50 - Computational thinking i fysikfaget

Fra makro til mikro med simuleringer i NetLogo

Af Jonas Ørbæk Hansen (JHA)

Indledning

Computersimuleringer spiller en stadig større rolle i naturvidenskab og i den moderne naturvidenskabelige forskning. Vi er rykket fra tiden, hvor nye naturvidenskabelige erkendelser nås alene med simpelt labudstyr og en kuglepen. Simuleringer er blevet en væsentlig naturvidenskabelig metode og bør derfor indgå i de naturvidenskabelige fag på lige fod med de andre naturvidenskabelige metoder.

Tag for eksempel den sensationelle måling af gravitationsbølger i 2015. Her kunne man ved at sammenligne det målte signal med signalet fra over 250.000 simuleringer fastslå, at signalet kom fra to sammensmeltende sorte huller med masser på hhv. 36 og 29 solmasser (Pedersen 2016). Ved at inddrage simuleringer kunne man altså opnå en dybere indsigt i fænomenet og give en fysisk forklaring (en mikroskopisk model) på det observerede.

Fordi simuleringer spiller så stor en rolle for moderne naturvidenskab, er det også en vigtig forudsætning for elevernes faglige dannelse, at de opnår en basal forståelse for, (1) hvordan simuleringer benyttes i moderne naturvidenskab, og (2) hvad der foregår i maskinrummet af en simulering.

Det er vigtigt, at eleverne bliver klædt på til at kunne vurdere simuleringsværktøjers rækkevidder og begrænsninger og blive reflekterede og kritiske i deres brug af simuleringer og data fra simuleringer. Skal de kunne dette, er det nødvendigt med et grundlæggende kendskab til programmering/kodning og en forståelse for algoritmer, og de skal præsenteres for simuleringsværktøjer, hvor de har adgang til maskinrummet (ikke tilfældet for de fleste online simuleringsværktøjer).

Ideelt set skal eleverne ned og arbejde med computerkoden i simuleringsværktøjet og lave deres egne værktøjer eller implementere ændringer i eksisterende. De skal rykkes fra at være *brugere* til at være *skabere* (eller *medskabere*) af de digitale værktøjer, som de arbejder med.

Denne måde at arbejde med simuleringer har yderligere nogle potentielle gevinster:

1. Eleverne udvikler nogle almene basiskompetencer inden for problemløsning, som kan udnyttes i andre fag og discipliner (computational thinking):
Der ligger mere i at kunne lave et velfungerende simuleringsværktøj end blot at kunne programmere. Man skal kunne nedbryde en opgave i mindre dele, tænke abstrakt og designe algoritmer. Det er her, begrebet *computational thinking* kommer ind i billedet, for der er flere anerkendte forskere, der taler for, at disse kompetencer er grundlæggende og almene basiskompetencer, som kan komme eleverne til gode bredere på tværs af fag og discipliner. Michael E. Caspersen skiver i "Gymnasiepædagogik - En grundbog", at (Caspersen 2017):

"... computational thinking er internationalt hastigt I færd med at blive en del af almendannelsen i skolen på alle klassetrin; mange mener, at det er (eller snart bliver) en lige så væsentlig grundlæggende kompetence som læsning, skrivning og matematik."

2. Det kan understøtte elevernes læringsproces og øge deres fysikfaglige forståelse:
At foretage ændringer i simuleringsværktøjer og tilpasse dem til nye problemstillinger kræver en grundig forståelse for de fysiske fænomener og processer, der finder sted i simuleringen, og de modeller, der inddrages. Det kan derfor give eleverne

en grundigere og dybere forståelse af den fysik og de fysikmodeller, der arbejdes med.

Den overordnede problemstilling

På baggrund af indledningen har jeg med udgangspunkt i et konkret fysikforløb med titlen *"Simuleringer og fysiske modeller (henfaldsloven)"* undersøgt følgende problemstilling:

Kan vi arbejde med computational thinking (CT) i fysikfaget på en måde, så det udover at styrke elevernes CT-kompetencer også bidrager til et øget fysikfagligt udbytte?

