Søknad om Meritteringsordning for utdanningsfaglig kompetanse ved Universitetet i Oslo

Morten Hjorth-Jensen*

Department of Physics and Center for Computing in Science Education,
University of Oslo, N-0316 Oslo, Norway and
Department of Physics and Astronomy and National Superconducting
Cyclotron Laboratory and Facility for Rare Ion Beams,
Michigan State University, East Lansing, MI 48824, USA

(Dated: May 15)

* mhjensen@uio.no

KORT INTRODUKSJON

Denne søknaden inneholder først en kort oversikt om undertegnede med bakgrunn og historikk. Deretter følger sjølve søknadsteksten og til slutt har jeg lagt ved den pedagogiske mappa.

BAKGRUNNSINFORMASJON

Jeg har vært ansatt ved Fysisk Institutt ved Universitet i Oslo siden januar 1999, først som førsteamanuensis og deretter som professor fra mai 2001.

Fra og med januar 2012 har jeg delt tida mi mellom Michigan State University (MSU) og UiO. Jeg har et professorat i fysikk begge steder og tilbringer tida januar-juni i USA og juli-desember i Norge. Jeg har en redusert stilling ved UiO. Jeg underviser på alle nivå begge steder og vegleder laveregradsstudenter, masterstudenter, PhD studenter og Post-docs både i Norge og i USA heile året.

Jeg er utdanna Sivilingeniør fra NTNU i Trondheim i 1988 og disputerte for PhD-graden i desember 1993 ved UiO.

I tida januar 1994 til desember 1998 var jeg post-doc ved European Center for Theoretical Studies in Nuclear Physics (Trento, Italia, 1994-1996) og deretter Nordita (København, Danmark, 1996-1998).

Jeg er innvalgt medlem av **Det Norske Videnskaps-Akademi** og **Det Kongelige Norske Videnskabers Selskab**.

Mitt engasjement i undervisning, undervisningsforskning og utvikling av aktive læringsmiljø, studiemiljø og studieprogrammer har resultert i flere utdanningspriser i Norge og USA. Mye av dette blir diskutert i sjølve søknaden også.

Utdanningsspriser

1. UiOs utdanningspris i 2000, (250kNOK)

2. UiOs utdanningspris i 2011 for utvikling of Computing in Science Education prosjektet (250kNOK)

3. NOKUTs pris for fremragende utdanning 2012 for **Computing in Science Education** prosjektet
4. UiOs utdanningspris 2015 for utvikling av studie og læringsmiljøet i Computational Physics ved Fysisk Institutt (250kNOK)
5. Favorite Graduate Teacher Award at Department of Physics and Astronomy, Michigan State University, 2016
6. Olav Thon Stiftelsen, Nasjonal utdanningspris 2018
7. Thomas H. Osgood Faculty Teaching award at Michigan State University, 2018

Jeg har siden jeg blei ansatt ved UiO i 1999 og ved Michigan State University i 2012, vegleda og vegleder over 100 Master, PhD studenter og post-docs. Over 50% av masterstudentene har fortsatt med PhD-studier enten med meg som vegleder eller andre i Norge og internasjonalt.

En fullstendig CV finnes på `http://mhjgit.github.io/info/doc/pub/cv/html/cv.html`.

SØKNADEN: BESKRIVELSE AV UNDERVISNINGSARBEID, UTVIKLING OG FORSKNING

I de 20 åra jeg har vært ansatt ved UiO og siden 2012 ved MSU, har jeg hatt og har et stort fokus på utdanning, utvikling av læringsmiljøer, nye undervisningsformer, innovasjon i utdanning, samt utvikling av utdanningsprogrammer i henhold til samfunnets kompetansekrav og spesifikke behov i ulike fagfelt.

Jeg har utvikla flere nye kurs, samt starta nye studieprogram og bidratt til å utvikle internasjonale kurstilbud i eget forskningsfelt, i tett samarbeid med kolleger på tvers av disipliner nasjonalt og internasjonalt.

Overordna målsettinger ved mitt utdanningsarbeid

Det er ti overordna målsettinger ved mitt undervisningsarbeid ved UiO og MSU:

1. Gi studentene ei forståing av den vitenskapelige metoden så tidlig som mulig i studieløpet. Dette lar seg gjøre spesielt gjennom prosjektorientert undervisning med

tett oppfølging og tilbakemelding til studentene. Her spiller mitt initiav til å starte Computing in Science Education prosjektet en sentral rolle. Dette er beskrevet nedenfor i større detalj.

2. Gi studentene kompetanse, faglig trygghet og innsikt som kreves for å løse naturvitenskapelige og teknologiske problem for det 21ste århundre, spesielt med tanke på digital kompetanse. Her har min rolle som programrådsleder i to store studieprogram vært, og er, svært viktig.

3. Gi studentene en god etisk holdning til deres arbeid, samt å utvikle kritisk tenkende mennesker med djup innsikt i alle sider av den vitenskapelige prosessen.

4. Å utvikle gode utdanningsprogrammer og faglig progresjon i studieløpa, i tett samarbeid med kolleger ved Fysisk institutt (både i Norge og i USA) og på tvers av disipliner.

5. Å sørge for at det faglige innholdet er i tråd med universitetenes samfunnsoppgave, ved å utdanne svært ettertraktede kandidater til forskning, utdanning, offentlig og privat sektor.

6. Å sørge for at det faglige innholdet har intellektuelle utfordringer og innhold i tråd med nåtidige og framtidige faglige behov og forskningsretninger.

7. Å utvikle ei kunnskapsbasert tilnærning til læring, samt utvikle forskningsprogram om hva som gir studentene økt innsikt og faglig forståelse.

8. Å utvikle et internasjonalt perspektiv på utdanninga vår.

9. Kunnskapen skal være fritt tilgjengelig for alle.

10. Pedagogiske tiltak skal være forskningsbaserte og/eller initiere ny forskning.

Disse ti overordna målsettingene er beskrevet i større detalj nedenfor. Flere av disse overordna målsettingene inngår i beskrivelsen av tiltak jeg har initiert, samt kurs og utdanningsmiljø jeg har utvikla.

Faglig fornying av utdanning og arbeid med studieprogram

For å oppnå måla om digital kompetanse, prosjektorientert utdanning, tett oppfølging av studenter og mange flere av de ti måla ovafor var jeg i 2003 en av de sentrale initiativtakerne til prosjektet **Computing in Science Education** (CSE), sammen med kolleger på Fysisk Institutt (Arnt Inge Vistnes), Matematisk Institutt (John Grue) og Institutt for Informatikk (Hans Petter Langtangen og Knut Mørken). Mye av dette skjedde via senteret for fremragende forskning (SFF) **Center for Mathematics for Applications** og i min rolle som programrådsleder for Bachelor programmet Fysikk, Astronomi og Meteorologi (FAM). De første midlene som blei tildelt til CSE prosjektet var via UiO tiltaket **Fleksibel læring** i 2004. Undertegnede var prosjektleder sammen med programrådsleder for MIT programmet John Grue. Jeg satt også som programrådsmedlem i MIT styret og var med å koordinere innføringa av CSE prosjektet. Senteret CMA spilte en sental rolle i begynnelsen og finasierte deler av stillinger også. Seinere kom også et anna SFF inn på banen, Physics of Geological Processes (PGP) og Anders Malthe-Sørenssen ved Fysisk Institutt. Anders var sentral i implemeteringen av CSE prosjektet i det første fysikkurset FYS-MEK1100.

Jeg var programrådsleder for FAM i tida 2002-2011 og var drifkraft og overordna ansvarlig for integrering av et beregningsperspektiv i FAM programmet. Dette har resultert i at fysikkfaget ved UiO har utdanna studenter med de digitale kompetansene som trengs for å møte de vitenskapelige og teknologiske utfordringene i det 21ste århundre. CSE prosjektet har fungert som inspirasjon for alle andre studieprogram ved MatNat-fakultetet og andre fakultet ved UiO og universitet nasjonalt.

CSE prosjeket er nå gjenspeilt i omtrent alle studieprogram ved MatNat-fakultetet ved UiO. I tillegg har jeg leda en komite ved fysisk institutt på Michigan State University (2018) om innføring av CSE-liknende tiltak ved dette universitetet. I vår underviste jeg et sentralt laveregrads kurs i klassisk mekanikk med full CSE implementering ved Michigan State University, til positiv respons fra studentene, se fagevalueringa i den pedagogiske mappa.

CSE prosjektet fikk UiO sin utdanningspris i 2011, samt NOKUT sin utdanningspris i 2012. Prosjektet resulterte i at UiO fikk et senter for fremragende utdanning i 2016, senteret for **Computing in Science Education** (CCSE). Senteret har blitt en stor suksess og har utvikla flere nye forskningsprosjekt om utdanning. Her er jeg involvert i prosjekt om kvantitativ utdanningsforksning samt kursutvikling.

Som et ledd i å styrke UiO sin utdanning i digital kompetanse, videreføre CSE prosjektet, samt utvikle masterprogrammer i beregninger og databehandling, tok jeg, sammen med kolleager fra Fysisk Institutt (Anders Malthe-Sørenssen), Institutt for Biovitenskap (Marianne Fyhn), Institutt for Matematikk (Knut Mørken) og Institutt for Informatikk (Hans Petter Langtangen) i mai 2015 initiativ til å etablere det nye Masterprogrammet Computational Science ved UiO. Programmet starta med det første kullet høsten 2018, og har hittil vært en stor suksess, og et viktig satsningsfelt for MatNat-fakultetet ved UiO. Jeg er programrådsleder for dette programmet og har jobba intenst med å utvikle og integrere kunnskap og kurs i beregningsorienterte fag. Jeg har sjøl utvikla et svært så populært kurs i maskinlæring (FYS-STK4155) for dette studieprogrammet.

Dette studieprogrammet er et samarbeid med alle institutt på MatNat-fakultetet unntatt Farmasi. CS-programmet har 10 studieretninger fordelt på sju institutt (ITA, Biovitenskap, Fysikk, Geovitenskap, Informatikk, Kjemi og Matematikk). Etter to års forarbeid (høst 2015-vår 2018) klarte vi å utvikle et faglig spennende studieprogram. Kandidatene har ofte jobbtilbud et år før de er ferdige med sin utdanning. Jeg har leda arbeidet på tvers av instituttgrenser og disiplinære grenser.

CS-programmet var også fra min side det første strategiske trinnet i å utvikle et heilhetlig tilbud til våre studenter i det som på engelsk kalles Computational Science and Data Science, på tvers av disipliner. Det neste steget var lanseringa av et nytt institutt i Computational Science og Data Science våren 2018, se materialet med white paper på `https://computationalscienceuio.github.io/CCAD/doc/pub/whitepaper/html/whitepaper-bs.html`. Mange av de opprinnelige initiativtakerne for CS-programmet var også sterkt delaktige i å utvikle grunnlagsmaterialet som seinere (høst 2018-vår 2019) resulterte i ei innstilling om et nytt senter i Data Science og Computational Science ved MatNat-fakultetet ved UiO. Senteret antas å ha oppstart i 2021 og vil spille ei viktig rolle i utdanningstiltak som fokuserer på digital kompetanse for heile universitetet. I tillegg planlegges det flere etterutdanningstiltak retta mot både privat og offentlig sektor.

Om dette lykkes, vil det bety ei videreføring og videreutvikling av CSE prosjektet til å dekke alle utdanningstrinn ved UiO. Å utvikle ansattes og studentenes digitale kompetanse er et sentralt element i vår faglige utvikling og vårt arbeid.

Utvikling av studiemiljø

Siden 2000 har jeg aktivt jobba for å utvikle et utdanningsmiljø i Computational Physics på mastergradsnivå. Fra og med 2007 har dette arbeidet vært gjort sammen med en nær kollega, Anders Malthe-Sørenssen, som også er direktør for CCSE. I 2015 blei dette arbeidet tilkjent UiO sin utdanningspris, se `https://www.uniforum.uio.no/nyheter/2015/10/instituttet-som-lofter-fram-gode-forelesere.html`.

Vi har spesielt vektlagt

1. Utvikle et godt sosialt miljø hvor deling av resultat og programvare står sterkt. Studentene definerer ofte sine egen prosjekt for masteroppgavene sine.

2. Oppbygging av lokaler som er imøtekommende og inkluderende. Bygd opp mange arenaer for sosialt samvær.

3. Studentene integreres tidlig i forskninga og studentene fungerer som læremestre for hverandre.

4. Studentene engasjeres tidlig i å utvikle læringsmateriale, spesielt som gruppelærere i kurs hvor beregninger (CSE prosjektet) er viktig.

Med etableringen av det nye CS-masterprogrammet har vi nå utvikla nye tiltak for å forbedre studiekvaliteten, med blant annet tett integrering av studentene i utforming av nye oppgaver og prosjekter til mastergradskursene, tett kopling mellom studenter og fagmiljø, samt mentorprogram i oppstart av masterprogrammet, med vekt på mulige fagvalg og karriereveger. Individuell oppfølging av studentene spiller en sentral rolle, spesielt også for studenter som sliter mentalt. Tett oppfølging og individuell tilrettelegging er et sentralt aspekt som vektlegges i utviklinga av et godt studiemiljø.

Studieprogrammet har sosiale spillkvelder med faglige seminar annen hver fredag. Her møter studentene forskere og/eller potensielle oppdragsgivere fra privat og offentlig sektor. Og i en del tilfeller har det resultert i sommerjobber og kanskje fast ansettelse seinere.

