visualisere data      | Implementere digitale modeller                         | Utvikle digitale løsningsmodeller           | Definere systemer og håndtere kompleksitet  |
|                       |  | Utvikle digitale abstraksjoner              |   |
|                       |  | Feilsøking og feilretting                   |   |

Tabell 1: Figuren er en oversikt over Weintrop et al. (2016, s. 135) sin taksonomi for «computational thinking». Oversatt av Nordby (2019, s. 28) og videreutviklet av meg.

Kategoriene er i tabellen listet øverst og inneholder **datahåndtering**, **modellering og simulering**, **digital problemløsning** og **systemtenkning**. **Datahåndtering** handler i stor grad om det første steget i en programmeringsprosess. I tilfeller der man gjør empiriske forsøk, er det viktig at man finner en måte å *samle dataen* på som er hensiktsmessig. I relative og teoretiske forsøk må man i større grad produsere og/eller *generere data*. Dette kan for eksempel være under en simulering av et tenkt tilfelle, der forsøket faktisk ikke har blitt gjort og dataen dermed ikke eksisterer. Når dataen er samlet inn, er det neste steget å kunne *behandle data*. I mange tilfeller der man samler inn data, kan man få variabler av ulik relevans for oppgavens formål. Her må man kunne gjøre avgjørelser på hva som er relevant å bruke osv. Et av datamaskinens største fortrinn er evnen til å hente ny informasjon gjennom *analyse av data*. I analysen av data gjelder det å sette den behandlede dataen opp mot hverandre på en hensiktsmessig og informativ måte. En måte å kunne avgjøre om informasjon er nyttig og informativ på, er ved å *visualisere data*. Ved å velge riktig visualiseringsverktøy vil både behandling og analyse av data bli mer intuitiv. I tillegg er visualisering av data et viktig hjelpemiddel når det kommer til å kommunisere resultatene sine (Weintrop et al., 2016, s. 135-136).

Under kategorien **modellering og simulering** presenteres det fem praksiser. Overordnet skriver Weintrop et al. at en modell er en forenklet fremstilling av virkeligheten

som fremhever enkelte egenskaper ved et fenomen, samtidig som den neglisjerer andre (2016, s. 137). Ved å *bruke digitale modeller for å forstå et begrep*, menes det at denne digitale tilnærmingen til virkeligheten og egenskapene den fremhever er med på å øke forståelsen til elevene om fenomenet eller konseptet modellen undersøker. Hvis elevene kan å *bruke digitale modeller til å finne og teste løsninger*, kan de bekrefte eller avkreft sitt svar. Man kan for eksempel finne en løsning gjennom en analytisk tilnærming, for så å lage en digital numerisk modell for å finne ut om den samsvarer med den analytiske løsningen. En viktig egenskap når man jobber med modeller er å kunne *vurdere* de. Her gjelder det å ha kunnskap om styrker og svakheter i modellen, for så å vurdere om modellen representerer virkeligheten på tiltenkt måte. Eksempelvis kan man i en kompleks modell ha neglisjert en egenskap av virkeligheten som ville ha påvirket modellen. Det å kunne stille seg kritisk til sin egen eller andres modell, er en egenskap Weintrop et al. mener er en viktig egenskap i modellering- og simuleringsarbeid (2016, s. 137). Når man *utformer digitale modeller* må man ta metodiske, teknologiske og konseptualistiske valg. Dette kan knyttes opp mot Sørby og Angells tanker om nødvendigheten av en plan for å kunne lage en effektiv kode (2012, s. 290-291). Når man *implementerer digitale modeller* bør man ta i betraktning at modellen skal kunne leses og oppfattes av en datamaskin.

**Digital problemløsning** er kategorien med flest praksiser. I mange tilfeller er det først i denne kategorien at selve programmeringen starter. Når man *forbereder problemer til å løse dem digitalt* er det viktig at modellen er utviklet på en måte som gjør det mulig for datamaskinen å kunne bearbeide modellen. Den utviklede modellen kan ha som formål å løse et større problem. Dette problemet må i mange tilfeller deles opp i flere mindre problemstillinger, slik at en datamaskin skal kunne løse de. Når man således *programmerer* må man kunne gi datamaskinen direkte instruksjoner. I denne praksisen handler det ikke bare om å kunne programmere, men også det å forstå allerede eksisterende kode slik at man for eksempel kan videreutvikle den. Ved å *velge effektive beregningsverktøy* vil programmet som blir utviklet kunne gjøre den jobben det skal gjøre på mest mulig effektiv måte. Dette krever dermed en evne til å *vurdere effektive beregningsverktøy*. Disse beregningsverktøyene kan for eksempel være importerte pakker til et programmeringsspråk. Det å ha kunnskap til ulike beregningsverktøy og bruken av dem, vil hjelpe elevene til å nettopp velge korrekt og effektive verktøy. Når man da *utvikler digitale løsningsmodeller* er det en fordel at programmet blir skrevet på en måte som gjør det mulig å bruke disse på nytt. Hvis man har utviklet modellen sin effektivt og med hensyn til at den skal løses digitalt, vil programmet

bestå av mange små deler som til sammen kan forklare hele modellen hensiktsmessig. Det er dermed en fordel at de tidligere praksisene og kategoriene har blitt gjennomført på en tilfredsstillende måte. Praksisen å *utvikle digitale abstraksjoner* handler om å kunne nytte digitale verktøy til å presentere en idé ved å fremheve noen egenskaper, samtidig som man skyver andre i bakgrunnen. Abstraksjonen som blir presentert vil kunne knytte forskjellige fenomener opp mot hverandre på måter som kanskje ikke er overfladisk sammenlignbare. I arbeidet med digitale modeller møter man ofte på feil. *Feilsøking og feilretting* er dermed en stor del av å nettopp å utvikle en digital modell. Ved å ha kunnskap til programmering, digitale beregningsverktøy og koden som helhet, vil en feilmelding kunne gi verdifull informasjon (Weintrop et al., 2016, s. 138-140).

**Systemtenkning** er den siste kategorien Weintrop et al. (2016) presenterer. Systemtenkning handler om å se på et fenomen som et konsept bestående av flere faktorer og variabler som stadig påvirker hverandre. Ved å *undersøke et komplekst system som en helhet* tar man et steg tilbake. Man tar ikke for seg hver lille bestanddel og deres rolle, men ser heller på hva som er den totale input og output. Noen ganger er det best å se på systemet som helhet, mens andre ganger må man *forstå sammenhenger innad i et system*. Digitale verktøy er ofte særs egnet til formålet om å forstå sammenhenger i et system, da datamaskinen ofte krever innsikt i hver enkelt bestanddel i et system og sammenhengen mellom dem. I algoritmisk tenkning og videre systemtenkning er det viktig å kunne *tenke i nivåer*. Dette innebærer å kunne vite hvordan hver mikro-bestanddel bidrar til forskjellige makro-bestanddeler som utgjør hele systemet. I tillegg til å visualisere data under kategorien «datahåndtering», innebærer det å *kommunisere informasjon om et system* også ofte om visualisering. Der man under datahåndtering visualiserer mye for å kunne ta beslutninger om videre analyser og modelleringer, er det å visualisere data under systemtenkning en form for å kommunisere det som har blitt lært og funnet ut av. Det er ikke bare visualisering som kommuniserer dette, men presise og gode forklaringer er også med på å kommunisere systemer. Til slutt er det å kunne *definere systemer og håndtere kompleksitet* en praksis. I virkeligheten kan alle fysiske fenomen bli sett på som et system. Da gjelder det til å kunne definere fornuftige systemer med bestanddeler som faktisk tjener sin hensikt (Weintrop et al., 2016, s. 140-142).

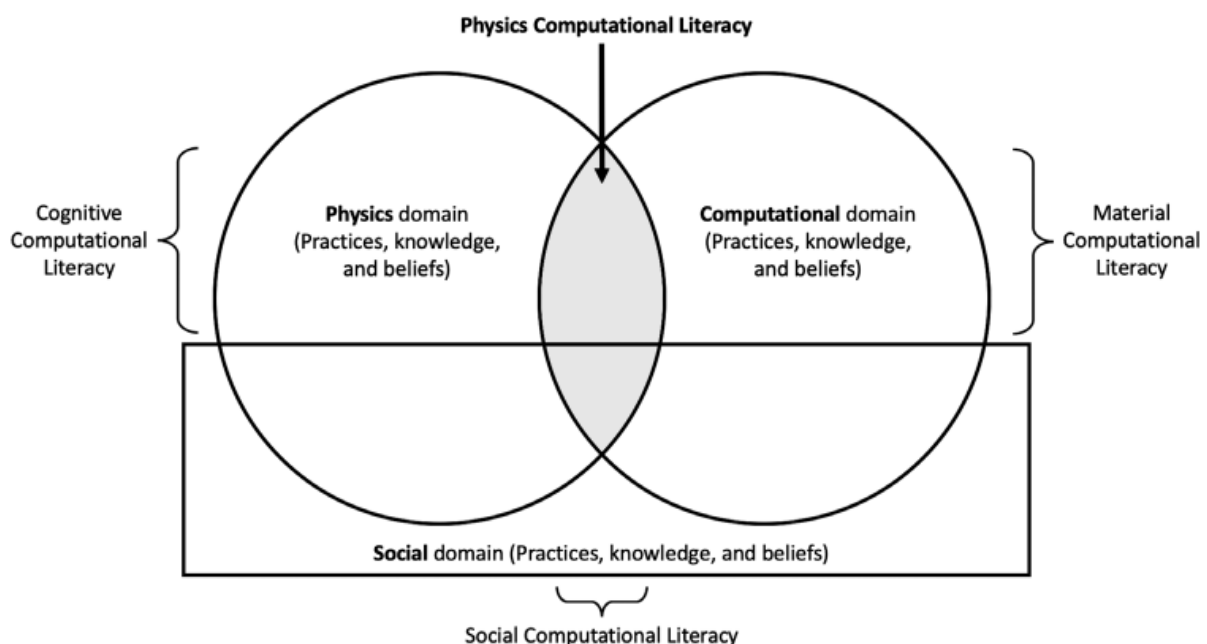
Weintrop et al. sin taksonomi er sentral i denne studien da den ble brukt som verktøy for utviklingen av undervisningopplegget (se kapittel 3.3). Det er likevel begrensninger med taksonomien, da den i liten grad tar høyde for andre digitale beregningsverktøy utenom programmering, kognitive læringsprosesser og didaktiske hensyn. Weintrop et al. skriver selv

at en svakhet med taksonomien er at det ikke er selvforklarende hvordan praksisene i taksonomien konkret kan brukes i undervisningssammenheng (2016, s. 128). Denne studien gir i kapittel 3.3 et eksempel på hvordan bruken av taksonomien konkret kan manifesteres i et undervisningsopplegg. Wings begrep «computational thinking» får også kritikk av diSessa (2018) for å være for overfladisk og ikke vise hvordan begrepet kan brukes til å tilegne seg andre kunnskaper enn bare datavitenskap. Det neste delkapittelet tar for seg et bredere begrep som kan nyttes generelt i undervisningssammenhenger hvor digitale beregninger foretas, og som i tillegg tar i betraktning perspektiver på læring og hvordan kunnskap tilegnes.

## 2.4 Digital beregningskompetanse

I boken “Changing Minds: Computers, Learning, and Literacy” introduserer Andrea A. diSessa (2000) et rammeverk for *digital beregningskompetanse (computational literacy)*, grunnet samfunnets og dermed klasserommets omfattende digitalisering. Begrepet digital beregningskompetanse innebærer ifølge diSessa mer enn bare å kunne skru på en datamaskin, bruke tastatur og datamus til å skrive noen enkle linjer. Digitale beregningsverktøy (f.eks. datamaskiner) krever ferdigheter innen skriftspråket (koder, siffer osv.), innsikt og kunnskap om hvordan skriftspråket fungerer (syntaks) og kunnskap om kodens formål (programutvikling, kalkulasjoner osv.). Digital beregningskompetanse som begrep utvikler seg proporsjonalt med den digitale vitenskapen, og grunnet det allmenne samfunnets manglende kompetanse innenfor digitale beregningsferdigheter, argumenterer diSessa for at det er behov for en ny innføring av begrepet (2000, s. 1-5). diSessa definerer tre pilarer som begrepet digital beregningskompetanse krinser rundt; den *materielle*-, den *kognitive*- og den *sosiale* pilaren. Den materielle pilaren består av konkrete og visuelle tegn, symboler, skildringer og representasjoner. Dette rommer kunnskap om syntaks og enkle digitale operasjoner (løkker, «if»-betingelser osv.). Disse tegnene og symbolene som står skrevet, har ingen betydning for en som ikke har kjennskap til språket. Det er leserens evne til tolkning og oppfatning som gir den materielle pilaren mening og verdi. De egenskaper som kreves av et individuelt menneske for å hente eller tilføre kunnskap gjennom bruken av digitale beregningsverktøy, definerer diSessa som egenskaper innunder den kognitive pilaren. diSessa argumenterer for at ny materiell kunnskap ikke kan oppstå fra individet alene. I en eller annen form interagerer man enten med personer direkte, eller baserer seg på tidligere kunnskap fra andre. Denne nødvendige kommunikasjonen for å kognitivt utvikle kunnskap gjennom den materielle pilaren, plasserer diSessa i den sosiale pilaren (2000, s. 6-11). Til sammen er pilarene alt som rommer digital beregningskompetanse.

diSessas definisjon av de tre pilarene er utviklet for hele det digitale samfunnet. Det er med andre ord flere disipliner hvor man benytter seg av digital beregningskompetanse. I 2019 kom Tor Ole Odden et al. med en videreutvikling og spesifisering av rammeverket rundt digitale beregningskompetanse i henhold til fysikk. I artikkelen blir det argumentert for at siden diSessas omfattende teori ble introdusert i 2000, har utviklingens omfang av digital vitenskap, både i form av programmerings bruk i fysikken, men også som hjelpemiddel for læring i form av simulasjoner og visualisering, økt i så stor grad at en egen teori innen fysikken var nødvendig. Digital beregningskompetanse skiller seg fra begrepet og taksonomien for algoritmisk tenkning i den grad at digital beregningskompetanse omfatter mer enn det materielle, samtidig som at digital beregningskompetanse ikke bare omhandler programmering (Odden et al., 2019, s. 18). En annen grunn til at Odden et al. ønsket en spesifisering av begrepet, var at digital beregning ikke ble tilstrekkelig anvendt i fysikkundervisningssammenheng på universitetsnivå. Videreutviklingen av rammeverket til diSessa skulle dermed også legge til rette for mer aktiv bruk av digitale beregningsverktøy hos fysikkstudenter. Teorien ble kalt *physics computational literacy* (Odden et al., 2019, s. 1), og vil videre bli referert til som *digital beregningskompetanse i fysikk*.

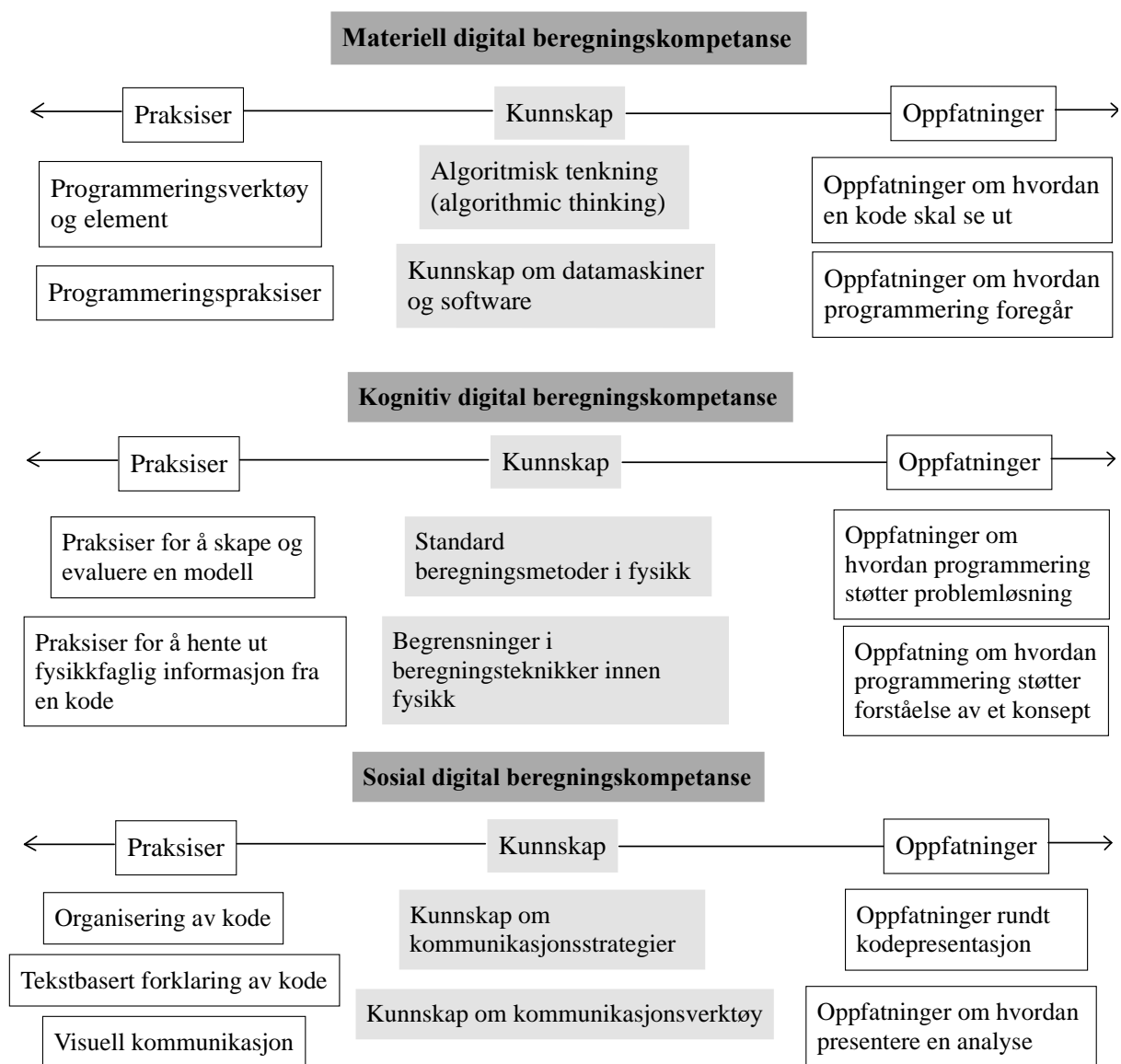


Figur 2: Teoretisk plassering av digital beregningskompetanse i fysikk (Odden et al., 2019, s. 3)

Den digitale beregningskompetansen i fysikk blir definert til å gjelde når digitale beregningsverktøy blir brukt i fysikken, både i og utenfor en sosial setting (se Figur 2 ). Ut fra



resultatene fra en innleveringsoppgave gitt til fysikkstudenter på sitt 3. semester på universitetet i Oslo, er diSessas teori videreutviklet. Elementer tilhørende hver pilar er gruppert i tre grupper (se Tabell 2). De tre gruppene rangert fra minst komplekst til mest komplekst er henholdsvis praksiser (practices), kunnskap (knowledge) og oppfatninger (beliefs). Praksiser rommer de empirisk observerbare aktivitetene studentene foretok seg, fra skriving av konkret kode til større observerbare mønster som feilsøking osv. Innunder kunnskap ligger studentenes kunnskaper og tanker om konkrete konsept eller idéer. Den siste gruppen er oppfatninger, der elevens generelle oppfatninger om digital beregning inngår. Eksempelvis vil store tanker om programmeringens natur og digital beregning i fysikk inngå i oppfatninger.



Tabell 2: Odden et al. sitt utvidede rammeverk for de tre pilarer (2019, s. 8). Egen oversettelse.

Bruken av denne utvidede modellen kan ifølge Odden et al. (2019, s. 17-19) rettferdiggjøre bruken av digitale beregningsverktøy i fysikkundervisning, da mangfoldet av egenskaper og potensialet er såpass bredt. Man kan også bruke modellen som et analyseverktøy til å for eksempel belyse deler av undervisning som kanskje i større grad trengs å fokuseres på, eller til å se utfordringer og problem i utviklingen og gjennomføringen av undervisning innen digital beregningskompetanse i fysikk.

## 2.5 Læreplanverkets tolkning av den algoritmiske tenkeren

Utdanningsdirektoratet har en egen definisjon om hva algoritmisk tenkning innebærer (Figur 3). Den originale modellen er hentet fra Barefoot Programming UK, og oversatt fra engelsk til norsk (Senter for IKT i utdanningen, 2016, s. 12-15). Den algoritmiske tenkeren er også i likhet med min oversettelse, oversatt fra *computational thinking*. Den algoritmiske tenkeren skal kunne dekomponere komplekse problemer til mindre delproblemet ved å analysere og organisere informasjon for å fremlegge konstruktive måter å bearbeide denne informasjonen. Den algoritmiske tenkeren kan også modellere abstrakte systemer ved å isolere mindre relevante egenskaper ved systemet, men også føre konkrete modeller til abstrakte systemer. Disse egenskapene skal utvikles gjennom å arbeide systematisk og analytisk, men også gjennom eksperimenterende og utforskende arbeidsmetoder. Ved å arbeide slik vil man også møte på feil og utfordringer. Den algoritmiske tenkeren gjenkjenner utfordringene som en del av prosessen og innehar standhaftigheten til å arbeide seg forbi disse utfordringene. Til slutt er den algoritmiske tenkeren en lagspiller som bruker og hjelper andre algoritmiske tenkere for å komme frem til gode løsninger. Den algoritmiske tenkningen blir sett på som en problemløsningsstrategi som ikke bare er relevant for elever på videregående skole, men i alle ledd av utdanningen på tvers av fag (Utdanningsdirektoratet, 2019).



Figur 3: Illustrasjon av den algoritmiske tenkeren (Utdanningsdirektoratet, 2019).

Man ser at koblingen til taksonomien for algoritmisk tenkning er sterk, da mange av kategoriene og praksisene fra taksonomien er nevnt, men hovedsakelig innen digital problemløsning (Weintrop et al., 2016). Koblingen mellom taksonomien og Utdanningsdirektoratets algoritmiske tenkning bekreftes av Vinnervik & Bungum (2022, s. 392-393), som skriver at mange av praksisene i taksonomien for algoritmisk tenkning finnes flere steder i den nye læreplanen. Studien til Vinnervik & Bungum er begrenset til grunnskoleopplæring (1. til 10. trinn), men statuerer relevansen for taksonomien til algoritmisk tenkning i det norske skolesystemet. Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019) beveger seg bort fra taksonomien til Weintrop et al., da den vektlegger arbeidsmetoder som å fikle, samarbeide og holde ut. Selv om de nevnte arbeidsmetodene implisitt kan inngå i taksonomien, nevnes ikke dette eksplisitt. Utdanningsdirektoratet fremhever også arbeidsmetodene som kreative og eksperimentelle, der samarbeidet er en viktig faktor for å kunne komme frem til nye gode løsninger. Dette samsvarer i større grad med *digital beregningskompetanse*, der den sosiale pilaren er grunnleggende for tilegning av ny kognitiv kunnskap (diSessa, 2000; Odden et al., 2019). Den algoritmiske tenkeren er som tidligere skrevet beslektet til det engelske begrepet «computational thinking». En generell kritikk av det engelske begrepet er at det er for snevert og begrenser seg i stor grad til programmeringsrelaterte situasjoner, samtidig er for abstrakt og vanskelig å benytte i undervisningssammenheng (diSessa, 2018; Odden et al., 2019). Utdanningsdirektoratet bruker dette begrepet og lar det gjelde for alle fag på alle nivå. Man

kan argumentere for at Utdanningsdirektoratet tar et begrep som i sin natur er fôr spesifisert til programmering, fjerner programmering og da står igjen med et begrep som er for snevert og ikke omhandler sitt egentlige felt. Odden et al. (2019) skriver at det vi på norsk kaller algoritmisk tenkning, kan innlemmes i begrepet digital beregningskompetanse, som gir en mer konkret og helhetlig oversikt av hvordan egenskaper arbeid med digitale beregningsverktøy kan brukes i undervisningssammenheng.

## 2.6 Nasjonal implementering av LK20

Overordnet del av læreplanen ble gjeldende fra 2020, der de fagspesifikke læreplanene skulle implementeres gradvis over en treårsperiode (Utdanningsdirektoratet, 2023), men grunnet pandemi varierte implementeringen fra skole til skole. Veiledning fra øvrig hold varierte, og det ble i stor grad opp til hver enkelt skole å bestemme hvordan den nye læreplanen skulle implementeres (Høvik, 2022). Elevene som skoleåret 2022/23 tar fysikk 1, ble først introdusert til den nye læreplanen skoleåret 2021/2022, i henhold til plan for innføring av den nye læreplanen (Utdanningsdirektoratet, 2023). Det vil si at siden elevene gikk i tiendeklasse skoleåret 2020/2021, fullførte ungdomsskolen med den gamle læreplanen (LK06), for så å begynne på videregående med den nye læreplanen. Det kan derfor tenkes at elevgruppen som ikke har fått grunnskoleopplæringen som LK20 legger opp til, men følger den kun i de siste tre årene av sitt utdanningsløp, er elever midt imellom to læreplaner. I tillegg har det vist seg at implementeringen av læreplanen, også i fysikk, i stor grad er avhengig av de enkelte lærernes tolkning og prioriteringer (Gjengset, 2022). Det er da veldig vanskelig å kartlegge elevene i forkant av undervisningsopplegget uten inngående samtaler med faglærere, da samfunnet ifølge DiSessa (2018) er sterkt preget av datavitenskapen og bruken av digital beregningskompetanse, mens den gamle læreplanen (Utdanningsdirektoratet, 2006) ikke legger tilstrekkelig opp til bruken av digitale hjelpemiddel i dens helhet. Noen elever kan dermed ha lært seg mye programmering på egenhånd, mens andre nesten ikke har rørt eller sett det.

I kapittel 3.2 vil det presenteres ytterligere teoretiske betraktninger og avveininger tilknyttet det konkrete undervisningsopplegget i denne masterstudien. De aktuelle teoretiske perspektivene er sterkt knyttet til de didaktiske valgene som ble gjort i utviklingen av undervisningsopplegget. Det er derfor mest hensiktsmessig å presentere den teorien i kapitlet om didaktiske valg, og ikke i teorikapitlet som har som formål å belyse forskningsspørsmål 1 (se Figur 1).

### 3 Casebeskrivelse

I dette kapittelet beskrives casen for denne studien. Teori bidrar til å belyse de didaktiske, faglige og praktiske valgene som ble tatt i forbindelse med utviklingen av undervisningsopplegget i casen. I tillegg bidrar erfaringer fra et pilotprosjekt som ble gjennomført i forkant av dette masterprosjektet med verdifull innsikt i utforming av undervisningsopplegget: Jeg fikk mulighet til å gjennomføre et pilotprosjekt for en gruppe studenter på forkurs for ingeniør- og sivilingeniørutdanning. Studentene gjennomførte et undervisningsopplegg i Jupyter Notebook med samme format, programmeringsspråk og veiledning som denne masterstudien, etterfulgt av en spørreundersøkelse og noen individuelle intervju. Denne gjennomkjøringen gav verdifull informasjon som for eksempel hvor mye veiledning eventuelle oppgavetekster skulle ha, hva studenter uten særlig programmerings erfaringer opplevde som utfordrende og hvordan de arbeidet for å komme seg videre.

Undervisningsopplegget for denne masterstudien ble utviklet og gjennomført for en elevgruppe på opp til 30 elever. Alle elevene studerte studiespesialiserende på en videregående skole med rundt 1000 elever, som etter norsk målestokk ligger i en stor by. Blant de to fysikklassene med totalt 30 elever, varierte programmerings erfaringene og fysikkferdighetene i stor grad. Det var omtrent halvparten av elevene som hadde valgfaget Teknologi og forskningslære, der de hadde programmert litt, men i et annet språk enn Python. Klassen hadde litt programmerings erfaring i Python fra matematikkfagene, men lite i fysikkundervisningen, og den programmeringen de hadde forholdt seg til hadde stort sett handlet om å lese kode og forstå hvordan kodene fungerer.

I desember 2022 møtte jeg faglærerne i fysikk 1 på den aktuelle skolen, Der fikk jeg informasjon om elevgruppen, tidligere undervisning og råd om hvordan undervisningsopplegget burde planlegges.. Fra møtet kom det frem at systemer med luftmotstand var det mest hensiktsmessige temaet for bruk av programmering. I fysikk 1 er det kun et enkelt kompetansemål som nevner programmering eksplisitt, hvilket omhandler å bruke programmering til å utforske systemer der akselerasjonen ikke er konstant (Utdanningsdirektoratet, 2021). Det ble også bestemt at de to fysikklassene skulle samles til en tretimersøkt. Dette ble gjort av praktiske hensyn, da organisering av timer og progresjon i

faget gjorde det lettest å sette av en felles tid. Samtlige elever hadde lastet ned programvaren Anaconda Navigator, der man videre kan benytte plattformen Jupyter Notebook. Elevene hadde ikke programmert i Jupyter Notebook tidligere, så en introduksjon av plattformen var nødvendig.

### 3.1 Generelt om undervisningsopplegget

Oppgaven elevene forsøkte å løse kan finnes i sin helhet i Vedlegg A (elevversjon) og Vedlegg B (løsningsforslag). Python ble valgt som programmeringsspråk siden eksempelkoder gitt i elevenes lærebok, Kraft 1, er skrevet i Python (Fossum et al., 2021). Python er også programmeringsspråket som i større og større grad blir brukt på universitetsnivå (Piatetsky, 2017). Det vil si at når elevene lærer Python, lærer de seg et språk de kan få bruk for senere som for eksempel fysikk- eller ingeniørstudenter. I tillegg er Python å regne som et intuitivt programmeringsspråk, med et stort nytteområde både i objekt- og prosedyreorientert programmering (Grandell et al., 2006). Videre ble Jupyter Notebook valgt som plattformen elevene skulle programmere i. Grunnen til dette er at Jupyter Notebook kombinerer celler med vanlig tekst og programmeringsceller på en hensiktsmessig måte. Det gjør at når elevene utfører oppgavene, så slipper de å bytte mellom faner, men kan gjøre oppgavene i samme program som de leser oppgaveteksten.