Computational thinking (CT)

Begrebet computational thinking (CT) blev introduceret af Seymour Papert i 1980 (Papert 1980), men det var først, da Jeannette M. Wing relancerede begrebet i 2006 (Wing 2006), at det for alvor begyndte at komme på dagsordenen i undervisningssystemet verden over (Caspersen 2017). Wing betegner computational thinking som:

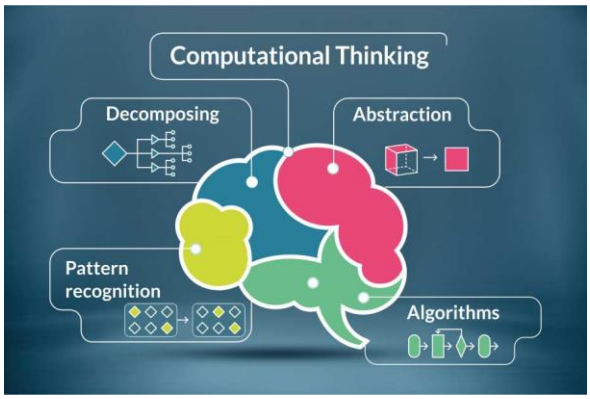
"De tankeprocesser, der foregår, når en problemstilling formuleres og de tilhørende løsninger udtrykkes på en sådan måde, at en computer – menneske eller maskine – effektivt kan udføre dem."[1]

Wing og flere argumenterer for, at computational thinking dækker over nogle almene basiskompetencer, som alle kan have glæde af, og som alle børn som voksne derfor bør lære (Wing 2006, Caspersen 2017, Google 2018).

Der er forskellige bud på, hvad disse almene kompetencer/tankeprocesser dækker over, men disse fire går igen mange steder (Google 2018):

- **Nedbrydning:** Nedbryde data, processer eller problemer i mindre, håndterbare dele.
- **Mønstre og generaliseringer:** Finde mønstre, tendenser og regelmæssigheder i data.
- **Abstraktion:** Identificere de generelle principper, der genererer disse mønstre.
- **Algoritmisk tænkning:** Udvikle trin-for-trin instruktioner til at løse dette og lignende problemer.

Hertil kunne man tilføje logisk ræsonnement og evaluering (Barefoot 2018) samt evnen til at beskrive problemer og problemløsningsstrategier meget præcist, detaljeret og struktureret, og en arbejdspraksis, hvor man iterativt analyserer og forbedrer sit produkt.



Figur 1: Computational thinking dækker over nogle almene basiskompetencer, som kan hjælpe eleverne til at løse problemer på tværs af fag og discipliner. Lånt fra Next Gurukul (2017).

Computational thinking er således mere end blot programmering. Begrebet dækker over en bred vifte af problemløsningskompetencer. Michael E. Caspersen beskriver det som (Caspersen 2017):

"... evnen til at repræsentere, analysere, bearbejde og præsentere data og dataprocesser gennem passende abstraktioner i modeller og simuleringer, samt automatiseret problemløsning gennem algoritmisk tænkning."

[1] *"Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry it out."* (Wing 2014). Den danske oversættelse er fra rapporten "Sammen om naturvidenskab" (Astra 2017).

Computational thinking skal altså forstås som en arbejdsform og en tænkeform, som kan indgå i alle fag og ikke kun i de naturvidenskabelige. Computational thinking handler således ikke om at lære elever at programmere for programmeringens skyld, men at eleverne – gennem det at lære at programmere – udvikler deres problemløsende kompetencer og abstrakte tænkning og lærer at se på problemer, emner og verden på en ny måde (Petropouleas 2018).