Computational Physics miljøet har siden 2003 utdanna over 100 mastergradsstudenter, og veldig mange av disse (over 50%) har fortsatt med PhD studier. Det er tett samarbeid mellom bachelor studenter, mastergrads studenter, PhD studenter og Post-docs.

Utvikling av kurs med prosjektbasert innhold

Over to tiår har jeg utvikla kurs med en prosjektbasert profil.

Da jeg underviste Kvantefysikk FYS2140 i perioden 1999-2004 introduserte jeg numeriske prosjekt som studentene jobba med. På den tida var dette ganske nytt ved UiO og resulterte blant annet i UiO sin utdanningspris i 2000, etter bare litt over ett år som ansatt ved UiO. Mye av dette arbeidet la grunnlaget for ideer og tiltak rundt CSE prosjektet i 2003. Numeriske prosjekt tillater studenter å utdjupe sin faglige innsikt på et heilt anna vis enn gjennom tradisjonelle papir og blyant oppgaver. En har en stor mulighet til å teste på et djupere nivå studentenes innsikt i et fysisk fenomen og er et ypperlig pedagogisk verktøy for å utvikle studentenes innsikt og forståelse av den vitenskapelige metoden. Med programmering har en også mulighet til å bringe inn mer realistiske problemstillinger på et tidligere stadium av utdanninga. Ofte møter ikke studentene forskninsgrelatert utdanning før de begynner på sine masterprosjekter. Med et prosjektbasert løp spiller tilbakemeldinger på arbeidene deres og tett kontakt med faglærere en sentral rolle. Tilbakemeldingene fra studentene er også avgjørende for forbedring av læringsmål og utdanningsmateriale.

Prosjektbasert undervisning er også sentralt i kursene jeg har utvikla fra scratch, FYS3150 Computational Physics I, FYS4411 Computational Physics II og FYS-STK4155 Applied Data Analysis and Machine Learning som jeg underviser ved UiO. FYS3150 og FYS-STK4155 er kurs med over 100 studenter hver og studentene jobber med 3-5 prosjekter gjennom heile semesteret. Prosjekta er lagt opp som vitenskapelige arbeider og for det siste prosjektet kan studentene ofte definere tema sjøl. Dette gir studentene en stor frihet i å utforske egne veger og kople undervisninga opp til eventuell forskning. Omfattende tilbakemeldinger på prosjekta med tanke på evalueringa spiller ei sentral rolle og er en tidkrevende, men en viktig del av disse kursa. Disse kursa er faktisk noen av de få ved UiO hvor studentene får lange tilbakemeldinger med begrunnelse for endelig karakter.

Alt utdanningsmateriale, forelesningsnotater, programmer, kildekode og mye mer er fritt tilgjengelig for studentene. Studentene kan feks bruke kildekoden til notata til å lage sine egne elektroniske notatbøker (ofte i form av en såkalla jupyter-notebook). Det styrker den ovennevnte delingskulturen og fungerer som et godt eksempel for studentene om deling og samarbeid om vitenskapelige resultat. I tillegg gir moderne versjonskontroll programvare en unik mulighet til å fokusere på utvalgte aspekt av vitenskapelig etikk, som reproduserbarhet av vitenskapelige resultat og korrekt behandling av kildemateriale. Eksempler på hvordan materiale er gjort fritt tilgjengelig kan ses på min GitHub adresse `https://github.com/mhjensen`. Her ligger undervisningsmateriale for flere kurs, og ved hjelp av versjonskontroll programvaren **git** kan alt materiale lastes ned med enkle tastetrykk.

At alt utdanningsmateriale er fritt tilgjengelig forenkler også læringsprosessen for studentene.

Det neste utvklingstrinnet i min undervisning er å utvikle et miljø for ei såkalla **Flipped Classroom** tilnærming. Dette blir også gjort som et mulig tiltak i anledning et eventuelt fullt eller delvis digitalt undervisningsløp høsten 2020. Dette blir implementert til høsten 2020 for kursene FYS3150 og FYS-STK4155 og innebærer ny produksjon av visuelt materiale i form av videoer som er kopla opp mot allerede eksisterende digitalt materiale. Studentene foventes å gå gjennom ukentlig oppgitt materiale før de eventuelt møter i mindre grupper for å diskutere materialet og jobbe med prosjekter og oppgaver. Det gir oss som faglærere en mulighet til enda tettere kopling opp mot studentene for å følge deres læring. Flipped Classroom er et spennende pedagogisk område, med mange interessante forskningsbaserte resultat, blant anna med tanke på økt læringsutbytte for studentene.

Jeg har også utvikla en sterk prosjektbasert profil i kurset Classical Mechanics PHY 321 som jeg underviser ved Michigan State University, med veldig positive tilbakemeldinger fra studentene (se kursevaluering i pedagogisk mappe).

Til slutt har jeg laga et nytt forslag til første studieår i fysikk som åpner for en ny integrering av teori, eksperiment og beregninger. Dette er ganske nytt og banebrytende og danner også grunnlag for spennende forskningsprosjekter, spesielt med tanke på den overordna forståelsen av den vitenskapelige prosessen. Forslaget er beskrevet på `https://mhjensen.github.io/FirstYearPhysicsUiO/doc/pub/proposal/html/proposal-bs.html`.

Dette forslaget føyer seg inn i Fysisk Institutt sin strategiske fornyelse av bachelor programmet i Fysikk og Astronomi. En svakhet i dag er at det eksperimentelle elementet er mindre framtredende enn ved andre universitet. Og fysikk er et eksperimentelt fag. Med moderne programmeringsspråk, samt hardware som tillater en å gjøre mange eksperiment (feks aksellerometer i mobiltelefon) kan en integrere beregninger, eksperiment og teori på et heilt anna vis enn tidligere. Studentene kan gjøre mange av eksperimenta med feks mobiltelefonene sine, nesten hvor som helst. Dette åpner opp for en tydeligere integrering av alle steg i et vitenskapelig studie og kan implementeres allerede fra første semester.

Internasjonale tiltak

Sammen med kolleger fra flere land starta jeg og etablerte et internasjonalt initiativ i 2010 kalt **Nuclear TALENT (Training in Advanced Low-Energy Nuclear Theory)** hvis mål er å styrke en faglig bredde i kjernefysikk internasjonalt. Mange universitet har ikke nok vitenskapelig personale til å gi studentene på master og PhD nivå den nødvendige faglige bredde i feltet. Nuclear TALENTs mål er å tilby denne faglige bredden i form av et titalls avanserte kurs som undervises på et intensivt vis over tre uker ulike steder i verden (Nord-Amerika, Europa og Asia). Siden sommeren 2012 har vi organisert over 15 slike kurs og jeg har undervist og organisert 5 av disse kursene og organisert tre andre. Dette tiltaket har vært en enorm suksess med over 500 deltakere totalt siden 2012. Pga COVID-19 er alle tre kurs i år utsatt til 2021, men kurset jeg har ansvaret for i år om maskinlæring anvendt på kjernefysikk tilbys digitalt i juni-juli 2020, se URL:"http://www.ectstar.eu/node/4472".

For mer informasjon om Nuclear TALENT, se https://fribtheoryalliance.org/TALENT/.

Ellers leder jeg et større INTPART prosjekt om Computing in Science Education mellom CCSE ved UiO, Michigan State University, Oregon State University og University of Colorado ved Boulder. Et viktig mål med dette prosjektet er å utvikle et program i kvantitativ utdanningsforskning. Vår første workshop om dette blei dessvere avlyst i år pga COVID-19 situasjonen.

Jeg har også etablert flere internasjonale utvekslingsprogrammer for studenter i Oslo, spesielt mot USA og Europa.

Utdanningsforskning

CCSE senteret spiller en avgjørende rolle i forskning rundt beregninger (Computing in Science Education) i utdanning. Jeg er spesielt interessert i forskning rundt faglig innsikt og studentenes innsikt om den vitenskapelige metoden. Et av målene er å kunne utvikle et forskningsbasert program i kvantitativ utdanningsforskning.

Her spiller feks maskinlæring en viktig rolle og sammen med PhD student John Aiken, Prof Danny Caballero fra Michigan State University og andre kolleger hat vi nå utvikla flere prosjekt for å nå disse måla. Å kunne gi kvantitave og kvalitative mål på hva som virker er sentralt for mange utdanningstiltak.

PEDAGOGISK MAPPE

Utdanningsverv og redaktøransvar for lærebøker i fysikk

- 2002-2011: Programrådsleder i bachelorprogrammet Fysikk, Astronomi og Meteorologi, et samarbeid mellom tre institutt. I tillegg til jobben som programrdsleder hadde jeg det overordna ansvaret for den faglige helheten samt innføring av Computing in Science Education prosjektet.
- 2003-2006: Styremedlem i programrådet Matematikk, Informatikk og Teknologi
- 2003-nå: Var med å starte Computing in Science Education prosjektet
- 2010-nå: Initierte Nuclear Talent prosjektet med kolleger fra Nord-Amerika, Europa og Asia. Leda prosjektet fra 2010 til 2015. Styremedlem 2016-2020. Har undervist og organisert flere tre-ukers avanserte intensive kurs. Se lista lenger ned over kurs jeg har organisert og undervist.
- 2015: Tok initiativ til, sammen med kolleger fra Fysisk Institutt, Institutt for Biovitenskap, Matematisk Institutt og Institutt for Informatikk for å etablere det nye masterprogrammet Computational Science. Er programrådsleder siden oppstart 2017 og har jobba med kolleger fra sju institutt ved MatNat-fakultetet for å skape et godt faglig program.

I tillegg har jeg viktige verv i utvikling av faglig litteratur i fysikk for Springer. Jeg er medredaktør i fem bokserier og flere kolleger ved UiO har fått publisert sine bøker via Springer.

- Editorial Board member of Springer's Lecture Notes in Physics, 2010-present
- Editorial Board member of Springer's Undergraduate Lecture Notes in Physics, 2014-present
- Editorial Board member of Springer's University Texts in Physics, 2015-present
- Editorial Board member of Springer's Undergraduate Texts in Physics, 2016-present
- Editorial Board member of Springer's Graduate Texts in Physics, 2018-present

Overordna målsettinger med utdanningsaktiviteten min

De ti overordna målsettingene med min utdanningsaktivitet og forskning er gjengitt her:

1. Gi studentene ei forståing av den vitenskapelige metoden så tidlig som mulig i studieløpet. Dette lar seg gjøre spesielt gjennom prosjektorientert undervisning med tett oppfølging og tilbakemelding til studentene. Her spiller mitt initiav til å starte Computing in Science Education prosjektet ei sentral rolle. Dette er beskrevet nedenfor i større detalj.

2. Gi studentene kompetanse, faglig trygghet og innsikt som kreves for å løse naturvitenskapelige og teknologiske problem for det 21ste århundre, spesielt med tanke på digital kompetanse. Her har min rolle som programrådsleder i to store studieprogram vært, og er, svært viktig.

3. Gi studentene en god etisk holdning til deres arbeid, samt å utvikle kritisk tenkende mennesker med djup innsikt i alle sider av den vitenskapelige prosessen.

4. Å utvikle gode utdanningsprogrammer og faglig progresjon i studieløpa, i tett samarbeid med kolleger ved Fysisk institutt (både i Norge og i USA) og på tvers av disipliner.

5. Å sørge for at det faglige innholdet er i tråd med universitetenes samfunnsoppgave ved å utdanne svært ettertraktede kandidater til forskning, utdanning, offentlig og privat sektor.

6. Å sørge for at det faglige innholdet har intellektuelle utfordringer og innhold i tråd med nåtidige og framtidige faglige behov og forskningsretninger.

7. Å utvikle ei kunnskapsbasert tilnærning til læring, samt utvikle forskningsprogram om hva som gir studentene økt innsikt og faglig forståelse.

8. Å utvikle et internasjonalt perspektiv til utdanninga vår.

9. Kunnskapen skal være fritt tilgjengelig for alle.

10. Pedagogiske tiltak skal være forskningsbaserte og/eller initiere ny forskning.

De ulike aktivitetene i den pedagogiske mappa gjenspeiler disse ti overordna målsettingene. Jeg har også prøvd etter beste evne å flette disse overordna målsettingene inn i kriteriene for tildeling om a) Fokus på studentenes læring, b) En klar utvikling over tid, c) En forskende tilnærming og d) En kollegial holdning og praksis.

Prosjektbasert undervisning og vegen videre

Min tilnærming til undervisning er sterkt inspirert av en serie med vitenskapelige artikler fra 90-tallet og seinere om prosjektbasert læring. Studentene jobber i stor grad sjølstendig med ulike prosjekt. Prosjekta ender ofte med å være nær endelig vitenskapelige resultat som er publisert i forskningslitteraturen. Og i noen tilfeller har også studentprosjekta endt opp som vitenskapelige publikasjoner.

Alle kursa jeg har utvikla har prosjekt og ei prosjektbasert tilnærming som grunnfilosofi. Ved å inkludere numeriske metoder og beregninger (engelsk Computational Science) lærer studentene å studere vitenskapelige problem med alle mulige verktøy, fra papir og blyant til numeriske metoder. Det gir studentene en unik mulighet til å utforske sin forståing av den gitte disiplinen og utdjupe sin innsikt både om utvalgte fenomen, samt å utvikle ei større forståing for den vitenskapelige prosessen. I prosjektbasert undervisning kan alle de 10 overordna målsettingene bakes inn.