Undervisningsøkten skulle som tidligere nevnt vare i tre skoletimer. Siden alle elevene skulle få tilbudet om å svare på spørreundersøkelsen, ble de siste 20 minuttene satt av til besvarelse av denne. Undervisningsopplegget tok utgangspunkt i kompetansemålet om at elevene skal kunne «bruke numeriske metoder og programmering til å modellere og utforske bevegelse i situasjoner der akselerasjonen ikke er konstant» (Utdanningsdirektoratet, 2021). Etter råd fra faglærerne ble kompetansemålet nyttet i en situasjon der en person skulle utføre et fallskjermhopp med luftmotstand. I denne situasjonen er det naturlig å bruke numeriske metoder for problemløsning, da en analytisk tilnærming vil være veldig kompleks og utfordrende (Angell et al., 2019, s. 196), i tillegg til at bevegelse med luftmotstand er nevnt i læreverket skolen benyttet seg av (Fossum et al., 2021).

Ut fra problemstillingen til masteren og fra samtaler med faglærerne til elevgruppen ble formålet å utvikle et undervisningsopplegg der alle elevene fikk prøve å skrive egen kode for å løse problemer i fysikken. Hver elev fikk laste ned en Jupyter Notebook-fil der oppgaveteksten med veiledning var gitt, og hver elev skulle gjøre disse oppgavene på sine egne datamaskiner. Siden det var jeg som utviklet opplegget, ønsket jeg selv å være

klasseleder for opplegget. Lærerne syntes det hørtes ut som en god idé, men de ville også være til stede av interesse for hvordan opplegget gikk. Deres rolle var da å være hjelpelærere som kunne bistå elevene med veiledning.

Oppgaven (se Vedlegg A) starter med en introduksjon som forteller at personen Tore har planer om å hoppe i fallskjerm fra et fly. Tore trenger hjelp til å gjøre noen beregninger slik at han er forberedt til sitt fallskjermhopp. Elevene får oppgitt noen relevante konstanter og blir bedt om å regne ut tiden fallskjermhoppet tar og hvilken fart Tore vil ha når han lander på bakken om han hopper uten fallskjerm og om man ser bort fra luftmotstand. Den påfølgende oppgaven ber elevene om å finne frem til terminalfarten Tore vil oppnå om han ikke bruker fallskjerm, men om man tar hensyn til luftmotstand med en gitt luftmotstandskoeffisient. Etter dette skal elevene regne ut hvordan posisjonen, farten og akselerasjonen utvikler seg gjennom hele hoppet uten fallskjerm. Fra oppgave 4 kobler man inn fallskjermen i undervisningsopplegget, der elevene ved hjelp av grafene de har plottet i oppgave 3b og terminalfarten med fallskjerm, skal avgjøre på hvilket tidspunkt Tore skal løse ut sin fallskjerm. Avslutningsvis blir elevene bedt om å finne ut hvordan posisjonen, farten og akselerasjonen utvikler seg gjennom helle fallskjermhoppet, da først uten fallskjerm, så med fallskjerm etter et selvbestemt tidspunkt.

## 3.2 Didaktiske valg

I utviklingen av et undervisningsopplegg er det mange valg som må tas. Enkelte er praktiske, andre er didaktiske og noen er personlige preferanser. Valgene kan påvirke oppgavens innhold og oppsett, i tillegg til arbeidsøktens flyt og gang. De valgene som bevisst ble tatt i utviklingen av opplegget, vil her presenteres og teoretisk forankres.

Problemstillingen som elevene skulle arbeide rundt, handlet som nevnt om et fallskjermhopp. Denne situasjonen passet godt til kompetansemålet (Utdanningsdirektoratet, 2021). Realistiske situasjoner kan motivere elevene i større grad enn abstrakte og teoretiske tilfeller (Brown & Wilson, 2018, s. 4-5). Oppgaveteksten inneholdt bl.a. følelser knyttet til et fallskjermhopp, ikke bare relevant informasjon, nettopp for at elevene lettere skal kunne få en nærmere tilknytning til situasjonen (se Figur 4).

## Tores fallskjermhopp

Tore Mannlig Aberge skal hoppe i fallskjerm. Han er fryktelig nervøs, og for å forberede seg best mulig tenker han at han må gjøre noen beregninger. Han åpner et pythonprogram, men sliter med programmeringen. Her kommer du inn for å hjelpe han!

Tore skal følge et fly opp til  $h_0 = 1565$  moh. Der skal han slippe seg ut med startfart  $v_0 = 0$  m/s. Han vil i første omgang simulere noen situasjoner uten fallskjerm. Du kan regne med tyngdeakselerasjonen  $g = 9.81$  m/s<sup>2</sup> i alle situasjoner.

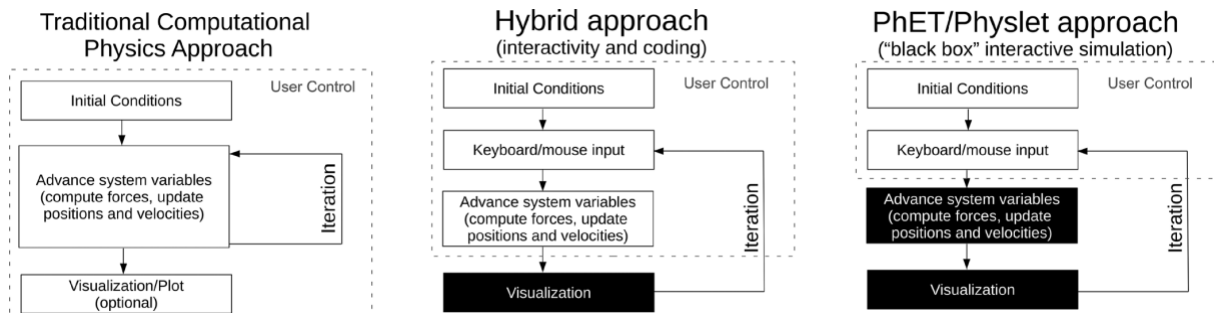
Figur 4: Introduksjonsteksten til undervisningsopplegget.

Siden elevene i den aktuelle klassen hadde varierende erfaring med å programmere og ingen hadde programmert mye i fysikkundervisningen, bestemte jeg at oppgavene burde kreve lite programmeringskunnskaper i starten, for så å bli mer og mer krevende. Yaşar (2013) viser at et for enkelt opplegg kan føre til kjedsomhet blant elevene, noe som viser at for enkle oppgaver for lenge, kan medføre at enkelte elever ikke blir motiverte til å fullføre oppgavene. Opplegget ble inspirert av *LIST-oppgavene* Wæge og Nosrati gjennomgår i *Motivasjon i matematikk* (2018, s. 83-88). Selv om LIST (lav inngangsterskel, stor takhøyde) er utviklet for enkelte oppgaver, ikke hele opplegg, er idéen om at det skal være lett å komme i gang, men ikke begrense elevene, fortsatt overførbart til større opplegg (Sengupta et al., 2013). Undervisningsopplegget kunne blitt gitt som en stor problemstilling, men med elevenes bakgrunnskunnskaper tatt i betraktning ble problemstillingen delt opp i flere små oppgaver. Å dele opp i små oppgaver kan gå på bekostning av enkelte kategorier i taksonomien for algoritmisk tenkning (Weintrop et al., 2016), men kan også sørge for at elever med lite programmerings erfaring får en kognitiv avlastning (Brown & Wilson, 2018). I tillegg bør læreren kunne mestre beregningsverktøyet for å lettere hjelpe elevene (Sengupta et al., 2013, s. 362). Ved at oppgaveteksten deler systemet og modellen opp i mindre problemstillinger, vil variasjonen i utfordringer kunne minskes noe og det blir lettere for meg som veileder å sette meg inn i de aktuelle utfordringene til elevene.

Undervisningsopplegget legger til rette for at elevene selv skal skrive Python-kode. I enkelte oppgaver er noe kode gitt, og elevene skal enten bygge videre eller endre på denne forhåndsgitte koden. Orban et al. (2018) foreslår konkrete tilnærminger for å bruke programmering i fysikkundervisning. Tilnærmingene har ulik grad av påvirkning fra brukeren, og krever dermed ulik grad av ferdigheter i programmeringen. Den tradisjonelle tilnærmingen gir brukeren tilgang til å endre og/eller skrive hele koden i sin helhet og alle detaljer, den hybride legger til rette for at brukeren skal kunne endre koden, men ikke selve visualiseringen, mens tilnærmingen som kalles Physlet (interaktiv simulering) kun lar brukeren endre på de grunnleggende betingelsene og input (Orban et al., 2018). En grunn til at noe kode er gitt på forhånd, er for å dempe den kognitive belastningen til elever med lite



programmeringserfaring. Orban et al. løser dette ved å skjule deler av koden for brukeren (Orban et al., 2018, s. 837).



Figur 5: En oversikt over hvilke deler av programmeringen i et program brukeren har tilgang til i ulike tilnærminger (Orban et al., 2018, s. 832).

Dette undervisningsopplegget kan betraktes som et undervisningsopplegg med tradisjonell tilnærming (se Figur 5). All kode som er skrevet blir presentert for elevene, og de kan endre på det de ønsker. Samtidig er det ikke meningen at deler av koden som er skrevet skal endres. For eksempel er koden til den første grafen på oppgave 3b (Figur 8) gitt. Videre skal elevene bruke denne koden for å plote de andre grafene. Hvilken kode som ble gitt på forhånd, ble bestemt ut fra hvilken bakgrunnskunnskap elevene hadde og hvor jeg følte det var hensiktsmessig for elevene å skrive koden. Jeg visste for eksempel at elevene ikke var kjent med `matplotlib.pyplot` fra før, samtidig følte jeg at dette ikke var vanskelig kode å sette seg inn i. Derfor ble koden til den første grafen gitt, mens de to andre grafene skulle elevene skrive selv.

### 3.2.1 Samarbeid

Den overordnede arbeidsmetoden for elevene ble planlagt å være samarbeid. Elevene hadde i utgangspunktet hver sin datamaskin, men ble oppfordret til å samarbeide med hverandre. Pultene i klasserommet var plassert i rader, der to og to pulter var samlet. Det ble derfor tenkt at det var naturlig at en elev samarbeidet med eleven ved siden av, siden de pultene ville være samlet. Denne tiltenkte sammensetningen vil refereres til som *par* i masterstudien, og om paret samarbeidet med paret foran eller bak seg, kalles sammensetningen for en *gruppe*.

Hutchins et al. (2018) skriver at elever som samarbeider rundt modellering klarer å dekomponere fysiske fenomen og at samarbeidet bidrar positivt til progresjon i oppgaver og tolkning av instruksjoner. Siden undervisningsopplegget handler om å videreutvikle digitale modeller om fysiske fenomen, og at målet er å få et faglig utbytte i fysikk, kan det sies at elevsamarbeid er en arbeidsmetode som komplimenterer undervisningsopplegget. Sins et al. (2005) skriver at ved å samarbeide får elever utveksle ideer, komme med for- og

motargumenter, i tillegg til å konkretisere sin digitale beregningsmodell gjennom diskusjonene. Dette gjorde det ytterligere hensiktsmessig å legge til rette for samarbeid i undervisningsopplegget.

Selv om samarbeid ble sett på som fordelaktig som en overordnet arbeidsmetode, kan det medføre at elevenes besvarelser blir likere (Kortemeyer & Kortemeyer, 2018). Dette kan gi en negativ konsekvens både i læringsutbytte og opplevd læringsutbytte blant elevene. Om en elev har kopiert all sin kode fra medelever, vil personen kanskje ikke ha satt seg inn i de fysikkfaglige aspektene. At koden til elevene i seg selv er like, er ikke en utfordring i denne masterstudien, da hverken besvarelsene blir samlet inn og fokuset ikke ligger på det programmeringstekniske. Det er heller et faremoment at elevene ikke får satt seg inn i hvordan fysikken representeres i koden. Til tross for denne usikkerheten, ble det antatt at elever som jobber sammen vil delta i en fysikkfaglig diskusjon som kan gi et hensiktsmessig læringsutbytte som ikke vises i koden.

Selv om elevene skulle samarbeide i par og grupper, ble det tiltenkt at de skulle programmere på hver sin datamaskin. Dette ble gjort for at alle elevene skulle få mulighet til å faktisk utvikle den digitale modellen, ikke se på at en annen gjorde det. Tatt Kortemeyer og Kortemeyers (2018) funn om like koder i betraktning, ble det sett på som hensiktsmessig at elevene programmerte på hver sin pc. Om en elev i et par ikke var nødt til å skrive inn sin kode, ville det være mulig for eleven å melde seg helt ut. Tiltaket bidrar derfor til å engasjere alle elevene.

I likhet med andre studier (diSessa, 2000; Sørby & Angell, 2012) var det sosiokulturelle perspektivet også dominerende i utviklingsfasen og påvirket oppleggets oppgaver og helhet. I det neste delkapittelet vil denne teoriens kjennetegn presenteres, samt min tolkning og bruk av sosiokulturell læringsteori.

## Sosiokulturell læringsteori

Den sosiokulturelle læringsteoriens budskap er at læring er avhengig av en spesifikk sosial kontekst. Læring oppstår som en konsekvens av kommunikasjon med andre, ikke som en individuell prosess (Wertsch, 1985). Både sosiokulturell og sosiokonstruktivistisk læringsteori stammer fra behaviorismen, men sosiokulturell læringsteori vektlegger i større grad den brede sosiale og kulturelle konteksten et individ kan befinne seg i, som igjen kan være en katalysator for læring. Den sosiokonstruktivistiske læringsteorien baserer seg i større grad på hvordan læring i en sosial situasjon utvikles fra individets tidligere erfaringer (Scott &

Palincsar, 2013). Denne masterstudien ser på læring med fokus informantenes egne erfaringer med programmering. Erfaringene til informantene baserer seg videre på kulturen de er en del av, og i hvilken grad programmering har vært en del av deres undervisning tidligere i fysikk eller andre fag. Den sosiokulturelle læringsteorien har vært avgjørende for synet på hvordan programmering kan brukes til å lære fysikk og har medført at arbeidsmetoden i undervisningsopplegget ble planlagt til å være samarbeid. Læringsteorien har også blitt tatt i betraktning i utførelsen av intervjuene og i analysen av datamateriale.

### 3.2.2 Veiledning

I en muntlig introduksjon til arbeidsøkten ble det formidlet at selv om elevene programmerte på hver sin maskin, var det meningen at man skulle arbeide sammen. Det ble også sagt at selv om oppgaven virket lang, var det ikke meningen alle oppgavene skulle utføres. Det ble utviklet en lysbildefremvisning (se Vedlegg C) der informasjon om forskningsprosjektet og hva oppgavene handlet om skulle gjennomgås. Generelt under timen skulle jeg og to hjelpelærere (faglærerne) gå rundt og bistå elevene om de skulle ha spørsmål. De første oppgavene (se Vedlegg A) bestod som tidligere nevnt av små introduksjonsoppgaver. Om forhåndsgitt kode var skrevet, ble de gitt med en forklaring for hva kodelinjene gjorde i programmet. I tillegg, som kapittel 3.3 vil belyse, ble den digitale modellen brutt opp i flere oppgaver for elevene. Det var ikke et stort problem de skulle løse, men flere små. Dette tiltaket for veiledning er som tidligere nevnt med på redusere den kognitive lasten for elevene (Brown & Wilson, 2018).

## 3.3 Bruken av taksonomi for algoritmisk tenkning

I utviklingen av opplegget ble Weintrop et al. sin for algoritmisk tenkning brukt som en veiledning (2016). Det ble sett på som fordelaktig om opplegget la til rette for at elevene skulle få håndtere data, modellere og simulere, løse problemer digitalt og tenke i systemer. Inn under hver kategori ble praksisene i taksonomien sett på som en sjekkliste. For eksempel skal ifølge Weintrop et al. elevene generere data når de bruker konstanter og formler til å regne ut terminalfarten til fallskjermhopperen (2016, s. 135-142). Da har elevene jobbet med datahåndtering i sitt opplegg. Taksonomien for algoritmisk tenkning fungerer på denne måten som en sjekkliste, der målet var at elevene skulle få oppholde seg i ulike praksiser innunder hver kategori i taksonomien.

Fokuset var ikke nødvendigvis å være innom så mange praksiser som mulig. Elevene måtte også få mulighet til å faktisk få utbytte fra de aktuelle praksisene. Når elevene skulle

jobbe på sin modell, ble det i planleggingsfasen lagt stor vekt på at de også skulle bruke tid på å vurdere den digitale modellens egenskaper. Taub et al. belyser i en artikkel fra 2015 at elever som gjennomgår fysikkundervisning ved hjelp av digitale beregningsverktøy, kan få kognitiv overbelastning. I dette begrepet ligger det at i slik undervisning kan bli for mange momenter for elevene å fokusere på, og et opplegg som potensielt kunne ha gitt stort læringsutbytte, ender med å nettopp ikke gi dette læringsutbyttet (Taub et al., 2015). Dette gjorde det ytterligere hensiktsmessig å fokusere på enkelte praksiser i taksonomien for algoritmisk tenkning i dette undervisningsopplegget.

Enkelte av kategoriene i taksonomien til Weintrop et al. (2016) er mer omfattende enn andre og krever lengre tid til både planlegging og gjennomføring. Derfor ble enkelte praksiser og kategorier fokusert mer på enn andre. Opplegget la til rette for utbytte innen alle de fire kategoriene i en eller annen form, og derunder flere praksiser. De tre kategoriene **datahåndtering, modellering og simulering** og **digital problemløsning** ble integrert kronologisk i undervisningsopplegget. Elevene skulle først arbeide med datahåndtering, så med praksiser i modellering og simulering og til slutt digital problemløsning. Kategorien **systemtenkning** ble sett på som en overordnet kategori der elevene skulle benytte seg av praksiser gjennom hele opplegget, ikke bare i konkrete deler av enkelte oppgaver.

Nedenfor er en tabell (se Tabell 3), tilsvarende Tabell 1, men som viser hvilke praksiser som ble fokusert på i utviklingen av undervisningsopplegget og som er gjennomgått (markert i grønn). Det er ikke utelukket at andre praksiser kan være relevante til opplegget, men dette var i så tilfelle ikke planlagt på forhånd.

| <b>Datahåndtering</b> | <b>Modellering og simulering</b>                       | <b>Digital problemløsning</b>               | <b>Systemtenkning</b>                       |
|-----------------------|--|---|---|
| Samle data            | Bruke digitale modeller for å forstå et begrep         | Forberede problemer til å løse dem digitalt | Undersøke et komplekst system som en helhet |
| Generere data         | Bruke digitale modeller til å finne og teste løsninger | Programmere                                 | Forstå sammenhenger innad i et system       |
| Behandle data         | Vurdere digitale modeller                              | Velge effektive beregningsverktøy           | Tenke i nivåer                              |
| Analysere data        | Utforme digitale modeller                              | Vurdere effektive beregningsverktøy         | Kommunisere informasjon om et system        |
| Visualisere data      | Implementere digitale modeller                         | Utvikle digitale løsningsmodeller           | Definere systemer og håndtere kompleksitet  |
|                       |  | Utvikle digitale abstraksjoner              |   |
|                       |  | Feilsøking og feilretting                   |   |

Tabell 3: En oversikt over hvilke praksiser fra taksonmien for algoritmisk tenkning (fra Weintrop et al., 2016, s. 135) som blir brukt i oppgaven (markert i grønn).

Taksonmien for algoritmisk tenkning (Weintrop et al., 2016) var den viktigste kilden i utviklingen av oppleggets innhold. I de neste underkapitlene vil kategoriene og praksisene som ble benyttet gjennomgå med relevante eksempler fra oppgaven.

### 3.3.1 Datahåndtering

Undervisningsopplegget starter med introduksjonsoppgaver der elevene skal løse analytiske problemer, men ved bruk av Jupyter Notebook som en kalkulator. Det er først på oppgave 3a at de numeriske metodene kommer til sin rett. Elevene har forhåpentligvis blitt kjent med situasjonen, og får utdelt et rammeverk der de tre siste linjene skal fylles ut (se Figur 6).

### Oppgave 3

Tore ønsker å vite hvor lenge han er i luften, selv om han ikke har på seg fallskjerm. Under er et rammeverk der Tore tar høyde for luftmotstand, men han mangler den siste biten for å finne tiden hoppet tar.

#### 3a)

Fyll ut for-løkken (linje 17-19) slik at koden kjører

```
1 #Noen konstanter
2 T = 70 #Dette er et overestimat, som sikrer at simuleringen foregår lenge nok til at Tore treffer bakken.
3 dt = 0.0001
4 n = int(T / dt)
5
6 #Lager lister med nuller som senere skal fylles med henholdsvis posisjons-, farts- og akselerasjonsverdier.
7 s = np.zeros(n)
8 s[0]=h_0
9 v = np.zeros(n)
10 v[0]=v_0
11 a = np.zeros(n)
12 a[0] = -9.81 #Merk at a=-g. Dette kan du endre på om du vil
13
14 t = np.linspace(0, T, n) #En liste med tidsverdier fra 0 til og med T.
15
16 for i in range(1, n): #vi begynner på 1, siden første verdien hver liste er bestemt fra før.
17     s[i] =
18     v[i] =
19     a[i] =
```

Figur 6: Oppgave 3a (elevversjon).

Listene på linje 7, 9 og 11 er fylte med  $n$  nuller, nummerert fra 0 til  $n$ . Disse listene skal så fylles ut med tall som skapes ut fra formler for posisjon, fart og akselerasjon (se Figur 7). Listene ser først slik ut:

$$s = [0, 0, 0, 0, \dots, 0]$$

$$v = [0, 0, 0, 0, \dots, 0]$$

$$a = [0, 0, 0, 0, \dots, 0]$$

```
16 for i in range(1, n): #vi begynner på 1, siden første verdien i hver liste er bestemt fra før.
17     s[i] = s[i-1] + v[i-1]*dt
18     v[i] = v[i-1] + a[i-1]*dt
19     a[i] = a_g + k_uten_fallskjerm * v[i-1]**2/m
```

Figur 7: Oppgave 3a, linje 16-19 (løsningsforslag).

Hvert siffer  $i$  i listene vil så for steg ( $i$ ) byttes ut med nye tall som regnes ut fra verdier for posisjon, fart og akselerasjon fra det forrige steget ( $i - 1$ ), ganget med tidsintervallet  $dt$ . Når operasjonene er blitt gjort en gang, starter programmet en ny *iterasjon* ( $i = i + 1$ ), helt til telleren ( $i$ ) er nådd den gitte grensen for gjentakelser/iterasjoner ( $i = n$ ). Denne numeriske metoden tar utgangspunkt i *Eulers metode* i et endimensjonalt hopp, en metode med forskjellige navn som «ny = gammel + endring» (Yaşar, 2013) og «tikkemodellen» (diSessa, 2018). I mitt løsningsforslag definerte jeg positiv retning oppover og fikk negative verdier for fart og akselerasjon. Listene etter utbyttingen av verdier så dermed slik ut:

```

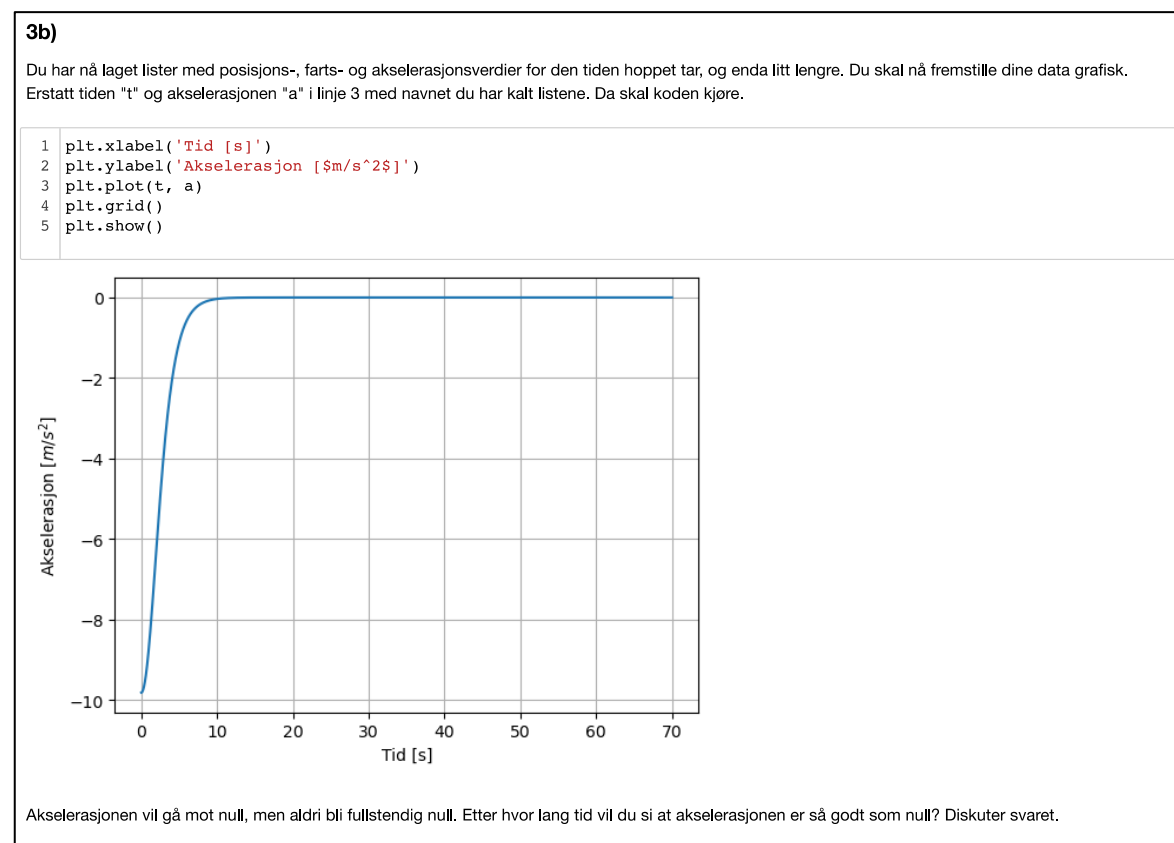
s = [1565, ..., 700.0, ...300.0, ..., -340.5]
v = [0.000, ..., -10,00, ...-28.02, -28.02]
a = [-9.810, ..., -7.800, ..., 0.000, 0.000]

```

Det er verdt å legge merke til at den siste verdien i listen for posisjon ( $s$ ) er negativ. Dette skjer fordi tidsintervallet som er gitt ( $T$ ) er lengre enn det fallskjermhoppet tar å gjennomføre. På denne måten får man et krysningspunkt ved bakkenivå og kan estimere hvor lang tid fallskjermhoppet tar.

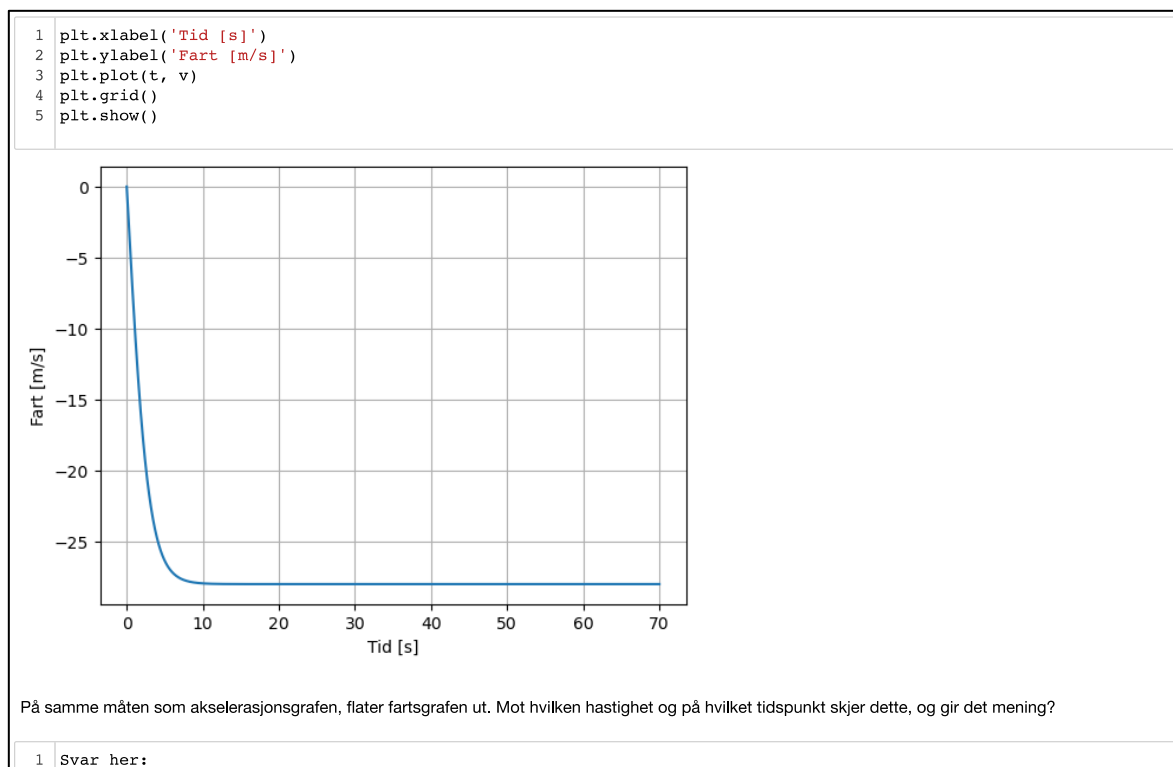
Det som blir gjort i oppgave 3a er at man gjennom numeriske metoder for beregning generer data. Hvis man skulle samlet reelle data i en slik situasjon, måtte man observert eller fått data fra et faktisk fallskjermhopp der luftmotstand var en betydelig faktor. Dette ville være omfattende og praksisen å *samle data* ble derfor valgt bort (Weintrop et al., 2016, s. 136).

I oppgave 3b skulle elevene bruke dataen de hadde generert og *visualisere* den. Dette gjorde de ved å bruke *plot*-funksjonen fra tilleggspakken *matplotlib.pyplot*. Koden for akselerasjonsgrafene var gitt, men koden forutsatte at elevene ikke hadde endret navnet på listen  $a$  eller listen  $t$  (se Figur 8). Elevene skulle så skrive tilsvarende kode for å visualisere dataen for posisjon ( $s$ ) og fart ( $v$ ).