Agentbaseret modellering (ABM)

Simuleringerne i forløbet er lavet i programmet NetLogo. Netlogo er agentbaseret, hvilket vil sige, at programmeringen af en simulering tager udgangspunkt i de enkelte "agenter" i modellen (fx atomkerner, kanonkugler, fugle, etc.). Det gør arbejdet med og programmeringen af modeller meget intuitivt og let at gå til for eleverne. Det centrale i programmeringen består således i at definere agenterne og udstyre dem med de rette "egenskaber" (fx udseende, position, retning og fart) og den rette "adfærd" (fx frastødning/tiltrækning, kemisk reaktion, henfald, etc.). Agenternes egenskaber og adfærd definerer, hvordan de ser ud og opfører sig, samt hvordan de interagerer med hinanden og deres omgivelser.

Med agentbaseret modellering kan store, komplekse fænomener reduceres til et spørgsmål om, hvordan de enkelte agenter agerer i bestemte situationer. Det giver eleverne mulighed for at arbejde med relativt komplekse problemstillinger (fænomener/processer), som ellers ville være vanskelige at behandle i undervisningen.

NetLogo er dermed et ideelt værktøj til at arbejde med forståelsen af fysikmodeller og til at undersøge *"hvilke interessante makroskopiske fænomener der fremtræder ud fra agenternes mikroskopiske adfærd"* (CCTD 2018), og dermed belyse sammenhængen mellem mikroskopiske og makroskopiske modeller.

Planlægning af forløbet

På baggrund af den overordnede problemstilling og de faglige mål i fysik har jeg formuleret seks læringsmål for forløbet. Eleverne skal:

1. **Lære at programmere (forstå og skrive computerkode).** Eleverne skal ikke være eksperter i programmering, men de skal stifte bekendtskab med programmering og kunne løse simple programmeringsopgaver.

2. **Skabe et nyt simuleringsværktøj ved at ændre i koden af et eksisterende (være digitale *medskabere*).** Eleverne skal kunne implementere fagligt begrundede ændringer i en udleveret simulering, så den kan anvendes til at undersøge en ny problemstilling.

3. **Kunne forklare fysiskmodellen anvendt i simuleringen.** Eleverne skal kunne udlede henfaldsloven ud fra "eksperimenter" lavet med simuleringsværktøjet samt forklare sammenhængen mellem henfaldsloven (den makroskopiske model) og den fysik/adfærd, der gælder for den enkelte atomkerne (den mikroskopisk model).

4. **Kunne diskutere og vurdere gyldigheden af den anvendte mikroskopiske model i simuleringen.** Det skal gøres med udgangspunkt i en sammenligning af resultater fra eksperiment (som de selv udfører) og simuleringer.

5. **Få en forståelse for den store rolle, som computersimuleringer spiller for moderne naturvidenskabelig forskning (og i samfundet).** Eleverne skal lære, hvad en simulering er, og møde eksempler på, hvor simuleringer bliver anvendt i samfundet. Eleverne skal lære, at resultaterne af en simulering afhænger af de startværdier, man giver den, og af de modeller, som simuleringen bygger på (ligger gemt i computerkoden), og at disse valg har betydning for simuleringens rækkevidde og begrænsninger.

6. **Få en oplevelse af, at simuleringer kan benyttes til bedre at forstå et fysisk fænomen eller en fysisk proces.** Det kan være gennem en visualisering og levendegørelse af processen eller ved at simuleringen muliggør undersøgelse af komplekse sammenhænge, som ellers ville kræve svær matematik. Eller ved at tydeliggøre en ellers abstrakt sammenhæng mellem den makroskopiske og den mikroskopiske model.

Med udgangspunkt i ovenstående læringsmål har jeg planlagt et undersøgelsesbaseret (induktivt) forløb, hvor eleverne med udgangspunkt i simuleringer i NetLogo arbejder med henfaldsloven og model-begrebet i fysik samt arbejder med programmering og de centrale CT-kompetencer. Jeg har skitseret forløbet og dets aktiviteter i figur 2.

