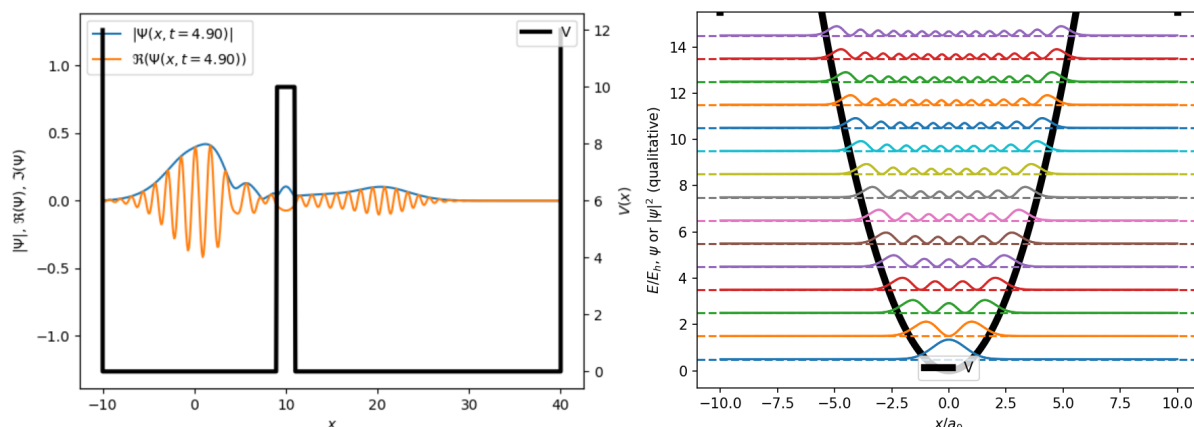


Rapport: utvikling av numeriske øvinger til kurs i kvantefysikk

Herman Sletmoen

19. januar 2020



Figur 1: De stasjonære tilstandene i en harmonisk oscillator og tunnellingen av en gaussisk bølgepakke gjennom en potensialbarriere er to eksempler på fysiske situasjoner som studentene skal utforske i de numeriske øvingene.

- **Oppdragsgiver:** Jon Andreas Støvneng, foreleser i *Innføring i kvantefysikk*
- **Oppdragstaker:** Herman Sletmoen, 3. årsstudent i *Fysikk og matematikk*
- **Oppdragets varighet:** kontinuerlig gjennom høstsemesteret 2019

1 Oppdrag og oppdragets mål

Oppdraget har bestått av å utvikle tre numeriske øvinger til kurset *Innføring i kvantefysikk* (TFY4215) med tilhørende løsningsforslag. Hver øving tar i hovedsak for seg (1) utvikling av en “maskin” som er i stand til å løse Schrödingers tidsuavhengige eller tidsavhengige ligning,

$$\hat{H}\psi = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2} + V\psi = E\psi \quad \text{eller} \quad i\hbar \frac{\partial \Psi}{\partial t} = \hat{H}\Psi,$$

for et vilkårlig potensiale V og en vilkårlig initialbetingelse $\Psi(x, t = 0)$ og (2) bruk av denne “maskinen” på en rekke konkrete eksempler med relevante potensialer og initialbetingelser. Øvingene kommer i form av Jupyter Notebooks og forklarer all nødvendig teori bak fysikken og de numeriske metodene, som dessuten vil være kjent for studentene fra forelesninger. Studentenes jobb er å sette seg inn i teorien og skrive nødvendig programkode for å løse oppgavene.

Det primære målet er at studentene gjennom arbeid med øvingene skal bruke numeriske metoder som et *verktøy* til å *forstå* kvantemekanikk bedre. Øvingene er bevisst strukturert slik at studentene skal se Schrödingerligningene og løsninger av disse fra et generelt perspektiv. Forhåpentligvis vil det være en tilfredsstillende tankevekker for mange å se hvordan numeriske metoder gjør det mulig å løse ligningene “en gang for alle”, i motsetning til de spesielle og vanskelige (eller umulige!) analytiske løsningsmetodene. Gjennom denne prosessen vil studentene også utvikle sine ferdigheter innen programmering og numerikk.