Figur 8: Akselerasjonsgrafene på oppgave 3b (løsningsforslag).

De respektive grafene av den genererte dataen skulle så *analyseres*. Elevene blir mot slutten av oppgave 3b spurt om dataen gir mening (se Figur 9). Her kreves det at elevene kan se den matematiske sammenhengen mellom akselerasjon, fart og posisjon, i tillegg til en viss intuisjon om hvordan et fallskjermhopp kan foregå. Hvis elevene oppdager en feil i koden sin, må kode fra oppgave 3a og 3b gjennomgås for å sørge for at den genererte dataen og visualiseringen stemmer overens. Selv om elevene ikke bestemmer hvordan dataen skal visualiseres, må de forstå hvordan visualiseringen fungerer, i tillegg til å analysere det som blir visualisert (Weintrop et al., 2016, s. 135-136)



Figur 9: Fartsgraf og tilleggsspørsmål. Oppgave 3b (løsningsforslag).

### 3.3.2 Modellering og simulering

Undervisningsopplegget kan i sin helhet ses på som en simulering av en situasjon der relevant informasjon knyttet til et fallskjermhopp med luftmotstand skal utvikles og vurderes.

Kategorien modellering og simulering omfatter blant annet å *implementere digitale modeller* (Weintrop et al., 2016, s. 136-137). Modellen for situasjonen er avklart på forhånd for elevene, da de får gitt hvilke variabler i systemet som er relevante og skal regnes ut. Elevene må selv finne ut hvordan informasjonen fra variablene skal komme frem, da ved å bestemme formler, konstanter og hvordan man skal skrive koden (se Figur 10).



| Oppgave 1   |  |
|---|--|
| <b>1a)</b>  |  |
| Hvor lang tid tar det fra han hopper til han treffer bakken? Du kan se bort i fra luftmotstand. |  |
| 1   | #Hvilke konstanter vil du legge inn? Skriv koden din i denne cellen og trykk "Shift"+"Enter" for å kjøre |
| 1   | #Hvilke(n) formler vil du bruke? Skriv koden din her   |
| <b>1b)</b>  |  |
| Hva er farten i det Tore treffer bakken?  |  |
| 1   | #Skriv koden din her   |

Figur 10: Oppgave 1a og 1b (elevversjon).

Oppgave 1a og 1b er et eksempel på hvor elevene implementerer en digital modell. Situasjonen er forklart, og sentrale størrelser i modellen som starthøyde og startfart er gitt. Elevene blir bedt om å definere konstanter de selv anser som relevant og de må finne en måte de kan beregne variabler som tid og fart. Oppgave 1 og 2 (se Vedlegg A og B) går stegvis mot en mer realistisk modell. Svarene fra de tidligere oppgavene er lagret, slik at elevene kan bruke de senere. Når oppgave 3 ber elevene om å implementere numeriske beregninger, kan de forrige oppgavene brukes videre i oppgave 3 til å *finne og teste løsninger* for å se om svarene de får stemmer overens. Det ville for eksempel vært urealistisk om terminalfarten med luftmotstand var høyere en terminalfarten uten luftmotstand. Når elevene gjør dette, *vurderer* de samtidig sin digitale modell. I oppgave 4b (se Vedlegg A) blir elevene bedt om å bruke sine tidligere løsninger til å bestemme tidspunktet fallskjermen skal utløses. Her bruker elevene sin digitale modell konkret for å komme med en antagelse for når fallskjermen skal utløses.

### 3.3.3 Digital problemløsning

Under kategorien digital problemløsning skal elevene *forberede problemer til å løse dem digitalt*. Dette kommer tydelig frem i oppgave 1 og oppgave 2, der det er lagt opp til at elevene må kunne å bruke blant annet bevegelsesligningene og en formel for konstant luftmotstand. Bevegelsesligningene er gjerne oppgitt slik i formelheftet til fysikk på videregående skoler:

$$v = v_0 + at \quad (1)$$

$$s = v_0t + \frac{1}{2}at^2 \quad (2)$$

$$s = \frac{1}{2}(v + v_0)t \quad (3)$$

$$v^2 = v_0^2 + 2as \quad (4)$$

Når elevene da blir bedt om å finne tiden må de finne ut hvilken formel som kan brukes og forberede formelen til å kunne løses digitalt. Det innebærer å omforme formelen til å gi det svaret den ønsker, samt å definere kjente konstanter for å gi datamaskinen siffer å regne med. I oppgave 1a (se Vedlegg B) er det mest hensiktsmessig å bruke bevegelsesligning 3 for å finne tiden. Formelen må da gjøres om før man programmerer den inn, og gi datamaskinen informasjon om hvilken starthastighet ( $v_0$ ), sluttastighet ( $v$ ) og strekning ( $s$ ) den skal benytte i sin utregning.

I noen deler av undervisningsopplegget er deler av koden (oppgave 3a) eller hele koden (oppgave 3b) gitt på forhånd. Dette er kode som henholdsvis skal videreutvikles eller gjengis. Det krever dermed at elevene forstår hva som skjer i koden som allerede er gitt og matematikken bak, for så å kunne bruke det på en konstruktiv måte. Dette er egenskaper som faller under praksisen å *programmere* (Weintrop et al., 2016, s. 139). I oppgave 3b (se Figur 9) er koden for akselerasjonsgrafene gitt, men elevene skal bruke den samme koden til å plote fart og posisjonsgrafene videre. Her må elevene sette seg inn i hva de ulike kodelinjene gjør, for så å kunne endre de.

```
plt.plot(t, a)
```

Linjen over er gitt til elevene og skal endres til:

```
plt.plot(t, v)
```

```
plt.plot(t, s)
```

Elevene må forstå koden representerer tiden ( $t$ ) langs x-aksen og akselerasjonen ( $a$ ) langs y-aksen.

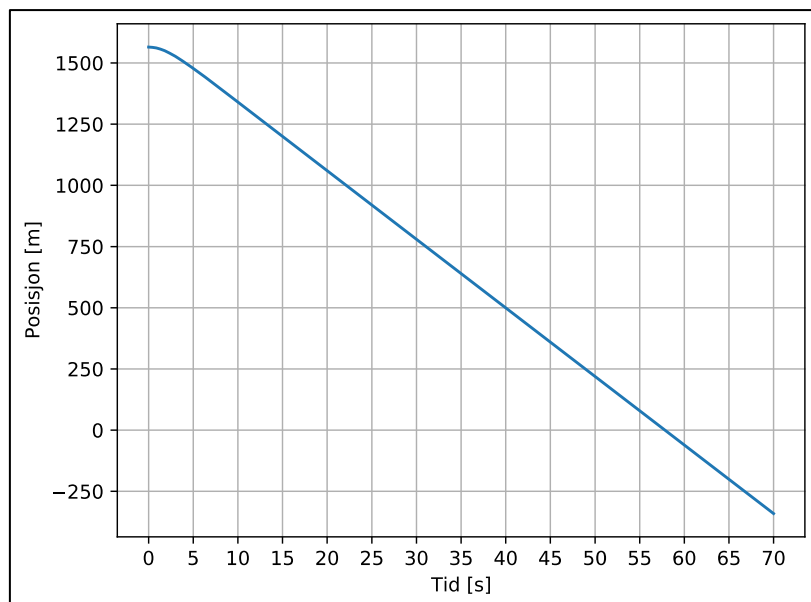
Hver enkelt oppgave bygger på hverandre. I de første oppgavene etablerer man konstanter for startfart ( $v_0$ ), akselerasjon ( $a_0$ ), starthøyde ( $h_0$ ) osv. Disse blir brukt til å gjøre overordnede beregninger som stadig blir mer realistiske. I oppgave 3b skal som nevnt elevene visualisere sine genererte data ved hjelp av grafer. Det er her de først får en oppfatning av hvordan dataen ser ut. Det kan dermed være tilfelle at grafen ser annerledes ut enn det man kunne forventet. Det finnes flere metoder for å løse slike feil, men grunnleggende er det lagt opp til at elevene må benytte seg av *feilsøking* og *feilretting* for å jobbe seg gjennom problemene (Weintrop et al., 2016, s. 140). Om for eksempel en kodelinje ikke vil kjøre, vil man få opp en feilmelding. Elevene blir da nødt til å sette seg inn i feilkoden for å prøve å

rette den opp. Hvis koden kjører, men grafen blir feil, er det kanskje mer hensiktsmessig å gjennomgå de fysiske formlene og implementeringen av dem.

### 3.3.4 Systemtenkning

Systemtenkning handler i stor grad om egenskaper og tenkemåter som brukes i digital vitenskap (Weintrop et al., 2016). En egenskap dette undervisningsopplegget legger til rette for er å *forstå sammenhenger innad i et system*. På oppgave 3b når elevene plotter en akselerasjonsgraf (Figur 8), en fartsgraf (Figur 9) og en posisjonsgraf (Figur 11). Elevene blir i tillegg spurt om resultatet gir mening. Her blir det lagt opp til at elevene må forstå formlene fra oppgave 3a (se Figur 6 og Figur 7) og den matematiske sammenhengen om at akselerasjonen er den deriverte av farten som igjen er den deriverte av posisjonen:

$$a = \dot{v} = \ddot{s}$$



Figur 11: Oppgave 3b - Posisjonsgraf (løsningsforslag)

Den samme oppgaven, i likhet med andre, legger opp til at elevene må kunne *kommunisere informasjon om systemet*. Opplegget oppfordrer til samarbeid, og spørsmålene om svarene elevene har kommet frem til gir mening, gjør at en form for kommunikasjon, enten med hverandre eller til datamaskinen, kreves. De ulike sammenhengene i systemet kommuniseres også gjennom blant annet *visualisering av data* (Weintrop et al., 2016, s. 141).



## 4 Metode

Masterstudien kan betraktes som en kvalitativ intervensjonsstudie med et fleksibelt forskningsdesign. En intervensjonsstudie består i hovedsak at forskeren tar utgangspunkt i en utfordring, for så å undersøke og forsøke å utbedre utfordringen (Fraser & Galinsky, 2010). I denne masterstudien er utfordringen bruken av programmering som et verktøy for læring av fysikk. Jeg utvikler så et undervisningsopplegg som bruker programmering som nettopp et verktøy for læring av fysikkfaglig kompetanse, ikke et undervisningsopplegg med formål om å bare lære seg programmering. Studien har et fleksibelt design, der metodiske, teoretiske og praktiske valg er en del av en dynamisk prosess i utvikling (Robson & McCartan, 2016, s. 146-147). Selv om metodene for datainnsamling var fastsatt, var problemstillingen med tilhørende forskningsspørsmål stadig i utvikling gjennom hele forskningsprosessen.

Elevenes oppfatninger og utfordringer utgjør en viktig dimensjon når det kommer til evaluering av et undervisningsopplegg. Deres opplevelser kan knyttes til motivasjon, mestring, forventninger til å utforske løsninger på egenhånd, opplevelse av relevans og læring (Wæge & Nosrati, 2018). Det ble derfor naturlig at elevene skulle få ytre sine erfaringer. Den primære metoden for datainnsamling er semistrukturerte intervju av elever som har gjennomført undervisningsopplegget. Det ble i tillegg ansett som hensiktsmessig å utvikle en spørreundersøkelse som ble tilbudt hele klassen, slik at alle kunne få svare på spørsmål og dele tanker om undervisningsopplegget. I tillegg til intervjuene og spørreundersøkelsene, vil observasjoner gjort i timen kunne være med på å gi resultatene ytterligere kontekst og mening. Det ble gjennomført tre intervju, derav et gruppeintervju av to elever som samarbeidet tett gjennom hele opplegget.

Studien bruker som nevnt både spørreundersøkelse, intervju og observasjoner som metoder for datainnsamling. Det å blande kvantitative og kvalitative metoder for datainnsamling kan medføre praktiske og prinsipielle utfordringer. Robson & McCartan (2016, s. 174-182) presenterer ulike utfordringer og fordeler med blandede metoder for datainnsamling. Blant annet argumenterer Sale, Lohfeld og Brazil (2002) for at kvantitative og kvalitative metoder er så forskjellig av natur, at det å kombinere de ikke er mulig. På den andre siden gir variasjonen i innsamlingsmetoder en større grad av triangulering, og man kan bruke kvalitative metoder til å undersøke funn fra kvantitative data nærmere. Det viktigste når

man kombinerer metoder for datainnsamling, er at forholdet mellom de er tydelig avklart i planleggingsfasen og i utføringen (Robson & McCartan, 2016, s. 180). Spørreundersøkelsen i masterstudien har som mål å samle inn generelle data blant et utvalg som ikke kan generaliseres til å gjelde en hel befolkning, men dekker fortsatt en større gruppe elever enn intervjuenes utvalg. Grunnet utvalget til spørreundersøkelsen og de spørsmålene som ble stilt, kan spørreundersøkelsen betraktes som en kvalitativ metode for datainnsamling. Flere metoder for datainnsamling er anbefalt i et fleksibelt design (Robson & McCartan, 2016, s. 147), da det som tidligere nevnt gir en økt grad av triangulering og dermed troverdighet. De neste delkapitlene vil gjennomgå metodiske valg angående datainnsamling og analyse av datamaterialet. I tillegg drøftes etiske implikasjoner og hensyn som ble tatt, før troverdigheten til datamaterialet og analysen blir gjennomgått.

## 4.1 Innsamling av data

Datainnsamlingen bestod av blant annet en spørreundersøkelse som hele klassen gjennomførte. Spørreundersøkelsen var en avsluttende del av undervisningsopplegget. Den ble gjennomført i Nettskjema.no og ingen personopplysninger ble samlet inn. Undersøkelsen bestod av enkle spørsmål som for eksempel hvor mange oppgaver elevene gjennomførte og hvor langt de kom seg i opplegget. I tillegg ble det stilt noen mer åpne spørsmål der elevene kunne svare hvordan de opplevde opplegget (se Vedlegg D). Formålet med spørreskjemaet var å danne seg et overordnet bilde av hvordan opplegget opplevdes blant elevene. Intervjuene ble dermed veiledet av spørreundersøkelsen. Siden informantenes overordnede oppfatninger og utfordringer var belyst allerede, kunne intervjuet fokusere mer på enkelte deler. Om det var interessante funn fra spørreundersøkelsen, kunne dette undersøkes nærmere og mer utdypende i intervjuet. For eksempel kom det frem fra spørreundersøkelsen at de fleste elevene ikke hadde kom seg til lengre enn til oppgave 3a, noe som gjorde det interessant å undersøke nærmere hvorfor elevene ikke kom seg videre til oppgave 4 og 5. I tillegg ville resultatene fra spørreundersøkelsen kunne bekrefte eller avkrefte påstander i intervjuet. Dette er et av resultatene i kapittel 5.2.4, der spørreundersøkelsen gav informantenes besvarelser i intervjuene en kontekst.

Fra et sosiokulturelt perspektiv er det nettopp gjennom å interagere med andre mennesker i en sosial kontekst at kunnskap blir kommunisert og tilgitt mening (Robson & McCartan, 2016, s. 24-25). I en undersøkelse der et mål er å finne ut hvilke erfaringer elevene sitter igjen med etter et undervisningsopplegg de selv har gjennomført, ble derfor det

semistrukturerte intervjuet sett på som ideelt og videre bestemt til å være den primære metoden for datainnsamling.

Intervjuene ble holdt som semistrukturerte intervju, da det gav mulighet for å kunne spille videre på informantenes besvarelser, samtidig som en progresjon i intervjuet ble sikret (Robson & McCartan, 2016, s. 284-293). Det kvalitative intervjuet ble regnet som hensiktsmessig, da faktorer som kroppsholdning, tonefall osv. kan gi forskeren ytterligere informasjon. For eksempel gav tonefallet og kroppsholdning i to av intervjuene informasjon om holdningen de hadde til undervisningsopplegget. Ordene «det var greit» ble i intervju med eleven Dani tolket i en negativ favør, da tonefallet var monotont med manglende øyekontakt i kontrast til andre besvarelser. De samme ordene ble sagt av eleven Alex i et annet intervju, men gitt et mer engasjert og givende tonefall ble ordene tolket i positiv favør.

Intervjuene ble i utgangspunktet også utviklet som individuelle intervju, da spørreundersøkelsen også var individuell. Siden informantene hadde tatt stilling til undervisningsopplegget allerede kunne de besvare mange av spørsmålene uten lengre diskusjoner og resonnementer med andre informanter. Det ene intervjuet ble derimot holdt med to informanter, da de var det eneste paret som samarbeidet rundt en datamaskin, noe som gjorde det hensiktsmessig å intervjuer begge to samtidig. Det semistrukturerte intervjuets form ble dermed ytterligere hensiktsmessig, da man enklere kunne omforme spørsmålene til å passe bedre for gruppen av informanter. I studier med fleksible design, flere metoder for datainnsamling og situasjoner der forskeren er sterkt involvert i forskningsprosessen er semistrukturerte intervju ofte brukt (Robson & McCartan, 2016, s. 290-291).

Intervjuet kan ses på som et intervju i to faser, der de første fem spørsmålene er spørsmål for å kunne avdekke bakgrunn og generelle holdninger til fysikk og programmering, i tillegg til å få samtalen i gang. Den fullstendige intervjuguiden finnes som vedlegg (se Vedlegg E). Deretter følger spørsmål som omhandler undervisningsopplegget. Hvert spørsmål har et overordnet tema der informanten etterhvert skal komme med konkrete eksempler. For eksempel da det i spørsmål 9 blir spurt om hvordan informanten arbeidet i timen. Den vil da forhåpentligvis svare overordnet om hvordan arbeidsmetodikken var i timen, før det blir spurt om informanten kan utdype og gi konkrete eksempler.

Intervjuguiden ble utviklet basert på Sørby og Angells (2012) studier gjort med lignende oppgaveformat på studenter på universitetsnivå. Fra denne studien så Sørby og Angell at studentene jobbet i såkalte moduser; matematikkmodus, fysikkmodus eller

programmeringsmodus. Intervjuet i dette masterarbeidet tok dermed utgangspunkt i at elevene måtte oppholde seg i disse modusene på ulike tidspunkt i løpet av opplegget, og at spørsmålene skulle avdekke innenfor hvilken modus ulike deler av opplegget stimulerte til. Sørby og Angells moduser ble også sett på som mer anvendelig å avdekke i en intervjusammenheng enn Weintrop et al. sin taksonomi for algoritmisk tenkning, som er mer omfattende og intrikat. Taksonomien er fortsatt å regne som masterstudiens teoretiske fundament, da utviklingen av undervisningsopplegget og resultatene ses i lys av den.

#### 4.1.1 Utvalg

Undervisningsopplegget med undersøkelsen ble gjennomført i en fysikk 1 klasse på en offentlig skole der ferdighetsnivået blant elevene ifølge faglærerne varierte i stor grad. Hele klassen som gjennomførte opplegget ble tilbudt å frivillig gjennomføre spørreundersøkelsen. Det viste seg at 22 av 22 elever som deltok i undervisningsopplegget også svarte på spørreundersøkelsen.

Siden deltagelse til intervjuet var basert på frivillighet, kan utvalget ses på som et bekvemmelighetsutvalg (Robson & McCartan, 2016, s. 280-281). Det ble bestemt at tre intervju ville kunne gi tilstrekkelig datamateriale. Med andre ord er det 4 av 22 elever som stilte til intervju. Alle de fire informantene hadde gode erfaringer med fysikk og anså seg selv som kompetente i faget. Det er i transkripsjonsarbeidet tildelt fiktive og kjønnsnøytrale navn til informantene, da kjønnsaspekter ikke er en del av min problemstilling. Hver informant ble spurt innledende spørsmål som omhandlet interesser, tanker om fysikkfaget generelt osv. Spørsmålene gav informasjon om bakgrunnen til informantene (se Tabell 4). Personen som blir omtalt som Alex har mye programmeringserfaring, da både i programmeringsspråkene Python og noe C++. Alex synes at fysikk er veldig gøy, fordi hen blir utfordret og som regel klarer å jobbe seg forbi utfordringene. Oppgaver som kanskje er vanskelige og tidkrevende er ikke å så gøy, men Alex mener at det er disse oppgavene som er ekstra givende når man omsider får de til. Dani har også litt programmeringserfaring, da fra TOF (Teknologi og forskningslære) som er et valgfag på skolen. Dani valgte fysikk fordi naturfag var artig og fordi ingeniørstudiene krever det. Hen synes at det er dumt at mange fysikktimer består av individuell oppgaverregning der man ender opp med å ikke prate med noen en hel arbeidsøkt. Kim og Sam samarbeidet tett i timen. Ingen av de har noe programmeringserfaring utover det de har gjort i matematikken fra førsteklasse. Begge anser seg som kompetente i fysikk, men synes programmering er vanskelig. Informantene er til vanlig fordelt på fysikkklasser, men samtlige har lite programmeringserfaring fra fysikkfaget. At elevene anser seg som



kompetente i fysikkfaget, kan gjøre at holdningen til arbeid i fysikk blir fremstilt mer positivt enn det ville ha gjort hos en informant som mente at fysikkfaget var mer utfordrende.

| <b>Fiktive navn</b> | <b>Bakgrunnsinformasjon</b>  |
|---------------------|--|
| Alex                | Har ikke faget Teknologi og forskningslære.<br>Mye programmeringserfaring (Python, C++).<br>Liker fysikk på grunn av utfordringer.<br>Glad i utfordringer.<br>Vil studere medisin.   |
| Dani                | Har faget Teknologi og forskningslære.<br>Valgte fysikk fordi naturfag var artig.<br>Ønsker å studere innen en ingeniørretning.<br>Mener fysikktimene ofte bestod av ensformig oppgaveregning.   |
| Kim                 | Har ikke faget Teknologi og forskningslære.<br>Anser seg som kompetent i fysikkfaget.<br>Har ingen programmeringserfaring utover matematikkfagene på skolen.<br>Liker fysikk fordi hen får det til.<br>Samarbeidet tett med Sam, rundt en felles datamaskin. |
| Sam                 | Har ikke faget Teknologi og forskningslære.<br>Har ingen programmeringserfaring utover matematikkfagene på skolen.<br>Anser seg som kompetent i fysikkfaget.<br>Samarbeidet tett med Kim, rundt en felles datamaskin.  |

Tabell 4: En oversikt med bakgrunnsinformasjon for hver informant til intervjuene.

Hverken spørreundersøkelsen eller intervjuet har et utvalg som i seg selv kan generaliseres til å gjelde for eksempel alle elever i den norske videregående skolen. Selv om funnene ikke er statistisk generaliserbare, mener jeg at utvalget bidrar med oppfatninger og betraktninger som er av verdi til masterarbeidet (Robson & McCartan, 2016, s. 166). Kvale og Brinkmann (2015, s. 290-291) argumenterer for at selv om resultatene fra kvalitative intervjuer ikke kan generaliseres globalt, kan kunnskapen som formidles i intervjuene gjelde i andre relevante situasjoner. Med andre ord kan betraktningene som informantene i intervjuene kommer med, medføre en kunnskap som også gjelder andre elever i samme situasjon som informantene. Funnene fra denne studien kan derfor anses som analytisk generaliserbare, der leseren selv bestemmer i hvilken grad resultatene kan brukes som veiledning i relevante situasjoner, basert på diskusjonens argumenter (Kvale & Brinkmann, 2015, s. 291).

## 4.2 Analyse av datamaterialet

I analysearbeidet ble alle lydfilene transkribert, men ikke gjengitt til hver minste detalj. Uklare ord eller setninger er transkribert som «[ugjenkjennelig]», lange pauser der informanten stammer eller stotrer er utelatt. Tolkninger er blitt gjort om konteksten har vært tydelig, som for eksempel «[egentlig mening]» og «[presiserende ord]» om informanten for eksempel sier «akselerasjon» når hen tydelig referer til en fartsgraf. De egentlige meningene og de presiserende ordene blir presentert rett bak informantens ord i løpende tekst. Renskrivning kan gjøre at verdifull informasjon forsvinner, men det ble ansett som mer verdifullt å fokusere på informantens ord og mening, ikke pauser eller lignende. Ord for å beskrive tydelige emosjonelle tegn fra informanten eller fra meg som forsker, som for eksempel «[latter]», er inkludert i transkripsjonen. Lengre pauser er markert med «...» og onomatopoetikon som «ehm» er inkludert der ordene tydelig kommer frem, da dette kan vise at informanten eller forskeren er usikker eller tenker lenge. Slike avgjørelser kan følgelig påvirke troverdigheten til resultatene, da det er meg som forsker som tar avgjørelser om hva som skal renskrives, inkluderes eller ekskluderes (Robson & McCartan, 2016, s. 305).

Etter transkripsjonsarbeidet var gjennomført ble intervjuene analysert ved bruk av tematisk analyse i analyseverktøyet NVivo (NTNU, 2022). Den tematiske analysen har sin styrke i at den er fleksibel i alle deler av analysearbeidet, og passer derfor et fleksibelt forskningsdesign godt (Braun & Clarke, 2012, s. 290-291). Dette ble det gjort nytte av, da forskningsspørsmålene og videre tematisk nettverk ble videreutviklet parallelt.

Den tematiske analysen består av seks steg der det første steget handler om å bli kjent med datamaterialet gjennom for eksempel transkripsjon og lesing av transkripsjonen flere ganger (Braun & Clarke, 2006). I denne masterstudien ble dette steget overholdt, da transkripsjonen av intervjuene ble gjort to ganger for å sikre presise gjengivelser. Dette resulterte i at jeg ble godt kjent med datamaterialet jeg skulle analysere videre. Det andre steget består av å danne de første kodene. Disse kodene kan være gjengivelser av det som er blitt sagt og trenger ikke å være så dyptgående. Eksempler på de første kodene i min tematiske analyse var «Problem med lister», da Alex i sitt intervju forteller om hvordan hen forklarer utfordringene sine i 3a. Dette er mer eller mindre en oppsummering av Alex sitt sitat og gir ingen videre innsikt i hvor problemet lå.

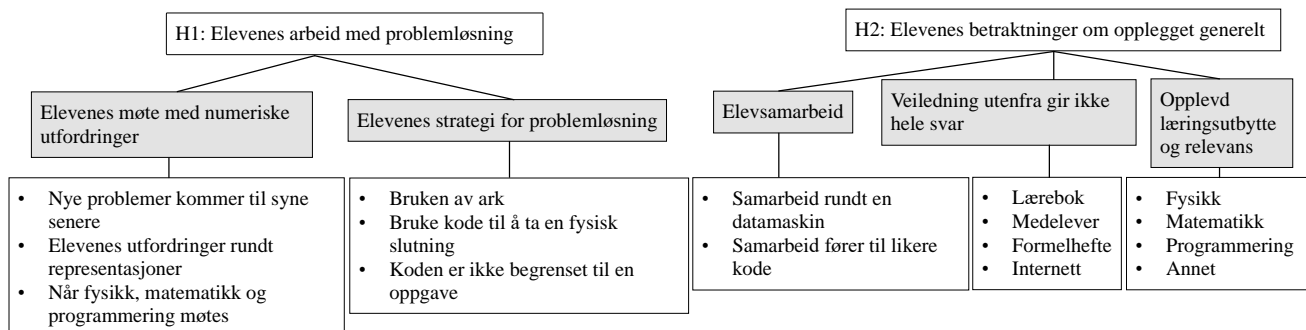
Det tredje steget i den tematiske analysen er å begynne å søke etter temaer. De første kodene kan være mange, overlappende med hverandre osv., og i denne fasen gjelder det å få

samlet kodene i temaer. Disse temaene skal i det fjerde steget videreutvikles. I tillegg må man vurdere troverdigheten til temaene. Om temaene gjengir datamateriale likestilt og på en korrekt måte, kan man gå videre til det femte steget. Det femte steget handler om å finjustere på temaene og etablere navn på de som forklarer hva de inneholder (Braun & Clarke, 2006). Steg tre til fem var i denne studien svært tidkrevende. Mine første koder var overlappende og overfladisk deskriptive og krevde en nøyere analyse av hvilke resultater datamateriale faktisk bidro med. Som nevnt tidligere i metodekapittelet bestod intervjuene av to faser, der første del av intervjuet også omhandlet hva elevene syntes om opplegget, før spørsmål om konkrete oppgaver ble stilt. Dette resulterte i et datamateriale som gav resultater i to dimensjoner. Disse to dimensjonene ble derfor valgt som hovedtemaer: «Elevenes arbeid med problemløsning» (H1) og «Elevenes betraktninger om opplegget generelt» (H2). Sitatet nedenfor er et eksempel på datamateriale som fungerer i to dimensjoner:

*I: Hva fikk du ut av den hjelpen du fikk av lærerne?*

*Dani: Det var jo begrensa. Du prøvde i hvert fall å gi meg en del hint. Det var denne oppgaven [3a]. Programmet ville ikke gi meg riktig svar, selv om jeg hadde skrevet nesten nøyaktig det samme som en annen. [...]*

Spørsmålet som ble stilt var ment å gi svar på noe om opplegget generelt, hvorav Dani sier at veiledningen gav hen noen hint. Dette ble kodet inn i det andre hovedtemaet (H2). I tillegg gir sitatet informasjon om at oppgave 3a var utfordrende og krevde veiledning, samt at informanten sammenlignet kode med andre. Dette kan ses opp mot andre sitat og gi informasjon om hvordan oppgave 3a ble løst av informantene, noe som gjorde at sitatet også ble kodet inn i det første hovedtemaet (H1). Selv om datamateriale kan gi overlappende koder og temaer, har et tematisk nettverk blitt utviklet:

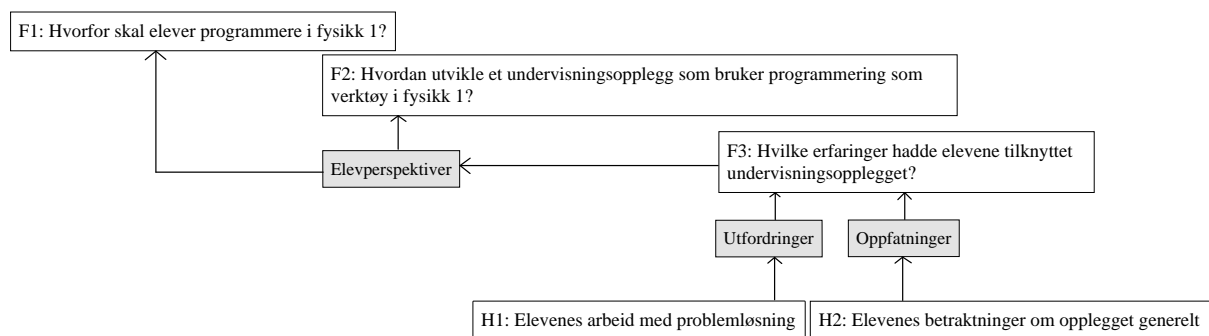


Figur 12: Det tematiske nettverket utviklet fra intervjuene.