	Nr.	Beskrivelse af aktivitet (i overskrifter)	Progression og fane	CT-begreber i anvendelse
Indledende aktiviteter	1a	Indledende oplæg: Modelbegrebet og simuleringers rolle for naturvidenskabelig forskning og i samfundet.	- -	-
	1b	Lege med simuleringen. Beskrive hvordan antallet af radioaktive kerner aftager med tiden. Aflæse halveringstid.	*"use"* *Interface*	-
	1c	Første simple ændringer af computerkoden i simuleringen (ændre atomkernernes udseende).	*"modify" (let)* *Interface + Code*	*Mønstre*
Kobling af mikroskopisk og makroskopisk model	2a	Læse om antagelsen i simuleringen (den mikroskopiske model) samt hvilke variabler, der kan ændres.	- *Info*	*Abstraktion*
	2b	Køre simuleringen med forskellige startværdier. Lave regression og udlede henfaldsloven (makroskopisk model).	*"modify" (let)* *Interface + Code*	*Mønstre og abstraktion*
	2c	Lave eksperiment (Ba*-137). Sammenligne med simuleringerne og diskutere den mikroskopiske models gyldighed.	- -	*Mønstre og abstraktion*
	2d	Læse om henfaldsloven, aktivitet og halveringstid i bogen. Udlede udtryk for halveringstid.	- -	*Abstraktion*
Anvendelse 1: Kulstof 14-datering	3a	Læse om kulstof 14-datering og løse opgaver (uden brug af simuleringer).	- -	-
	3b	Studere koden i simuleringen i større detalje. Arbejde med at forstå algoritmerne (vha. rutediagrammer).	*"modify" (middel)* *Code*	*Abstraktion, mønstre og algoritmedesign*
	3c	Ændre i koden og tilpasse simuleringen, så den kan anvendes til at undersøge C-14 indholdet i organismer.	*"modify" (middel)* *Interface + Code*	*Nedbrydning, abstraktion, mønstre og algoritmedesign*
Anvendelse 2: Henfaldskæde	4a	Læse om Tjernobyl-ulykken. Opskrive henfaldskæden for Te-131 og de tilhørende henfaldsskemaer.	- -	-
	4b	Tilpasse simuleringen og undersøge aktiviteten fra en radioaktiv isotop (I-131) midt i en henfaldskæde.	*"modify" (svær)* *Interface + Code*	*Nedbrydning, abstraktion, mønstre og algoritmedesign*
Afrund-ing	5a	Afrunding og perspektivering. Fordele og begrænsninger ved simuleringer. Modellernes rolle for simuleringen.	- -	*Abstraktion*

Figur 2: Oversigt over forløbets aktiviteter (ikke en moduloversigt). Farverne indikerer de forskellige dele af forløbet. Under "Progression og fane" kan det ses, hvor i progressionen fra figur 3 (use-modify-create) vi arbejder, samt hvilke faner i NetLogo der arbejdes med (Interface, Info eller Code). I den sidste kolonne har jeg forsøgt at angive, hvilke CT-kompetencer der kommer i spil. Forud for forløbet havde vi et forløb om atomets opbygning, henfaldstyper, bevarelsessætninger og henfaldskæder. Desuden havde klassen haft et forløb i matematik om eksponentialfunktioner. Begreberne mikroskopisk og makroskopisk model blev introduceret undervejs i forløbet. Forløbet strakte sig over 12 moduler.

Arbejdet med at tilpasse simuleringerne (læringsmål 2) kræver en grundlæggende forståelse for programmering (det at forstå og skrive computerkode) og algoritmer (den trinvise beskrivelse af handlinger, der udføres i simuleringen). Arbejdet med simuleringerne tager udgangspunkt i "use-modify-create"-modellen vist på figur 3 (Lee 2011).