Kopla opp med numeriske metoder og beregninger, har en mulighet til å studere system som ikke har analytiske løsninger (som det er veldig få av) og/eller krever kompliserte matematiske triks for å finne en eventuell løsning. Med et diskretisert matematisk problem kan en feks fokusere på overordna fysiske aspekt som feks hva er kreftene som virker på et system, hva er randbetingelsene og initialbetingelsene og mer. Dette gir studentene unike muligheter til å fokusere på faglig forståelse i stedet for ulike matematiske triks.

Studentene lærer også å dele kode og diskutere med andre studenter, de forstår bedre betydninga av reproduserbarhet av vitenskapelige resultat og ulike vitenskapelige etiske aspekt.

Et prosjektbasert løp tar også bort stresset fra en standard eksamenssistuasjon, hvor en i løpet av noen få timer skal reprodusere store deler av pensum. Et prosjektbasert løp gir mulighet for faglig refleksjon som et jag fra oppgave til oppgave ofte ikke gir.

Studentene utvikler ofte eierskap til prosjekta og gjør som regel mye mer enn det en hadde forventa som lærer. Studentene får dermed mulighetene til utforske sine egne veger, ofte til stor personlig tilfredsstillelse, og ny lærdom for både lærere og studenter.

Alle disse observasjonene er grundig diskutert i forskningslitteraturen og har for min del vært ei viktig ledestjerne i mitt pedagogiske arbeid.

Alle kursene jeg har utvikla de siste 20 åra har prosjektbaserte element hvor klassiske ukentlige oppgaver, standard eksamener og midtermeksamener er erstatta av prosjekt. Kursa er:

- FYS2140 Kvantefysikk (1999-2004). Her utvikla jeg og erstatte de tradisjonelle midterm eksamenene med numeriske prosjekt samt at de ukentlige innleveringene hadde numeriske element. Tradisjonell skriftlig eksamen blei beholdt. Mye av erfaringene her leda til mitt arbeid med Computing in Science Education prosjektet. For arbeidet med dette kurset fikk jeg UiO sin undervisningspris i 2000. Den var delt med Arnt Inge Vistnes som da underviste FYS1120, Elektromagnetisme. Jeg gjorde betydelige forandringer på kurset og innførte gruppetimer med tett kontakt mellom studenter og lærere og obligatorisk innlevering av oppgaver.

- FYS3150/4150 Computational Physics I: Dette er et kurs jeg utvikla som nyansatt fra bunnen av i 1999. Det er et av Fysisk institutts mest populære kurs, med ca 100 studenter fra omtrent alle institutt ved MatNat-fakultetet. Kurset er fullstendig prosjektdrevet, med regulære forelesninger og gruppetimer. Studentene får omfattende tilbakemeldinger på prosjektene og kravet til prosjektene er at de skal se ut som vitenskapelige rapporter. Det gir også studentene en betydelig skrivetrening og start på deres masterprosjekter. Det er ingen eksamen i faget, kun prosjekter og mange av prosjektene ender opp som små forskningsprosjekter. Studentene lærer sentrale programmeringsspråk som C++ og Python, samt sentrale numeriske algoritmer. Siden dette er et kurs som tas av studenter fra heile MatNat-fakultetet, blir ofte de siste prosjektene tilpassa deres faglige bakgrunn og interesser. Jeg har også undervist dette kurset i tre år (2016-2018) ved Michigan State University. Kurset har som kode PHY480/PHY905 Computational Physics.

- FYS4411/9411 Computational Physics II er en fortsettelse av FYS3150/4150 og har fokus på beregningsorientert kvantemekanikk, ofte med tette koplinger til studentenes master eller PhD prosjekter. Dette kurset er også fullstendig prosjektbasert og i en del tilfeller har prosjektene resultert i publikasjoner. Jeg starta dette kurset i 2004. Omfattende tilbakemedlinger fra undertegnede og gruppelærere er gjennomgående her også.

- FYS-STK3155/4155 Applied Data Analysis and Machine Learning er et heilt nytt kurs som inngår som et obligatorisk kurs i Computational Science master programmet ved UiO. Jeg utvikla dette kurset fra bunnen av i 2017 og underviste det første gang høsten 2018. I fjor fullførte 129 studenter kurset, med studenter fra omtrent alle institutt fra MatNat-fakultetet og andre fakultet ved UiO. Vi forventer flere studenter i 2020. Styrken igjen er prosjektorientering med de samme grunnleggende prinsippene; studentene får omfattende tilbakemeldinger samt at de kan definere egne prosjekt. Flere av sluttprosjektene har resultert i vitenskapelige publikasjoner eller har vært vesentlige deler av masteroppgaver og Phd avhandlinger. Kurset spiller en viktig rolle i CS programmet da det er et av treffstedene for studentene som ellers ville ha tilbragt sin tid på sitt respektive institutt.

- FYS-KJM4480/9480 Quantum Mechanics for Many Particle Systems er også et kurs jeg utvikla fra bunnen med flere prosjekt. Kurset er dessverre lagt ned da undertegnede underviser allerede tre andre 10 ECTS kurs ved UiO (og har 50% stilling). Omtrent samme opplegg som de andre kursene men med endelig skriftlig eller muntlig eksamen. Prosjektbasert ellers.

- PHY 981 Nuclear Structure ved Michigan State University (2013-2016) og PHY 989 Nuclear Forces ved Michigan State University (høst 2017). Disse kursene hadde ukentlige innleveringsoppgaver (vanlig papir og blyant arbeid) samt to større numeriske prosjekt som la grunnlaget for avsluttende muntlig eksamen.

- PHY 321 Classical Mechanics ved Michigan State University, første gang i år (se emneevaluering som er vedlagt). Jeg leda en komite ved MSU i 2018 som la fram forslag om integrering av numeriske metoder i sentrale fag i en fysikk bachelorgrad. Dette svarer til implementering av Computing in Science Education liknende tiltak. PHY 321 er det første viktige fysikkurset som studenter som planlegger en bachelor grad i fysikk må ta. Sammen med fem andre kurs er dette det første fysikkurset hvor studenter nå møter numeriske oppgaver og prosjekt. De ukentlige oppgavene inneholdt nå numeriske oppgaver og de vanlige skriftlige midtermeksamenene var erstatta med to en-ukers lange prosjekt. Pga. COVID-19 blei universitetet stengt fra 11 mars og for min del blei den skriftlige avsluttende eksamenen erstatta med et ukeslangt prosjekt, til stor glede for studentene. Tilbakemeldingene fra studentene viser hvor mye friheten rundt det å jobbe med prosjekter betyr for egen læring.

Min erfaring etter 20 år med prosjektbasert undervisning er at dette gir studentene mye større mulighet for faglig refleksjon, utvikle djupe innsikter om et bestemt fag og gjøre naturvitenskap slik den gjøres i forskning, enten det er i akademia, offentlig eller privat sektor. Alle fordeler med gruppearbeid, omfattende tilbakemeldinger, mulighet for studentene til å komme med tilbakemeldinger som forbedrer kursa, tett samarbeid student og lærere, og mer, gir et mye bedre læringsmiljø.

Hittil har alle kursa jeg underviser hatt regulære forelesninger. Vegen videre for min del er å ta det prosjektbaserte arbeidet over til det som kalles Flipped Classroms. Her finnes det også omfattende forskning som viser fordelene ved dette. Mitt første steg i høst blir å utvikle videomateriale sammen med det allerede eksisterende digitale materiale. På grunn av usikkerheten rundt semesterstart, enten fullt digitalt eller hybrid løsning, vil et initiatv av typen Flipped Classroms være en ypperlig måte for studentene å komme i kontakt med faglærere i mindre grupper. Tanken er å videreutvikle dette til et mer varig tiltak for kursene FYS3150/4150, FYS4411/9411 og FYS-STK3155/4155. Studentene vil da gå gjennom materialet før de møter til gruppeundervisning. I gruppeundervisninga gjennomgås og oppklares uklarheter rundt det ukentlige materialet, i dialog med faglærere, gruppelærere og Learning Assistants. Deretter starter arbeidet med de ulike oppgavene og prosjektene. Målet er at studentene kommer forberedt til gruppene og at gruppene utvikles til gode faglige diskusjonsfora, med tett dialog mellom studenter og lærere.

Computing in Science Education

Et sentralt tema i prosjektorientert utdanning har vært innføringa av numeriske prosjekt og oppgaver. Mitt arbeid med kurset FYS2140 i tida 1999-2004 og FYS3150/4150 la mye av grunnlaget og forståelsen for visjonen om å integrere beregninger i vanlige fysikkurs. Ei systematisk og helheltig tenkning rundt digital kompetanse har lagt grunnlaget for at numeriske beregninger oppfattes av studentene som en naturlig del av en naturviter sin verktøykasse. Et prosjektorientert studieløp krever også ei tettere oppfølging av studentene og legger grunnlaget for et mentorprogram og en-til-en kopling mellom lærere og studenter. Det gir mulighet for å utforme et mer individualisert studieløp.

For å oppnå måla om digital kompetanse, prosjektorientert utdanning, tett oppfølging av studenter og mange flere av de ti måla ovafor var jeg i 2003 en av de sentrale initiativtakerne til prosjektet **Computing in Science Education** (CSE), sammen med kolleger på Fysisk Institutt (Arnt Inge Vistnes), Matematisk Institutt (John Grue) og Institutt for Informatikk (Hans Petter Langtangen og Knut Mørken). Mye av dette skjedde via senteret for fremragende forskning (SFF) **Center for Mathematics for Applications** og i min rolle som programrådsleder for Bachelor programmet Fysikk, Astronomi og Meteorologi (FAM). De første midlene som blei tildelt til CSE prosjektet var via UiO tiltaket **Fleksibel læring** i 2004. Undertegnede var prosjektleder sammen med programrådsleder for MIT programmet John Grue. Jeg satt også som programrådsmedlem i MIT styret og var med å koordinere innføringa av CSE prosjektet. Senteret CMA spilte en sental rolle i begynnelsen og finasierte deler av stillinger. Seinere kom også et anna SFF inn på banen, Physics of Geological Processes (PGP) og Anders Malthe-Sørenssen ved Fysisk Institutt. Anders var sentral i implemeteringen av CSE prosjektet i det første fysikkurset FYS-MEK1100.

Jeg var programrådsleder for FAM i tida 2002-2011 og var drifkraft og overordna ansvarlig for integrering av et beregningsperspektiv i FAM programmet. Dette har resultert i at fysikkfaget ved UiO har utdanna studenter med de digitale kompetansene som trengs for å møte de vitenskapelige og teknologiske utfordringene i det 21ste århundre. CSE prosjektet har fungert som inspirasjon for alle andre studieprogram ved MatNat-fakultetet og andre fakultet ved UiO og universitet nasjonalt.

CSE prosjeket er nå gjenspeilt i omtrent alle studieprogram ved MatNat-fakultetet ved UiO. I tillegg har jeg leda en komite ved fysisk institutt på Michigan State University

(2018) om innføring av CSE-liknende tiltak ved dette universitetet. I vår underviste jeg et sentralt laveregrads kurs i klassisk mekanikk med full CSE implementering ved Michigan State University, til positiv respons fra studentene, se fagevalueringa i den pedagogiske mappa.

CSE prosjektet fikk UiO sin utdanningspris i 2011, samt NOKUT sin utdanningspris i 2012. Prosjektet resulterte i at UiO fikk et senter for fremragende utdanning i 2016, senteret for **Computing in Science Education** (CCSE). Senteret har blitt en stor suksess og har utvikla flere nye forskningsprosjekt om utdanning. Jeg er medlem av CCSE og jobber både med kursinnhold med tanke på innføring av beregninger samt nye tiltak rundt første studieår for Fysikk og Astronomi programmet. I tillegg jobber jeg med å utvikle forskningsaktivitet i kvantitativ utdanningsforskning.

Utvikling av studiemiljø

Siden 2000 har jeg aktivt jobba for å utvikle et utdanningsmiljø i Computational Physics på mastergradsnivå. Fra og med 2007 har dette arbeidet vært gjort sammen med en nær kollega, Anders Malthe-Sørenssen, som også er direktør for CCSE. I 2015 blei dette arbeidet tilkjent UiO sin utdanningspris, se `https://www.uniforum.uio.no/nyheter/2015/10/instituttet-som-lofter-fram-gode-forelesere.html`.

Vi har spesielt vektlagt

1. Utvikle et godt sosialt miljø hvor deling av resultat og programvare står sterkt. Studentene definerer ofte sine egne prosjekt for masteroppgavene.

2. Oppbygging av lokaler som er imøtekommende og inkluderende. Bygd opp mange arenaer for sosialt samvær.

3. Studentene integreres tidlig i forskninga og studentene fungerer som læremestre for hverandre.

4. Studentene engasjeres tidlig i å utvikle læringsmateriale, spesielt som gruppelærere i kurs hvor beregninger (CSE prosjektet) er viktig.