Figur 12 viser det tematiske nettverket i sin helhet. Under hovedtemaet «Elevenes arbeid med problemløsning» (H1), finnes temaene «Elevenes møte med numeriske

utfordringer» og «Elevenes strategi for problemløsning» som henholdsvis tar for seg hvordan elevene gjenkjente en utfordring og hvordan de oppstod, før hvordan elevene arbeidet for å løse utfordringene og problemene inngår i elevenes strategi for problemløsning. Hovedtemaet «Elevenes betraktninger om opplegget generelt» (H2) sine undertema kan ses på som mer deskriptive, der analysearbeidet i stor grad handler om å diskutere elevenes betraktninger opp mot hverandre og opp mot relevant teori i resultat- og diskusjonskapittelet.

I det siste steget i den tematiske analysen (Braun & Clarke, 2006) skal rapporten produseres, og analysearbeidet skal presenteres. Formålet med den tematiske analysen er å fortelle den kompliserte historien dataen forteller på en organisert og beskrivende måte. Alle de seks stegene er stadig i utvikling og det er en del av prosessen å gå frem og tilbake mellom de ulike stegene (Braun & Clarke, 2012). Bevegelsen frem og tilbake i stegene i den tematiske analysen, ble i mitt analysearbeid tydeligst da jeg så sent som i rapportskrivningen gjorde endringer i mitt tematiske nettverk for å knytte temaene opp mot forskningsspørsmålene på en organisert måte.



Figur 13: En oversikt over hvordan hovedtemaene besvarer forskningsspørsmålene.

De to hovedtemaene (se Figur 13) dekker begrepet *erfaringer* i forskningsspørsmål 3 ved å gi perspektiver på *utfordringer* og *oppfatninger* elevene hadde. De to hovedtemaene gir også *elevperspektiver* som er med på å belyse forskningsspørsmål 1 og 2 direkte.

Nedenfor (se Tabell 5) er en oversikt over hva som kjennetegner en god tematisk analyse (Braun & Clarke, 2006, s. 96). Av 15 punkter som bidrar til en god tematisk analyse, er det kun ett av punktene (punkt 6) som var utfordrende i denne masterstudien.

| Process        | No. | Criteria   |
|----------------|-----|--|
| Transcription  | 1   | The data have been transcribed to an appropriate level of detail, and the transcripts have been checked against the tapes for 'accuracy'.                        |
| Coding         | 2   | Each data item has been given equal attention in the coding process.   |
|                | 3   | Themes have not been generated from a few vivid examples (an anecdotal approach), but instead the coding process has been thorough, inclusive and comprehensive. |
|                | 4   | All relevant extracts for all each theme have been collated.   |
|                | 5   | Themes have been checked against each other and back to the original data set.   |
| Analysis       | 6   | Themes are internally coherent, consistent, and distinctive.   |
|                | 7   | Data have been analysed – interpreted, made sense of – rather than just paraphrased or described.  |
|                | 8   | Analysis and data match each other – the extracts illustrate the analytic claims.  |
| Overall        | 9   | Analysis tells a convincing and well-organized story about the data and topic.   |
|                | 10  | A good balance between analytic narrative and illustrative extracts is provided.   |
|                | 11  | Enough time has been allocated to complete all phases of the analysis adequately, without rushing a phase or giving it a once-over-lightly.                      |
| Written report | 12  | The assumptions about, and specific approach to, thematic analysis are clearly explicated.   |
|                | 13  | There is a good fit between what you claim you do, and what you show you have done – ie, described method and reported analysis are consistent.                  |
|                | 14  | The language and concepts used in the report are consistent with the epistemological position of the analysis.   |
|                | 15  | The researcher is positioned as <i>active</i> in the research process; themes do not just 'emerge'.  |

Tabell 5: En gjennomgang av hva som kjennetegner en god tematisk analyse (Braun & Clarke, 2006, s. 96).

Tabellen viser at kodene bør være distinkte og ikke overlappe med hverandre (punkt 6). Dette var som tidligere nevnt en utfordring, men gitt forskningsspørsmålene og intervjuets dimensjoner, kan datamateriale og koder overlappe og samtidig gi ny innsikt alt etter hvilket perspektiv de ses i lys av. Som det tematiske nettverket (se Figur 12) viser, er temaene distinkte, men kan inneholde overlappende data.

| Hovedtemaer:                               | Antall kodeenheter |
|--|--------------------|
| Elevenes arbeid med problemløsning         | 58                 |
| Elevenes betraktning om opplegget generelt | 73                 |

Tabell 6: En oversikt over hvor mange koder hvert av hovedtemaene bestod av.

Tabellen ovenfor (se Tabell 6) viser fordelingen av koder i hovedtemaene. En «kodeenhet» er et direkte sitat eller deler av et sitat som er kodet inn i temaer i det tematiske nettverket (se Figur 12). Fordelingen er relativ jevn, noe som viser at det tematiske nettverket gir resultater som er vektet likt i begge hovedtemaer. Dette viser videre at metoden for datainnsamling kan bidra til besvarelse av forskningsspørsmålene fra perspektiver om elevenes arbeid med problemløsning og elevenes betraktning om opplegget generelt.

### 4.3 Etske betraktninger

Masterprosjektet med metoder for datainnsamling og innsamling av personopplysninger fra intervjuet, har blitt godkjent av Sikt–Kunnskapssektorens tjenesteleverandør (2023). All data

som har blitt samlet inn og blir presentert i studien, vil være i tråd med søknaden (se Vedlegg F) og godkjenningen fra Sikt.

Som tidligere nevnt, ble ingen personopplysninger samlet inn fra spørreundersøkelsen. Det er likevel etiske hensyn som må tas i betraktning. Det ble for eksempel presisert i forkant av spørreundersøkelsen at det var frivillig å delta og at det ikke ville ha noen negative konsekvenser å ikke svare på den. Det er fortsatt mulig at siden det ble satt av tid i timen til å gjennomføre spørreundersøkelsen, at noen informanter kan ha følt seg presset til å delta.

Intervjuene ble alle tatt opp på en diktafon uten nettilgang for å sørge for en presis gjengivelse i transkripsjonsarbeidet. I forkant av intervjuet måtte informantene lese og skrive under på et samtykkeskjema i tråd med Sikts retningslinjer (se Vedlegg F). Det ble i tillegg gitt et muntlig sammendrag av det som stod i informasjonsskrivet, som for eksempel at alle særnavn og andre personopplysninger ville bli anonymisert i en transkripsjon. Det ble også informert om at mot slutten av intervjuet skulle det bli spurt om det var noe informanten ønsket å trekke fra sin besvarelse. I transkripsjonsarbeidet ble alt av personopplysninger utelatt. Navnene som er valgt for å gjengi sitatene er Alex, Dani, Kim og Sam. Disse navnene er ikke tilknyttet noe kjønn, men er valgt for å gi leseren mer oversikt. For informantenes personvern vil ikke transkripsjonen i sin helhet bli presentert, men relevante sitater brukes for å utdype og eksemplifisere resultatene.

De etiske betraktningene og tiltakene som er blitt gjort i denne masterstudien, kan påvirke troverdigheten til resultatene i både positiv og negativ retning. I det neste delkapittelet vil derfor troverdigheten til datamaterialet og resultatet fra analysen belyses.

#### 4.4 Troverdighet

I denne masterstudien brukes begrepet troverdighet i stedet for mer kvantitative ladede begreper som validitet og reliabilitet. Troverdighet omfatter dermed validitet, reliabilitet og generalisering (Robson & McCartan, 2016, s. 168-173). Tiltak som kan øke graden av troverdighet i en studie er blant andre triangulering i metoder for datainnsamling, datainnsamling over en lengre periode, forskertriangulering og det å ta lydopptak av intervjuene. Av etiske hensyn er lydopptakene fra intervjuene i sin helhet ikke tilgjengelig for leseren av personvernsgrunner, men sitatene er gjengitt i resultatdelen så fullstendig som mulig, og alltid med den aktuelle konteksten eller spørsmålet tilknyttet sitatet. Faktorer som

kan svekke troverdigheten i denne masterstudien og tiltak som ble gjort for å øke graden av troverdighet, vil bli drøftet i de neste avsnittene.

Masterstudien med det semistrukturelle intervjuet er som tidligere nevnt inspirert av sosiokulturalismen, noe som impliserer at forskeren til en grad er delaktig i forskningsprosessen (Robson & McCartan, 2016, s. 24-25). I tillegg er jeg også delaktig i undervisningsøkten som klasseleder. Min rolle som forsker og samtidig en lærer vil kunne virke inn på informantenes besvarelser. Under opplegget var jeg en klasseleder som gjorde alle didaktiske grep for at timen skulle fungere så godt som mulig. Informantene var klar over at opplegget de nå gjennomførte, ville hjelpe meg i mitt masterarbeid. Når de da besvarer spørreundersøkelsen, kan de dermed ha blitt påvirket av den kortvarige relasjonen som er blitt opparbeidet, både på godt og vondt. Det samme gjelder for intervjuene. Et tiltak som kunne blitt gjort for å minke deltakerbias blant informantene, kunne være å samle inn data over et lengre tidsperspektiv enn to dager. Grunnet praktiske hensyn opp mot skolens time- og fremgangsplaner, var det ikke mulig å samle inn data over lengre tid. To dager med to metoder for datainnsamling, er likevel et tiltak som øker graden av troverdighet.

Rollen som delaktig lærer og forsker, vil også kunne påvirke min tilnærming til intervjuene og analysen av datamaterialet. Betydningen av relasjonen mellom forsker og informantene kan ifølge Robson og McCartan minskes dersom forskeren avklarer sine verdier (2016, s. 148-149). Det er derfor viktig at spørsmålene er formulert på en måte som ikke vektlegger eller gir uttrykk for et ønskelig svar. Dette er imidlertid vanskelig å gjennomføre, siden det semi-strukturelle intervjuet åpner opp for oppfølgingsspørsmål som ikke nødvendigvis er planlagt like nøye som de forhåndsbestemte spørsmålene. Det ble derfor i tillegg til spørsmålene, utviklet en begrunnelse til hvorfor jeg stiller utvalgte spørsmål (se Vedlegg E). Dette skulle i større grad forhindre at oppfølgingsspørsmålene tjente en annen hensikt enn de opprinnelige spørsmålene. I tillegg til dette var forslag til oppfølgingsspørsmål og hva de kunne omhandle foreslått i intervjuguiden. Det viste seg likevel at enkelte oppfølgingsspørsmål var ledende, og troverdigheten på svarene etterfulgt av spørsmålene måtte tas i betraktning i analysearbeidet. Dette ble gjort ved å rett og slett utelate enkelte besvarelser på de ledende oppfølgingsspørsmålene, med mindre de bidro med aspekter utover oppfølgingsspørsmålets tematikk. Et konkret eksempel på dette er:

*I: Synes du at 3a var vanskelig?*

*Alex: Ja, den var jo vanskeligere enn de andre oppgavene, eller, jeg kom jo til oppgave 4a. Sånn jeg ser det. Jo mer tid jeg bruker på en oppgave, jo mer givende er det å få den til. Jeg fikk et dytt i riktig retning, jeg fikk ikke svaret, og da synes jeg at det er enda bedre at når jeg prøver meg frem med mine egne metoder, at jeg da får det riktige svaret. Så, det synes jeg var veldig greit.*

I et slikt sitat kan ikke det konkluderes med at Alex synes oppgave 3a var vanskelig, men hen gir innsikt i hvordan hen arbeidet for å løse et problem og hvordan det føltes for Alex å løse en oppgave som hen brukte mye tid på.

Selv om det kvalitative intervjuet bringer frem utfordringer som kan svekke troverdigheten til resultatene, medfører intervjuformen også noen fordeler. Kvalitative intervju er ifølge Kvale og Brinkmann (2015, s. 42-43) hensiktsmessige om informantens eget perspektiv er ønskelig, noe masterstudiets fokus retter seg mot. Semistrukturelle intervju gir en naturlig form for samtale, men for å utføre intervjuet på en strukturert og profesjonell måte som gir hensiktsmessige resultat, kreves det en form for strategi og struktur. Kvale og Brinkmann eksemplifiserer dette med å gi åpne spørsmål som informantene kan svare på innledningsvis, for så å stille mer spesifiserte spørsmål etter hvert. Dette ble tilstrebet i mine intervju, noe som viste seg nødvendig da informantene ikke alltid forstod de åpne spørsmålene. Eksempelvis spurte jeg Kim og Sam om hvordan de angrep oppgavene:

*I: I starten på opplegget, hvordan angrep dere oppgavene?*

*Kim: Ja, husker du?*

*Sam: Jeg tror vi bare tenkte på hvilke bevegelsesligninger vi måtte bruke, så definerte vi de vi trengte. De er egentlig oppgitt i boken, så jeg tror vi bare la de inn. Og så skrev vi inn formelen, og så ble det ordna.*

*I: Omformet dere bare formelen i hodet?*

*Kim: Hvis det var den fjerde, gjorde vi det sikkert på ark.*

Informantene trenger et oppfølgingsspørsmål for å komme på hvordan de implementerte bevegelsesligningene fra læreboken til koden sin. Slike oppfølgingsspørsmål var ofte nødvendig, da de første spørsmålene ofte viste seg å være vanskelig for informantene



å tolke og videre svare på. Dette kan medføre at spørsmålene blir mer ledende, som belyst tidligere i delkapittelet svekker troverdigheten til resultatene.

Triangulering i metode for datainnsamling er en faktor som kan øke troverdigheten til besvarelsene (Robson & McCartan, 2016, s. 147, 179). Hvis en informant i intervjuet for eksempel sier noe generelt om opplegget, kan dette sjekkes opp mot de andre intervjuene, samt opp mot spørreundersøkelsen for å sjekke om dette var noe som gjaldt flere enn bare denne ene informanten. I tillegg er transkripsjonene av intervjuene og alt av analysearbeid med etterfølgende resultater diskutert og gjennomgått med veiledere, en forskertriangulering som kan minke risikoen for forsker-bias (Roberts & Priest, 2006). Jeg oppdaget allerede i utføringen av undervisningsøkten at jeg hadde stor tro på oppgavene jeg hadde utviklet. Det ble derfor vektlagt at jeg i utførelsen av intervjuene måtte forholde meg upersonlig til undervisningsopplegget, med spørsmål av en nøytral art. Dette ble gjort ved å spesifisere bakgrunnen for hvert spørsmål i intervjuguiden (se Vedlegg E), og i ettertid analysere spørsmålene som faktisk ble stilt i intervjuet. Som sitatet ovenfor viser, ble spørsmålet om oppgave 3a var vanskelig, ansett som ledende da jeg fra spørreundersøkelsen hadde en anelse om at elevene generelt synes oppgave 3a var utfordrende, men det er ikke dermed gitt at Alex synes den var vanskelig.

Et annet tiltak som kan øke graden av troverdighet, er *negativ analyse*. I dette begrepet ligger det å inneha et kritisk blikk, og analysere og tolke datamaterialet i motsatt retning enn det som anses som ønskelig. Selv om dette gir motstridende resultater, er det med på å gi forskeren en større innsikt i hva som tolkes og hvordan det tolkes (Robson & McCartan, 2016). Bevisstheten rundt egen tolkning er blitt presentert i tidligere avsnitt, da jeg stadig forhørte meg med veiledere om resultatene bar preg av mitt forskerbias. Det ble ikke gjennomført en dedikert negativ analyse, men alle funn ble individuelt sjekket om de bar preg av overtolkning.

Ved å belyse utfordringene av studiens overordnede design, kvantitativ metode for datainnsamling, forsker- og lærerrollen, er ønsket å gi leseren en best mulig innsikt i hvilke utfordringer som burde tas i betraktning om ønsket for å gjennomføre lignende studier skulle finne sted. Kvalitativ forskning medfører alltid en form for tolkning som kan svekke troverdigheten til studien (Johnson, 2013; Robson & McCartan, 2016). Tar man den sosiokulturelle læringsteorien i betraktning, kan det argumenteres for at et likt kvalitativt intervju ikke kan medbringe nøyaktig samme resultater, da kunnskapen fra intervjuet er

avhengig av den sosiale og kulturelle konteksten både jeg som intervjuer, men også informanten medbringer. Nyanser i min tolkning i analysen av datamaterialet vil sannsynligvis også skille seg fra andre sin tolkning av det samme datamaterialet. Med de metodiske grepene tatt i betraktning, mener jeg at analysen er gjennomført på en troverdig og transparent måte og at resultatene som presenteres er gyldige og forsvarlige.

## 5 Resultater

I dette kapittelet vil resultatene med tilhørende analyser fra intervjuene presenteres. Kapittelet vil introduseres med funn fra spørreundersøkelsen som påvirket intervjuene. Deretter presenteres funn fra intervjuene. Resultatene er organisert etter de to hovedtemaene fra analysen: *Elevenes arbeid med problemløsning (H1)* og *elevenes oppfatninger om opplegget generelt (H2)*. Disse temaene skal sammen belyse de ulike forskningsspørsmålene fra elevenes perspektiv, med hovedvekt på forskningsspørsmål 3: «Hvilke erfaringer hadde elevene tilknyttet undervisningsopplegget?» (se Figur 13).

### 5.1 Elevenes arbeid med problemløsning

Hovedtemaet *elevenes arbeid med problemløsning (H1)* inneholder elevers betraktninger om konkrete oppgaver, da i form av innhold, løsning og løsningsmetode. Delkapitlene tar stegvis for seg elevenes møte med utfordringer, derav hvilke oppgaver som utfordret dem og hvorfor, etterfulgt av hvordan de arbeidet mot en løsning på utfordringene. Hovedtemaets formål er å belyse forskningsspørsmål 1 og 2 fra elevenes perspektiv. Elevenes arbeid med problemløsning skal også dekke deler av begrepet *erfaringer* fra forskningsspørsmål 3, da med vekt på *utfordringer* (se Figur 13).

#### 5.1.1 Elevenes møte med numeriske utfordringer

Det ble som tidligere nevnt satt av 20 minutter til gjennomføringen av spørreundersøkelsen. Besvarelsene ble levert på nettskjema.no, slik at ingen personopplysninger ble samlet inn. Av de 22 som deltok i undervisningsøkten, svarte alle på spørreundersøkelsen. 11 elever hadde Teknologi og forskningslære som valgfag, noe som betød at de hadde programmeringserfaring i andre språk enn Python. De resterende 11 hadde ikke dette valgfaget, så all programmeringen de hadde fått i undervisningssammenheng kom primært fra matematikkundervisningen. Tabell 7 viser en oversikt over hvilke oppgaver elevene gjennomførte.

| Hvilke oppgaver fullførte du? |    |    |    |   |   |
|-------------------------------|----|----|----|---|---|
| Oppgave                       | 1  | 2  | 3  | 4 | 5 |
| Antall elever                 | 21 | 20 | 20 | 8 | 1 |

Tabell 7: En oversikt over hvilke oppgaver elevene gjennomførte.

Fra tabellen kommer det frem at det var mange elever som fullførte oppgave 1, 2 og 3, men færre som fullførte oppgave 4, 5. Det er kun 1 person som fullførte oppgave 5, noe som tyder på at opplegget var omfattende nok for de fleste. Dette forsterkes av at ingen elever har svart «For mye tid» på spørsmålet om hvor mye tid de hadde til oppgavene (se Vedlegg D). Når resultatene fra spørreundersøkelsen var klare, var det interessant å undersøke nærmere hvorfor så mange elever fullførte oppgave 3, men ikke mer.

Under intervjuene fikk alle informantene elevversjon av oppgavene og løsningsforslag i papirform, slik de kunne referere til oppgavene. Den ene informanten, Alex, hadde ikke Teknologi og forskningslære som valgfag, men hadde programmert relativt mye på fritiden siden ungdomsskolen. Hen syntes at programmering er artig, og var glad i å utfordre seg selv i fysikk- og matematikktimene. Alex ble spurt om det var noe hen synes var lett i opplegget.

*Alex: Jeg pløyde gjennom alt frem til oppgave 3, men bakgrunnen min hjelper jo på. Så da går det litt raskere, men oppgave 3 krevde litt mer tenking.*

Alex sier her at oppgave 3 var den første oppgaven hvor hen møtte på en utfordring. Dette kan forklare hvorfor så mange har arbeidet med oppgave 3, men ikke oppgave 4. Dani forsterker denne oppfatningen da hen svarer følgende på hvor hen møtte på et problem:

*Dani: Der var det [3a]. For resten var ganske intuitive, men sånn med lister. Vi hadde kanskje litt av det i fysikk, men jeg tror det er få folk som husker det.*

I likhet med Alex, sier Dani at han møter på en utfordring først på oppgave 3. Hen synes listene var vanskelig å forstå. Dani utbroderer ytterligere utfordringer med oppgaven:

*Dani: Og så skjønnte jeg ikke alt av koden, men det trengtes kanskje ikke heller. Du skulle vel bare skjønne sann grovt hva koden gjorde for at du kunne redigere den. Men nøyaktig hva du gjorde her, det står for så vidt forklart da, men jeg hadde ikke klart å ordne det selv. Eller å endre masse på den.*

Dani sier hen brukte tid på å skjønne den forhåndsgitte koden på oppgave 3a (se Figur 6). Selv om den forhåndsgitte koden kommer med forklarende kommentarer for hva som skjer for hver setning, brukte Dani tid for å kunne bruke koden for videre utfylling av linje 17-19.

Oppgave 3a er den første oppgaven der bruken av numeriske metoder kommer til sin rett, da elevene må iterere seg gjennom listene for å generere data fra fysiske formler. Kim og Sam ble intervjuet sammen og forteller at de brukte læreboken Kraft 1, som forklarer et lignende eksempel på side 79-80 (Fossum et al., 2021).

*I: Hvordan fikk dere koden fra eksempelet i boka og ned på datamaskinen?*

*Kim: Det var nesten bare å kopiere.*

*I: Ja, men da ble det noe feil?*

*Kim: Ja, det ble noe feil med de i-ene.*

Problemet med i-ene ble ikke oppdaget med en gang, men etter å ha spurt lærer om råd og etter å ha tenkt litt, skjønnte Kim og Sam at det var itereringen og indekseringen som var problemet. Selv om koden fra læreboken ligner, er indekseringen til listene forskjellig. Dette ble gjort for at elevene ikke skal kunne kopiere koden direkte, men være nødt til å sette seg inn i hva koden skal gjøre. Kim og Sam har prøvd å kopiere koden for å se hvilken feilmelding som kommer opp, og det er nettopp indekseringen i listene i for-løkken som ble feil for Kim og Sam. Den samme utfordringen forteller Alex om, da hen blir spurt om hen fikk mange feilmeldinger i løpet av opplegget:

*Alex: Ikke på oppgave en og to, men på oppgave tre fikk jeg en error-melding jeg ikke er vant til. Da fikk jeg «invalid value» [Informanten mener mest sannsynlig «Index error»]. Det er jeg ikke vant til å se.*

Spørreundersøkelsen viser at oppgave 3 var en mer utfordrende oppgave enn de foregående. Utdragene fra intervjuene bekrefter resultatet fra spørreundersøkelsen. Intervjuene viser at en grunn til dette er at oppgave 3a er sammensatt. Dani bruker tid på å skjønne den forhåndsgitte koden for å forstå hvordan listene fungerer. Dette innebærer både å forstå hvordan for-løkken itererer seg gjennom listene, i tillegg til hvordan indekseringen av listene fungerer. Selv om alle informantene navngir forskjellige utfordringer, henger utfordringene sammen med hverandre. For å få en kode som vil kjøre, må man forstå både indeksering og iterering, i tillegg til at kodespråket Python må skrives korrekt.

Nye problemer kommer til syne i senere oppgaver

Undervisningsopplegget (se Vedlegg A og B) bestod av oppgaver som bygget videre på tidligere kode. Et eksempel på dette var oppgave 3b, der elevene skulle visualisere sine genererte data fra oppgave 3a.

*Alex: Da begynner mysteriet på 3b. Når du har blitt så glad at du har fått til 3a, og så er grafen din feil.*

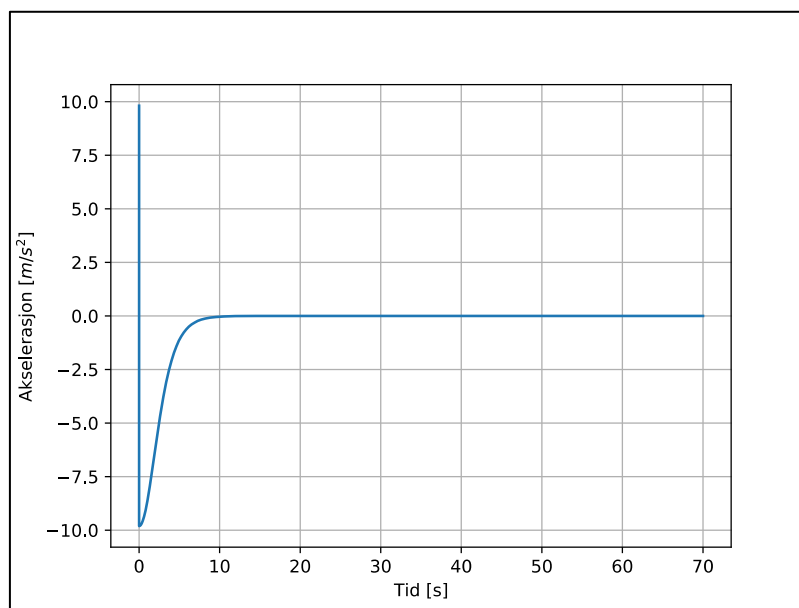
Sitatet ovenfor er sagt etter Alex har gått i dybden på hvordan hen arbeidet for å løse feilmeldingen på oppgave 3a. Alex har kanskje tatt mye av det hen har skrevet for gitt, så lenge koden kjører. Når hen kommer til oppgave 3b og får plottet grafen, så ser hen at noe ikke stemmer med dataen fra oppgave 3a (se Figur 14). Dani gir også uttrykk for at grafene ikke stemte.

*Dani: Når jeg kom til neste oppgave [3b] så jeg at farten hans går jo nedover når han faller. Så man fikk først vite etter oppgaven [3a] at det var feil, så måtte man tilbake.*

Det er først i oppgave 3b man får visualisert sin genererte data. Her får elevene mulighet til å kontrollere at dataen de har generert, gir mening. Problemet blir identifisert på følgende måte:

*Sam: Det var litt feil en periode, fordi det ble valgt motsatt positiv retning enn det vi hadde brukt før. Så jeg tror kanskje vi måtte endre noe her [3a] for at det skulle bli det samme som vi hadde gjort før.*

Sam sier at de definerte positiv retning nedover, mens den forhåndsgitte koden definerte positiv retning oppover. Når akselerasjonsgrafene plottes vil dermed et tilnærmet diskontinuerlig hopp skje fra  $-9.81 \text{ m/s}^2$  til  $9.81 \text{ m/s}^2$  (se Figur 14). De må da tilbake til oppgave 3a for å endre enten sin egen eller den forhåndsgitte koden.



Figur 14: En visualisering som viser en akselerasjonsgraf der den første verdien har motsatt fortegn i forhold til de andre verdiene. Grafen gjør da et hopp fra positiv til negativ verdi.

Ovenfor vises det en akselerasjonsgraf som er feil, fordi den første verdien til akselerasjonen har motsatt fortegn til de resterende verdiene. For at dette skal være tilfelle, må fallskjermhopperen først ha en akselerasjon i positiv retning, for så tilnærmet umiddelbart akselerere i negativ retning. Det er dette problemet informantene også kommer over. De innser at dette ikke stemmer og går tilbake til koden på oppgave 3a for å undersøke hva som skaper problemet.

### Elevenes utfordringer rundt representasjoner

Dette delkapittelet tar for seg resultater som omhandler utfordringer rundt fysikken i oppgave 3b (se Vedlegg A og B), og da mer konkret hvordan informantene hentet informasjon ut fra grafene som ble plottet.

*I: Hva gjorde du da, når du så at du hadde feil?*

*Dani: Jeg måtte jo gå tilbake da og prøve å tenke meg om. Så tror jeg at jeg hadde definert tyngdekraften negativt, eller den var definert negativt. Det var noe sånt. Så det var ikke noe krisefeil, men jeg måtte skjønne litt.*

Dani forteller om oppgave 3b og at oppgaven krevde en forståelse av fysikken for å skjønne hva som var problemet. Hen sier også at problemet i seg selv ikke var vanskelig å rette på. Dette indikerer at når Dani har satt seg inn i koden i oppgave 3a, så er det fysikken som gir videre utfordring. Indikasjonen forsterkes når Dani sier:

*Dani: På 3a, der brukte jeg lang tid på en oppgave som jeg i ettertid ikke synes var så vanskelig. For jeg skjønnte ikke hvordan liste-konseptet var.*

Etter hen har satt inn i koden en gang, så kan Dani bruke mye kortere tid på lignende oppgaver. Dette er ikke representativt for alle, men gir medhold i forklaringen på at utfordringen på oppgave 3b handlet om det fysikkfaglige.

Alex sier hen også støtte på utfordringen med motsatt definert retning. Hen sammenlignet kode med de andre, diskuterte og resonnererte med seg selv for å få et riktig svar. Alex bli så spurt om hvordan han kunne vite at hen hadde fått riktig graf.