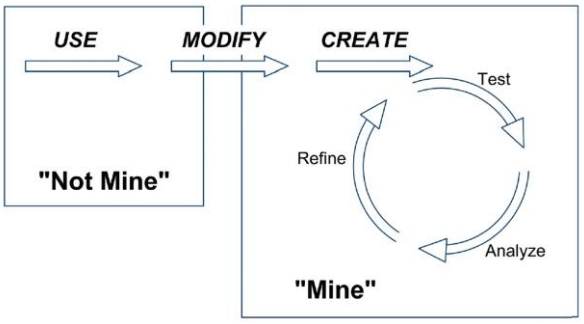
Målet med de første opgaver (aktivitet 1b) er, at eleverne bliver fortrolige med at bruge simuleringsværktøjet og med det fysiske fænomen, der simuleres (radioaktivt henfald). Eleverne er *brugere* af simuleringen (*"use"*).

Derefter skal eleverne i gang med at lave simple ændringer af koden i simuleringen (*"modify"*). Eleverne starter med at ændre de radioaktive kerners udseende (form og farve) og simuleringens startværdier (aktivitet 1c-2b). Her er målet, at eleverne får en klar ide om, at der er en sammenhæng mellem, hvad der står i koden, og hvad der sker i simuleringen. De får snust til koden og ændret i den, uden at de forstår alle dele af koden.

De næste opgaver bliver gradvist mere omfattende og komplicerede (både hvad angår den fysiske forståelse af processerne og hvad angår de ændringer, der skal laves i simuleringskoden). Forud for disse gik en analyse af algoritmerne i den udleverede kode vha. rutediagrammer.

I dette forløb når eleverne kun til "*modify*"-fasen og kommer ikke til at lave et simuleringsværktøj fra bunden ("*create*") - det kommer måske i 2g. I de sidste opgaver vil eleverne dog alligevel komme til at udføre *test-analyse-refine*-cykler, som vist på figur 3 under "*create*"-fasen. De vil i aktivitet 3c og 4b støde på nogle udfordringer, der gør, at deres simulering i første omgang vil fejle. De må derfor af flere omgange analysere, fejlfinde, forbedre og teste deres simulering.

Da programmering er helt nyt for de fleste elever – og mange af eleverne desuden udtrykker en forventning om, at det er *meget svært* – er programmeringsaktiviteterne planlagt med henblik på at give eleverne hurtige mestringsoplevelser (Bandura 2013, Krogh 2017).

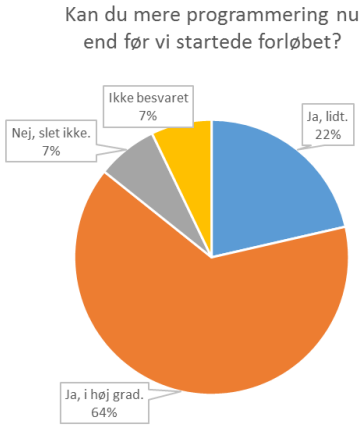


Figur 3: Progression for arbejdet med at tilpasse simuleringerne (ændre i computerkoden). Lånt fra Lee 2011.

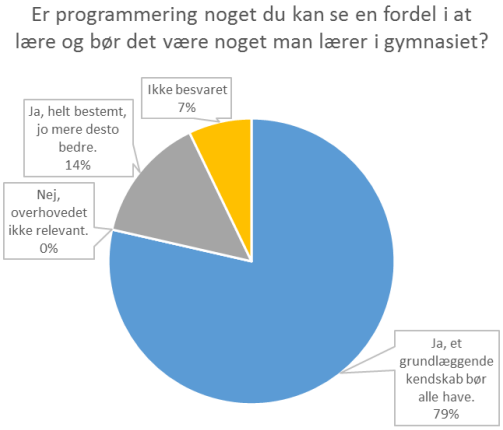
Evaluering

Læringsmål 1:

Eleverne lærte at programmere på et niveau, så de kunne arbejde med simuleringen og implementere ændringer deri. Den bedste indikation herpå er, at alle grupper blev færdige med opgaverne i aktivitet 1c, 2b, 3c og 4b (figur 2). Selv svarer de:



Figur 4: Resultat fra anonym spørgeskemaundersøgelse efter forløbet.



Figur 5: Resultat fra anonym spørgeskema-undersøgelse efter forløbet.