Med etableringen av det nye CS-masterprogrammet har vi nå utvikla nye tiltak for å forbedre studiekvaliteten, med blant anna tett integrering av studentene i utforminga av nye oppgaver og prosjekt til mastergradskursene, tett kopling mellom studenter og fagmiljø, samt mentorprogram i oppstart av masterprogrammet, med vekt på mulige fagvalg og karriereveger. Individuell oppfølging av studentene spiller en sentral rolle, spesielt også for studenter som sliter mentalt. Tett oppfølging og individuell tilrettelegging er et sentralt aspekt som vektlegges i utviklinga av et godt studiemiljø.

Studieprogrammet har sosiale spillkvelder med faglige seminar annen hver fredag. Her møter studentene forskere og/eller potensielle oppdragsgivere fra privat og offentlig sektor. Og i en del tilfeller har det resultert i sommerjobber og kanskje fast ansettelse seinere.

Computational Physics miljøet har siden 2003 utdanna over 100 mastergradsstudenter, og veldig mange av disse (over 50%) har fortsatt med PhD studier. Det er tett samarbeid mellom bachelor studenter, mastergrads studenter, PhD studenter og Post-docs.

Studentene våre deltar også i å videreutvikle det gode læringsmiljøet og spiller en avgjørende rolle som gruppelærere.

Nye tiltak

Studentene kan med dagens teknologi (feks via applikasjoner på smarttelefoner) utføre ulike eksperiment hjemme, samle inn data og analysere og diskutere dataene som er samla inn. Et slikt løp innebærer også en tettere kontakt mellom student og lærer, med større oppfølging av den enkelte students læring.

Nylig lanserte jeg, sammen med kolleger på Fysisk Institutt, se `https://mhjensen.github.io/FirstYearPhysicsUiO/doc/pub/proposal/html/proposal-bs.html` for mer detaljer, et forslag til nytt første studieår i Fysikk og Astronomi hvor målet er å utvikle eksempler på eksperiment som kan integreres i ulike fysikk kurs,samt hvordan en kan integrere disse eksperimentene med numeriske metoder og programmerings kunnskapene til studentene. Det vil innebære både en revisjon av kurs samt utvikling av nytt læringsmateriale. Denne aktiviteten vil også integreres tett opp mot pågående og ny forskning ved Center for Computing in Science Education. Kvantitativ utdanningsforskning rundt temaer om hva som gir økt innsikt for studentene om fysiske prosesser er sentrale og nye temaer i utdanningsforskning. Hvordan en definerer kvantitativ utdanningsforskning er et åpent tema hvor et slikt prosjekt kan være med å bringe fram verdifull innsikt om studentenes læring.

Det siste argumentet bringer oss over til neste punkt.

Ei forskningsbasert tilnærming til utdanning

Alle kurs jeg har utvikla i min tid ved UiO og MSU har hatt et prosjektbasert perspektiv. All erfaring tyder på at studentene setter pris på denne måten å tilegne seg ny kunnskap og innsikt om et fagfelt. I tillegg gir det studentene stor frihet i å definere egne prosjekter og utvikle eierskap til egen læring. Jeg ser også tydelig at med klare læringsmål er det lett å oppnå mange av de ønska effektene.

Kvalitativt ser vi tydelig at studentene via ei prosjektbasert tilnærming har en bedre mulighet til å utvikle djupere innsikter og forståelser om et fagfelt og den vitenskapelige prosessen. Vi ser også fra våre daglige vekselvirkninger med studentene at et studieløp hvor numeriske oppgaver inkluderes fra dag en, gir store muligheter for å utvikle videre studentenes innsikter og bringe resultat fra aktuell forskning inn i et tidligere stadium i studieløpet.

Men hvordan vi kan kvantisere denne økte innsikten gjenstår å se. Et av de faglige måla til CCSE senteret er nettopp å utvikle et slikt forskningsbasert program for kvantitativ utdanningsforskning. Dette er et tema i forskningsfronten for feks. forskning rundt fysikk utdanning på universitetsnivå.

Her spiller feks. maskinlæring en viktig rolle og sammen med PhD student John Aiken ved CCSE, Prof Danny Caballero fra Michigan State University og CCSE og andre kolleger hat vi nå utvikla flere prosjekt for å nå disse måla. Å kunne gi kvantitave og kvalitative mål på hva som virker er sentralt for mange utdanningstiltak.

De to vedlagte artiklene ved slutten av dette dokumentet, og inkludert her, viser hvordan vi kan bruke feks kvantitative metoder for å kunne si noe om hva som virker eller ikke ved ulike utdanningstiltak. Den andre artikkelen her setter opp den generelle ramma for arbeidet rundt et kvantitativt forskningsprogram mens den første artikkelen viser hvordan maskinlæring kan brukes til å forstå studieprogresjon for studenter. Vårt mål er å videreutvikle ei slik kvantitativ tilnærming til å kunne si noe om studentene faktisk opplever økt faglig innsikt. Det er en lang veg å gå for å oppnå dette, men vi har starta.

1. John M. Aiken, Riccardo De Bin, Morten Hjorth-Jensen, Marcos D. Caballero, *Predicting time to graduation at a large enrollment American university*, arXiv:2005.05104

2. Marcos Daniel Caballero, Morten Hjorth-Jensen, *Integrating a Computational Perspective in Physics Courses*, arXiv:1802.08871

I tillegg har jeg skrevet flere bøker med fokus på beregningsorienterte metoder i fysikk. Her følger ei liste over aktuelle bøker for den pedagogiske mappa.

Bøker:

1. Morten Hjorth-Jensen, *Computational Physics, an introduction*, to be published by IOP in 2020, 500 pages.

2. Morten Hjorth-Jensen, *Computational Physics, an advanced course*, to be published by IOP in 2020, 400 pages

3. Morten Hjorth-Jensen, *Nuclear many-body physics, a computational perspective*, in preparation for Lecture Notes in Physics by Springer.

4. Morten Hjorth-Jensen, M.P. Lombardo and U. van Kolck, *Computational Nuclear Physics-Bridging the scales, from quarks to neutron stars*, Lectures Notes in Physics by Springer, Volume **936** (2017).

Studenter og Post-Docs

Jeg har vegleda og vegleder over 100 studenter på alle nivå. Flere av PhD studentene har endt opp i vitenskapelige stillinger i inn og utland. Simen Kvaal (nå ansatt ved Kjemisk Institutt UiO) fikk et ERC stipend og Gaute Hagen (ansatt ved Oak Ridge National Laboratory) fikk det prestisjefylte Young Investigator Award fra Department of Energy. Omtrent 50% as masterstudentene har fortsatt med PhD studier enten med meg eller andre som vegledere.

Nåværende PhD studenter.

1. Benjamin Hall, Michigan State University, started 2018.

2. Jane Kim, Michigan State University, started 2018.

3. Julie Butler, Michigan State University, started 2018.

4. Omokuyani C. Udiani , Michigan State University, started 2017, co-supervisor

5. Danny Jammoa, Michigan State University, started 2020, co-supervisor
6. Øyvind Sigmundsson Schøyen, University of Oslo, started 2019
7. John Mark Aiken, University of Oslo, started 2017, defends thesis September 2020

Nåværende masterstudenter.

1. Eina Jørgensen, University of Oslo, (2019-2021), co-supervisor
2. Morten Hemmingsen, University of Oslo, (2019-2021), co-supervisor
3. Huying Zhu, University of Oslo, (2019-2021), co-supervisor
4. Jens Due Bratten, University of Oslo, (2019-2021), co-supervisor
5. Gabriel Cabrera, University of Oslo, (2019-2021), co-supervisor
6. Kristian Wold, University of Oslo, (2019-2021)
7. Martin Krokan Hovden, University of Oslo, (2019-2021)
8. Oliver Hebnes, University of Oslo, (2019-2021), co-supervisor
9. Mohamad Ismail, University of Oslo, (2019-2021), co-supervisor
10. Heine Aabø, University of Oslo, (2018-2020)
11. Stian Bilek, University of Oslo, (2018-2020)
12. Thomas Sjåstad, University of Oslo, (2018-2020)
13. Halvard Sutterud, University of Oslo, (2018-2020)
14. Stian Isachsen, University of Oslo, (2018-2020), co-supervisor
15. Marius Holm, University of Oslo, (2018-2020), co-supervisor
16. Halvard Sutterud, University of Oslo, (2018-2020)
17. Geir Utvik, University of Oslo, (2018-2020)
18. Markus Asprusten, University of Oslo, (2018-2020), co-supervisor

Tidligere PhD studenter og deres nåværende stillinger.

1. Justin Lietz (PhD MSU 2019), now post-doctoral fellow at Oak Ridge National Laboratory
2. Samuel Novario (PhD MSU 2018), now post-doctoral fellow at Oak Ridge National Laboratory
3. Fei Yuan (PhD MSU 2018), employed at Google
4. Gustav Baardsen (PhD UiO 2014), Research Scientist at Varian Medical Systems, Helsinki, Finland
5. Simen Kvaal (PhD UiO 2009), now associate professor of chemistry, Department of Chemistry, University of Oslo. Recipient of an ERC starting grant
6. Gustav Jansen (PhD UiO 2012), now permanent position as scientist at the Computational Science Division of Oak Ridge National Laboratory
7. Torquil MacDonald Sørensen (PhD UiO 2012), post-doctoral fellow at the Department of Mathematics, UiO
8. Jon Kerr Nilsen (PhD UiO 2010), senior engineer at the University of Oslo center for information technologies (co-supervisor)
9. Marius Lysebo (PhD UiO 2010), now Associate Professor at Oslo University College, (co-supervisor)
10. Elise Bergli (PhD UiO 2010), teacher Ås high school, Norway
11. Eirik Ovrum (PhD UiO 2007), now Associate Professor at the University College of Southeast of Norway
12. Gaute Hagen (PhD UiB and UiO 2005), now permanent position as scientist at the Physics Division of Oak Ridge National Laboratory. Recipient of the Department of Energy Early career award
13. Maxim Kartamyshev (PhD UiO), now at the Bank of Norway as senior analyst

14. Øystein Elgarøy (PhD UiO 1999), now professor of Theoretical Astrophysics at the University of Oslo, Norway (co-supervisor)

15. Lars Engvik (PhD UiO 1999), now Associate Professor at Sør-Trøndelag University College, Trondheim, Norway, (co-supervisor)

Post-docs og deres nåværende stillinger.

1. Andreas Ekstrøm (UiO and MSU 2010-2014), now Associate Professor at Chalmers Technological University in Gothenburg, Sweden

2. Øyvind Jensen (UiO 2011), now researcher at the Institute for Energy Technology

3. Simen Kvaal (UiO 2008-2012), now associate professor of chemistry, Department of Chemistry, University of Oslo. Recipient of an ERC starting grant

4. Elise Bergli (UiO 2010-2011), now teacher at Ås high school, Norway

5. Sølve Selstø (UiO 2008-2010), now Professor at Oslo Metropolitan University

6. Nicolas Michel (MSU 2013), now senior researcher at Langzhou Nuclear Physics Laboratory, China

Masterstudenter som har avlagt eksamen.

1. Vebjørn Gilberg, University of Oslo, (2017-2020), co-supervisor

2. Kari Eriksen, University of Oslo, (2017-2020)

3. Robert Solli, University of Oslo, (2017-2019)

4. Andreas Lefdalsnes, University of Oslo, (2017-2019)

5. Joseph Knutson, University of Oslo, (2017-2019)

6. Bendik Samseth, University of Oslo, (2017-2019)

7. Even Nordhagen, University of Oslo, (2017-2019), (now PhD student)

8. Øyvind Schøyen Sigmundson, University of Oslo, (2017-2019), (now PhD student)

9. Sebastian Gregorius Winther-Larsen, University of Oslo, (2017-2019), (now PhD student)
10. Giovanni Pederiva, University of Oslo, (2016-2018), co-supervisor, (now PhD student)
11. Anna Gribovskaya, University of Oslo, (2016-2018)
12. Andrei Kucharenka, University of Oslo, (2016-2018)
13. Vilde Moe Flugsrud, University of Oslo, (2016-2018)
14. Alfred Alocias Mariadason, University of Oslo, (2016-2018)
15. Marius Jonsson, University of Oslo, (2016-2018), (now PhD student)
16. Hans Mathias Vege Mamen, University of Oslo, (2016-2019), co-supervisor
17. Alexander Fleischer, University of Oslo, (2015-2017)
18. Håkon Emil Kristiansen, University of Oslo, (2015-2017), (now PhD student)
19. Morten Ledum, University of Oslo, (2015-2017), (now PhD student)
20. Håkon Treider Vikør, University of Oslo, (2015-2017), co-supervisor
21. Jon-Andreas Stende, University of Oslo, (2015-2017), co-supervisor
22. Sean Bruce Sangholt Miller, University of Oslo, (2015-2017), (now PhD student)
23. Christian Fleischer, University of Oslo, (2015-2017)
24. John Bower, Michigan State University, (2014-2017)
25. Wilhelm Holmen, University of Oslo (2014-2016)
26. Roger Kjøde, University of Oslo, (2014-2016)
27. Håkon Sebatian Mørk, University of Oslo, (2014-2016)
28. Jonas van den Brink, University of Oslo, (2014-2016), co-supervisor, (now PhD student)
29. Marte Julie Sætra, University of Oslo, (2014-2016), co-supervisor, (now PhD student)