Øvingene oppfordrer til bruk av programmeringsspråket Python med bibliotekene Matplotlib, Numpy og Scipy for plotting, numerikk og vitenskapelige beregninger, og det forventes at studentene selv finner nødvendig funksjonalitet og dokumentasjon i bibliotekene. Samtidig er øvingene forsøkt formulert uten knytning til én konkret teknisk løsning, og det er fullt mulig å lese oppgaveteksten og deretter løse oppgaven med andre språk og biblioteker.

Øvingene er tilgjengelige gjennom ressursene til slutt i denne rapporten.

2 Bakgrunn

Det er for tiden et sentralt mål å integrere numeriske metoder i grunnleggende fysikkurs ved NTNU og andre norske universiteter. Majoriteten av TFY4215-kursets deltakere er (tidligst) i 3. semester og har gjennomført numeriske øvinger i flere fysikkurs i de foregående semesterene. De numeriske øvingene i TFY4215 er utformet med en bevisst tanke om at de skal føye seg inn i denne rekken som en naturlig fortsettelse med større arbeidsmengde, krav til selvstendighet og vanskelighetsnivå enn i de første semestrene.

3 Arbeidsforhold

Arbeidet har vært av en svært selvstendig karakter og gitt meg stor frihet under ansvar. Oppdragsgiver Jon Andreas overlot til meg å selv bestemme innholdet i øvingene, men var alltid tilgjengelig for å diskutere, godkjenne og foreslå endringer til innholdet. Denne arbeidsmåten har vært svært motiverende for meg, da jeg føler jeg har fått bidra til utviklingen av kurset med mine egne idéer.

Arbeidstiden har vært svært fleksibel. Den store friheten har latt meg arbeide med øvingene når jeg selv har hatt tid, og Jon Andreas har hatt forståelse for min allerede travle hverdag. Vi to har møttes for å diskutere framgangen i arbeidet noen ganger i løpet av semesteret da vi begge hadde tid, og holdt ellers en løpende dialog på epost.

Det ble i forkant bestemt at arbeidet skulle belønnes med en lønn tilsvarende 60 arbeidstimer, og jeg ble bedt om å bruke dette som en øvre begrensning på arbeidstiden min. Jeg anslår å ha brukt omtrent like mange timer på arbeidet. Takket være friheten i arbeidet, har jeg hatt god mulighet til å kontrollere arbeidsmengden selv.

4 Erfaringer

I skrivende stund har øvingene ennå ikke vært prøvd ut i praksis, så det har foreløpig ikke vært mulig å hente inn tilbakemeldinger fra studenter. Allikevel har jeg gjennom utviklingen av øvingene opparbeidet en rekke erfaringer i forhold til numeriske øvinger generelt og bruk av Jupyter Notebooks som format for slike øvinger.

- En betydelig arbeidsmengde har vært knyttet til å løse tekniske utfordringer relatert til Jupyter Notebook og kompatibilitet mellom Jupyter Notebook og Jupyter Lab – et nytt lignende program som på sikt skal erstatte Jupyter Notebook. Det har vært viktig å unngå å snekre sammen upålitelige “ad-hoc”-løsninger, men heller lete fram riktige og robuste offisielle løsninger som er ment å fungere både nå og i framtiden, slik at studentene ikke skal bruke tid på teknisk feilsøking. Alle utfordringene har imidlertid latt seg løse på tilfredsstillende måter. Etttersom skiftet til Jupyter Lab gjennomføres og programvaren modnes ytterligere, vil forhåpentligvis slike tekniske irritasjonsmomenter forsvinne og heller ikke kreve inngrep fra brukersiden.

- For at figurene som produseres av Matplotlib skal være mest mulig tilfredsstillende, må det brukes en *interaktiv backend*. Dette gjelder spesielt ved animering av figurer. I Jupyter Notebook aktiveres en interaktiv backend enkelt med kommandoen `%matplotlib notebook`, og alt fungerer slik det burde uten behov for ytterligere inngrep fra brukeren. I Jupyter Lab bør man bruke `%matplotlib widget`, men dette krever installasjon av tillegget `ipyml` i Jupyter Lab. Siste løsning i begge programmene er å falle tilbake til den ikke-interaktive backenden `%matplotlib inline`, men dette gir mindre tilfredsstillende figurer og større begrensninger

ved animering av figurer. I framtiden vil det forhåpentligvis ikke være nødvendig med verken utførelse av en slik kommando eller installasjon av tilleggsprogramvare for å produsere optimale figurer.