Det er interessant at lægge mærke til, at eleverne finder det relevant at bruge tid på at lære at programmere, og at de har en forventning om, at det er noget, som de kan bruge senere i deres (studie)liv.

De ved nu lidt om, hvad programmering er. Det betyder forhåbentlig, at barrieren for en anden gang at kaste sig ud i et projekt, der kræver programmering, er sænket.

Læringsmål 2:

Jeg kan konstatere, at alle grupper blev færdige med opgaverne og havde velkørende simuleringer, der viste de rigtige grafer og fysiske sammenhænge. De kunne efterfølgende anvende disse nye simuleringsværktøjer til at løse de stillede fysikfaglige opgaver inden for emnet.

Den generelle oplevelse blandt eleverne var, at det var lidt svært, men også sjovt og afvekslende, at arbejde med at lave ændringer i simuleringen. Eleverne tog ret hurtigt ejerskab over deres produkter (de tilrettede simuleringer) og lod sig ikke genere af, at det meste af koden i simuleringen trods alt stadig var urørt af dem. De følte tydeligt, at det var *deres* værk. For eksempel udtaler en af eleverne:

"Jeg har lært at finde rundt i simuleringer. Derudover så har jeg lært selv at programmere en simulering."

Det er tydeligt, at eleverne hurtigt føler sig som *skabere* nærmere end *brugere* af simuleringen.

De sidste opgaver (aktivitet 3c og 4b) krævede mere komplicerede og omfattende ændringer af koden og en dybere forståelse for algoritmerne i simuleringen og fysikken bag. Et par af grupperne blev forbløffende hurtigt færdige, hvilket vidner om et ret godt overblik over koden, algoritmerne og fysiskmodellen. Andre grupper havde brug for mere hjælp til analysen af algoritmerne og nedbrydningen af problemet i mindre dele (en fælles strategi), men kunne dernæst også implementere de rette ændringer i koden.

De ændringer, som nogle af eleverne implementerede i simuleringen, forudsætter, at de har et grundlæggende kendskab til det at læse, forstå og skrive computerkode. Men det kræver mere end blot det. De skal kunne analysere opgaven (det fysikfaglige og det programmeringsmæssige) og dele den op i mindre, veldefinerede dele, som kan løses hver for sig (fx Hvilke dele af den eksisterende kode skal kopieres? Hvorhen i koden skal det kopieres? Hvad skal efterfølgende rettes til?). For at kunne løse opgaven må de også have en god forståelse for den fysik, der er skrevet ind i koden. De skal også kunne forstå algoritmerne i simuleringen (fx hvordan tiden avancerer, og hvordan rækkefølgen af de forskellige handlinger er), identificere dem i koden og tilføje de rette trin de rigtige steder. De skal kunne kæde den virkelige verden sammen med modellen, og modellen sammen med computerkoden. Det kræver et højt niveau af abstraktion. Og er der noget, der ikke virker, som det skal, så skal de analysere koden og finde frem til, hvor fejlen er. Det er centrale CT-kompetencer, der kommer i spil og trænes i denne proces.

Der var undervejs mange gode, konstruktive og lærerige diskussioner om koden, algoritmerne og fysikken i simuleringen, og der blev foreslået mange løsninger, som blev diskuteret, afprøvet og analyseret. En af fordelene ved at anvende simuleringer på denne måde er, at der er meget hurtig feedback på elevernes løsning og deres faglige forståelse. Kan simuleringen køre? Er resultaterne fornuftige? Hvis ikke, skyldes det så en fejl i implementeringen eller fejl i den fysiske forståelse?

Læringsmål 3 og 4

Helt kort kan man sige, at eleverne i hvert fald har lært henfaldsloven på et niveau, som svarer til det, de ville have fået ud af et mere traditionelt forløb om henfaldsloven. Desuden nævner flere af eleverne, at den dynamiske visualisering af processen hjælper dem til en bedre forståelse af henfaldsloven:

"Processen med henfald er visualiseret for en, så det bliver mere håndgribeligt og derfor nemmere at forstå."