30. Audun Skau Hansen, University of Oslo, (2013-2015), (now PhD student)
31. Henrik Eiding, University of Oslo, (2012-2014)
32. Svenn-Arne Dragly, University of Oslo, (2012-2014), defended PhD
33. Milad Hobbi Mobarhan, University of Oslo, (2012-2014), defended PhD
34. Ole Tobias Norli, University of Oslo, (2012-2014)
35. Filip Sand, University of Oslo, (2012-2014), co-supervisor
36. Emilie Fjørner, University of Oslo, (2012-2014), co-supervisor
37. Jørgen Høgberget, University of Oslo, (2011-2013), , defended PhD
38. Sarah Reimann, University of Oslo, (2011-2013), defended PhD
39. Karl Leikganger, University of Oslo, (2011-2013), defended PhD
40. Sigve Bøe Skattum, University of Oslo, (2011-2013), defended PhD
41. Veronica Berglyd Hansen, University of Oslo, (2010-2012), defended PhD
42. Camilla Nestande Kirkemo, University of Oslo, (2010-2012), co-supervisor
43. Christoffer Hirth, University of Oslo, (2009-2011)
44. Marte Hoel Jørgensen, University of Oslo, (2009-2011)
45. Yang Min Wang, University of Oslo, (2009-2011), defended PhD
46. Ivar Nikolaisen, University of Oslo, (2009-2011), began on PhD
47. Vegard Amundsen, University of Oslo, (2008-2010)
48. Håvard Sandsdalen, University of Oslo, (2008-2010)
49. Lars Eivind Lervåg, University of Oslo, (2008-2010)
50. Magnus Lohne Pedersen, University of Oslo, (2008-2010)
51. Simen Sørby, University of Oslo, (2008-2010), co-supervisor

52. Sigurd Wenner, University of Oslo, (2008-2010), co-supervisor, defended PhD
53. Lene Norderhaug Drøsdal, University of Oslo, (2007-2009), defended PhD
54. Islen Vallejo, University of Oslo, (2007-2009), works at the Norwegian Institute for Air Research
55. Jacob Kryvi, Norwegian University of Science and Technology, (2007-2009), co-supervisor, defended PhD
56. Rune Albrigtsen, University of Oslo, (2007-2009)
57. Johannes Rekkedal, University of Oslo, (2007-2009), began PhD
58. Patrick Merlot, University of Oslo, (2007-2009), began PhD
59. Gustav Jansen, University of Oslo, (2006-2008), defended PhD
60. Ole Petter Harbitz, University of Oslo, (2006-2008)
61. Sutharsan Amurgian, University of Oslo, (2005-2007)
62. Jon Thonstad, University of Oslo, (2005-2007)
63. Espen Flage-Larsen, University of Oslo, (2003-2005), defended PhD
64. Joachim Berdahl Haga, University of Oslo, (2004-2006), defended PhD
65. Jon Kerr Nilsen, University of Oslo, (2002-2004), defended PhD
66. Simen Kvaal, University of Oslo, (2002-2004), defended PhD
67. Simen Reine Sommerfelt, University of Oslo, (2002-2004), defended PhD
68. Mateuz Marek Røstad, University of Oslo, (2002-2004)
69. Victoria Popsueva, University of Oslo, (2002-2004), defended PhD
70. Eivind Brodal, University of Oslo, (2001-2003), defended PhD
71. Eirik Ovrum, University of Oslo, (2001-2003), defended PhD
72. Ronny Kjelsberg, Norwegian University of Science and Technology, (2001-2003)

Medlem av PhD komiteer ved MSU

1. Justin Lietz, chair, defended thesis June 2019.
2. Fei Yuan, chair. Defended thesis January 24 2018.
3. Sam Novario, chair. Defends thesis February 7 2018.
4. John Bower, chair together with Scott Bogner. Master of Science thesis May 2017.
5. Adam Jones, committee member. Master of Science thesis July 2017.
6. Chris Sullivan, committee member. Defended thesis January 2018.
7. Thomas Redpath, committee member. Defended thesis October 2019.
8. Sean Sweany, committee member
9. Rachel Taverner, committee member. Defended thesis May 2019.
10. Nathan Parzuchowski, committee member. Defended thesis April 2017.
11. Titus Morris, committee member. Defended thesis May 2016
12. Kenneth Whitmore, committee member. Defended thesis June 2016
13. Alex Dombos, committee member. Defended thesis May 2018.
14. Josh Bradt, committee member, Defended thesis July 2017
15. Charles Loelius, committee member, Defended thesis May 2017
16. Safwan Shanab, committee member. Defended thesis January 2020.
17. Hao Lin, committee member
18. Mao Xingze, committee member. Defended thesis May 2020.
19. Amy Lovell, committee member. Defended thesis January 24 2018.
20. Debra Richman, committee member
21. Roy Ready, committee member

22. Nathan Watwood, committee member

23. Ben Hall, chair

24. Udiani Omokuyani, committee member

25. Jane Kim, chair

Internasjonale utdanningstiltak

Sammen med kolleger fra flere land starta jeg og etablerte et internasjonalt initiativ i 2010 kalt **Nuclear TALENT (Training in Advanced Low-Energy Nuclear Theory)** hvis mål er å styrke en faglig bredde i kjernefysikk internasjonalt. Mange universitet har ikke nok vitenskapelig personale til å gi studentene på master og PhD nivå den nødvendige faglige bredde i feltet. Nuclear TALENTs mål er å tilby denne faglige bredden i form av et titalls avanserte kurs som undervises på et intensivt vis over tre uker ulike steder i verden (Nord-Amerika, Europa og Asia). Siden sommeren 2012 har vi organisert over 15 slike kurs og jeg har undervist og organisert 5 av disse kursene og organisert tre andre. Dette tiltaket har vært en enorm suksess med over 500 deltakere totalt siden 2012. Pga COVID-19 er alle tre kurs i år utsatt til 2021, men kurset jeg har ansvaret for i år om maskinlæring anvendt på kjernefysikk tilbys digitalt i juni-juli 2020, se `http://www.ectstar.eu/node/4472`.

For mer informasjon om Nuclear TALENT, se `https://fribtheoryalliance.org/TALENT/`.

Ellers leder jeg et større INTPART prosjekt om Computing in Science Education mellom CCSE ved UiO, Michigan State University, Oregon State University og University of Colorado ved Boulder. Et viktig mål med dette prosjektet er å utvikle et program i kvantitativ utdanningsforskning. Vår første workshop om dette blei dessvere avlyst i år pga COVID-19 situasjonen.

Jeg har også etablert flere internasjonale utvekslingsprogrammer for studenter i Oslo, spesielt mot USA og Europa.

Her følger en liste over skoler jeg har organisert.

Organisering av skoler og foredrag ved skoler

1. Morten Hjorth-Jensen, Nuclear Talent Course on Machine Learning in Nuclear Physics for the Erasmus+ program European Master in Nuclear Physics, University of Basse-Normandie and GANIL, January 20-31, 2020. 45 lectures and 45 exercise sessions. Main teacher

2. Morten Hjorth-Jensen, Matthew Hirn, Michelle Kuchera, and R. Ramanujan, FRIB TA Summer School - Machine Learning Applied to Nuclear Physics, Facility for Rare Isotope Beams (FRIB) on the Michigan State University campus in East Lansing, MI from May 20 to 23, 2019. Main organizer and teacher.

3. Morten Hjorth-Jensen, Nuclear Talent Course on Machine Learning in Nuclear Physics for the Erasmus+ program European Master in Nuclear Physics, University of Basse-Normandie and GANIL, January 21-February 1, 2019. 45 lectures and 45 exercise sessions. Main teacher

4. Nuclear Talent course on Many-body methods for nuclear physics, from Structure to Reactions at Henan Normal University, P.R. China, July 16-August 5 2018. Teachers: Kevin Fossez, Morten Hjorth-Jensen, Thomas Papenbrock, and Ragnar Stroberg.

5. Alex Brown, Alexandra Gade, Morten Hjorth-Jensen, Gustav Jansen, Robert Grzywacz, Nuclear Talent course on Nucleartheory for Nuclear Structure Experiments, July 3-21 2017. Main organizer and teacher with in total fifteen hours of lectures.

6. Hjorth-Jensen, Morten, High performance computing in Nuclear Physics, Lecture at the *Advanced Computational Research Experience* at Michigan State University, East Lansing, Michigan, June 1, 2017.

7. Hjorth-Jensen, Morten, How to write good code, Lecture at the *Advanced Computational Research Experience* at Michigan State University, East Lansing, Michigan, May 24, 2017.

8. Hjorth-Jensen, Morten, Computational Nuclear Physics and Post Hartree-Fock Methods. Configuration Interaction Theory, Many-Body Perturbation Theory and Coupled

Cluster Theory, five lectures at 28th Indian-Summer School on Ab Initio Methods in Nuclear Physics, Prague, Czech Republic, August 29 - September 2, 2016.

9. Hjorth-Jensen, Morten, Computational Physics and Quantum Mechanical Systems, one week course on Computational Physics at the University of Tunis El Manar, Tunis, Tunisia, May 16-20, 2016. In total 15 hours of lectures and 15 hours of computer lab and exercises.

10. Co-organizer with Giuseppina Orlandini and Alejandro Kievsky of Nuclear Talent course Few-body methods and nuclear reactions, ECT*, Trento, Italy, July 20-August 7 2015

11. Carlo Barbieri, Wim Dickhoff, Gaute Hagen, Morten Hjorth-Jensen, and Artur Polls, Nuclear Talent course on Many-body methods for nuclear physics, GANIL, Caen, France, July 5-25 2015. Main organizer and teacher with in total five hours of lectures.

12. Hjorth-Jensen, Morten, ECT* Doctoral Training Program 2015 on Computational Nuclear Physics, April 13- May 22, ECT*, Trento, Italy. I taught the last week of the lecture series. In total I have ten one hour lectures.

13. Hjorth-Jensen, Morten, Nuclear Talent School in Nuclear Astrophysics, co-organizer with Richard Cyburt and Hendrik Schatz of the Nuclear Talent course on Nuclear Astrophysics, Michigan State University, May 26 - June 13, 2014.

14. Hjorth-Jensen, Morten, Nuclear Talent course on Density Functional theories, co-organizer with Scott Bogner, Nicolas Schunck, Dario Vretenar and Peter Ring, European Center for Theoretical Nuclear Physics and Related Areas, Trento, Italy, July 13 -August 1 2014.

15. Hjorth-Jensen, Morten, Nuclear Talent Course Introduction on High-performance computing and computational tools for nuclear physics; ECT*, Trento, Italy, June 24 - July 13 2012. Main organizer and teacher together with Francesco Pederiva, Kevin Schmidt and Calvin Johnson.

16. Hjorth-Jensen, Morten. Computational environment for Nuclear Structure, five lectures in Nuclear Physics at Universidad Complutense Madrid; 2011-01-17 - 2011-02-09

17. Hjorth-Jensen, Morten, organizer with David Dean, Thomas Papenprock and Gaute Hagen. Third MSU-UT/ORNL-UiO winter school in nuclear physics; Oak Ridge National Lab, Tennessee, January 2012

18. Hjorth-Jensen, Morten, organizer with Alex Brown and teaching five lectures. Second MSU-UT/ORNL-UiO winter school in nuclear physics, East Lansing, Michigan, USA; 2011-01-03 - 2011-01-07

19. Hjorth-Jensen, Morten, organizer, First MSU-UT/ORNL-UiO winter school in nuclear physics, Wadahl, Norway, January 4-10 2010

20. Hjorth-Jensen, Morten. Five lectures on Theory of shell-model studies for nuclei. CERN/Isolde course on nuclear structure theory; 2010-03-01 - 2010-03-04

21. Hjorth-Jensen, Morten. Six lectures on Nuclear interactions and the Shell Model. 8th CNS-EFES International Summer School, Riken, Tokyo, Japan, 2009-08-26 - 2009-09-01

22. Hjorth-Jensen, Morten. Five lectures on nuclear theory at the 20th Chris Engelbrecht Summer School in Theoretical Physics, Stellenbosch, South Africa, 2009-01-19 - 2009-01-28

23. Hjorth-Jensen, Morten. Nuclear many-body theory, five lectures at the UK Postgraduate Nuclear Physics Summer School, Leicester, UK, 2009-09-12 - 2009-09-23

24. Hjorth-Jensen, Morten. Nuclear many-body methods. Lectures series at Lund University; 2008-05-04 - 2008-05-07

25. Hjorth-Jensen, Morten. Trends in Nuclear Structure Theory. Workshop at the University of Lund; 2008-05-07 - 2008-05-07

26. Hjorth-Jensen, Morten. Trends in Nuclear Structure Theory. Physics Division Seminar; 2008-04-17 - 2008-04-17

27. Hjorth-Jensen, Morten. Trends in nuclear structure theory. Lecture series at the University of Padova and Legnaro National Laboratory, Padova Italy; 2008-07-16 - 2008-07-19

28. Hjorth-Jensen, Morten. Five lectures on Monte Carlo methods and applications in the physical sciences. eScience Winther School 2007; Geilo, Norway 2007-01-28 - 2007-02-02

29. Hjorth-Jensen, Morten. Five lectures at the ISOLDE Spring School in Nuclear Theory; CERN, Switzerland, 2007-05-21 - 2007-05-26

30. Hjorth-Jensen, Morten. Ten lecures at ECT* Doctoral Training Programme 2007; Trento, Italy, April 16-20

31. Hjorth-Jensen, Morten. From the nucleon-nucleon interaction to a renormalized interaction for nuclear systems. Lecture series at Michigan State University; April 2005

32. Hjorth-Jensen, Morten. CENS: A computational Environment for Nuclear Structure. Isolde Lecture series; 2004-11-11 - 2005-11-25

Undervisningsrelevante foredrag

1. Hjorth-Jensen, Morten, Århus University, Denmark, workshop and Ole Rømer Colloquium: Integrating a Computational Perspective in Physics (and Science) Courses, October 23, 2019

2. Hjorth-Jensen, Morten, Computing in Science Education, seminar at the Department of Physics, University of Trento, Trento, Italy, March 5, 2019.