*Alex: Den gav mening, fordi at akselerasjonen [mener farten] vil fortsette å stige til terminalfarten. Så de første grafene jeg hadde, spesielt den flate, vil ikke han holde konstant akselerasjon [mener fart] gjennom hele perioden.*

Alex resonnerer logisk og bruker sin kunnskap om fartsgrafer. Når startfarten  $v_0 = 0$ , må farten på et tidspunkt være 0 og bevege seg mot terminalfarten ( $v_{terminal}$ ), ikke være konstant gjennom hele hoppet. Alex fikk flere varianter av fartsgrafer og forteller videre:

*Alex: Og den som gikk motsatt, så mister du ikke fart når du faller i fritt fall. Så, da ser jeg at det er feil, og ved å se på de tidligere tallene jeg har fått, så klarer jeg å lage en forståelse om at «Det her er, om ikke riktig, så ligner det veldig.*

Alex fortsetter sitt logiske resonnement ved å bruke sin kunnskap fra fysikk og hverdagen for å bekrefte eller avkrefte om grafen gjengir situasjonen korrekt. I tillegg sier hen at hen bruker tall fra tidligere for å ytterligere verifisere sitt resultat. Alex vet ikke om det han har fått er helt riktig, men synes at fartsgrafen gir mening ut fra sin oppfatning om hvordan farten vil utvikle seg, i tråd med for eksempel slutfarten som hen regnet ut i oppgave 1b (se Vedlegg B). Av resultatene fremkommer det at utfordringene rundt oppgave 3b ikke lå i selve kodingen for grafene, men hvordan fysikken formidles i oppgave 3a. I likhet med Alex, Dani, Kim og Sam, undersøker vi oppgave 3a på nytt i neste delkapittel.

### Når fysikk, matematikk og programmering møtes

Alex blir spurt om hvordan hen tenkte for å løse indekseringsproblemet i oppgave 3a. Hen forteller at hen går til toppen av programmeringscellen og undersøker om det er skrevet noe



feil og at det som er skrevet gir mening. Alex bruker også feilmeldingen som peker på hvilken linje det er problemer med. Videre forteller Alex hvordan hen tenkte:

*Alex: Så får jeg at jeg har en feil verdi, og da tenker jeg at visst det står at maks-verdien er 700 000 og jeg ikke har satt maks-verdi på 700 000. Så må grunn-verdien være 0 og maks grunn-verdien er 700 000, så da tenker jeg at jeg på et eller annet vis har overskrevet denne verdien. Så det er da jeg kommer i en diskusjon med de bak meg og så kommer vi frem til at vi starter med 1 og ikke 0. [...] Så det var slik vi kom frem til  $i+1$ .*

Alex gir her en innsikt i hvordan for-løkken itererer seg gjennom listene. Hen sier at listene har plass til 700 000 tall, og at tellingen i for-løkken starter på 1, ikke 0 som hen først trodde. Det er dette som er opphavet til problemet, da Alex forsøker å sette inn et siffer på plass nummer 700 001 i en liste med kapasitet til 700 000. Gjennom diskusjon med medelever og tips fra lærer beveger de seg stadig mot en løsning på problemet. Kim forklarer samme problem med tilhørende løsning:

*Kim: Det var noen lignende formler som var i boka, så vi skrev ned dem, men så måtte vi plusse på «1» på alle de, for det var på en måte en «i» bak.*

Dani gir de samme kodelinjene et fysisk perspektiv:

*Dani: Du ser jo at strekninga i dette tidspunktet er lik strekninga i forrige tidspunkt, pluss farten i forrige tidspunkt. [...] Men,  $a[i]$  var litt vanskeligere, for da måtte jeg begynne med en formel som jeg fant i boka.*

Dani forklarer formlene på kodelinje 17-19 (se Figur 7) og hvordan formlene oppdateres for hver iterasjon i for-løkken ved bruk av verdiene fra den forrige iterasjonen. Det er denne numeriske tilnærmingen Alex, Kim og Sam også får utfordringer med. Kim og Sam identifiserer problemet ved å si at det ble «kluss med i-ene» og at de måtte «plusse på 1», Alex forklarer hvordan datamaskinen vil plassere disse verdiene, mens Dani forklarer hvordan verdiene genereres. Alle informantene forteller om samme utfordring, men resonnementene stammer fra forskjellige disipliner, følgelig programmering, numerisk matematikk og fysikk.

Undertemaene som har blitt presentert så langt, har omhandlet elevenes møte med utfordringer. Oppgave 3a var et skille i grad av utfordring for elevene, der de måtte inneha kunnskaper både om programmering, numerisk matematikk og fysikk. Første gang elevene arbeidet med oppgave 3a, handlet problemene stort sett om det programmeringstekniske.

Visualiseringen av dataen på oppgave 3b førte til at elevene på nytt måtte tilbake til oppgave 3a. Denne gangen handlet problemene mer om den numeriske matematikken og fysikken og hvordan det implementeres i koden. Resultatene har som sagt belyst møtet med utfordringene, mens de neste delkapitlene skal belyse elevenes strategi for problemløsning og metode for arbeidet med oppgavene.

### 5.1.2 Elevenes strategi for problemløsning

Elevene brukte ulike strategier for problemløsning. Den overordnede arbeidsmetoden var samarbeid i *par* og *grupper*, men det finnes flere konkrete strategier for problemløsning som inngår i samarbeid. For eksempel kom det frem fra alle intervjuene at både Alex, Dani, Kim og Sam hadde brukt papir til å forberede ulike problemer før de skrev de ned i programmet. Alex og Dani omformet bevegelsesligningene slik at de gav de parameterne de ønsket. Kim og Sam brukte papir for å visualisere hvordan listene i oppgave 3a så ut konkret. Dette viser at muligheten til å bruke kjente hjelpemidler som papir, kan være et steg på veien å forenkle programmeringen. De neste delkapitlene tar for seg flere strategier for problemløsning.

#### Bruke programmet til å ta en fysisk slutning

Etter oppgave 4a, der elevene regner ut terminalhastigheten Tore har etter at fallskjermen er utløst, skal elevene i oppgave 4b anbefale et tidspunkt for Tore å løse ut sin fallskjerm. Dani begrunner sin slutning slik:

*Dani: Jeg fant ut når han datt med en fart slik at han ikke skulle få så mye rykk på seg. Så jeg regnet ut sånn ca. når han fikk en fart. For med fallskjerm så faller han med så og så stor fart, så venta jeg bare til han kom til den farten og så skulle han trekke ut fallskjermen. Det ble litt tidlig da men, det var nå sånn jeg løste oppgaven*

Alex snakker også om oppgave 4b:

*Alex: [...] Da fant vi ut hva terminalfarten hans var i det tidspunktet, jeg tror det var 6,26. Når du får det tallet, så ser du på fartsgrafen, så skal du prøve å finne ut når jeg ville ha sluppet fallskjermen. Det blir jo veldig nøyaktig da.*

*I: Hva valgte dere da?*

*Alex: Vi valgte på 6,26 sekunder, for da vil du ikke merke endringen fra fallet ditt til fallskjermen. Det er jo ikke så lett da men.*

Både Dani og Alex har samme tanker, da de ønsker at Tore ikke skal merke overgangen fra ingen fallskjerm til fallskjerm. Oppgaven forutsetter at du har anerkjent resultater fra tidligere oppgaver og klarer å knytte dette opp mot hverdagslige preferanser. Alex tar i intervjuet et ytterligere steg bak, og sier at det å løse ut fallskjermen på et så nøyaktig tidspunkt, momentant, ikke er lett å gjennomføre i praksis. Sitatene viser at elevene klarer å knytte oppgaveløsningen og programmeringen til en fysisk virkelighet.

### Koden er ikke begrenset til en deloppgave

På den foregående oppgaven (4a) skal elevene regne ut terminalfarten med en ny luftmotstandskoeffisient (se Vedlegg A). Alex løste denne oppgaven slik:

*Alex: [...] Så da gikk vi videre dit og brukte samme utregning i oppgave 2 som i 4, bare at vi endra verdien for luftmotstandskoeffisienten. Da fant vi ut hva terminalfarten hans var i det tidspunktet [...].*

Alex sier at det eneste hen trengte å endre for å løse oppgaven var å kopiere koden fra oppgave 2 og erstatte verdien for luftmotstandskoeffisienten. Dette svaret bruker hen videre for å avgjøre tidspunktet Tore skal løse ut fallskjermen sin. Alex har med andre ord oversikt over tidligere kode og skjønner hvilken kode som er relevant å bruke på nytt.

*I: [...] Hvordan arbeidet dere for å løse oppgavene?*

*Kim: Først så må man definere bokstaver og sånt, sånn man kan bruke de videre. Og så ble det bare å sette inn formler som vi har fra fysikken, egentlig. [...]*

Kim snakker her om andre situasjoner der hen har skrevet en kode som kan nyttes i en ny sammenheng senere. Kim starter oppgaven med å definere konstanter tilknyttet variabelnavn. Disse konstantene nytter hen senere i for eksempel bevegelsesligningene. Selv om oppgaveteksten ber elevene om å definere konstanter i starten, har Kim skjønnet hvorfor dette gjøres, nemlig at tidligere skrevet kode ikke er nødvendig å gjengi hver gang. Dette viser innsikt i programmeringens natur og programmeringspraksis.

## 5.2 Elevenes betraktninger om opplegget generelt

Dette hovedtemaet tar for seg betraktninger angående faktorer som var gjeldende under hele opplegget, som for eksempel samarbeid, ekstern veiledning og momenter i undervisningsopplegget elevene ser forbedringspotensialet i. Elevenes betraktninger om opplegget generelt skal dekke den andre delen av begrepet *erfaringer* fra forskningsspørsmål 3, da med vekt på *oppfatninger*.

### 5.2.1 Elevsamarbeid

Fra intervjuene kommer det frem at samtlige informanter har samarbeidet i sine respektive *par*. Kim og Sam ble bedt om å samarbeide, da de kun hadde en datamaskin til rådighet. I den muntlige introduksjonen ble det oppfordret til samarbeid, noe både Alex og Dani valgte å gjøre med sin partner, altså i *par*. De samarbeidet også med andre par, noe som vil bli referert til som *grupper*. Fra observasjoner gjort i timen var det kun en person som ikke åpenbart samarbeidet med noen, da hen ønsket å sitte alene. Vedkommende ble ikke intervjuet. I intervjuene blir informantene spurt om hvordan det var å samarbeide i par og grupper. Hvert sitat er hentet fra ulike intervju, bortsett fra Alex sine utsagn som er hentet fra to forskjellige tidspunkt fra det samme intervjuet:

*Alex: Det var veldig greit.*

*Sam: Det var greit å samarbeide, men det hadde vi kanskje gjort uansett.*

*Alex: Og så sitter jeg med mange kloke hoder. Så da er det jo greit å få snakket sammen og diskutert litt.*

*Dani: Det fungerte greit. Det å sitte med en person gjør at det ikke blir like kjedelig når du kan diskutere eller spør. Da blir det ikke helt stille i klasserommet.*

Alex gir uttrykk for at hen likte å samarbeide og sier at hen generelt diskuterte mange oppgaver i grupper. Sam sier at selv om hen og Kim ikke var nødt til å samarbeide, så hadde de kanskje samarbeidet likevel. Dani sier i sitt intervju at ved å ha en sidekamerat å snakke med, gjør at timen føles mindre kjedelig. Dani utdyper hva hen mener i en annen del av intervjuet, der han forklarer at det ble mye «stillesitting». Dani blir så bedt om å utdype hva hen mener med begrepet «stillesitting».

*Dani: [...] For vanligvis så får vi bare beskjed om å gjøre oppgave 9.6 til 9.20. Så sitter du der i tre skoletimer i nesten total stillhet. Du spør partneren din av og til, men for det meste sitter du bare for deg selv og skriver og rekker opp hånda.*

Dani sier at hen gjerne ønsker å arbeide i et mer dynamisk arbeidsmiljø enn det tradisjonell oppgavearbeid ofte tillater. Selv om dette opplegget kun består av oppgavearbeid, opplevdes samtalen som oppstod med og rundt Dani i timen som bedre enn stillhet.

### Samarbeid rundt en datamaskin

For Kim og Sam som kun hadde en enkelt datamaskin på deling, ble dynamikken kanskje annerledes enn for andre par. Jeg var interessert i å vite mer om tilgangen på kun en datamaskin gav noen fordeler:

*I: Siden dere bare hadde en pc; følte dere at det var en fordel eller en ulempe?*

*Kim: Det var vel en ulempe, vil jeg si i hvert fall.*

*Sam: Ja, du fikk jo i hvert fall ikke gjort det sjøl.*

*Kim: Ja, for når man taster det selv, så blir hjernen mer aktivert, føler jeg da. For da ser man alt man skriver og får oversikt.*

Kim gir uttrykk for at hen gjerne skulle programmert alt selv for å ha vært mer aktiv i prosessen. Kim blir så stilt et oppfølgingsspørsmål om hvorfor hen ønsker å skrive kode selv:

*Kim: Man husker det bedre når man skriver det selv. I tillegg så har man hele koden foran seg, man trenger ikke å se på skriveboka si, så se på pc-skjermen til venstre og tilbake til skriveboka.*

Ved å skrive koden selv mener Kim at hen får bedre oversikt, at man husker det man lærer bedre, i tillegg til at arbeidsprosessen blir mindre oppstykket. Alex blir i sitt intervju spurt om hvordan hen synes det var å samarbeide, men fortsatt skrive på hver sin pc:

*Alex: Jeg synes det fungerte bra. Man føler jo litt mens man holder på, at du ser at du får det til. Så det er jo en mestringsfølelse i det [...], at du klarer å finne ut av det selv om du ikke har gjort det så mye før.*

Alex mener at en fordel med at hver enkelt elev programmerer på sin egen pc, er at man vil kunne få en større mestringfølelse når man faktisk har kodet noe som fungerer, i motsetning til å se på noen programmere. Alex blir videre spurt om hvordan hen tror det hadde fungert om elevene parvis skulle ha samarbeidet på en datamaskin:

*Alex: Det tror jeg spriker veldig fra par til par. Fordi noen er på forskjellige nivå når det kommer til fysikk- og/eller kodeegenskaper. Hen [en elev] som sitter bak meg er veldig god på koding, hen [en annen elev] som sitter bak meg er veldig god på fysikk. Så de tror jeg kunne laget en god duo.*

Ifølge Alex er et slikt samarbeid avhengig av parets individuelle egenskaper. Et samarbeid kan fungere godt, men er avhengig av at noen kompetanser hver enkelt elev besitter komplimenteres av andre kompetanser hos andre elever.

### Samarbeid fører til likere kode

Elevene samarbeidet tett med også med andre par, eller grupper. En av observasjonene jeg gjorde underveis var at mange elever snudde seg til paret bak eller foran seg for å diskutere. Noen elever gikk også vekk fra sine plasser for å diskutere med andre grupper. Dette nevner Dani i sitt intervju:

*Dani: Nei, jeg spurte folk rundt. Gikk og så på noen som heller ikke hadde fått det til. Til slutt så kom du til en av de, og da spurte jeg hva de hadde gjort og så prøvde de å forklare det.*

Dani sier at et resultat av det hen kaller for «stillesitting», et begrep som tolkes som fysisk stillesitting uten faglig progresjon eller sosial interaksjon, var at hen gikk rundt til andre par og grupper for å forhøre seg om de hadde forslag til løsning. Når jeg som lærer i klasserommet hadde veiledet et par, fikk Dani høre fra disse hva de nå tenkte.

Når Alex forklarer hvordan hen løser sitt fortegnspørsmål med tyngdeakselerasjonen, sier hen:

*Alex: [...] Så da fant jeg ut at jeg hadde satt +9.81, og at det var det motsatte av alle andre. Men jeg hadde en lik kode som alle andre. [...] Så, da gikk jeg litt tilbake og tenkte: «Okey. Nå ligner kodene våres veldig, men jeg har satt +9.81. Så hvis jeg prøver meg litt frem med fortegn osv.».*

Alex sammenligner sin kode med andre, endrer den for å få den til å ligne enda mer på andres besvarelse som gir et fornuftig svar, for så å eksperimentere videre med fortegn. Kim og Sam blir spurt om de tror at gruppenes besvarelse på tvers varierte mye:

*Sam: Ja, eller vi endte i hvert fall opp med å gjøre det samme.*

*I: Tror dere det var stor forskjell i gruppa [mener hele elevgruppen]?*

*Kim: Nei.*

*Sam: Det var sikkert litt forskjell. Jeg tror at man generelt har gjort det samme, men kanskje med litt ulik rekkefølge. Stort sett det samme.*

I intervjuet med Kim og Sam kommer det ytterligere frem at det var flere par som konsulterte dem om løsningen på for eksempel oppgave 3a. Kim og Sam tror at par og grupper har endt opp med stort sett samme kode, men med små variasjoner. Kim og Sam sine påstander om at kodene er like på tvers av parene, underbygges av Alex sitt resonnement, der hen forteller at hen endrer koden sin for å få den til å se helt lik ut som de andre.

### 5.2.2 Veiledning utenfra gir ikke hele svar

Fra spørreundersøkelsens spørsmål om hvordan elevene oppsøkte hjelp, har de fleste svar at de enten oppsøkte en lærer, medelever og/eller internett for veiledning. Jeg ble derfor interessert i å finne ut mer om hvordan den eksterne veiledningen fungerte:

*I: Så du var ikke på internett?*

*Alex: Nei, ikke overordna. Litt, fordi vi søkte opp en eksempelkode, men den ligna ikke i det hele tatt. Men det var litt der vi fikk idéen om å ta  $[i-1]$  fordi at  $n=1$ . Fordi jeg hadde egentlig basert hele koden min på at det var tatt høyde for allerede. Så da fant vi ut at det var slik vi skulle gjøre det.*

Alex brukte ikke internett som sin primære kilde til veiledning, men fikk noe veiledning fra internettsøkene. Fra eksempelkoden de søkte opp, fikk de idéen om å hente verdier fra forrige iterasjon. Arbeidet med å implementere koden på en hensiktsmessig måte var fortsatt opp til Alex og samarbeidspartnerene.

*Dani: Men,  $a[i]$  var litt vanskeligere, for da måtte jeg begynne med en formel som jeg fant i boka.*

*I: Så du brukte boken der også?*

*Dani: Ja, det var en eller annen formel der tror jeg. For en lignende oppgave. Fordi  $s[i]$  og  $v[i]$  var ganske ok, og  $a[i]$  var det litt mer pes. For  $s$  og  $v$  er ganske like, bare med andre benevnelser.*

Dani forteller at hen brukte boken som et hjelpemiddel for å finne en eksempelkode til å finne en løsning til formelen for akselerasjonen  $a[i]$  i oppgave 3a. Oppgaven var lignende, men ikke lik. Dette er enda et eksempel på hvordan elementer fra eksempelkoder kan gi veiledninger på enkelte deler av koden. Dani forklarer også i intervjuet hvordan hen ser på eksempelkoder:

*I: Brukte du internett til andre ting?*

*Dani: Nei, egentlig ikke. For med mindre noen har gjort en helt lik oppgave, så er det litt vanskelig. [...] Så jeg søkte egentlig ikke noe mer opp, for det var oppgaver som ikke var så lett å søke opp.*

Dani har brukt læreboken (Fossum et al., 2021) og internett som veiledning, og mener at resultatet fra internettsøket ikke er av stor nok verdi, med mindre eksempelkoden er helt lik. Dette strider mot Alex sin oppfatning, som har brukt internett på samme måte som Dani har brukt læreboken.

Kim og Sam brukte internett til mer enn å hente eksempelkode:

*I: Brukte dere internett noe?*

*Kim: Jo, vi søkte opp litt de forskjellige kommandoene. Numpy skjønnte vi ikke hva var, og hvordan man får forskjellige ting opp.*

Internett ble også brukt til å søke opp programmeringstekniske uklarheter, noe som hjalp elevene med å forstå hva de ulike kommandoene fungerte. I tillegg til læreboken, samarbeidspartnere og internett, var det tre lærere tilstede i undervisningsøkten. Dani forklarer hvordan veiledningen var, og hvordan det fungerte:

*I: Hva fikk du ut av den hjelpen du fikk av lærerne?*

*Dani: Det var jo begrensa. Du prøvde i hvert fall å gi meg en del hint. [...] Men jeg fikk jo hjelp, og du gav meg ikke bare svaret, du gav meg litt sånne hint: «Det her er nesten rett.», «Du må gjøre det på den måten.».*



*I: Var det irriterende?*

*Dani: Ja, på det tidspunktet, men det er jo sånn man tenker at man lærer. Hvis jeg bare hadde fått svaret så hadde jeg vel bare gått videre til neste oppgave.*

På samme måte som ved bruken av internett og læreboken, gav ikke jeg som lærer Dani nødvendigvis svaret hen var ute etter, men kun noen veiledende råd. Selv om Dani kanskje synes det var irriterende å ikke få utlevert svaret, anerkjenner hen at det er en del av læringsprosessen.

Spørreundersøkelsen viser at elevgruppen har brukt flere eksterne kilder til veiledning. Informantene sier i intervjuene hvordan denne veiledningen foregikk. Selv om informantene har benyttet seg av forskjellige veiledende kilder, er bruken av de relativt like. Ingen har fått utlevert et svar direkte, men fått råd og inspirasjon som har blitt brukt til videreutvikling av sine løsninger.

### 5.2.3 Opplevd læringsutbytte

Mot slutten av hvert intervju stilte jeg informantene spørsmål om hva de føler de lærte av undervisningsopplegget.

*I: Hva følte dere at dere lærte av timen?*

*Kim: Blir mer kjent med kommandoene. Litt mer komfortabel med programmering.*

*Sam: Komfortabel med programmering vet jeg ikke om jeg skal si, men følte at det ble litt bedre enn det ble.*

Kim og Sam føler de selv har blitt litt mer komfortable med programmering og hvordan ulike kommandoer fungerer. Det kommer frem fra intervjuet at de føler de hadde kontroll på fysikken opplegget omhandlet på forhånd, så det opplevde læringsutbyttet ble hovedsakelig programmeringsteknisk.

*I: Hva følte du at du lærte av timen?*

*Alex: Å diskutere fysikk, vil jeg si. Diskutere det på en litt mer uvant måte. Diskutere på en måte kodepunkter, fordi vi er vant til å diskutere.. Nå som*

*vi har om termofysikk og litt klimaendringer. Så vi er jo vant til å diskutere mer på teorisiden, ikke på programmeringssiden. [...]*

Alex legger mer vekt på arbeidsmetoden i sin forklaring. Hen mener at fysikken som ble diskutert under dette opplegget, er en annen type fysikk enn det hen er vant til. Alex skildrer diskusjonen som mindre teoretisk der man diskuterer kodepunkter. Sitatene viser at et undervisningsopplegg i fysikk hvor programmering står sentralt, ikke bare krever at elevene blir utfordret på å forstå og skrive kode, men også utfordrer elevene til å diskutere problemløsningen og fysikken på nye måter.

*I: Hva føler du at du lærte av timen?*

*Dani: Jeg fikk «refresha» litt tema som vi ikke har hatt på lenge i fysikken. Og programmering. Jeg vil ikke si at jeg lærte så sykt mye nytt, for vi har hatt det meste av det her før, bortsett fra listeskriving og plotting.*

For Dani var timen en måte å få frisket opp materiale hen allerede følte hen hadde kontroll på. I tillegg til dette lærte hen hvordan lister og plotting fungerer.

Informantene ble avslutningsvis i intervjuene spurt om hva de synes kunne forbedres i opplegget, der samtlige ønsket flere tilgjengelige lærere for veiledning. Dette viser at elevene hadde et stort behov for ytterligere hjelp og at undervisningsøkten kunne i større grad lagt til rette for at alle elevene fikk tilstrekkelig veiledning.

#### 5.2.4 Opplevd relevans

I spørreundersøkelsen (se Vedlegg D) spørres det om hvor relevant timen følte for elevene. Det er stor overvekt i elever som svarer ja på dette spørsmålet (se Tabell 8). Elevene blir i det neste spørsmålet spurt om å forklare utsagnene. Elevene sine besvarelser inneholder blant annet resultater om at programmering hjalp for å forstå fysikken bedre, at det følte relevant for problemløsning, at man kunne bruke programmering i flere fag og i fremtidig jobb. Det var også besvarelser som hintet om at timen ikke var fullt så relevant for dem, da de ennå ikke var sikker på om dette var noe de kom til å få bruk for.

| Føltes timen relevant for deg? |                |
|--------------------------------|----------------|
| Svaralternativer:              | Antall elever: |
| Ja                             | 17             |
| Nei                            | 1              |
| Vet ikke                       | 4              |

Tabell 8: En oversikt over om timen føltes relevant for elevene.

Relevansen til programmering i fysikk 1 ble også et tema i intervjuene og informantene ble spurt om hvorfor de tror at elever skal bruke programmering i fysikk 1:

*Kim: Jeg gjetter at det ikke har en direkte tilkobling til fysikk [mener fysikk 1], men heller at det er viktig å kunne forstå programmering fordi det er en veldig viktig del av samfunnet. Som vi må forstå de tingene vi bruker, på en måte.*

Kim ser ikke nødvendigvis relevansen til fysikk 1, noe hen utdyper senere ved å si at fysikken de lærer i fysikk 1 ikke er så avansert at man behøver å benytte seg av digitale beregningsverktøy. Programmering føles uansett relevant for Kim, da samfunnet blir mer og mer preget av ulike verktøy og hjelpemidler som programmeres, der en kunnskap om programmering vil kunne gi en forståelse av de verktøyene vi omgir oss med i hverdagen.

Alex forteller tidlig i sitt intervju at hen har en del programmeringserfaring. Det blir derfor et naturlig å spørre hen hvilke tanker hen har om hvorfor det blir programmert i fysikk 1:

*Alex: Det er jo for eksempel at man korter ned regnetiden. For eksempel med grafene så slippe man alle mellomregningene. Men også så føler jeg at når man kommer til fysikken, så når man programmerer så lager man seg et program og man beholder det programmet. Det programmet går jo an å bruke på nytt, eller bygge på det til et større program. Så kan man bruke det samme programmet til å regne ut flere ting.*

Alex føler at tiden det tar å regne ut store regneoperasjoner for hånd, kortes ned om man programmerer. I tillegg ser hen nytten av å utvikle et program som hen kan bruke på nytt og videreutvikle.

Noen av elevene fra spørreundersøkelsen som har svart at timen føles relevant, angir varierende begrunnelser for hvorfor de mener dette. Det er også elever som har svart at timen

føltes relevant, men uttrykker det motsatte når de blir bedt om å begrunne dette svaret. Det fremkommer fra intervjuet med Kim og Sam, at Kim føler at fysikken de lærer nå ikke er avansert nok til å kunne få et konstruktivt utbytte av programmering som digitalt beregningsverktøy. Et resultat fra analysen av intervjuene og spørreskjemaet er derfor at timen ikke føles relevant for enkelte elever i fysikk 1, da elevene mener fysikken de lærer ikke krever numeriske programmeringsoperasjoner. Ifølge informantene er opplegget derimot relevant fordi de kan bruke og videreutvikle programmet ved en senere anledning, samt at samfunnet beveger seg i en digital retning som krever en dypere innsikt i programmering. I diskusjonskapittelet vil oppleggets relevans diskuteres opp mot relevant teori og statens direktiver.

## 6 Diskusjon

Kapittel 6 vil drøfte resultatene fra denne studien med henblikk på å besvare de tre forskningsspørsmålene for denne studien:

*F1: Hvorfor skal elever programmere i fysikk 1?*

*F2: Hvordan utvikle et undervisningsopplegg som bruker programmering som verktøy i fysikk 1?*

*F3: Hvilke erfaringer hadde elevene tilknyttet undervisningsopplegget?*

Forskningsspørsmål 1 og 2, som tidligere er gjort rede for i kapittel 2 og kapittel 3, vil bli diskutert ved hjelp av elevenes perspektiver og erfaringer og teori. Forskningsspørsmål 3 vil ikke bli diskutert for seg selv, men preger hele diskusjonen med formål om å belyse forskningsspørsmål 1 og 2 (se Figur 1). Etter dette vil en mer generell refleksjon rundt mulige forbedringspunkter ved undervisningsopplegget gjennomgås, etterfulgt av en lignende refleksjon rundt masterstudien som helhet. Avslutningsvis vil enkelte implikasjoner fra studien bli belyst.

### 6.1 Elevers perspektiver på programmering i fysikk 1

Numeriske metoder og programmerings tiltenkte rolle i norsk undervisning, samt teori om programmerings egenskaper har blitt presentert i kapittel 2. Odden et al. (2019) beskriver hvordan digital beregningskompetanse som helhetlig begrep har blitt så omfattende at en innføring og spesifisering av begrepet er nødvendig for fysikkdisiplinen. Hensikten med spesifiseringen er blant annet å synliggjøre programmerings rolle og relevans i fysikkundervisning. Programmerings relevans til fysikk er teoretisk etablert ut fra den digitale beregningskompetansen i fysikk, der man ser hvordan elever og studenter kan bruke praksiser, kunnskap og oppfatninger i arbeidet med programmering i fysikk. Arbeidet med digitale modeller for å utforske fysiske konsept er sentralt i Weintrop et al. sin taksonomi for algoritmisk tenkning (2016), en taksonomi som er beslektet til Utdanningsdirektoratet definisjon av den algoritmiske tenkeren (2019). Selv om teorikapittelet argumenterer for at relevante kunnskaper og egenskaper for fysikk 1 kan komplimenteres av programmering,

viser mine resultatet at dette ikke nødvendigvis er argumenter elever i fysikk 1 selv anerkjenner.

Det fremkommer i kapittel 5.2.4 at Kim sier at programmering er en viktig del av samfunnet som helhet, hvilket er bakgrunnen for det meste av sentral teori som er gjennomgått i denne studien (diSessa, 2000; Weintrop et al., 2016; Wing, 2006). Selv om Kim uttrykker denne relevansen, ser ikke hen parallellen til fysikk 1 som eksplisitt relevant. Så hvorfor skal elever programmere i faget fysikk 1, ikke bare i andre fag som matematikk?