Derudover har alle elever fået en generel forståelse for forskellen på mikroskopiske modeller (her modellen for de enkelte radioaktive atomkerner - *at hver radioaktiv kerne for hver tidsenhed har en vis sandsynlighed for at henfalde*) og makroskopiske modeller (her henfaldsloven - *grafen/formlen der beskriver, hvordan antallet af radioaktive kerner aftager med tiden*). De har oplevet, hvordan ændringer i den ene model (fx sandsynligheden for, at der sker et henfald) ændrer den anden (fx grafen for henfaldsloven og halveringstiden). De kan generelt i deres afleveringer forklare, at henfaldskonstanten *k* i henfaldsloven er den samme som den sandsynlighed "Sandsynlighed-for-henfald-N1", som indgår i simuleringen.

Læringsmål 5 og 6:

Eleverne har fået en godt førstehåndsindtryk af, hvad en simulering er. For eksempel skriver en elev:

"*En simulering består af en computerkode og nogle modeller/antagelser. Der skal være et input, en algoritme og et output. Det er en efterligning af virkeligheden / en virkelig proces.*"

Og en anden:

"Vi har lært, hvorfor computeren forstår det, vi fortæller den, og at de følger en bestemt fremgangsmåde (Algoritmer). Vi har lært at forholde os kritisk til computerprogrammer, og at simuleringer kun er så gode som antagelserne."

Eleverne fik præsenteret eksempler på, hvor simuleringer anvendes, både inden for fysikfaget og mere generelt i samfundet (fx samfundsøkonomiske simuleringer i Finansministeriet, klimasimuleringer til at undersøge konsekvenserne af et øget CO_2-indhold i atmosfæren, simuleringer af vejret til vejrudsigten, styrkesimuleringer af bygningskonstruktioner og simuleringer inden for bioinformatik).

Konklusion

Der er en god opfyldelse af de opstillede læringsmål, og det er mit klare indtryk, at eleverne har fået en dybere forståelse af henfaldsloven ved at arbejde med den på denne måde, end de ville have fået ved en mere traditionel gennemgang. Der er for de fleste elever etableret en klar forståelse for forskellen (og sammenhængen) mellem mikroskopiske og makroskopiske modeller og mere konkret, for hvordan henfaldsloven skyldes, at de radioaktive kerner har en bestemt sandsynlighed for at henfalde per tidsenhed - noget der kun står nævnt i en enkelt sætning i lærebogen (FysikABbogen 2). Eleverne har desuden kunnet undersøge mere komplekse processer, end vi ellers ville have kunnet (da den fysikfaglige forståelse nemt ville drukne i matematiske udfordringer).

Eleverne har generelt meldt positivt tilbage på forløbet. Arbejdet med simuleringer i NetLogo blev set som en udfordrende, men relevant og afvekslende aktivitet. Både det at programmere og det at arbejde med simuleringer som værktøj til at undersøge og forklare fysiske fænomener og processer. Mange har fået pirret nysgerrigheden og har meldt tilbage, at de gerne vil arbejde med simuleringer i NetLogo igen.

I arbejdet med simuleringerne har eleverne udviklet væsentlige kompetencer i programmering og inden for computational thinking. Jeg er ikke i tvivl om at disse færdigheder vil komme dem til gode i fysikfaget senere hen. Det store spørgsmål er, hvorvidt eleverne kan udnytte disse kompetencer i andre fag? Og hvordan vi i så fald kan evaluere det? Det er interessante spørgsmål, som jeg ikke har noget godt svar på (endnu).

Alt i alt er svaret på problemstillingen altså: Ja, det er muligt at arbejde med computational thinking (CT) i fysikfaget på en måde, så det udover at styrke elevernes CT-kompetencer også bidrager til et øget fysikfagligt udbytte! Der er god synergi mellem arbejdet med de faglige mål i fysik og arbejdet med computational thinking, og den induktive, undersøgende tilgang kombineret med inddragelsen af simuleringsværktøjer, som eleverne selv kan ændre og tilpasse, har vist sig at være en stærk kombination.