3. Hjorth-Jensen, Morten, "Integrating Computations in Physics Courses, Workshop on New Horizons in Teaching Science: 18th-19th, June 2018, University of Messina, Italy"

4. Hjorth-Jensen, Morten, Computing in Science Education; how to integrate computing in Science courses across disciplines, seminar at the University of Surrey, UK, November 28 2017

5. Hjorth-Jensen, Morten, Computing in Physics Education, Invited talk at the 103rd National congress of the Italian Physical Society, Trento, September 11-15, 2017, Italy

6. Hjorth-Jensen, Morten, Integrating a Computational Perspective in the Basic Science Education, Special Lectures and Events, Notre Dame University, South Bend, Indiana, March 30 2015.

7. Hjorth-Jensen, Morten, Computing in Science Education. Integrating a Computational Perspective in the Basic Science Education, Physics Colloquium, Central Michigan University, Mt Pleasant, March 19 2015.

8. Hjorth-Jensen, Morten, Computing in Science Education. Integrating a Computational Perspective in the Basic Science Education, condensed matter seminar, Ohio University, Athens, Ohio, February 26 2015.

9. Hjorth-Jensen, Morten, Computing in Science education, how to introduce a computational perspective in the basic science education, special colloquium Department of Physics, Lousiana State University, Baton Rouge, Lousiana, April 4 2014.

10. Hjorth-Jensen, Morten. Educating the next generation of nuclear scientists; how can a center like the ECT* aid in developing modern nuclear physics educational programs?. ECT* 20th anniversary colloquium; 2013-09-14 - 2013-09-14

11. Hjorth-Jensen, Morten. Computing in Science Education. Seminar at college of engineering; 2012-03-15 - 2012-03-15

12. Hjorth-Jensen, Morten. Computing in Science Education, a new way to teach science?. Institute seminar The Ohio State University; 2012-02-28 - 2012-02-28

13. Hjorth-Jensen, Morten. Computers in Science Education; a new way to teach Science?. Institute seminar; 2011-03-21 - 2011-03-21

14. Hjorth-Jensen, Morten. Computers in Science Education; a new way to teach Science?. Seminar at Universidad Complutense Madrid; 2011-01-24 - 2011-01-24

15. Hjorth-Jensen, Morten. Computers in Science Education. Institute seminar at the university of Trento, Italy; 2010-05-05 - 2010-05-05

16. Hjorth-Jensen, Morten. Datamaskiner i realfagsopplæringen, en ny måte å undervise realfag på?. Institutt kollokvium; 2009-02-13 - 2009-02-13

17. Hjorth-Jensen, Morten. Computers in Science Education. Guest lecture at Michigan State University; 2008-03-30 - 2008-03-30

18. Hjorth-Jensen, Morten. Computers in Science Education. Forelesning ved UniK, Kjeller; 2008-10-23 - 2008-10-23

19. Hjorth-Jensen, Morten. Computers in Science education, a new way to teach science?. eNORIA: Workshop on eScience in Higher Education; 2008-10-07 - 2008-10-07

20. Hjorth-Jensen, Morten; Langtangen, Hans Petter; Malthe-Sørenssen, Anders; Mørken, Knut Martin; Vistnes, Arnt Inge. Computers in Science Education, a new way to teach physics and mathematics?. April Meeting of the American Physical Society; 2008-04-11 - 2008-04-15

21. Hjorth-Jensen, Morten; Mørken, Knut Martin. Computers in Science Education A New Way to Teach Science?. "I POSE OG SEKK" - Kvalitet i både forskning og utdanning. Er det mulig?; 2008-11-12 - 2008-11-13

22. Hjorth-Jensen, Morten; Mørken, Knut Martin. Computers in Science Education A New Way to Teach Science?. Møte i Nasjonalt råd for teknologisk utdanning; 2008-11-11 - 2008-11-11

23. Hjorth-Jensen, Morten. Computeres in Science Education, a new way to teach science?. Institute seminar; 2007-05-15 - 2007-05-15

24. Hjorth-Jensen, Morten. Computers in Science Education, a new way to teach science?. EUPEN's 9th General Forum - EGF2007; 2007-09-06 - 2007-09-08

25. Hjorth-Jensen, Morten. Computers in Science Education: realfagsundervisning på en ny måte?. Pedagogisk modul for MN-fak; 2007-04-11 - 2007-04-11

26. Hjorth-Jensen, Morten. How to Integrate Parallel Computing in Science Education?. High-Performance and Parallel Computing; 2007-10-24 - 2007-10-24

27. Hjorth-Jensen, Morten; Mørken, Knut Martin. Computers in Science Education, realfag på en ny måte?. Realfag – nøkkelen til fremtidens kunnskapssamfunn; 2007-03-23 - 2007-03-23

28. Hjorth-Jensen, Morten; Mørken, Knut Martin. Computers in Science Education: Realfagsundervisning på en ny måte?. Presentasjon for Abelia og NHO; 2007-08-14 - 2007-08-14

29. Hjorth-Jensen, Morten. Computers in Science Education. CMA workshop on 'Computers, computations and science education'; 2005-09-30 - 2005-09-30

30. Hjorth-Jensen, Morten. Kvalitetsreformen, nye Muligheter for Samarbeid mellom Universitet og Næringsliv. Industridag, rom for muligheter, Universitetet i Oslo; 2005-09-16 - 2005-09-16

31. Hjorth-Jensen, Morten. Økt innsikt og læring ved hjelp av IKT i Fysikk. Det Umuliges kunst? IKT i utdanning - kvalitetetsreformen i praksis; 2004-04-28 - 2004-04-28

32. Vistnes, Arnt Inge; Hjorth-Jensen, Morten. Numerical methods as an integrated part of physics education. 9th Workshop on Multimedia in Physics Teaching and Learning; 2004-09-09 - 2004-09-11

33. Hjorth-Jensen, Morten. Bruk av numeriske verktøy i undervisningen. Pedagogisk modul i 'Undervisning i matematiske og naturvitenskapelige fag', UNiversitetet i Oslo; 2003-05-23 - 2003-05-23

Instructor course sections

SIRS SUMMARY REPORT FOR: PHY 321 001 (TERM: SS20)

- Instructor: MORTEN HJORTH JENSEN
- Number of students enrolled: 61
- Number of replies: 45
- Date generated: 5/17/2020 8:36:27 PM

Show Form Questions (opens in a new window)

SOCT

In addition to results from SIRS Online surveys, the SOCT survey (Students' Opinion of Courses and Teaching) results are also available. SOCT is a brief, optional survey that students can take. Students, faculty and the MSU community can view results from this survey at **soct.msu.edu**.

	Superior	Above average	Average	Below average	Inferior	OMIT	MEAN	STD. Deviation
INSTRUCTION	1	2	3	4	5	6	7	8
1 The instructor's enthusiasm when presenting course material.	84.4%	11.1%	4.44%	0%	0%	0%	1.2	0.49
2 The instructor's interest in teaching.	82.2%	13.3%	4.44%	0%	0%	0%	1.22	0.51
3 The instructor's use of examples or personal experiences to help get points across in class.	70.4%	22.7%	6.81%	0%	0%	2.22%	1.36	0.60
4 The instructor's concern with whether the students learned the material.	91.1%	4.44%	4.44%	0%	0%	0%	1.13	0.45
5 Your interest in learning the course materials.	60%	28.8%	11.1%	0%	0%	0%	1.51	0.68
6 Your general attentiveness in class.	42.2%	33.3%	22.2%	2.22%	0%	0%	1.84	0.84
7 The course as an intellectual challenge.	55.5%	31.1%	8.88%	4.44%	0%	0%	1.62	0.82
8 Improvement in your competence in this area due to this course.	53.3%	33.3%	13.3%	0%	0%	0%	1.6	0.71
9 The instructor's encouragement to students to express opinions.	75.5%	17.7%	6.66%	0%	0%	0%	1.31	0.58
10 The instructor's receptiveness to new ideas and others' viewpoints.	77.2%	15.9%	6.81%	0%	0%	2.22%	1.29	0.58
11 The student's opportunity to ask questions.	73.3%	20%	6.66%	0%	0%	0%	1.33	0.59
12 The instructor's stimulation of class discussion.	53.4%	25.5%	18.6%	2.32%	0%	4.44%	1.69	0.85
13 The appropriateness of the amount of material the instructor attempted to cover.	57.7%	31.1%	11.1%	0%	0%	0%	1.53	0.68
14 The appropriateness of the pace at which the instructor attempted to cover the material.	59.0%	29.5%	11.3%	0%	0%	2.22%	1.52	0.69
15 The contribution of homework assignments to your understanding of the course materials relative to the amount of time required.	75.5%	17.7%	6.66%	0%	0%	0%	1.31	0.58
16 The appropriateness of the difficulty of assigned reading topics.	60%	22.2%	17.7%	0%	0%	0%	1.57	0.77
17 The instructor's ability to relate the course concepts in a systematic manner.	60%	35.5%	4.44%	0%	0%	0%	1.44	0.57
18 The course organization.	60%	24.4%	15.5%	0%	0%	0%	1.55	0.74
19 The ease of taking notes on the instructor's presentation.	50%	29.5%	18.1%	0%	2.27%	2.22%	1.75	0.90
20 The adequacy of the outlined direction of the course.	55.5%	28.8%	15.5%	0%	0%	0%	1.6	0.74

	Superior	Above average	Average	Below average	Inferior	OMIT	MEAN	STD. Deviation
INSTRUCTION	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
21 Your general enjoyment of the course.	64.2%	23.8%	11.9%	0%	0%	6.66%	1.47	0.69

COMPOSITE PROFILE FACTORS

Category	Items	Mean	Standard Deviation
Instructor Involvement (HJORTH JENSEN,MORTEN)	Items 1-4	1.22	0.52
Student Interest (Non-Instructor)	Items 5-8	1.64	0.77
Student Instructor Interaction (HJORTH JENSEN,MORTEN)	Items 9-12	1.40	0.68
Course Demands (Non-Instructor)	Items 13-16	1.47	0.69
Course Demands (HJORTH JENSEN,MORTEN)	Items 13-16	1.52	0.69
Course Organization (Non-Instructor)	Items 17-20	1.57	0.74
Course Organization (HJORTH JENSEN,MORTEN)	Items 17-20	1.59	0.77

STUDENT BACKGROUND

						OMIT	MEAN	STD. Deviation
BACKGROUND QUESTIONS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
22 Was this course required in your degree program?	100%	0%				0%	1	0
23 What is your sex?	84.4%	15.5%				0%	1.15	0.36
24 What is your overall GPA?	2.22%	0%	2.22%	24.4%	71.1%	0%	4.62	0.73
25 What is your class level?	6.81%	47.7%	36.3%	9.09%	0%	2.22%	2.47	0.75

RESPONDERS' COMMENTS FOR PHY 321 001 (SS20)

Question 26: Do you have any comments or suggestions for organizing the course?

- - Project-based assignments were a nice change of pace from more stressful exams - professor's leniency with due dates due to personal circumstances was much appreciated. I am very grateful - coding exercises were nice but sometimes an excessive amount of plots were required to show all results asked (I recall having to make ~30 different plots for a single problem on one of the homeworks)
- amazing prof and did a great job teaching the course
- BNM
- I had a really good time and enjoyed this class. I also was able to stay awake during the class time. Huge plus.
- I think offering some extra credit on assignments is a good idea, but I think Morten went a bit overboard in that respect. Other than that, this was a great course. We learned how to practically solve physics problems, consider different systems and models, and compare analytical solutions with numerical ones. He once said something along the lines of "I want everyone to leave this class loving physics." I think that sums up his passion for teaching and how great of a professor he was. He was incredibly nice, and wanted all of us to learn and succeed.
- Mort was awesome! Would highly recommend!
- Morten has by far been the best professor I have ever had
- Morten was great! He was super understanding about my lack of coding skills and was always available for help and questions. His transition to online lessons was fine, and he helped me learn a lot.
- Morten was one of the best professors that I have ever had in college. He was passionate about the class and drawing on experiences and examples that made sense. He made a great effort to meet outside of class and help in any way possible. He made great adaptations once classes were shifted online and has done an incredible job to spark interest and further learning
- Morten was really great.
- Nah, it was good.
- no
- None
- The best professor I have had since I have been at MSU, he loves his students and wants to see them succeed and is willing to do everything in his power to help. A truly amazing professor.
- This class is very difficult in an online setting but I believe that Morten made this transition the easiest. He was my favorite professor I had this semester. Going to his classes were so much fun, he made the in-class lectures very interesting and he had a great attitude every day. Absolutely amazing professor and person.
- This was my favorite course I've ever taken. I can't speak highly enough of it.
- While it may seem like i just selected all Superior ratings for Professor Jensen, it is just me being completely honest. This class was phenomenal. Morten clearly has a passion for physics and and has clearly inspired his students. With switching to online classes this is the only class that had consistently almost full attendance (minus 1-2 students occasionally) compared to classes I had with mandatory attendance.