Utdanningsdirektoratet skriver at elever i fysikk 1 blant annet skal lære om hvordan teknologisk utvikling påvirker individet og samfunnet, i tillegg til å opparbeide seg kompetanse innen analytisk tenkning og problemløsning. Fysikkfaget skal også bidra til at elevene ser en verdi i samarbeid og idéutveksling. Utdanningsdirektoratet nevner programmering under kjerneelementer, derunder praksiser og tenkemåter i fysikk, samt som en del av digitale ferdigheter som er relevant for fysikkelever (Utdanningsdirektoratet, 2021). Det er ikke å forvente at man i løpet av en undervisningsøkt tilegner seg alle relevante kompetanser og egenskaper i fysikkfaget (Kunnskapsdepartementet, 2017; Utdanningsdirektoratet, 2021), men undervisningsøkten kan gi et bidrag i denne retningen. Resultater fra informantene viser imidlertid at elevene har samarbeidet tett. Gjennom diskusjon, idéutveksling, prøving og feiling har elevene arbeidet med fysikkfaglige oppgaver ved hjelp av programmering.

En annen grunn til at enkelte elever kanskje ikke ser en direkte relevans til fysikkfaget, kan være at elevene ikke har programmert nevneverdig i fysikkundervisningen tidligere. Når elevene i tillegg har gjennomgått bevegelsesligningene og formelen for terminalfart med luftmotstand tidligere på høsten ved bruk av analytiske arbeidsmetoder, kan dette forsterke inntrykket av at programmering og numeriske metoder ikke er nødvendig for å oppnå videre fysikkfaglig kompetanse. Ingen av informantene anerkjente at oppgave 3a krevde en numerisk løsning. Det kan være at en presisering under selve undervisningsøkten om at oppgave 3a nettopp krever en numerisk tilnærming kunne gi elevene en mer opplevd relevans og dermed rettferdiggjøre bruken av programmering i fysikk 1 undervisning for elevene selv. På den andre siden, ved å ikke presisere dette fikk man muligens mer ærlige og selvproduserte resonnement fra informantene i intervjuene, ikke en mening utenfra som de har blitt bedt om å videreformidle.

Dani sier at hen fikk frisket opp fysikk de hadde lært tidligere når det spørres om hvilket læringsutbytte hen fikk av undervisningsopplegget. Utsagnet fastslår at informanten har arbeidet med fysikk. Dani snakker om utfordringene rundt det programmeringstekniske i store deler av intervjuet, men mener fortsatt at noe av læringsutbytte var repetisjon av fysikken. Dette viser at til tross for programmering som aktivitet, så har elevene oppholdt seg i fysikkens verden. Figur 2 viser domenet den digitale beregningskompetansen i fysikk finner sted (Odden et al., 2019). Svaret til Dani viser at elevene har oppholdt seg både i det digitale domenet, i fysikkdomenet og i en kombinasjon av disse.

Oppsummert skriver Utdanningsdirektoratet hva elever i fysikk 1 skal lære seg av faglig innhold og hvilke egenskaper faget skal legge til rette for utvikling av. Programmering kan ifølge teori stimulere og bidra til denne faglige kunnskapen og egenskapene elevene i fysikk 1 skal opparbeide seg (Weintrop et al., 2016). Informantene selv ser ikke eksplisitt nødvendigheten av programmering i dette faget, men viser gjennom resonnement at programmering er viktig å lære seg. Resultatene viser også til at elevene har fått et faglig utbytte av undervisningsopplegget som er av relevans i fysikk 1. Det kan ikke forventes av informantene at de skal gjenkjenne de underliggende kompetanser og egenskaper (for eksempel vitenskapelig- og kritisk tenkemåte) som manes frem i fysikkfaget, og om programmering har bidratt med dette. For å belyse de underliggende kompetansene, burde flere inngående spørsmål blitt stilt i intervjuet. Elementene og arbeidsmetodene som har bidratt til en fysikkfaglig forståelse i dette undervisningsopplegget, vil i de neste delkapitlene diskuteres opp mot blant annet taksonomien for algoritmisk tenkning (Weintrop et al., 2016) og Sørby og Angells tre moduser (2012).

## 6.2 Fysikk, matematikk og programmering i samme kodelinje

Resultatene fra spørreundersøkelsen i denne studien antyder at de fleste hadde utfordringer med oppgave 3a (se Vedlegg A). Intervjuene frembringer resultater som bekrefter den samme utfordringen, men med ulike utgangspunkter, da med perspektiver fra matematikken, fysikken og programmeringen. En teori som er interessant å sette funnet opp mot, er Sørby & Angells innføring av matematikk-modus, fysikk-modus og programmeringsmodus (2012). Funnene fra denne masterstudien bekrefter at elevene på videregående i likhet med studenter på høyere nivå jobber i de samme tre modusene. Funnet angående oppgave 3a fremhever imidlertid modusenes tette og intrikate samspill. Elevene må i samme kodelinje kombinere kunnskaper om det programmeringstekniske, den numeriske matematikken og fysikken. Om dette er å

foretrekke i motsetning til å adskille de tre modusene er usikkert, men samspillet mellom flere moduser argumenterer for programmeringens relevans til fysikk. Resultatene viser at selv om noen elever gjenkjenner utfordringen og problemet som programmeringsteknisk, krever løsningen av problemet kunnskap innen både numerisk matematikk og fysikk.

### 6.3 Algoritmisk tenkning i praksis

Et interessant funn fra resultatene kommer fra Alex sitt utsagn om at hen føler hen har diskutert fysikk på en annen måte. Alex forklarer at de tidligere har diskutert en mer teoretisk preget fysikk. En tolkning av dette kan være at fysikkundervisning kan bestå av hypotetiske tankeeksperiment om abstrakte modeller som ikke manifesterer seg fysisk for elevene. Programmering kan være en måte å konkretisere abstrakte modeller med tilhørende begreper. Fra Weintrop et al. sin taksonomi for algoritmisk tenkning (2016) er nettopp modeller et sentralt begrep. Ved å videreutvikle, implementere og vurdere digitale modeller i fysikken, må elevene ta stilling til konkrete problemer og ta avgjørelser som vil utgjøre en forskjell i koden og videre den digitale modellen. Dette funnet støttes også opp av Sins, Savelsbergh og van Joolingen som sier at ved å samarbeide direkte med hverandre, får elevene en mulighet til å konkretisere sin digitale beregningsmodell, komme med eksempler og argumentere for eller i mot hvorvidt konkrete ideer fungerer eller ikke (2005).

Informantene forteller at når grafene på oppgave 3b var plottet, var det nødvendig å utbedre koden i oppgave 3a. Den første gangen de gjennomførte oppgave 3a, handlet de fleste utfordringene om det programmeringstekniske, mens den andre gangen var det grafen som gav elevene informasjon som gjorde at de ønsket å endre koden. Visualiseringen er med andre ord noe som kan få elevene til å revurdere sine digitale modeller, der elevene får brukt sin fysikkfaglige kunnskap til å avgjøre om koden genererer fornuftige data opp mot sin digitale modell.

Intervjuene bringer frem resultater som viser at den overordnede metoden for problemløsning for elevene ifølge informantene er samarbeid, da i form av diskusjon med medelever. Når elevene feilsøker og feilretter (Weintrop et al., 2016, s. 140), går eleven Alex systematisk gjennom koden sin, linje for linje, for å se om der er noe som ikke stemmer. Når hen så har gjort det og fortsatt får feil, diskuterer hen med medelever og sammenligner kodene for å finne en løsning. Et slik samarbeid har gitt Alex følelsen av å diskutere fysikk på en ny måte.



Både Alex, Dani, Kim og Sam forteller i sine intervju at de sammenlignet kode med andre par. For Dani kom deler av dette samarbeidet med andre grupper som en konsekvens av det hen kaller for «stillesitting», en situasjon der Dani ikke har progresjon i opplegget. Det er mulig at elever har sammenlignet kode med andre som kanskje har kommet lengre i oppgavene, bare for å komme seg videre i opplegget. Alex snakker også om at etter hen hadde gått systematisk igjennom koden sin, så endret hen koden til å ligne på koden til de rundt seg. En konsekvens av å endre koden sin uten å begrunne endringene i fysikken eller i det programmeringstekniske, er at oppgavene ikke gir et like stort læringsutbytte som det kunne ha gjort. Dette er igjen en mulig negativ konsekvens av det tette elevsamarbeidet.

Sørby & Angell (2012, s. 294) skriver at studentene ofte tyr til spontane og enkle løsninger og forklaringer for å rettferdiggjøre sin digitale modell. Når en feil oppstår, forekommer det en mer tilfeldig form for prøving og feiling for å prøve å rette opp koden. Resultatene fra informantene viser til at de hadde en plan i det de feilsøkte, men at de etter hvert gikk over til mer prøving og feiling om koden ikke stemte. Dette kan komme av at den kognitive lasten (Orban et al., 2018) blir for stor når elevene ikke har kontroll på det programmeringstekniske og samtidig må forholde seg til fysikk. Selv om feilsøking og feilretting er en praksis i taksonomien for algoritmisk tenkning (Weintrop et al., 2016), er ikke denne prøvingen og feilingen medregnet, og gir ikke det samme faglige utbytte som systematiske og bevisste feilsøkingstrategier. Det må videre tas i betraktning at samtlige informanter anså seg selv som kompetente i fysikkfaget. Feilsøkingen kan dermed ha vært preget av mer tilfeldig prøving og feiling hos andre elever. En løsning på problemet kan være å gi elevene en opplæring om hva som kan gjøres og hvordan de kan gå frem når de søker etter feil i koden. Problemet i oppgaven løste seg for mange elever ved å slette kjørehistorikken, da programmet vil huske gamle konstanter og variabler om man ikke overkjører de.

## 6.4 Programmering i fysikkundervisning: Et verktøy for læring eller læring av et verktøy?

Diskusjonskapittelet har så langt tatt for seg temaer om hvorfor elever i fysikk 1 skal programmere og hvordan arbeidet med programmering kan stimulere til ulik innsikt i fysikkfaget. Resultatene viser som tidligere nevnt at elevene har samarbeidet, diskutert, utviklet idéer, i tillegg til å formidle for- og motargumenter i et opplegg om fysikk gjennom programmeringsaktiviteter. Noen av utfordringene og diskusjonene har vært fysikkfaglige,

andre har handlet om det programmeringstekniske. Det kan virke som at fysikken og programmeringen på mange måter går hånd i hånd i dette opplegget. Informantene diskuterer i oppgave 3b med utgangspunkt i sin fysikkforståelse rundt grafen og hvordan fysikken formidles i programmeringen i oppgave 3a. På denne måten er programmeringen et verktøy for å fremstille fysiske fenomen og begrep. Oppgaveaktivitetene har også manet frem diskusjoner på en annerledes måte, noe som blir anerkjent av Alex i sitt intervju. Dette viser til at programmeringen er et verktøy for å gi elevene nye kompetanser i fysikkundervisningens øyemed. Da elevene selv kanskje ikke konkret kan formidle alle argumenter for eller imot bruken av programmering i fysikk 1, fremkommer det av spørreundersøkelsen og intervjuene at elevene stort sett ser verdien i å lære seg programmering generelt i fysikken og samfunnet. Selv om undervisningsopplegget har vist seg å kunne gi et læringsutbytte, er der flere elementer som kan forbedres og videreutvikles.

## 6.5 Forbedringsmuligheter rundt undervisningsopplegget

I utviklingsfasen av undervisningsopplegget ble det lagt opp til en tradisjonell tilnærming, noe som er diskutert i kapittel 3.2. Det ble bestemt at elevene skulle kode selv og få tilgang til koden som var forhåndsskrevet. Dette gjaldt også i oppgavene der dataen skulle visualiseres, fremfor en hybrid tilnærming som i større grad ville ha gjort koden for visualiseringen utilgjengelig for elevene (Orban et al., 2018). Dani snakker i sitt intervju om at hen brukte mye tid på å forstå den forhåndsgitte koden på oppgave 3a, selv om hen innser at hen ikke skulle endre på den. Dette resultatet strider imot min intensjon i planleggingen av undervisningsopplegget, da den forhåndsgitte koden (se Figur 6) ble presentert for elevene for å lette deres kognitive belastning. Det er selvsagt slik at når elevene skal sette inn verdier i listene som er forhåndsgitt, må de fortsatt forstå hvordan lister og for-løkker fungerer. Andre konstanter som ble oppgitt var for eksempel tidsstegene for hver iterasjon, noe informantene ikke nevner i sitt intervju. Dette antyder at deler av den forhåndsgitte koden ikke skapte utfordringer for elevene, eller at elevene bevisst eller ubevisst ikke forholdt seg til disse delene av koden. I begge tilfeller kan den forhåndsgitte koden ha bidratt til å redusere den kognitive belastningen noe.

Selv om den forhåndsgitte koden i oppgave 3a krever en forståelse av lister, er som tidligere nevnt tidssteget ( $dt$ ) for hver iterasjon ikke noe informantene spesifikt nevner at de har brukt tid på å forstå. Det ville nok gitt et større direkte fysikkfaglig utbytte for elevene å ta hensyn til tidsstegene. Samtidig er listene et programmeringsteknisk verktøy elevene faktisk

er nødt til å forstå for å få koden til å kjøre. Tidsstegene er ikke nødvendigvis det. Selv om elevene benytter seg av  $dt$  (se Figur 7), kan dette brukes uten å ta videre hensyn til hvor stort tidssteget er. En løsning på dette kan være å gi elevene en felles innføring av liste-verktøyet, for så å lage oppgavene slik at elevene må ta stilling til tidsstegene ved å ikke oppgi dette.

I oppgave 3b ble koden for å visualisere akselerasjonen presentert for elevene, og det ble lagt opp til at elevene skulle bruke denne koden for å visualisere fart- og posisjonsgrafene. Orban et al. (2018) velger i sin hybride tilnærming å skjule koden for visualisering. Grunnen til at koden i denne studien ikke ble skjult, var at elevene selv skulle ta stilling til hvilke data som skulle visualiseres, langs hvilken akse osv. Det var imidlertid en fare for at elevene ikke ville forstå koden og dermed bruke lang tid på å prøve å få grafene plottet. Fra resultatene kommer det frem at ingen av informantene har brukt tid på å få koden i oppgave 3b til å kjøre. Elevene fikk en graf som gav dem informasjon om forrige deloppgave. Om informantene har knyttet det programmeringstekniske opp mot fysikken slik som det var tenkt, er mer usikkert, men resultatene tilsier at å vise koden ikke var en stor utfordring i seg selv. Det er selvsagt annerledes for større simulasjoner og visualiseringer der flere variabler plottes på mer utradisjonelle måter. Da ville det ha vært mer hensiktsmessig å skjule koden for elevene. Plottene i dette undervisningsopplegget er 2D-plot som elevene er vant til å se i fysikk- og matematikkundervisningen, og det kan virke som at avgjørelsen om å ikke skjule koden, ikke medførte store negative konsekvenser i arbeidstid og kognitiv belastning. En annen grunn til å vise koden i oppgave 3b, var for å gi elevene en større innsikt i programmeringen bak et plot. Kunnskapen vil de forhåpentligvis kunne bruke i andre sammenhenger. Gjenbruken av kode anerkjenner blant andre Alex, når hen begrunner hvorfor hen føler de skal programmere i fysikk 1. Om koden hadde blitt skjult, hadde elevene åpenbart ikke fått sett koden og hvordan den fungerer. Når i tillegg informantene later til å ikke ha store utfordringer med den aktuelle koden, kan det konkluderes med at den tradisjonelle tilnærmingen på oppgave 3b kan være hensiktsmessig for elevene.

Når jeg spurte informantene om hva de tror kunne gjort undervisningsopplegget bedre, svarte alle at de ønsket flere tilstedeværende lærere. Dette viser at selv om det var tre tilgjengelige veiledere, så ble ikke behovet for veiledning tilstrekkelig dekket. Det er selvsagt ikke praktisk gjennomførbart i en lærerhverdag å være tre lærere i hver fysikktime, men det finnes andre tiltak som kunne blitt gjennomført for å lette på behovet for fysiske lærere. Jeg som klasseleder hadde ingen fellesgjennomgang utover en liten introduksjon, noe som kunne ha oppklart utfordringer flere elever slet med. Som tidligere nevnt kunne jeg for eksempel

introdusert ulike funksjoner og verktøy i Python, som lister og for-løkker. Oppgaveteksten kunne gitt enda mer konkret veiledning og bestått av flere mindre oppgaver. En viss veiledning og oppdeling av oppgaver ble gjort, da for eksempel i introduksjonsoppgavene (se Vedlegg A og B), der elevene stegvis blir bedt om å definere konstanter, for så å finne fart og tid uten luftmotstand, deretter terminalfart osv. Det viser seg at oppgave 3a kan deles opp i flere deloppgaver. Dette vil bli tatt med videre til neste gang jeg utfører et lignende undervisningsopplegg.

I studien fremkommer det at elevene selv ikke nødvendigvis ser bruken av programmering som nødvendig for å løse problemer i fysikk 1. Et forslag til endring har vært å presisere overfor elevene at for eksempel oppgave 3a kun kan løses numerisk, ikke analytisk. Tiltaket kan som tidligere nevnt påvirke elevenes inntrykk av opplevd relevans, noe som ikke var ønskelig i denne studien. Tiltaket er derimot mer hensiktsmessig i en vanlig undervisningssammenheng for å gjøre elevene oppmerksom over at oppgaven bruker numeriske metoder for løsning av et problem i fysikken.

## 6.6 Metodisk refleksjon

Programmeringen og dens rolle er utbredt og godt etablert i vitenskapen, og det har i lang tid blitt argumentert for at programmering burde inkluderes i undervisningen av vitenskapelige skolefag (diSessa, 2000; Papert, 1980). Til tross for at fagfornyelsen ble innført i 2020, har denne implementeringen skjedd i så ulik grad over hele landet, at en intervensjonsstudie gir et verdifullt bidrag og en innsikt til hvordan programmering faktisk kan foregå for elever med lite programmeringserfaring i undervisningssammenheng.

Problemstillingen har ble delt inn i tre forskningsspørsmål, der det første forskningsspørsmålet skal veilede forskningsspørsmål 2, som igjen blir testet i forskningsspørsmål 3. Det siste forskningsspørsmålet har deretter belyst de to andre i diskusjonskapittelet. Utdanningsdirektoratets direktiver har vært utgangspunktet for masterstudien. Videre hviler både undervisningsopplegget og studien på et teoretisk grunnlag, med mål om å besvare hvilke egenskaper ved programmering som kan fremme elevers kompetanse som er relevant for fysikk 1. Litteratursøket har bidratt til en bratt læringskurve, der jeg uten forkunnskaper har måttet lære meg søkestrategier for å finne relevant litteratur. Jeg har vært lovprist av gode veiledere som stadig har kommet med råd om litteratur og søkestrategier, men det er fortsatt sannsynlig at gode arbeider og studier som kunne gitt verdifull innsikt i teorien, har gått upåaktet hen.

Bakgrunnen for studien og ønsket om å få elevenes perspektiver stod sterkt gjennom hele studien og kan sies å ha farget arbeidet i stor grad. Det er nettopp elevenes perspektiv på overordnede tanker om programmering og hvordan de opplevde å arbeide med undervisningsopplegget som sørger for de mest verdifulle bidragene fra studien. Studiens design med forskningsspørsmålenes sammenheng til hverandre, gjør at elevenes perspektiver lett kan koples opp mot teori. En svakhet er derimot forskjellen i min metode satt opp imot metoden brukt i studier fra teorien. Metoden fra brorparten av litteraturen har brukt samtaler mellom studenter mens de har arbeidet med oppgaver, mens jeg har samlet inn data fra elever i etterkant av selve opplegget. Dette ble valgt da intervjuformen gir mulighet til å spørre elevene om hvordan de opplevde undervisningsopplegget, samt en mulighet til å få innsikt i hvilke utfordringer og strategier for problemløsning elevene benyttet seg av. En mulig forbedring av metoden for datainnsamling kunne være å inkludere lydopptak fra selve undervisningsopplegget og/eller samlet inn oppgavebesvarelsene fra elevene, i likhet med blant annet Weintrop et al. (2016). Mengden datamateriale og analysearbeid dette ville medført, kunne potensielt ha overskygget studiens egentlige formål; nemlig elevenes erfaringer. Dette leder oss inn annen konsekvens av metodevalgene, som er at teorien som danner det teoretiske grunnlaget, i stor grad handler om hvordan elever arbeider, ikke hvordan de opplever å arbeide på denne måten. Dette har ført til at analyseprosessen i masterstudien har måtte foregå i to dimensjoner for å knytte det hele sammen: En dimensjon for å se hvordan elevene arbeider og om dette kan stemme opp imot teori, en annen dimensjon for hvordan elevene opplever å arbeide på denne måten. Selv om dette har vært utfordrende, har det også gitt innsikt og kunnskap om hvordan teori om en målgruppe kan ses i lys av opplevelsene hos målgruppen selv.

De forskjellige dimensjonene i den tematiske analysen har krevd mye tid og omtanke (se Figur 12). Om man ser det tematiske nettverket opp mot Braun og Clarke (2006) sin oversikt over en god tematisk analyse (se Tabell 5), er et punkt at kodene skal være distinkte. Dette har vært en utfordring å opprettholde, da dataen forteller hvilke utfordringer elevene har opplevd, samtidig kan den samme dataen gi innsikt i hvordan elevene har opplevd å stå i disse utfordringene. En løsning på dette problemet kunne ha vært å tilstrebe en mer deduktiv tilnærming med forhåndsbestemte koder, noe som kan ha gjort det tydeligere hvilket datamateriale som skal kategoriseres i hvilken dimensjon. På en annen side er det mulig å overse interessante funn som dukker opp med en induktiv analyse. Siden teorien ikke belyste

elevenes perspektiver, var det vanskelig å forutse hvilke betraktninger elevene ville komme med. Dermed var en induktiv tematisk analyse å anse som mest hensiktsmessig.

Utvalget til intervjuene bestod av elever som anså seg selv som relativt kompetente i fysikk 1, ikke av elever som muligens befinner seg på et lavere kompetansenivå i fysikk. Et lite varierende utvalg er alltid en risiko ved bekvemmelighetsutvalg med frivillig deltakelse. Selv om utvalget ikke har en stor variasjon i kompetansenivå, viste resultatene en variasjon i andre former. Kim og Sam måtte for eksempel arbeide på en felles datamaskin, noe som muligens kunne gi andre utfordringer og løsninger. Elevene hadde stor variasjon i tidligere programmeringserfaring og generelt opplevde de utfordringene ulikt, noe som er blitt gjennomgått i resultat og diskusjonskapittelet. Dette viser at selv med et lite bekvemmelighetsutvalg, er variasjonen i underliggende variabler stor, og det er ikke utenkelig at elever med et lavere kompetansenivå i fysikk kan komme med lignende betraktninger. Det er selvfølgelig ønskelig å dekke et så stort utvalg som mulig, men informantene bidrar med refleksjoner og oppfatninger som er av stor verdi for studien.

Til slutt er min rolle som både forsker og lærer noe som må tas i betraktning. Det ble bestemt at det var jeg som skulle ha hovedansvar for undervisningsøkten, da jeg var den som utviklet opplegget og hadde best kjennskap til det. Dette ble sett på som fordelaktig, da jeg hadde god innsikt i hvilke utfordringer som kunne oppstå. De to andre lærerne fungerte som hjelpelærere. Dette fungerte godt under selve undervisningsopplegget, men jeg skulle ønske at jeg hadde en bedre relasjon til elevene i forkant av undervisningsøkten. Selv om en nærere relasjon kan påvirke elevenes besvarelser i en retning som kan svekke troverdigheten, kan det også gjøre det motsatte. Jeg hadde ingen kjennskap til elevene før undervisningsopplegget, og en tretimersøkt med oppgaveregning gir ikke mulighet til nevneverdig relasjonsbygging. Den manglende relasjonen kan ha ført til at de individuelle intervjuene ble preget av å bli kjent i startfasen for å skape en økt tillit til meg som forsker. Om tilliten og relasjonen er opparbeidet fra før, kan det øke graden av ektefølte resonnement fra informantene og da enda dypere datamateriale. Selv om jeg oppfatter at resultatene fra studien gir tilstrekkelig innsikt med tanke på å besvare forskningsspørsmålene, utgjør betydningen av relasjonsbygging overfor elevene et moment som jeg i en forskerrolle burde ha tenkt grundigere gjennom i forkant av intervensjonen.

## 6.7 Implikasjoner

Det er enkelte hensyn jeg ønsker å presentere for videreutvikling av opplegget og studiet som helhet. Diskusjonen har i tillegg til å reflektere over hvorfor og hvordan elever kan programmere i fysikk 1, gitt en innsikt i forbedringspotensialet både i undervisningsopplegget og studiens oppbygning og metode. Dette kan ses på som en tilnærming til å gjøre undervisningen mer relevant for elevene, men spørsmålet som gjenstår å stille er: Hvordan skal lærere undervise fysikk 1 med bruken av programmering som verktøy?

Det er verdt å nevne at et undervisningsopplegg lignende det som har blitt presentert i denne studien, er omfattende både for elevene som gjennomfører det, men også for lærere som skal planlegge og legge opplegget til rette for elevene. Om man bruker for eksempel dette opplegget, så krever det en kunnskap og tilnærming til elevgruppen det omhandler, og videre da muligens en endring i oppgavesettet. Selv om dette undervisningsopplegget skulle passe perfekt for en tenkt elevgruppe, må læreren fortsatt sette seg inn i oppleggets innhold og bredde. For eksempel viser resultatene at elever har forskjellige utfordringer innad i samme oppgave og adresserer de samme utfordringene på forskjellige måter. En lærer må derfor ha innsikt i hvilke problemer og utfordringer som kan oppstå og hvordan elevene kan veiledes gjennom disse utfordringene.

Undervisningsopplegget krever også 2,5 skoletimer på 45 minutter, men ingen krav til særegne programmeringstekniske forkunnskaper. Det er likevel å foretrekke at elevene har noe erfaring med programmeringsspråket de opererer i. I tillegg vil kunnskaper om for eksempel for-løkker eller lister være fordelaktig, da programmeringsverktøyene kan skape ytterligere kognitiv belastning. Blant elever som deltok i undervisningsopplegget, var det elever som opplevde de hadde mye programmeringserfaring, men som fortsatt ikke ble ferdig med alle oppgavene. Med andre ord er opplegget også beregnet for elever med relativt gode programmeringskunnskaper. For slike elever er det kanskje i større grad fysikkunnskapene som vil bli satt på prøve, fremfor det programmeringstekniske.

Dette undervisningsopplegget ble gjennomført innen et tema i fysikken som var kjent for elevene fra før. I fremtiden, når læreplanen (Utdanningsdirektoratet, 2021) er fullstendig implementert blant elever som forhåpentligvis er bedre kjent med digitale beregningsverktøy for problemløsning, kan slike undervisningsopplegg integreres i større grad som en del av innlæringen av ny fysikk. Dette vil være i tråd med blant annet Odden et al. (2019) sin studie, som argumenterer for digitale beregningsverktøy som en naturlig del av fysikkundervisning.

Masterstudien bidrar til forskning med perspektiver på hvordan elevene opplever å arbeide med et undervisningsopplegg som er utviklet ved hjelp av teori. Studien kan dermed tas i betraktning i videreutviklingen av allerede eksisterende teori, eller for å utvikle teori om nettopp elevenes erfaringer av programmeringsarbeid i fysikkundervisning. Studien gir i tillegg et konkret eksempel på et undervisningsopplegg, som etter resultatene å dømme kan være en tilnærming på å bruke programmering som verktøy for læring i fysikk, og ikke bruke fysikktimen til utelukkende læring av et programmeringsverktøy.



## 7 Konklusjon

Med bakgrunn i den nye læreplanens økte fokus på digital beregningskompetanse og derav numeriske metoder og programmering for blant annet fysikkfaget, i tillegg til tidligere erfaringer og samtaler med videregåendelever, er problemstillingen for studien: *Hvordan kan programmering som verktøy brukes i undervisning i fysikk 1 på en måte som oppleves relevant for elevene?* Det teoretiske grunnlaget har gitt innsikt i programmeringens tiltenkte rolle i fysikk 1, samt hvordan programmering kan bidra til læringsutbytte av relevante kompetanser og egenskaper for elever i faget. Med utgangspunkt i Weintrop et al. sin taksonomi for algoritmisk tenkning (2016), har et undervisningsopplegg med formål om å bruke programmering som verktøy for å lære fysikk blitt utviklet. For å følge studiens logiske oppsett, har teori informert de to første forskningsspørsmålene slik:

Forskningsspørsmål 1: Utdanningsdirektoratet mener at elever i fysikk 1 skal utvikle en innsikt i moderne teknologi og tenkemåte. Den algoritmiske tenkeren er relevant for alle fag, derav fysikk 1. Elevene skal dermed utvikle egenskaper som å analysere og forutse, utvikle algoritmer, dekomponere osv., gjennom å samarbeide, fikle, skape, feilsøke, fortsette og prøve igjen. I tillegg til dette er programmering et konkret kompetansemål i fysikk 1, noe som fastslår Utdanningsdirektoratets tanker om at programmering er relevant i fysikk 1. Odden et al. (2019) videreutvikler diSessas (2000) omfattende begrep om digital beregningskompetanse, siden den digitale vitenskapen har utviklet seg i alle felt, derav fysikk. Teorien om digital beregningskompetanse i fysikk illustrerer at programmering ikke bare er en del av fysikken som disiplin, men også en viktig del av fysikkundervisningen. Odden et al. viser hvordan programmeringen komplimenterer egenskaper som er relevante i fysikk.

Forskningsspørsmål 2: Undervisningsopplegget ble utarbeidet med utgangspunkt i en samtale med faglærerne og Utdanningsdirektoratets kompetansemål som nevner programmering. Taksonomien for algoritmisk tenkning (Weintrop et al., 2016) var svært nyttig i utviklingen av undervisningsopplegget, da opplegget tilstrebet å inneholde praksiser fra alle kategoriene i taksonomien for å kunne gi elevene et mest mulig helhetlig faglig utbytte, og for å konkretisere programmeringen opp mot fysikken. I tillegg ble ytre didaktiske valg gjort i utviklingsfasen, som at elevene skal samarbeide og at opplegget skal tilstrebe å ha en lett inngang med høy takhøyde, da ved å gi introduserende og mindre oppgaver i starten,

for så å gradvis øke vanskelighetsgraden. Undervisningsopplegget ble gjennomført i en fysikk 1 klasse. En spørreundersøkelse og tre intervju har gitt resultater som gir følgende elevperspektiver på forskningsspørsmålene:

Forskingsspørsmål 3: Elevene har samarbeidet, diskutert, idémyldret og resonnert ved bruk av for- og motargumenter for hvordan programmering påvirker sin digitale modell. De har også diskutert hvordan modellen kan videreutvikles. Elever opplever et slik undervisningsopplegg som krevende, med en stor variasjon i hva som oppleves som utfordrende. Variasjonen i utfordringene finner sted selv i samme deloppgave. Enkelte elever anser programmering som en relevant arbeidsmetode for fysikk 1, mens andre ser viktigheten av å lære seg programmering om fysikken blir mer «avansert» og for å forstå det stadig mer digitaliserte samfunnet.