- Jupyter Notebook og Jupyter Lab viser på enkelte datamaskiner ikke matematiske ligninger på en tilfredsstillende måte. Det er MathJax som brukes for å vise ligningene, og resultatet kan på mange maskiner forbedres ved å høyreklikke på en ligning og bytte tegnemotor. MathJax fungerer som regel utmerket på andre nettsider, så dette burde være en feil som rettes på sikt.
 - Matplotlib er ikke spesielt godt egnet til animasjoner og interaktive figurer. Bruk av biblioteket til slike formål føles til tider “ad-hoc”, og det er heller ikke imponerende raskt. I disse øvingene løses dette problemet delvis ved å inkludere et fungerende animasjonsrammeverk (basert på Matplotlib) som studentene så skal modifisere til å utføre de ønskede handlingene, uten at de trenger å sette seg inn i alle de tekniske detaljene knyttet til dette. Det kan også være en idé å se til andre biblioteker som er bedre egnet for interaktive figurer og animasjoner. Det er derimot Matplotlib som vil være kjent for studentene fra tidligere fysikkurs, så et alternativt bibliotek bør i så fall være enkelt i bruk.
 - Levering av øvingene i notebook-format åpner for konflikter mellom versjoner av Python og de nødvendige bibliotekene, da programkoden i en notebook må kjøres for at resultatene skal vises. Dette løses enkelt ved å eksportere notebooken til et HTML-dokument før innlevering. Dokumentene inneholder all tekst, programkode og figurer og garanterer slik like resultater på alle skjermer uten behov for å kjøre andres programkode. Merk at animasjoner må bygges inn i HTML-filen som en JavaScript-animasjon eller videofil for at de skal følge med i eksporteringen (noe som i disse øvingene enkelt kan gjøres med det inkluderte animasjonsrammeverket). Jupyter Notebook og Jupyter Lab inkluderer også løsninger for eksportering til PDF, men dette krever installasjon av ekstra programvare og gir dessuten ikke tilfredsstillende resultater. Det er heller ingenting i veien for å gi studentene en alternativ mulighet til å få øvingene godkjent ved å vise fram arbeidet i en øvingstime.
- Det er vanskelig å finne oppgaver som passer alles ferdighetsnivå. Det er allerede forskjeller i fysikknivået blant studentene, men forskjellene forsterkes ytterligere i numeriske øvinger av en potensielt enda større forskjell i programmeringsferdigheter. Den generelle strukturen på oppgavene vil trolig også være utfordrende for enkelte studenter. For å gjøre øvingene overkommelige for alle, gis studentene frie tøyler til å løse oppgavene slik de selv vil gjennom en åpen oppgavetekst uten større spesifikke krav.
 - Jupyter Notebooks er et tilfredsstillende format for numeriske fysikkøvinger. Formatet lar en kombinere tekst, matematiske ligninger og programkode med genererte figurer, animasjoner og lignende i ett sammenhengende dokument. Dette er enklere enn å manuelt legge inn tekst, ligninger, programkode og figurer i et \LaTeX -dokument, og gjør det dessuten mulig å bruke interaktive figurer og animasjoner.
 - Gjennom bevisst strukturering av numeriske øvingsoppgaver, kan man understreke den generelle naturen til fysikkens grunnleggende ligninger. Utviklingen av en generell “maskin” som løser ligningene er i seg selv arbeidskrevende og må typisk gjøres parallellt med et eksempel eller to, men når dette først er gjennomført er det enkelt å “putte inn” vilkårlige eksempler i denne. Dette åpner for muligheten til å utforske mange konkrete fysiske situasjoner uten nevneverdig større arbeidsbelastning.

5 Nedlastning av øvingene

Oppgavene vil i løpet av våren 2020 bli tilgjengelig på kursets hjemmeside. De er også tilgjengelige fra folk.ntnu.no/hermas1/TFY4215_complex/. Løsningsforslagene vil ikke (alltid) være tilgjengelig disse stedene, da øvingene vil være obligatoriske innleveringsoppgaver. Ta kontakt med hermansletmoen@gmail.com om du også skulle være interessert i løsningsforslagene.