Chapter 1
Integrating a Computational Perspective in Physics Courses

Marcos Daniel Caballero and Morten Hjorth-Jensen

Abstract In this contribution we discuss how to develop a physics curriculum for undergraduate students that includes computing as a central element. Our contribution starts with a definition of computing and pertinent learning outcomes and assessment studies and programs. We end with a discussion on how to implement computing in various physics courses by presenting our experiences from Michigan State University in the USA and the University of Oslo in Norway.

1.1 Introduction

Many important recent advances in our understanding of the physical world have been driven by large-scale computational modeling and data analysis, for example, the 2012 discovery of the Higgs boson, the 2013 Nobel Prize in chemistry for computational modeling of molecules, and the 2016 discovery of gravitational waves. Given the ubiquitous use in science and its critical importance to the future of science and engineering, scientific computing plays a central role in scientific investigations and is critical to innovation in most domains of our lives. It underpins the majority of today's technological, economic and societal feats. We have entered an era in which huge amounts of data offer enormous opportunities. By 2020, it is also expected that one out of every two jobs in the STEM (Science, Technology, Engineering and Mathematics) fields will be in computing (Association for Computing Machinery, 2013, [1]).

These developments, needs and future challenges, as well as the developments that are now taking place within quantum computing, quantum information theory and data driven discoveries (data analysis and machine learning) will play an essential role in shaping future technological developments. Most of these developments require true cross-disciplinary approaches and bridge a vast range of temporal and spatial scales and include a wide variety of physical processes. To develop computational tools for such complex systems that give physically meaningful insights requires a deep understanding of approximation theory, high performance computing, and domain specific knowledge of the area one is modeling.

Marcos Daniel Caballero
Department of Physics and Astronomy, Michigan State University, East Lansing, 48824 Michigan, USA and Department of Physics and Center for Computing in Science Education, University of Oslo, N-0316 Oslo, Norway, e-mail: caballero@pa.msu.edu

Morten Hjorth-Jensen
Department of Physics and Astronomy and National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, 48824 Michigan, USA and Department of Physics and Center for Computing in Science Education, University of Oslo, N-0316 Oslo, Norway, e-mail: hjensen@frib.msu.edu

Computing competence represents a central element in scientific problem solving, from basic education and research to essentially almost all advanced problems in modern societies. These competencies are not limited to STEM fields only. The statistical analysis of big data sets and how to use machine learning algorithms belong to the set of tools needed by almost all disciplines, spanning from the Social Sciences, Law, Education to the traditional STEM fields and Life Science. Unfortunately, many of our students at both the undergraduate and the graduate levels are unprepared to use computational modeling, data science, and high performance computing, skills that are much valued by a broad range of employers. This lack of preparation is most certainly no fault of our students, but rather a broader issue associated with how departments, colleges, and universities are keeping up with the demands of these high-tech employers. It is through this integrated computational perspective that we aim to address this. Furthermore, although many universities do offer compulsory programming courses in scientific computing, and physics departments offer one or more elective courses in computational physics, there is often not a uniform and coherent approach to the development of computing competencies and computational thinking. This has consequences for a systematic introduction and realization of computing skills and competencies and pertaining learning outcomes.

The aim of this contribution is to present examples on how to introduce a computational perspective in basic undergraduate physics courses, basing ourselves on experiences made at the University of Oslo in Norway and now also at Michigan State University in the USA. In particular, we will present the **Computing in Science Education** project from the University of Oslo [2], a project which has evolved into a Center of Excellence in Education, the Center for Computing in Science Education [3]. Similar initiatives and ideas are also being pursued at Michigan State University. The overarching aim is to strengthen the computing competencies of students, with key activities such as the establishment of learning outcomes, how to develop assessment programs and course transformations by including computational projects and exercises in a coherent way. The hope is that these initiatives can also lead to a better understanding of the scientific method and scientific reasoning as well as providing new and deeper insights about the underlying physics that governs a system.

This contribution is organized as follows. After these introductory remarks, we present briefly in the next section what we mean by computing and present possible learning outcomes that could be applied to a bachelor's degree program in physics (Sec. 1.2), which are distinguished as more general competencies and course-specific ones. In Sec. 1.3, we discuss possible paths on how to include and implement computational elements in central undergraduate physics courses. We discuss briefly how to assess various learning outcomes and how to develop a research program around this. Several examples that illustrate the links between the learning outcomes and specific mathematics and physics courses are discussed in Sec. 1.4. Finally, in the last section we present our conclusions and perspectives.

1.2 Computing competencies

The focus of this article is on computing competencies and how these help in enlarging the body of tools available to students and scientists alike, going well beyond classical tools taught in standard undergraduate courses in physics and mathematics. We will claim through various examples that computing allows for a more generic handling of problems, where focusing on algorithmic aspects results in deeper insights about scientific problems.

With **Computing** we will mean solving scientific problems using all possible tools, including symbolic computing, computers and numerical algorithms, experiments (often of a numerical character) and analytical paper and pencil solutions. We will thus, deliberately, avoid a discussion of computing and computational physics in particular as something separate from theoretical physics and experimental physics. It is common in the scientific literature to encounter statements like *Computational physics now represents the third leg of research alongside analytical theory and experiments*. In selected contexts where say high-performance topics or specific computational methodologies play a central role, it may be meaningful to separate analytical work from computational studies. We will however argue strongly, in particular within an educational context, for a view where computing means solving scientific problems with all possible tools. Through various examples in this article we will show that a tight connection between standard analytical work, combined with various algorithms and a computational approach, can help in enhancing the students' understanding of the scientific method, hopefully providing deeper insights about the physics (or other disciplines). Whether and how we achieve these outcomes is the purpose of research in computational physics education.

The power of the scientific method lies in identifying a given problem as a special case of an abstract class of problems, identifying general solution methods for this class of problems, and applying a general method to the specific problem (applying means, in the case of computing, calculations by pen and paper, symbolic computing, or numerical computing by ready-made and/or self-written software).

This generic view on problems and methods is particularly important for understanding how to apply available generic software to solve a particular problem. Algorithms involving pen and paper are traditionally aimed at what we often refer to as continuous models, of which only few can be solved analytically. The number of important differential equations in physics that can be solved analytically are rather few, limiting thereby the set of problems that can be addressed in order to deepen a student's insights about a particular physics case. On the other hand, the application of computers calls for approximate discrete models. Much of the development of methods for continuous models are now being replaced by methods for discrete models in science and industry, simply because we can address much larger classes of problems with discrete models, often also by simpler and more generic methodologies. In Sec. 1.4 we will present several examples thereof. A typical case is that where an eigenvalue problem can allow students to study the analytical solution as well as moving to an interacting quantum mechanical case where no analytical solution exists. By merely changing the diagonal matrix elements, one can solve problems that span from classical mechanics and fluid dynamics to quantum mechanics and statistical physics. Using essentially the same algorithm one can study physics cases that are covered by several courses, allowing teachers to focus more on the physical systems of interest.

There are several advantages in introducing computing in basic physics courses. It allows physics teachers to bring important elements of scientific methods at a much earlier stage in our students' education. Many advanced simulations used in physics research can easily be introduced, via various simplifications, in introductory physics courses, enhancing thereby the set of problems studied by the students (see Sec. 1.4). Computing gives university teachers a unique opportunity to enhance students' insights about physics and how to solve scientific problems. It gives the students the skills and abilities that are asked for by society. Computing allows for solving more realistic problems earlier and can provide an excellent training of creativity as well as enhancing the understanding of abstractions and generalizations. Furthermore, computing can decrease the need for special tricks and tedious algebra, and shifts the focus to problem definition, visualization, and "what if" discussions. Finally, if the setup of undergraduate courses is properly designed, with a synchronization with mathematics and computational science courses, computing can trigger further insights in mathematics and other disciplines.

1.3 Learning Outcomes and Assessment Programs

An essential element in designing a synchronization of computing in various physics (and other disciplines as well) courses is a proper definition of learning outcomes, as well as the development of assessment programs and possibly a pertinent research program on physics education. Having a strong physics education group that can define a proper research program is an essential part of such an endeavor. Michigan State University has a strong physics education group involved in such research programs. Similarly, the University of Oslo, with its recently established center of excellence in Education [3], has started to define a research program that aims at assessing the relevance and importance of computing in science education.

Physics, together with basic mathematics and computational science courses, is at the undergraduate level presented in a very homogeneous way worldwide. Most universities offer more or the less the same topics and courses, starting with Mechanics and Classical Mechanics, Waves, Electromagnetism, Quantum physics and Quantum Mechanics and ending with Statistical physics. Similarly, during the last year of the Bachelor's degree one finds elective courses on computational physics and mathematical methods in physics, in addition to a selection of compulsory introductory laboratory courses. Additionally, most physics undergraduate programs have now a compulsory introductory course in scientific programming offered by the computer science department. Here, one encounters frequently Python as the default programming language. Moreover, one finds almost the same topics covered by the basic mathematics courses required for a physics degree, from basic calculus to linear algebra, differential equations and real analysis. Many mathematics departments and/or computational science departments offer courses on numerical mathematics that are based on the first course in programming.

These developments have taken place during the last decade and several universities are attempting to include a more coherent computational perspective to our basic education. In order to achieve this, it is important to develop a strategy where the introduction of computational elements are properly synchronized between physics, mathematics, and computational science courses. This would allow physics teachers to focus more on the relevant physics. The development of learning outcomes plays a central role in this work. An additional benefit of properly-developed learning outcomes is the stimulation of cross-department collaborations as well as an increased awareness about what is being taught in different courses. Here we list several possibilities, starting with some basic algorithms and topics that can be taught in mathematics and computational science courses. We end with a discussion of possible learning outcomes for central undergraduate physics courses

1.3.1 General Learning Outcomes for Computing Competence.

Here we present some high-level learning outcomes that we expect students to achieve through comprehensive and coordinated instruction in numerical methods over the course of their undergraduate program. These learning outcomes are different from specific learning goals in that the former reference the end state that we aim for students to achieve. The latter references the specific knowledge, tools, and practices with which students should engage and discusses how we expect them to participate in that work.

Numerical algorithms form the basis for solving science and engineering problems with computers. An understanding of algorithms does not itself serve as an understanding of computing, but it is a necessary step along the path. Through comprehensive and coordinated instruction, we aim for students to have developed a deep understanding of:

- the most fundamental algorithms for linear algebra, ordinary and partial differential equations, and optimization methods;
- numerical integration including Trapezoidal and Simpson's rule, as well as multidimensional integrals;
- random numbers, random walks, probability distributions, Monte Carlo integration and Monte Carlo methods;
- root finding and interpolation methods;
- machine learning algorithms; and
- statistical data analysis and handling of data sets.

Furthermore, we aim for students to develop:

- a working knowledge of advanced algorithms and how they can be accessed in available software;
- an understanding of approximation errors and how they can present themselves in different problems; and
- the ability to apply fundamental and advanced algorithms to classical model problems as well as real-world problems as well to assess the uncertainty of their results.

Later courses should build on this foundation as much as possible. In designing learning outcomes and course contents, one should make sure that there is a progression in the use of mathematics, numerical methods and programming, as well as the contents of various physics courses. This means also that teachers in other courses do not need to use much time on numerical tools since these are naturally included in other courses.

1.3.1.1 Learning Outcomes for Symbolic Computing

Symbolic computing is a helpful tool for addressing certain classes of problems where a functional representation of the solution (or part of the solution) is needed. Through engaging with symbolic computing platforms, we aim for students to have developed:

- a working knowledge of at least one computer algebra system (CAS);
- the ability to apply a CAS to perform classical mathematics including calculus, linear algebra and differential equations; and
- the ability to verify the results produced by the CAS using some other means.

1.3.1.2 Learning Outcomes for Programming

Programming is a necessary aspect of learning computing for science and engineering. The specific languages and/or environments that students learn are less important than the nature of that learning (i.e., learning programming for the purposes of solving science problems). By numerically solving science problems, we expect students to have developed (these are possible examples):

- an understanding of programming in a high-level language (e.g., MATLAB, Python, R);
- an understanding of programming in a compiled language (e.g., Fortran, C, C++);
- the ability to to implement and apply numerical algorithms in reusable software that acknowledges the generic nature of the mathematical algorithms;
- a working knowledge of basic software engineering elements including functions, classes, modules/libraries, testing procedures and frameworks, scripting for automated and reproducible experiments, documentation tools, and version control systems (e.g., Git); and

- an understanding of debugging software, e.g., as part of implementing comprehensive tests.

1.3.1.3 Learning Outcomes for Mathematical Modeling

Preparing a problem to be solved numerically is a critical step in making progress towards an eventual solution. By providing opportunities for students to engage in modeling, we aim for them to develop the ability to solve real problems from applied sciences by:

- deriving computational models from basic principles in physics and articulating the underlying assumptions in those models;
- constructing models with dimensionless and/or scaled forms to reduce and simplify input data; and
- interpreting the model's dimensionless and/or scaled parameters to increase their understanding of the model and its predictions.