Kan man som lærer anvende NetLogo i undervisningen, hvis man ikke i forvejen har erfaring med programmering? Ja, det kan man godt. Agentbaseret modellering har en lav indlæringstærskel (både for elever og lærere), og der er lavet mange simuleringer inden for mange forskellige fag, som man kan tage udgangspunkt i (se *Materialer* nedenfor).

Materialer

Du kan hente og installere NetLogo fra https://ccl.northwestern.edu/netlogo. Når du åbner NetLogo, kan du under *File* og *Models Library* finde et bibliotek af modeller (simuleringer) til forskellige fag. Der er endnu flere modeller at finde på siderne https://github.com/NetLogo/models, http://modelingcommons.org/ og https://ccl.northwestern.edu/netlogo/models/community. De NetLogo-filer, som jeg har anvendt til dette forløb, kan findes på SG's "Web 2.0"-blog https://potentialeriweb20.wordpress.com/ ved at søge på NetLogo. Her er der også vejledninger. De anvendte arbejdsark udleveres ved henvendelse.

Litteraturliste:

Astra (Nationalt Center for Undervisning i Natur, Teknik og Sundhed) (2017): *Sammen om naturvidenskab - Anbefalinger til en national strategi for de naturvidenskabelige fag.* Link: https://astra.dk/sites/default/files/nns_rapport_anbefalinger_final_web.pdf (besøgt 28/5, 2018).

Bandura, Albert (2013): "Self-efficacy", in: *Kognition & pædagogik*, Årg, 22, nr. 83, s. 16-35. Oversat af Tom Havemann.

Barefoot (2018): *Computational Thinking*. Link: https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/ (besøgt 10/6-2018).

Caspersen, Michael E. (2017): "Computational thinking", in: Dolin, Jens (red.) m.fl. (2017): *Gymnasiepæ-dagogik - En grundbog*, Hans Reitzels Forlag, 3. udg., s. 470-478.

CCTD (Center for Computational Thinking and Design), Aarhus Universitet (2018): *CT I Gymnasiefag (afsluttende rapport)*. Link: http://cctd.au.dk/fileadmin/user_upload/CT_i_Gymnasiefag-afslutningsrapport-ur.pdf (besøgt 27/5, 2018).

Google (2018): Computational Thinking for Educators. Link: https://computationalthinking-course.withgoogle.com/course (besøgt 10/6-2018).

Krogh, Lars Brian og Andersen, Hanne Møller (2017): "Motivation", in: Dolin, Jens (red.) m.fl. (2017): *Gymnasiepædagogik - En grundbog*, Hans Reitzels Forlag, 3. udg., s. 250-267.

Lee, Irene m.fl. (2011): "Computational Thinking for Youth in Practice", in: *ACM Inroads*, Vol. 2, No. 1, s. 32-37.

Next Gurukul (2017): *What is Computational Thinking?* Link: https://www.nextgurukul.in/Knowledge-World/computer-masti/what-is-computational-thinking/ (besøgt 11/6-2018).

Papert, Seymour (1980): *Mindstorms: Children, computers, and powerful ideas*, Basic Books.

Petropouleas, Eva (2018): *Computational Thinking*. Link: https://www.mv-nordic.com/dk/blog-gen/artikel/computational-thinking/ (besøgt 11/6-2018).

Wing, Jeannette M. (2006): "Computational thinking", in: *Communications of the ACM*, Vol. 49, No. 3, s. 33-35.

Wing, Jeannette M. (2014): "Computational Thinking Benefits Society", in: *Social Issues in Computing* (2014). Link: http://socialissues.cs.toronto.edu/index.html%3Fp=279.html (besøgt 28/5, 2018).

Pedersen, Jens Olaf Pepke og Andersen, Michael Cramer (2016): "Tyngdebølger observeret for første gang", in: *KVANT*, Årg. 27, nr. 1, s. 8-12.