Forskingsspørsmål 2: Enkelte elever synes at en undervisningsøkt der man kun regner oppgaver i flere skoletimer, kan være utmattende og lite givende. Arbeidsmetoden som konkrete diskusjoner med det formålet om å utvikle et konkret produkt som en digital modell er, kan være mer givende enn tradisjonell og individuell oppgaveregning på papir. Elevene selv anerkjenner ikke alle fordelene med å jobbe med programmering, men viser gjennom resonnement at de har arbeidet innen flere kategorier og praksiser innen taksonomien for algoritmisk tenkning. I tillegg anerkjenner elever flere positive sider med programmering, som å korte ned regnetiden og fordelen ved å ha et program som man kan utvikle videre og bruke i lignende tilfeller.

Forskingsspørsmål 1: Elever ser ikke selv de underliggende kompetanser som programmering stimulerer til, men anerkjenner programmering som et viktig verktøy å lære seg, enten om de selv skal studere innen en vitenskapelig retning videre, eller bare for å forstå mer av de digitale verktøyene de omgir seg med i hverdagen.

Til sammen kan det sies at programmering kan fungere som et verktøy for læring i fysikk 1 på en måte som oppleves relevant for elevene, men i ulik grad og med forskjellige argumenter for relevans. Jeg håper å ha fått vist at et undervisningsopplegg som benytter programmering som verktøy kan gi elever et utbytte de selv setter pris på, men at et slikt opplegg også er ressurskrevende å utvikle og gjennomføre. Jeg vil avslutte masteroppgaven med å stadfeste at det er behov for mer forskning på feltet, kanskje særlig forskning som fremmer elevenes perspektiver og erfaringer om opplevd læring og relevans, utfordringer, i tillegg til hvordan slik undervisning kan utføres.

## Referanseliste

- Andersen, A. H. P., Fiskum, T. A. & Rosenlund, M. R. (2018). Hva menes med undrende, utforskende og aktiviserende undervisning? Rosenlund, M.R., Gulaker, D. & Andersen, H. P. (red.) *Den engasjerte eleven: undrende, utforskende og aktiviserende undervisning i skolen*. 17-30
- Angell, C., Bungum, B., Henriksen, E. K., Kolstø, S. D., Persson, J. & Renstrøm, R. (2019). *Fysikkdidaktikk* (2 utg.). Cappelen Damm Akademisk.
- Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 77-101. <https://doi.org/10.1191/1478088706qp063oa>
- Braun, V., & Clarke, V. (2012). Thematic analysis. I H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.), *APA handbook of research methods in psychology, Vol. 2. Research designs: Quantitative, qualitative, neuropsychological, and biological* (s. 57–71). American Psychological Association. <https://doi.org/10.1037/13620-004>
- Brown, N. C. & Wilson, G. (2018). Ten quick tips for teaching programming. *PLoS computational biology*, 14(4), e1006023-e1006023. <https://doi.org/10.1371/journal.pcbi.1006023>
- diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Mit Press. <https://doi.org/10.7551/mitpress/1786.001.0001>
- diSessa, A. A. (2018). Computational literacy and “the big picture” concerning computers in mathematics education. *Mathematical thinking and learning*, 20(1), 3-31. <https://doi.org/10.1080/10986065.2018.1403544>
- Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A. & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4, 355-369. <https://doi.org/10.1007/s40692-017-0090-9>
- Fossum, J.-C., Sandstad, M., Dellnes, H. R. & Myhreagen, H. V. (2021). *Kraft 1: lærebok I fysikk 1: studiespesialiserende utdanningsprogram*. (2. utg.). Cappelen Damm.

- Fraser, M. W. & Galinsky, M. J. (2010). Steps in intervention research: Designing and developing social programs. *Research on social work practice*, 20(5), 459-466. <https://doi.org/10.1177/1049731509358424>
- Gjengset, F. R. (2022). *Shaping programming in physics education: A study on how teachers' conceptualization of computation in high school physics influences what is taught*. [Masteroppgave]. Universitetet i Oslo.
- Grandell, L., Peltomaki, M., Back, R.-J. & Salakoski, T. (2006). *Why complicate things?: introducing programming in high school using Python*, 18(1), 71-80. Presentert ved Australasian Conference on Computing Education, Hobart, Australia.
- Hutchins, N., Biswas, G., Conlin, L., Emara, M., Grover, S. & Basu, S. (2018). Studying synergistic learning of physics and computational thinking in a learning by modeling environment. *Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education*.
- Høvik, I. (2022). *Implementering og pandemi: En studie om implementeringen av LK20 under COVID-19-pandemien* [Masteroppgave]. Nord universitet.
- Johnson, B. R. (2013). Validity of Research Results in Quantitative, Qualitative and Mixed Research. (Kapittel 11) I B. R. Johnson & L. Christensen. *Educational Research: Quantitative, Qualitative and Mixed Approaches*, 277-316.
- Kortemeyer, G. & Kortemeyer, A. F. (2018). The nature of collaborations on programming assignments in introductory physics courses: a case stud. *European Journal of Physics*, 39(5), 55705. <https://doi.org/10.1088/1361-6404/aad511>
- Kunnskapsdepartementet. (2017). *Overordnet del - verdier og prinsipper for grunnopplæringen*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/overordnet-del/>
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (T. M. Anderssen & J. Rygge, Overs.; 3. utgave Gyldendal akademisk.
- Labusch, A., Eickelmann, B. & Vennemann, M. (2019). Computational thinking processes and their congruence with problem-solving and information processing. *Computational thinking education*, 65-78.
- Nordby, S. T. (2019). *Programmering og algoritmisk tenkning i fysikkundervisning* [Masteroppgave]. Norges teknisk-naturvitenskapelige universitet.
- NTNU. (2022, 17.05.2023). *NVivo*. Norges teknisk-naturvitenskapelige universitet. <https://i.ntnu.no/wiki/-/wiki/Norsk/NVivo>

- Odden, T. O. B., Lockwood, E. & Caballero, M. D. (2019). Physics computational literacy: An exploratory case study using computational essays. *Physical Review Physics Education Research*, 15(2), 020152.  
<https://doi.org/10.1103/PhysRevPhysEducRes.15.020152>
- Orban, C., Teeling-Smith, R. M., Smith, J. R. & Porter, C. D. (2018). A hybrid approach for using programming exercises in introductory physics. *American Journal of Physics*, 86(11), 831-838. <https://doi.org/10.1119/1.5058449>
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books, Inc., Publishers.
- Piatetsky, G. (2017, 28. august). *Python overtakes R, becomes the leader in Data Science. Machine Learning platforms*. Kdnuggets.  
<https://www.kdnuggets.com/2017/08/python-overtakes-r-leader-analytics-data-science.html>.
- Roberts, P. & Priest, H. (2006). Reliability and validity in research. *Nursing standard*, 20(44), 41-46.
- Robson, C. & McCartan, K. (2016). *Real world research : a resource for users of social research methods in applied settings*. (4. utg.). Wiley.
- Sale, J. E., Lohfeld, L. H. & Brazil, K. (2002). Revisiting the quantitative-qualitative debate: Implications for mixed-methods research. *Quality and quantity*, 36(1), 43-53.  
<https://doi.org/10.1023/A:1014301607592>
- Scott, S. & Palincsar, A. (2013). Sociocultural theory. I.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G. & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380.  
<https://doi.org/10.1007/s10639-012-9240-x>
- Sikt. (2023, 17.05.2023). *NSD. Sikt-Kunnskapssektorens Tjenesteleverandør*.  
<https://www.nsd.no/index.html>
- Sins, P. H., Savelsbergh, E. R. & van Joolingen, W. R. (2005). The Difficult Process of Scientific Modelling: An analysis of novices' reasoning during computer-based modelling. *International Journal of Science Education*, 27(14), 1695-1721.  
<https://doi.org/10.1080/09500690500206408>
- Sørby, S. A. & Angell, C. (2012). Undergraduate students' challenges with computational modelling in physics. *Nordic Studies in Science Education*, 8(3), 283-296.  
<https://doi.org/10.5617/nordina.534>

- Taub, R., Armoni, M., Bagno, E. & Ben-Ari, M. M. (2015). The effect of computer science on physics learning in a computational science environment. *Computers & Education*, 87, 10-23. <https://doi.org/10.1016/j.compedu.2015.03.013>
- Utdanningsdirektoratet. (2006). *Læreplan i fysikk - programfag i utdanningsprogram for studiespesialisering (FYS1-01)*. Utdanningsdirektoratet. [https://www.udir.no/kl06/fys1-01/hele/komplett\\_visning](https://www.udir.no/kl06/fys1-01/hele/komplett_visning)
- Utdanningsdirektoratet. (2019, 27. mars). *Algoritmisk tenkning*. Utdanningsdirektoratet. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2021). *Læreplan i fysikk (FYS01-02)*. Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/fys01-02>
- Utdanningsdirektoratet. (2023, 25. januar). *Innføring og overgangsordninger for nye læreplaner*. Utdanningsdirektoratet. <https://www.udir.no/laring-og-trivsel/lareplanverket/innforing-og-overgangsordninger-for-nye-lareplaner/#a166494>
- Vinnervik, P. & Bungum, B. (2022). Computational thinking as part of compulsory education: How is it represented in Swedish and Norwegian curricula? *Nordic Studies in Science Education*, 18(3), 384-400. <https://doi.org/10.5617/nordina.9296>
- Waters, J. B. (2020). *Programmering og dybdeløring i fysikk-en kvalitativ studie av elevers arbeid med programmering i fysikk 1* [Masteroppgave]. Universitetet i Oslo.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, 25, 127-147.
- Wertsch, J. V. (1985). *Vygotsky and the social formation of mind*. Harvard university press.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wæge, K. & Nosrati, W. (2018). *Motivasjon i matematikk*. Universitetsforlaget.
- Yaşar, O. (2013). Teaching science through computation. *International Journal of Science, Technology and society*, 1(1), 9-18. <https://doi.org/10.11648/j.ijsts.20130101.12>

# Vedlegg

# Vedlegg A: Undervisningsopplegg - Elevversjon

09.02.2023, 14:05

Undervisningsopplegg-elevversjon - Jupyter Notebook

## Tores fallskjermhopp

Tore Mannlig Abergé skal hoppe i fallskjerm. Han er fryktelig nervøs, og for å forberede seg best mulig tenker han at han må gjøre noen beregninger. Han åpner et pythonprogram, men sliter med programmeringen. Her kommer du inn for å hjelpe han!

Tore skal følge et fly opp til  $h_0 = 1565$  moh. Der skal han slippe seg ut med startfart  $v_0 = 0$ m/s. Han vil i første omgang simulere noen situasjoner uten fallskjerm. Du kan regne med tyngdeakselerasjonen  $g = 9.81$ m/s<sup>2</sup> i alle situasjoner.

In [3]:

```
1 #Husk å kjøre denne kodelinjen ("Shift"+"Enter")
2 import numpy as np
3 import matplotlib.pyplot as plt
4
```

## Oppgave 1

### 1a)

Hvor lang tid tar det fra han hopper til han treffer bakken? Du kan se bort i fra luftmotstand.

In [2]:

```
1 #Hvilke konstanter vil du legge inn? Skriv koden din i denne cellen og trykk "Shift"+"Enter" for å kjøre
```

In [3]:

```
1 #Hvilke(n) formler vil du bruke? Skriv koden din her
```

### 1b)

Hva er farten i det Tore treffer bakken?

In [1]:

```
1 #Skriv koden din her
```

Hvilket fortegn får du i oppgaven over for  $v$ ? Gir dette mening?

```
1 Svar her:
```

## Oppgave 2

Dette er jo ikke helt realistisk. Tore ønsker å ta hensyn til luftmotstanden, for å gjøre beregningene mer presise. Luftmotstandskoeffisienten er  $k_{\text{utenfallskjerm}} = 1.00$ N/(m/s)<sup>2</sup>. Hvilken terminalfart oppnår Tore. Tore med utstyr har massen  $m = 80$ kg.

In [ ]:

```
1 #Skriv koden din her. Om du vil ha flere celler, trykker du på plusstegnet oppe til venstre.
```

## Oppgave 3

Tore ønsker å vite hvor lenge han er i luften, selv om han ikke har på seg fallskjerm. Under er et rammeverk der Tore tar høyde for luftmotstand, men han mangler den siste biten for å finne tiden hoppet tar.

### 3a)

Fyll ut for-løkken (linje 17-19) slik at koden kjører



In [ ]:

```

1 #Noen konstanter
2 T = 70 #Dette er et overestimat, som sikrer at simuleringen foregår lenge nok til at Tore treffer bakken.
3 dt = 0.0001
4 n = int(T / dt)
5
6 #Lager lister med nuller som senere skal fylles med henholdsvis posisjons-, farts- og akselerasjonsverdier.
7 s = np.zeros(n)
8 s[0]=h_0
9 v = np.zeros(n)
10 v[0]=v_0
11 a = np.zeros(n)
12 a[0] = -9.81 #Merk at a=-g. Dette kan du endre på om du vil
13
14 t = np.linspace(0, T, n) #En liste med tidsverdier fra 0 til og med T.
15
16 for i in range(1, n): #vi begynner på 1, siden første verdien i hver liste er bestemt fra før.
17     s[i] =
18     v[i] =
19     a[i] =

```

**3b)**

Du har nå laget lister med posisjons-, farts- og akselerasjonsverdier for den tiden hoppet tar, og enda litt lengre. Du skal nå fremstille dine data grafisk. Erstatt tiden "t" og akselerasjonen "a" i linje 3 med navnet du har kalt listene. Da skal koden kjøre.

In [ ]:

```

1 plt.xlabel('Tid [s]') #Lager aksnavn på x-aksen
2 plt.ylabel('Akselerasjon [m/s^2]') #Lager aksnavn på y-aksen
3 plt.plot(t, a) #tiden t langs x-aksen, akselerasjonen a langs y-aksen
4 plt.grid() #Lager rutenettet
5 plt.show() #Viser grafen

```

Akselerasjonen vil gå mot null, men aldri bli fullstendig null. Etter hvor lang tid vil du si at akselerasjonen er så godt som null? Diskuter svaret.

Ups, Tore har desverre slettet grafen for farten. Bruk oppsettet fra akselerasjonsgrafene for å plote fartsgrafene. Husk å skrive korrekt navn på aksene.

In [ ]:

```
1 #Skriv koden din her
```

På samme måten som akselerasjonsgrafene, flater fartsgrafene ut. Mot hvilken hastighet og på hvilket tidspunkt skjer dette, og gir det mening?

```
1 Skriv ditt svar her:
```

Plot posisjonsgrafene til slutt

In [ ]:

```
1 #Skriv koden din her
```

Hvor lang tid vil du si det tar for Tore å treffe bakken, ut fra grafene over?

Under er kode som viser landingstidspunktet i siffer, gitt at du har fylt inn riktig i for-løkken tidligere.

In [12]:

```
1 print(f"t = {t[np.argmin(np.abs(s))]:.3} s")
```

```
t = 57.8 s
```

**Oppgave 4**

Tore har snart fått all den informasjonen han trenger, men vet ennå ikke på hvilket tidspunkt han burde utløse sin fallskjerm. Dette må du hjelpe han med å finne ut.

**4a)**

Hvor stor er terminalfarten Tore oppnår med fallskjermen utløst, gitt at luftmotstandskoeffisienten er  $k_{fallskjerm} = 20.0 N/(m/s)^2$ . Hint: Se oppgave 2

In [3]:

```
1 #Skriv koden din her
```

**4b)**

Gitt terminalfarten med fallskjerm du nettopp har funnet (Oppgave 4a) og fartsgrafen du fant i sted uten fallskjerm (Oppgave 3); på hvilket tidspunkt vil du anbefale Tore å løse ut fallskjermen, og hvorfor?

```
1 Diskuter og svar her:
```

**Oppgave 5**

Tore hører på ditt råd, og tenker å utløse fallskjermen på det tidspunktet du har kommet frem til. Bruk gjerne noe av koden fra oppgave 3 og plot posisjonsgrafen, fartsgrafen og akselerasjonsgrafen slik det er blitt gjort før, men ta i betraktning at Tore utløser fallskjermen på tidspunktet du har gitt. Tore ønsker ikke å besvime, så kraften i det han drar ut fallskjermen må ikke overskride 9G.

In [15]:

```
1 #Skriv koden din her
```

In [4]:

```
1 #Plot akselerasjonsgrafen her
```

In [5]:

```
1 #Plot fartsgrafen her
```

In [6]:

```
1 #Plot posisjonsgrafen her
```

Hva kan forklaringen på at akselerasjonsgrafen og fartsgrafen knekker plutselig?

```
1 Svar her:
```

**"TUSEN TAKK"**

Tore føler seg endelig forberedt nok. Han takker deg for den formidable innsatsen du har vist, spenner på seg selen og starter flymotoren. Nå kan han hoppe med et smil om munnen.

In [ ]:

```
1
```

## Vedlegg B: Undervisningsopplegg - Løsningsforslag

### Tores fallskjermhopp

Tore Mannlig Aberge skal hoppe i fallskjerm. Han er fryktelig nervøs, og for å forberede seg best mulig tenker han at han må gjøre noen beregninger. Han åpner et pythonprogram, men sliter med programmeringen. Her kommer du inn for å hjelpe han!

Tore skal følge et fly opp til  $h_0 = 1565$  moh. Der skal han slippe seg ut med startfart  $v_0 = 0$  m/s. Han vil i første omgang simulere noen situasjoner uten fallskjerm. Du kan regne med tyngdeakselerasjonen  $g = 9.81 \text{ m/s}^2$  i alle situasjoner.

```
In [1]: 1 #Husk å kjøre denne kodelinjen
        2 import numpy as np
        3 import matplotlib.pyplot as plt
```

### Oppgave 1

#### 1a)

Hvor lang tid tar det fra han hopper til han treffer bakken? Du kan se bort i fra luftmotstand.

```
In [2]: 1 g = 9.81
        2 a_g = -g #m/s**2, legg merke til at akselerasjonen er negativ
        3 #Positiv retning er valgt oppover i dette løsningsforslaget
        4 h_0 = 1565 #m
        5 v_0 = 0 #m/s
```

```
In [3]: 1 #Bruker bevegelsesligning nr. 3
        2 def t(v, v_0, s):
        3     return 2*s/(v+v_0)
        4
        5 #Bruker bevegelsesligning nr. 4
        6 def v(a,s,v_0):
        7     return np.sqrt(2*a*s + (v_0)**2)
```

#### 1 b)

Hva er farten i det Tore treffer bakken?

```
In [4]: 1 #Vi har definert positiv retning oppover, så v1 blir negativ
        2 v1 = -v(a_g, -h_0, v_0)
        3
        4 #Skriver -h_0 fordi vi har definert positiv retning oppover
        5 #Distansen tilbakelagt blir negativ
        6 print('Farten rett før han treffer bakken er v =',v1,'m/s')
        7
        8 t1 = t(v1, v_0, -h_0)
        9 print('Tiden det tar for Tore fra flyet til jordoverflaten er t =',t1,
        10       'sekund.')
```

Farten rett før han treffer bakken er  $v = -175.22927837550438$  m/s  
Tiden det tar for Tore fra flyet til jordoverflaten er  $t = 17.862311761009618$  sekund.

Hvilket fortegn får du i oppgaven over for  $v$ ?. Gir dette mening?

1 Svar her:

## Oppgave 2

Dette er jo ikke helt realistisk. Tore ønsker å ta hensyn til luftmotstanden, for å gjøre beregningene mer presise. Luftmotstandskoeffisienten er  $k_{utenfallskjerm} = 1.00 N/(m/s)^2$ . Hvilken terminalfart oppnår Tore. Tore med utstyr har massen  $m = 80\text{kg}$ .

In [5]:

```
1 #Definer nødvendige konstanter
2 m = 80 #kg
3 G = m*a_g #N
4 k_uten_fallskjerm = 1 #N/(m/s)**2
5
6 #Terminalfarten er når summen av kreftene er null. Dette fører til:
7 L = -G
8
9 #L=kv**2
10 v_terminal_uten_fallskjerm = -np.sqrt(L/k_uten_fallskjerm)
11 print('Terminalfarten er',v_terminal_uten_fallskjerm,'m/s')
```

Terminalfarten er -28.014282071829005 m/s

## Oppgave 3

Tore ønsker å vite hvor lenge han er i luften, selv om han ikke har på seg fallskjerm. Under er et rammeverk der Tore tar høyde for luftmotstand, men han mangler den siste biten for å finne tiden hoppet tar.

### 3a)

Fyll ut for-løkken (linje 17-19) slik at koden kjører

```

In [23]: 1 #Noen konstanter
2 T = 70 #Dette er et overestimat, som sikrer at simuleringen
3 #foregår lenge nok til at Tone treffer bakken.
4 dt = 0.0001
5 n = int(T / dt)
6 #Lager lister med nuller som senere skal fylles med
7 #henholdsvis posisjons-, farts- og akselerasjonsverdier.
8 s = np.zeros(n)
9 s[0]=h_0
10 v = np.zeros(n)
11 v[0]=v_0
12 a = np.zeros(n)
13 a[0] = a_g #Merk at a=-g. Dette kan du endre på om du vil
14
15 t = np.linspace(0, T, n) #En liste med tidsverdier fra 0 til og med T.
16
17 for i in range(1, n): #vi begynner på 1, siden første verdien i
18                       #hver liste er bestemt fra før.
19     s[i] = s[i-1] + v[i-1]*dt
20     v[i] = v[i-1] + a[i-1]*dt
21     a[i] = a_g + k_uten_fallskjerm * v[i-1]**2/m

```

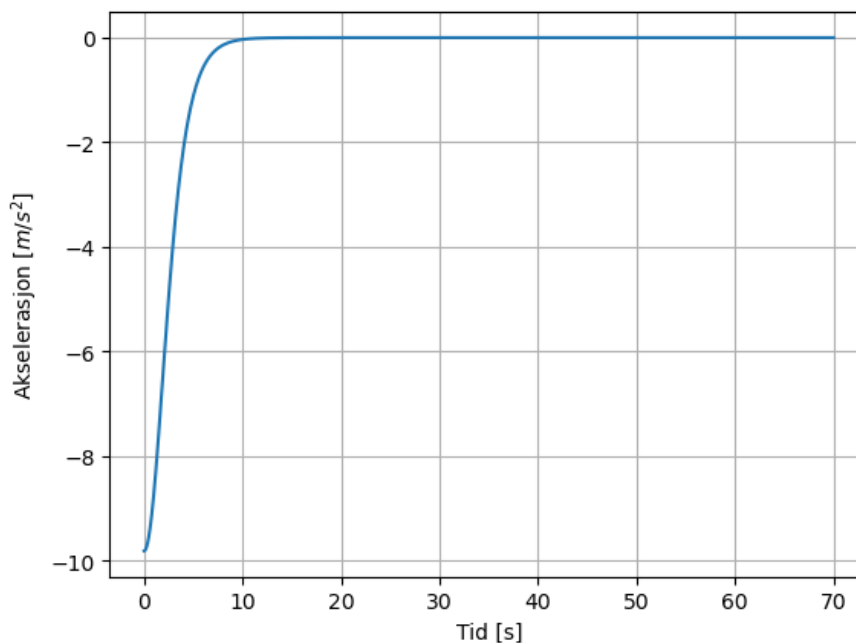
### 3b)

Du har nå laget lister med posisjons-, farts- og akselerasjonsverdier for den tiden hoppet tar, og enda litt lengre. Du skal nå fremstille dine data grafisk. Erstatt tiden "t" og akselerasjonen "a" i linje 3 med navnet du har kalt listene. Da skal koden kjøre.

```

In [24]: 1 plt.xlabel('Tid [s]')
2 plt.ylabel('Akselerasjon [m/s2]')
3 plt.plot(t, a)
4 plt.grid()
5 plt.show()

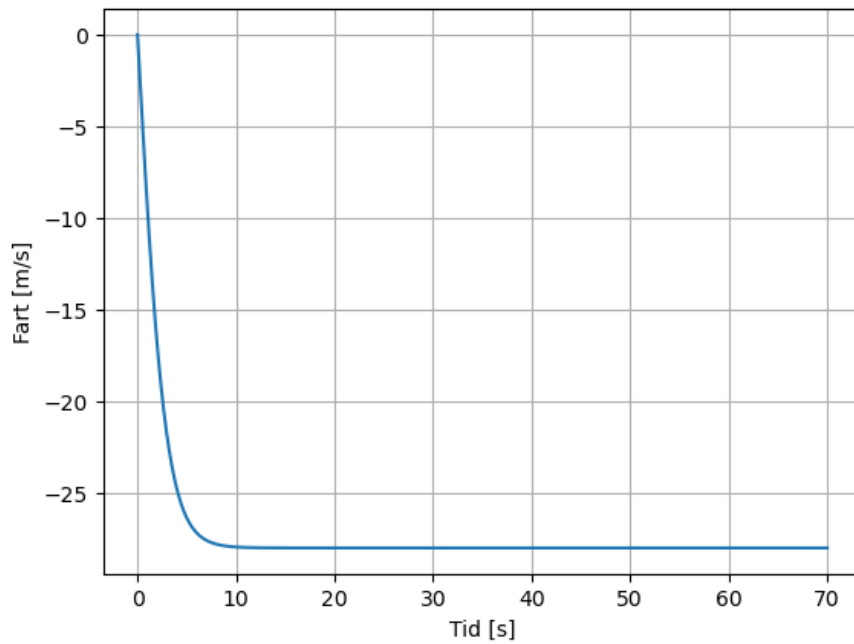
```



Akselerasjonen vil gå mot null, men aldri bli fullstendig null. Etter hvor lang tid vil du si at akselerasjonen er så godt som null? Diskuter svaret.

Ups, Tore har desverre slettet grafen for farten. Bruk oppsettet fra akselerasjonsgrafene for å plote fartsgrafen. Husk å skrive korrekt navn på aksene.

```
In [8]: 1 plt.xlabel('Tid [s]')
        2 plt.ylabel('Fart [m/s]')
        3 plt.plot(t, v)
        4 plt.grid()
        5 plt.show()
```

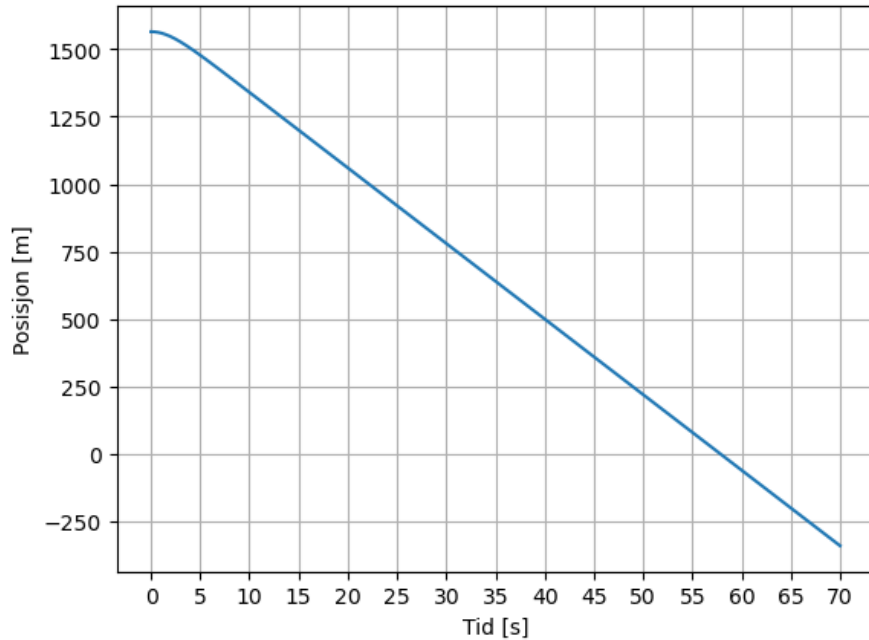


På samme måten som akselerasjonsgrafene, flater fartsgrafen ut. Mot hvilken hastighet og på hvilket tidspunkt skjer dette, og gir det mening?

1 Svar her:

Plot posisjonsgrafen til slutt

```
In [9]: 1 plt.xlabel('Tid [s]')
2 plt.ylabel('Posisjon [m]')
3 plt.plot(t, s)
4 plt.xticks(np.arange(0,T+1, step=5))
5 plt.grid()
6 plt.show()
```



Hvor lang tid vil du si det tar for Tore å treffe bakken, ut fra grafen over?

Under er kode som viser landingstidspunktet i siffer, gitt at du har fylt inn riktig i for-løkken tidligere.

```
In [10]: 1 print(f"t = {t[np.argmin(np.abs(s))]:.3} s")
t = 57.8 s
```

## Oppgave 4

Tore har snart fått all den informasjonen han trenger, men vet ennå ikke på hvilket tidspunkt han burde utløse sin fallskjerm. Dette må du hjelpe han med å finne ut.

### 4a)

Hvor stor er terminalfarten Tore oppnår med fallskjermen utløst, gitt at luftmotstandskoeffisienten er  $k_{fallskjerm} = 20.0 \text{ N/(m/s)}^2$ . Hint: Se oppgave 2

```
In [11]: 1 k_fallskjerm = 20 #N/(m/s)**2
2
3 #Terminalfarten er når summen av kreftene er null. Dette fører til:
4 L = -G
5
6 #L=kv**2
7 v_terminal_med_fallskjerm = -np.sqrt(L/k_fallskjerm) #Vil ha negativ fart
8 print('Terminalfarten er',v_terminal_med_fallskjerm,'m/s')
```

Terminalfarten er -6.26418390534633 m/s

#### 4b)

Gitt terminalfarten med fallskjerm du nettopp har funnet (Oppgave 4a) og fartsgrafen du fant i sted uten fallskjerm (Oppgave 3); på hvilket tidspunkt vil du anbefale Tore å løse ut fallskjermen, og hvorfor?

```
1 Diskuter og svar her:
```

### Oppgave 5

Tore hører på ditt råd, og tenker å utløse fallskjermen på det tidspunktet du har kommet frem til. Bruk gjerne noe av koden fra oppgave 3 og plot posisjonsgrafen, fartsgrafen og akselerasjonsgrafen slik det er blitt gjort før, men ta i betraktning at Tore utløser fallskjermen på tidspunktet du har gitt. Tore ønsker ikke å besvime, så kraften i det han drar ut fallskjermen må ikke overskride 9G.

```
In [12]: 1 t_ut=t[np.argmin(np.abs(v-v_terminal_med_fallskjerm))]
2 print("Tore bør løse ut fallskjermen etter", t_ut,
3       "sekund, for å unngå kraftig nedbremsing")
```

Tore bør løse ut fallskjermen etter 0.6495009278584684 sekund, for å unngå kraftig nedbremsing



```

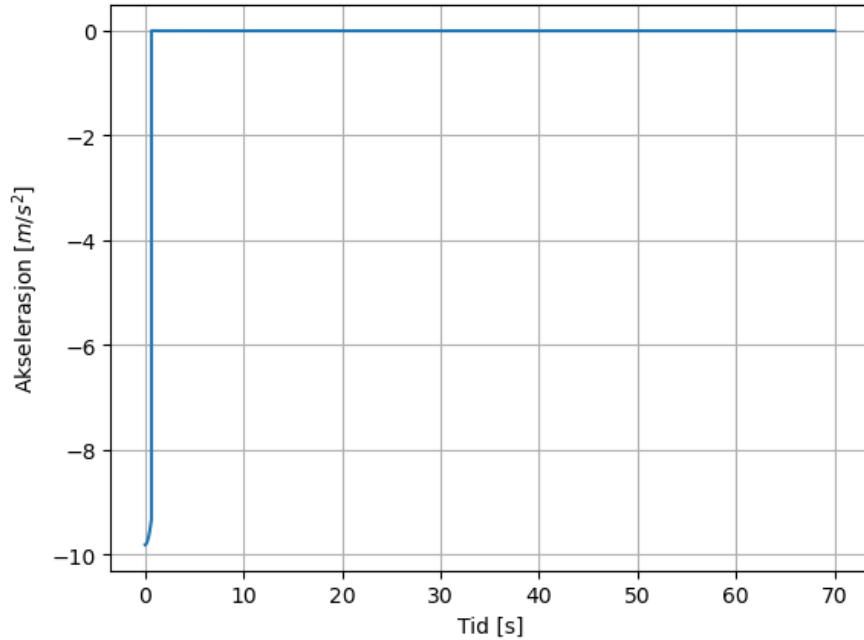
In [13]: 1 #Samme kode som i 3a
2 T = 70
3 dt = 0.0001
4 n = int(T / dt)
5
6 s = np.zeros(n)
7 s[0]=h_0
8 v = np.zeros(n)
9 v[0]=v_0
10 a = np.zeros(n)
11 a[0] = a_g
12
13 t = np.linspace(0, T, n)
14
15 for i in range(1, n):
16     s[i] = s[i-1] + v[i-1]*dt
17     v[i] = v[i-1] + a[i-1]*dt
18
19     #if-bestingelsen bestemmer hvilken luftmotstandskoeffisient vi bruker
20     if t[i] <= t_ut:
21         k = k_uten_fallskjerm
22
23     else:
24         k = k_fallskjerm
25
26     a[i] = a_g + k * (v[i-1]**2)/m #---
27

```

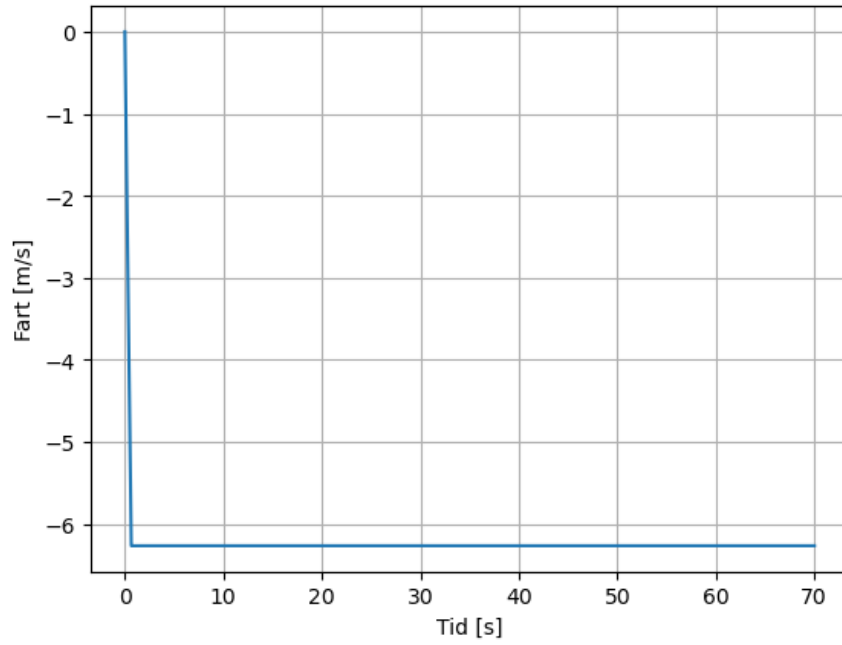
```

In [14]: 1 plt.xlabel('Tid [s]')
2 plt.ylabel('Akselerasjon [m/s^2]')
3 plt.plot(t, a)
4 plt.grid()
5 plt.show()

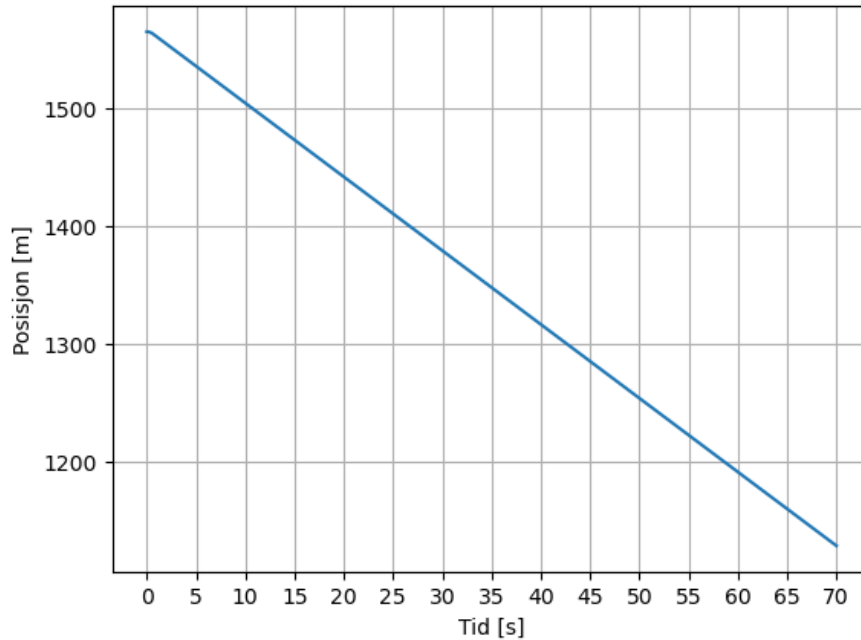
```



```
In [15]: 1 plt.xlabel('Tid [s]')
          2 plt.ylabel('Fart [m/s]')
          3 plt.plot(t, v)
          4 plt.grid()
          5 plt.show()
```



```
In [16]: 1 plt.xlabel('Tid [s]')
2 plt.ylabel('Posisjon [m]')
3 plt.plot(t, s)
4 plt.xticks(np.arange(0,T+1, step=5))
5 plt.grid()
6 plt.show()
```



Hva kan forklaringen på at akselerasjonsgrafen og fartsgrafene knekker plutselig?

1 Svar her:

## "TUSEN TAKK"

Tore føler seg endelig forberedt nok. Han takker deg for den formidable innsatsen du har vist, spenner på seg selen og starter flymotoren. Nå kan han hoppe med et smil om munnen.

In [ ]: 1

## Vedlegg C: Powerpoint-presentasjon

### Programmering i fysikk

Elias Odden Slettevoll



### Om timen



2 TIMER (++) MED  
OPPGAVEGJØRING



SPØRREUNDERSØKELSE



DELTAKERE TIL  
INTERVJU



VI BRUKER JUPYTER  
NOTEBOOK

## Jupyter Notebook

- Python
- Kodeceller
- Åpne programmet

## Tores fallskjermhopp

- Erfaringer
- Oppgavene
- Står du fast?



## **Spørreundersøkelse**

- Frivillig
- Individuelt
- Anonymt
- Svar ærlig
- Link: <https://nettskjema.no/a/312256>



## **Intervju / samtale**

- Frivillig
- Individuell
- Sikt.no (NSD)
- 30-40 min samtale



# Vedlegg D: Resultat fra spørreundersøkelse





## Spørreundersøkelse - vgs

Oppdatert: 19. mai 2023 kl. 13:40

Dette spørreskjemaet omhandler dine tanker rundt fysikktimen i dag.  
Først kommer noen spørsmål om dine forkunnskaper:





### Har du TOF som valgfag (Teknologi og forskningslære)?

Antall svar: 22

| Svar | Antall | % av svar |  |
|------|--------|-----------|--|
| Nei  | 11     | 50%       |  50% |
| Ja   | 11     | 50%       |  50% |






### I hvilken grad følte du at du hadde kontroll på temaet i fysikk, før timen i dag?

Antall svar: 22    Snitt: 3.59    Median: 4

| Svar | Antall | % av svar |   |
|------|--------|-----------|---|
| 5    | 2      | 9.1%      |  9.1%  |
| 4    | 11     | 50%       |  50%  |
| 3    | 7      | 31.8%     |  31.8% |
| 2    | 2      | 9.1%      |  9.1%  |
| 1    | 0      | 0%        | 0%  |

### I hvilken grad følte du at du hadde kontroll på programmering, før timen i dag?

Antall svar: 22    Snitt: 2.86    Median: 3

| Svar | Antall | % av svar |   |
|------|--------|-----------|---|
| 5    | 1      | 4.5%      |  4.5%  |
| 4    | 3      | 13.6%     |  13.6% |
| 3    | 11     | 50%       |  50%  |
| 2    | 6      | 27.3%     |  27.3% |
| 1    | 1      | 4.5%      |  4.5%  |

## Hvilke oppgaver gjennomførte du?

Antall svar: 22

| Svar      | Antall | % av svar |       |
|-----------|--------|-----------|-------|
| Oppgave 5 | 1      | 4.5%      | 4.5%  |
| Oppgave 4 | 8      | 36.4%     | 36.4% |
| Oppgave 3 | 20     | 90.9%     | 90.9% |
| Oppgave 2 | 20     | 90.9%     | 90.9% |
| Oppgave 1 | 21     | 95.5%     | 95.5% |

Spørsmålene på denne siden handler om timen i dag.

## Samarbeidet du med noen medelever?

Antall svar: 22

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| Nei  | 2      | 9.1%      | 9.1%  |
| Ja   | 20     | 90.9%     | 90.9% |

## Opplevde du at du fikk hjelp når du trengte det?

Antall svar: 22

| Svar                   | Antall | % av svar |       |
|------------------------|--------|-----------|-------|
| Jeg trengte ikke hjelp | 0      | 0%        | 0%    |
| Av og til              | 8      | 36.4%     | 36.4% |
| Nei                    | 0      | 0%        | 0%    |
| Ja                     | 14     | 63.6%     | 63.6% |







## Hvordan oppsøkte du hjelp i løpet av opplegget?

- Rakk opp hånda og spurte om hjelp
- Jeg spurte lærer om hjelp.
- Rakk opp hånden og brukte ChatGPT.
- Rakk opp handa
- Rakk opp hånda og spurte noen av de rundt meg
- Jeg spurte spørsmål.
- rakk opp hånda
- Jeg satt ved siden av noen som var god i koding og jeg spurte om hjelp fra lærer og student.
- Rakk opp handa
- Rekke opp hånda.
- Spurte de rundt meg om deres tanker og rakk opp hånda og fikk hjelp/et lite dytt fra læreren
- Jeg rakk opp hånden men det var for få lærere tilgjengelig.
- Rakk opp hånda.
- rakk opp hånda og spurte medelev
- Rakk opp hånden og brukte internett
- ba om det
- Spørre lærer
- rekke opp hånden
- rekte opp hånda

## I hvilken grad lærte du ny programmering?

Antall svar: **22**    Snitt: **3.36**    Median: **3**

| Svar | Antall | % av svar |  |
|------|--------|-----------|--|
| 5    | 1      | 4.5%      |  4.5%   |
| 4    | 9      | 40.9%     |  40.9% |
| 3    | 9      | 40.9%     |  40.9% |
| 2    | 3      | 13.6%     |  13.6%  |
| 1    | 0      | 0%        | 0%   |

## I hvilken grad har du økt din forståelse i fysikken du jobbet med?

Antall svar: 22    Snitt: 2.77    Median: 3

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| 5    | 1      | 4.5%      | 4.5%  |
| 4    | 3      | 13.6%     | 13.6% |
| 3    | 11     | 50%       | 50%   |
| 2    | 4      | 18.2%     | 18.2% |
| 1    | 3      | 13.6%     | 13.6% |

Under er det noen påstander. Ranger i hvilken grad du er enig med påstanden.

- 1 = helt uenig
- 2 = litt uenig
- 3 = nøytral / vet ikke
- 4 = litt enig
- 5 = helt enig

## I timen i dag fikk jeg til å programmere.

Antall svar: 22    Snitt: 3.68    Median: 4

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| 5    | 1      | 4.5%      | 4.5%  |
| 4    | 14     | 63.6%     | 63.6% |
| 3    | 6      | 27.3%     | 27.3% |
| 2    | 1      | 4.5%      | 4.5%  |
| 1    | 0      | 0%        | 0%    |

## I timen i dag ble jeg utfordret til å tenke nytt innen fysikken jeg jobbet med

Antall svar: 22    Snitt: 3.73    Median: 4

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| 5    | 4      | 18.2%     | 18.2% |
| 4    | 10     | 45.5%     | 45.5% |
| 3    | 6      | 27.3%     | 27.3% |
| 2    | 2      | 9.1%      | 9.1%  |
| 1    | 0      | 0%        | 0%    |

### I timen i dag brukte jeg kun programmering jeg kunne fra før

Antall svar: 22    Snitt: 2.27    Median: 2

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| 5    | 1      | 4.5%      | 4.5%  |
| 4    | 3      | 13.6%     | 13.6% |
| 3    | 3      | 13.6%     | 13.6% |
| 2    | 9      | 40.9%     | 40.9% |
| 1    | 6      | 27.3%     | 27.3% |

### I timen i dag brukte jeg kun fysikk jeg kunne fra før

Antall svar: 22    Snitt: 3.41    Median: 3

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| 5    | 5      | 22.7%     | 22.7% |
| 4    | 4      | 18.2%     | 18.2% |
| 3    | 9      | 40.9%     | 40.9% |
| 2    | 3      | 13.6%     | 13.6% |
| 1    | 1      | 4.5%      | 4.5%  |





### Timen i dag var interessant / gøy

Antall svar: 22    Snitt: 3.59    Median: 4

| Svar | Antall | % av svar |       |
|------|--------|-----------|-------|
| 5    | 2      | 9.1%      | 9.1%  |
| 4    | 10     | 45.5%     | 45.5% |
| 3    | 9      | 40.9%     | 40.9% |
| 2    | 1      | 4.5%      | 4.5%  |
| 1    | 0      | 0%        | 0%    |






### Timen i dag gjorde meg mer interessert i å bruke programmering i fysikk.

Antall svar: 22    Snitt: 3.27    Median: 3

| Svar | Antall | % av svar |  |
|------|--------|-----------|--|
| 5    | 3      | 13.6%     |  13.6%  |
| 4    | 7      | 31.8%     |  31.8% |
| 3    | 5      | 22.7%     |  22.7%  |
| 2    | 7      | 31.8%     |  31.8% |
| 1    | 0      | 0%        | 0%   |





### Ved å bruke programmering forstod jeg fysikken bedre.

Antall svar: 22    Snitt: 3.00    Median: 3

| Svar | Antall | % av svar |  |
|------|--------|-----------|--|
| 5    | 1      | 4.5%      |  4.5%     |
| 4    | 8      | 36.4%     |  36.4%   |
| 3    | 6      | 27.3%     |  27.3% |
| 2    | 4      | 18.2%     |  18.2% |
| 1    | 3      | 13.6%     |  13.6%  |

### Det er unødvendig å bruke programmering i fysikk.




Antall svar: 22    Snitt: 2.73    Median: 3

| Svar | Antall | % av svar |  |
|------|--------|-----------|--|
| 5    | 0      | 0%        | 0%   |
| 4    | 2      | 9.1%      |  9.1%   |
| 3    | 13     | 59.1%     |  59.1% |
| 2    | 6      | 27.3%     |  27.3% |
| 1    | 1      | 4.5%      |  4.5%   |

Under er det noen avkryssningsspørsmål etterfulgt av tekstspørsmål.

## Føltes timen relevant for deg?

Antall svar: 22



| Svar     | Antall | % av svar |  |
|----------|--------|-----------|--|
| Vet ikke | 4      | 18.2%     |  18.2%  |
| Nei      | 1      | 4.5%      |  4.5%   |
| Ja       | 17     | 77.3%     |  77.3% |

## Forklar:

- Handlet om det vi har jobbet med i fysikk
- Hvis programmering blir relevant for videre regning i fysikk er dette relevant for meg. Vi har fram til nå ikke hatt mye programmering, og derfor føles det mindre relevant.
- Ja, oppgavene handlet om ting vi har hatt om i fysikk.
- Vi har ikke om dette temaet nå
- Jeg synes det var veldig fint å få lære mer om programmering innen fysikk siden vi ikke har hatt så mye om det før.
- Det med programmeringen hjalp med forståelse til fysikken.
- Programmering er en del av læreplanen
- Artig og nyttig å lære programmering, både for problem løsning og evt. jobb
- Passet ikke akkurat tema. Men fikk oppsummert bevelgelsesligninger og programmering
- Brukte programmering vi ikke forstår.
- Det er alltid relevant å lære seg koding ettersom det kan brukes i flere fag. + Ved å tenke litt på jobb i framtiden kan kodingen jeg har lært være relevant for opplæringen jeg har da
- Synes det er viktig å lære å bruke andre fremgangsmetoder i fysikken. Det gjør det lettere å velge lettest mulig fremgangsmåte når man skal løse en oppgave.
- stor interesse for programering
- God trening for programmering, men jeg kan det meste fra før av og jeg forstår fysikk veldig godt.
- Gøy med variasjon
- vet ikke
- spennede og intressant
- vi trenger mer øving på programmering




## Timen i dag var:

Antall svar: 22

| Svar                 | Antall | % av svar |  |
|----------------------|--------|-----------|--|
| For utfordrende      | 7      | 31.8%     |  31.8%  |
| Passelig utfordrende | 15     | 68.2%     |  68.2% |
| For lite utfordrende | 0      | 0%        | 0%   |

## Synes du timen i dag var givende?

Antall svar: 22

| Svar               | Antall | % av svar |  |
|--------------------|--------|-----------|--|
| Nøytral / Vet ikke | 8      | 36.4%     |  36.4% |
| Nei                | 1      | 4.5%      |  4.5%   |
| Ja                 | 13     | 59.1%     |  59.1% |

## Forklar:

- Lærte noen nye ting om programmering men vet ikke om jeg mestret det helt
- Jeg har ikke hatt så mye programmering i både fysikk og matte, så for meg er programmering veldig nytt. Dermed var det litt givende for jeg lærte nye ting.
- Jeg fikk flere errorer jeg ikke forsto, og måtte få hjelp til å fikse på koden.
- Jeg forsto ikke så mye og de ulike lærerne sa at forskjellige ting var riktig som forvirret meg
- Lærte nye ting og fikk jobbet med relevant tema
- Fikk brukt forskjellig kunnskap til å gjøre ting jeg ikke har gjort før
- Vanskelig, men artig når jeg forsto
- Skjønte ikke alt vi gjorde, og hvorfor vi gjorde det slik.
- Syntes det var gøy å lære mer om programmering, og at det var interessant å tenke programmering både med og uten luftmotstand
- ble utfordret
- Lærte mye om hvordan man skulle tenke når man skulle gjøre slike oppgaver
- vet ikke
- Lærte en del nytt om hvordan man setter opp lister blant annet

## Hva tenker du kunne blitt gjort for å forbedre timen?

- Forklart hva som skjer i de kodene som var lagt inn fra før
- Lærer og forsker ga meg ulike råd på oppgave 2 , og det ble derfor vanskelig å vite hva som var rett.
- Enklere oppgaver og bedre forklaring på hva kodene betyr.
- vet ikke
- Kanskje en liten gjennomgang på slutten
- Å gi ut skjema av kommandoer i python, slik at man ser hva de forskjellige gjør, og vet hvilke forskjellige muligheter av kode man har for å løse oppgaven
- Litt lettere oppgaver
- Brukt programmering vi kan fra før.
- Litt flere lærere kanskje? Til tider litt lite hjelp pga. det tar mer forklaring å forklare programmering enn vanlig
- Flere lærere tilstede
- Kanskje hatt en gjennomgang på det grunnleggende
- enklere oppgaver eller ett løsningsforslag

## Hvor mye tid hadde du til oppgavene?

Antall svar: 22

| Svar             | Antall | % av svar |       |
|------------------|--------|-----------|-------|
| For mye tid      | 0      | 0%        | 0%    |
| Passelig med tid | 12     | 54.5%     | 54.5% |
| For lite tid     | 10     | 45.5%     | 45.5% |

## Vedlegg E: Utfyllende intervjuguide med teoretisk forankring

### Intervjuguide

Du har nå signert et samtykkeskjema som er godkjent av Sikt (NSD). Vi har gjennomgått en trippeltime i fysikk. Jeg ønsker gjerne å vite litt mer utdypende om hvordan du opplevde timen. Mot slutten av intervjuet vil jeg for ordenhets skyld spørre om det er noe du ønsker å trekke fra intervjuet, så vil jeg fjerne det i transkripsjonen. I tillegg vil alle særnavn, personlige pronomen eller andre identifiserende ord bli utelatt fra en transkripsjon.

1. Oppvarmingsspørsmål: Hva er dine interesser, hva skal du studere osv?
2. Har du TOF som valgfag?
3. Hva er fysikk 1 for deg? Hva liker du spesielt godt med fysikk?
  - a. Hvorfor valgte du fysikk 1?
4. Hva tenker du når du hører ordet programmering?
5. Hva er din erfaring med programmering fra tidligere?
6. Hvorfor tror du at man bruker programmering i fysikk 1?
7. Hvordan føler du det gikk med oppgavene i opplegget?
  - a. Hva var det som var vanskelig / lett for deg?
8. Hva var det i timen som var spesielt? (gøy, krevende osv.)
9. Hvordan arbeidet du i timen? (Samarbeid, individuelt osv.)
  - a. Hvordan føler du at det fungerte?
10. Hvordan angrep du/dere oppgaven?... (Hvordan tenkte dere i starten?)
  - a. Sett i ettertid, er det noe du tenker dere burde gjort annerledes i starten?  
(Begynte dere å programmere med en gang, eller skrev dere på papir osv?)
11. Hvordan gjorde dere tankene/planene om til faktisk kode?
  - a. Var der noe som var ekstra krevende med å skrive koden? (Kan du forklare litt dypere hva som var krevende?)
  - b. Var det lett å gjøre formler om for å få de til å kjøre osv.? (?)
  - c. Måtte dere endre noe av den foregående koden for at de skulle fungere?
12. Hvordan oppsøkte du/dere hjelp i løpet av opplegget?
  - a. Internett / chat gpt / lærer / sidekamerat
  - b. Hva fikk du ut av hjelpen?
13. Hva føler du at du lærte av timen?
14. Hvilke endringer tror du kunne blitt gjort for å gjøre timen bedre for deg?



15. Er det noe vi ikke har snakket om [tilknyttet trippeltimen] som du mener burde vært nevnt?

Tusen takk for at du deltok! Er det noe du ønsker å trekke fra din besvarelse?

### **Begrunnelse for spørsmål:**

1. Oppvarmingsspørsmål for å få samtalen i gang.
2. For å knytte spørreundersøkelsen opp mot gjeldende intervju
3. Holdning / Bakgrunnskunnskap
4. Bakgrunnskunnskap
5. Bakgrunnskunnskap
6. Sørby og Angell: Igjen om hvilke kunnskaper som programmeringen «tvinger» elevene til å benytte seg av for å forstå konseptene. Elevens holdning og tanker om programmeringens rolle i faget. Vil avhenge av hvordan programmering har blitt presentert og brukt tidligere.
7. Sørby og Angell: Hvilke kunnskaper er det informanten føler at hen brukte i timen. Måtte hen kombinere matematisk, fysisk og programmeringsforståelse for å komme noen vei.
8. For å bevege seg inn mot den gjeldende timen. For å knytte intervju mot spørreundersøkelse.
9. Sørby og Angell: Sosiokulturell teori om at kunnskap blir til gjennom sosial omgang.
10. Sørby og Angell: Spørsmålet er laget for at informanten skal forklare hvilke kunnskaper/modus som initialiserte læringsprosessen. Var det dagligdagse erfaringer, var det matematisk/analytisk kunnskap, eller var det programmeringsteknisk informanten tenkte først på.

Weintrop: I tillegg vil spørsmålet forhåpentligvis nevne Weintrops tanker om å angripe et system og diskusjoner om hvordan det er delt opp. Selv om jeg har delt opp systemet i mindre problem for elevene, må elevene planlegge hvordan koden skal samkjøres videre.

11. Weintrop: Planen er å få dokumentert hvordan elevene faktisk tenker «computational thinking». Denne faktiske kodingen er en viktig egenskap i det 21. århundre.
12. Spørsmålet er stilt for å få en innsikt i hvordan veiledningen i løpet av undervisningsopplegget fungerte for elevene, og hva de fikk ut av det

13. Sørby og Angell: Dette for å knytte de kunnskaper og moduser informanten forhåpentligvis har snakket om opp mot problemstillingen til masteren.
14. Spørsmålet skal gi innsikt i elevenes perspektiver for hvordan et undervisningsopplegg som bruker programmering som verktøy er ønsket. Det gir også direkte råd om hva som kan bedres i opplegget for elevene.
15. For å avdekke eventuelle hull i intervjuguiden.

Vedlegg F: Informasjonsskriv godkjent av Sikt

## Vil du delta i forskningsprosjektet *Programmering som verktøy i fysikk*

Dette er et spørsmål til deg om du vil delta i et forskningsprosjekt hvor formålet er å undersøke om programmering kan brukes som verktøy til å øke læringen i fysikk. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

### **Formål**

Prosjektet omhandler fysikkundervisning på videregående nivå. Bakgrunnen for prosjektet er de nye læreplanmålene som kom med den nye læreplanen. Programmering fikk da en vesentlig større rolle i undervisningen, særlig i matematikk og fysikk. Formålet er å undersøke hvordan dere som elever opplever bruk av programmering som verktøy i fysikk. Undersøkelsen vil danne grunnlaget for videre analyse og diskusjon til masterstudien min i fysikkdidaktikk.

### **Hvem er ansvarlig for forskningsprosjektet?**

Elias Odden Slettevoll, femteårsstudent ved Institutt for fysikk er ansvarlig for prosjektet.

### **Hvorfor får du spørsmål om å delta?**

Dere elever er valgt ut til å delta i dette prosjektet, da dere gjennomgår undervisning som er relevant for læreplanmålene tilknyttet denne studien. Det er derfor interessant å få vite mer om hva du som elev tenker rundt opplegg med formål å stimulere læreplanmålene.

### **Hva innebærer det for deg å delta?**

Hvis du velger å delta i prosjektet, innebærer det at du deltar på et personlig intervju. Intervjuet vil ta mellom 15-60 minutter. Intervjuet inneholder utdypende spørsmål som angår programmering i fysikk, din oppfatning av undervisningen, og litt om din holdning til programmering generelt. Intervjuet vil bli tatt opp. Dette intervjuet vil bli lagret ved hjelp av kryptert skylagring.

### **Det er frivillig å delta**

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg. Det vil eksempelvis ikke ha noe å si for din vurdering i faget i noen omfang om du velger å trekke deg.

### **Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger**

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Det er kun jeg som student som har tilgang til opptakene.
- Opplysningene/opptakene vil bli lagret på en kryptert mappe i en ekstern lagringsenhet.

I en eventuell publikasjon vil ingen personopplysninger om deg være mulig å spore. Dine personopplysninger blir ikke publisert.

### **Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?**

Masteroppgaven skal etter planen avsluttes 01.06.2023. Ved slutten av masteroppgaven vil dataene bli anonymisert, det vil si at koblingsnøkkelen som gjør det mulig å kombinere dine personopplysninger og dine data vil bli slettet. Lydopptakene vil også bli slettet.

### **Hva gir oss rett til å behandle personopplysninger om deg?**

Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra Institutt for fysikk ved NTNU har Sikt – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

### **Dine rettigheter**

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med:

- Student som gjennomfører prosjektet: Elias Odden Slettevoll, tlf: 41590928, e-post: elias.slettevoll@gmail.com
- Veileder: Trine Høyberg Andersen, tlf: 73558923, e-post: trine.andersen@ntnu.no
- Vårt personvernombud: Thomas Helgesen

Hvis du har spørsmål knyttet til Personverntjenester sin vurdering av prosjektet, kan du ta kontakt med:

- Personverntjenester på epost (personverntjenester@sikt.no) eller på telefon: 53 21 15 00.

Med vennlig hilsen

Elias Odden Slettevoll  
(Student/master)

---

## Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet *Programmering i fysikk*, og har fått anledning til å stille spørsmål. Jeg samtykker til (kryss av boksene):

- å delta i personlig intervju
- at det kan bli tatt opp lyd av en gruppediskusjon jeg deltar i

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

---

(Signert av prosjektdeltaker, dato)