1.3.1.4 Learning Outcomes for Verification

Verifying a model and the resulting outcomes it produces are essential elements to generating confidence in the model itself. Moreover, such verifications provide evidence that the work is reproducible. By engaging in verification practices, we aim for students to develop:

- an understanding of how to program testing procedures; and
- the knowledge of testing/verification methods including the use of:
  - exact solutions of numerical models,
  - classical analytical solutions including asymptotic solutions,
  - computed asymptotic approximation errors (i.e., convergence rates), and
  - unit tests and step-wise construction of tests to aid debugging.

1.3.1.5 Learning Outcomes for Presentation of Results

The results of a computation need to be communicated in some format (i.e., through figures, posters, talks, and other forms of written and oral communication). Computation affords the experience of presenting original results quite readily. Through their engagement with presentations of their findings, we aim for students to develop:

- the ability to make use of different visualization techniques for different types of computed data;
- the ability to present computed results in scientific reports and oral presentations effectively; and
- a working knowledge of the norms and practices for scientific presentations in various formats (i.e., figures, posters, talks, and written reports).

The above learning goals and outcomes are of a more generic character. What follows here are specific algorithms that occur frequently in scientific problems. The implementation of these algorithms in various physics courses, together with problem and project solving, is a way to implement large fractions of the above learning goals.

1.3.2 Central Tools and Programming Languages

We will strongly recommend that Python is used as the high-level programming language. Other high-level environments like Mathematica and Matlab can also be presented and offered as special courses. This means that students can apply their knowledge from the basic programming course offered by most universities. Many university courses in programming make use of Python, and extend their computational knowledge in various physics classes. We recommend that the following tools are used:

1. jupyter and ipython notebooks;
2. version control software like git and repositories like GitHub and GitLab;
3. other typsetting tools like LaTeX; and
4. unit tests and using existing tools for unit tests. Python has extensive tools for this.

The notebooks can be used to hand in exercises and projects. They can provide the students with experience in presenting their work in the form of scientific/technical reports.

Version control software allows teachers to bring in reproducibility of science as well as enhancing collaborative efforts among students. Using version control can also be used to help students present benchmark results, allowing others to verify their results. Unit testing is a central element in the development of numerical projects, from microtests of code fragments, to intermediate merging of functions to final tests of the correctness of a code.

1.3.3 Specific Algorithms for Basic Physics Courses

For a bachelor's degree in physics, it is now more and more common to require a compulsory programming course, typically taught during the first two years of undergraduate studies. The programming course, together with mathematics courses, lay the foundation for the use of computational exercises and projects in various physics courses. Based on this course, and the various mathematics courses included in a physics degree, there is a unique possibility to incorporate computational exercises and projects in various physics courses, without taking away the attention from the basic physics topics to be covered.

What follows below is a suggested list of possible algorithms which could be included in central physics courses. The list is by no means exhaustive and is mainly meant as a guideline of what can be included. The examples we discuss in Sec. 1.4, illustrate how these algorithms can be included in courses like mechanics, quantum physics/mechanics, statistical and thermal physics and electromagnetism. These are all core courses in a typical bachelor's degree in physics.

1.3.4 Central Algorithms

- Ordinary differential equations
  1. Euler, modified Euler, Verlet and Runge-Kutta methods with applications to problems in courses on electromagnetism, methods for theoretical physics, quantum mechanics and mechanics.
- Partial differential equations
  1. Diffusion in one and two dimensions (statistical physics), wave equation in one and two dimensions. These are examples of physics cases which could apper in courses on me-

chanics, electromagnetism, quantum mechanics, methods for theoretical physics and Laplace's and Poisson's equations in a course on electromagnetism.

- Numerical integration

  1. Trapezoidal and Simpson's rule and Monte Carlo integration. Here one can envision applications in statistical physics, methods of theoretical physics, electromagnetism and quantum mechanics.

- Statistical analysis, random numbers, random walks, probability distributions, Monte Carlo integration and Metropolis algorithm. These are algorithms with important applications to statistical physics and laboratory courses.
- Linear Algebra and eigenvalue problems.

  1. Gaussian elimination, LU-decomposition, eigenvalue solvers, and iterative methods like Jacobi or Gauss-Seidel for systems of linear equations. These algorithms are important for several courses, classical mechanics, methods of theoretical physics, electromagnetism and quantum mechanics.

- Signal processing

  1. Discrete (fast) Fourier transforms, Lagrange/spline/Fourier interpolation, numeric convolutions & circulant matrices, filtering. Here we can think of applications in electromagnetism, quantum mechanics, and experimental physics (data acquisition)

- Root finding techniques, used in methods for theoretical physics, quantum mechanics, electromagnetism and mechanics.
- Machine Learning algorithms and Statistical Data Analysis, relevant for laboratory courses

In order to achieve a proper pedagogical introduction of these algorithms, it is important that students and teachers see how these algorithms are used to solve a variety of physics problems. The same algorithm, for example the solution of a second-order differential equation, can be used to solve the equations for the classical pendulum in a mechanics course or the (with a suitable change of variables) equations for a coupled RLC circuit in the electromagnetism course. Similarly, if students develop a program for studies of celestial bodies in the mechanics course, many of the elements of such a program can be reused in a molecular dynamics calculation in a course on statistical physics and thermal physics. The two-point boundary value problem for a buckling beam (discretized as an eigenvalue problem) can be reused in quantum mechanical studies of interacting electrons in oscillator traps, or just to study a particle in a box potential with varying depth and extension. We discuss some selected examples in section 1.4. Our coming texbook [4] will contain a more exhaustive discussion of these, combined with a more detailed list of examples and a proper discussion of learning outcomes and possible assessment programs.

In order to aid the introduction of computational exercises and projects, there is a strong need to develop educational resources. Physics is an old discipline with a large wealth of established analytical exercises and projects. In fields like mechanics, we have centuries of pedagogical developments with a strong emphasis on developing analytical skills. The majority of physics teachers are well familiar with this approach. In order to see how computing can enlarge this body of exercises and projects, and hopefully add additional insights to the physics behind various phenomena, we find it important to develop a large body of computational examples. The PICUP project, Partnership for Integration of Computation into Undergraduate physics, develops such resources for teachers and students on the integration of computational material [5]. We strongly recommend these resources.

1.3.4.1 Advanced Computational Physics Courses

Towards the end of undergraduate studies it is useful to offer a course which focuses on more advanced algorithms and presents compiled languages like C++ and Fortran, languages our students will meet in actual research. Furthermore, such a course should offer more advanced projects which train the students in actual research, developing more complicated programs and working on larger projects.

1.3.5 Physics Education Research and Computing in Science Education

The introduction of computational elements in the various courses should be, if possible, strongly integrated with ongoing research on physics education. The Physics and Astronomy department at MSU is in a unique position due to its strong research group in physics education, the PERL group [6]. Together with the Center for Computing in Science Education at the University of Oslo [3], we are now in the process of establishing new assessments and assessment methods that address several issues associated with integrating computing into science courses. The issues include but are not limited to how well students learn computing, what new insights students gain about the specific science through computing, and how students' affective states (e.g., motivation to learn, computational self-efficacy) are affected by computing . Broadly speaking, these assessments should provide deeper insights into the integration of computing in science education in general as well as provide a structured framework for assessment of our efforts and a basis for systematic studies of student learning.

The central questions that our research must address are

1. how can we assess the effect of integrating computing into science curricula on a variety of learned-centered constructs including computational thinking, motivation, self-efficacy and science identity formation,
2. how should we structure assessments to ensure valid, reliable and impactful assessment, which provides useful information to our program and central partners, and finally
3. how can the use of these structured assessments improve student outcomes in teacher-, peer-, and self-assessment.

Addressing these questions requires a combination of qualitative techniques to construct the focus of these assessments, to build assessment items and to develop appropriate assessment methods, and quantitative techniques, including advanced statistical analysis to ensure validity and reliability of the proposed methods as well as to analyze the resulting data.

The learning objectives and learning outcomes for computational methods developed as part of the first objective form parts of the basis for the assessment program, and we will also investigate the assessment of non-content learning goals such as self-efficacy and identity formation. Identifying and investigating the role of such non-content factors will be critical to support all students in achieving our computational learning goals.

The effect of integration of computational methods into basic science courses have been sparsely studied, primarily because the practice is sparse. Further progress depends now on the development of assessments that can be used for investigative, comparative and/or longitudinal studies and to establish best practices in this emerging field. Some assessments will be developed for specific courses, but we will aim for broad applicability across institutions.

1.4 Examples on how to Include computing in Physics Undergraduate Programs

Having defined possible learning outcomes, we would like now to present some examples which reflect the discussions above. These examples are mainly taken from various courses at the University of Oslo, although some of them have been used at Michigan State University. Since 2003, first via the Computing in Science Education project [2] and now through the recently established center of excellence in education Center for computing in Science Education [3], computing has been introduced across disciplines in a synchronized way.

Central elements here are a compulsory programming course with a strong mathematical flavour. This course gives a solid foundation in programming as a problem solving technique in mathematics. The line of thought is that when solving mathematical problems numerically, this should enhance algorithmic thinking, and thereby the students' understanding of the scientific process. Secondly, mathematics is at least as important as before, but should be supplemented with development, analysis, implementation, verification and validation of numerical methods. Finally, these methods are used in modeling and problem solving with numerical methods and visualisation, as well as traditional methods in various science courses, from the physical sciences to life science.

Crucial ingredients for the success of the computing in Science Education project has been the support from governing bodies as well as extensive cooperations across departmental boundaries. And finally, the willingness of several university teachers and researchers to give priority to teaching reforms and course transformations.

In addition to the above, over the years we have coordinated the use of computational exercises and numerical tools in most undergraduate courses. Furthermore, via the computing in Science Education project and now the Center for computing in Science Education, we help in updating the scientific staff's competence on computational aspects and give support (scientific, pedagogical and financial) to those who wish to revise their courses in a computational direction. This may include the organization of courses for university teachers. Summer students aid in developing and introducing computational exercises and several new textbooks have been developed, from the basic mechanics course to a course in statistical physics [7–9].

1.4.1 The Physics Undergraduate Program at the University of Oslo.

The layout of the physics bachelor's degree program at the University of Oslo is shown in table 1.1.

Table 1.1 The bachelor's degree program in physics at the University of Oslo, Norway

6th Semester	Elective	Elective	Elective
5th Semester	FYS2160 Statistical physics	FYS3110 Quantum Mechanics	Elective
4th Semeters	FYS2130 Waves and Motion	FYS2140 Quantum physics	FYS2150 physics Laboratory
3rd Semester	FYS1120 Electromagnetism	MAT1120 Linear Algebra	AST2000 Intro to Astrophysics
2nd Semester	FYS-MEK1100 Mechanics	MEK1100 Vector Calculus	MAT1110 Calculus and Linear Algebra
1st Semester	MAT 1100 Calculus	MAT-INF1100 Modeling and Computations	IN1900 Intro to Programming
Credits	10 ECTS	10 ECTS	10 ECTS

In the first semester the students encounter the first level of syncronization between computing courses and mathematics courses. As an example, consider the numerical evaluation of an integral by the trapezoidal rule. Integral calculus is typically discussed first in the calculus course MAT1100. Thereafter, the algorithm for computing the integral using the trapezoidal rule for an interval $x \in [a,b]$

$$\int_a^b (f(x)dx \approx \frac{1}{2}\left[f(a)+2f(a+h)+\cdots+2f(b-h)+f(b)\right],$$

is discussed and developed in MAT-INF1100, the modeling and computations course that serves as an intermediate step between the standard calculus course and the programming course. Finally, the algorithm is implemented in IN1900, introduction to programming with scientific applications. We show here a typical Python code which exemplifies this.

```
from math import exp, log, sin
def Trapez(a,b,f,n):
  h = (b-a)/float(n)
  s = 0
  x = a
  for i in range(1,n,1):
     x = x+h
     s = s+ f(x)
  s = 0.5*(f(a)+f(b)) +s
  return h*s

def f1(x):
   return exp(-x*x)*log(1+x*sin(x))

a = 1; b = 3; n = 1000
result = Trapez(a,b,f1,n)
print(result)
```

Here we have defined an integral given by

$$I = \int_1^3 dx \exp\left(-x^2\right)\log\left(1+x\sin\left(x\right)\right).$$

Coming back to the above learning outcomes, we would like to emphasize that Python offers an extremely versatile programming environment, allowing for the inclusion of analytical studies in a numerical program. Here we show an example code with the trapezoidal rule using Python's symbolic package **SymPy** [10] to evaluate an integral and compute the absolute error with respect to the numerically evaluated one of the integral $\int_0^1 dx x^2 = 1/3$. This is in shown in the following code part

```
# define x as a symbol to be used by sympy
x = Symbol('x')
exact = integrate(function(x), (x, 0.0, 1.0))
print("Sympy integration=", exact)
```

where we have defined the function to integrate in the complete Python program that follows here.

```
from math import *
from sympy import *
def Trapez(a,b,f,n):
  h = (b-a)/float(n)
  s = 0
  x = a
  for i in range(1,n,1):
     x = x+h
     s = s+ f(x)
  s = 0.5*(f(a)+f(b)) +s
  return h*s
```

```
# function to compute pi
def function(x):
    return x*x

a = 0.0; b = 1.0; n = 100
result = Trapez(a,b,function,n)
print("Trapezoidal rule=", result)
# define x as a symbol to be used by sympy
x = Symbol('x')
exact = integrate(function(x), (x, 0.0, 1.0))
print("Sympy integration=", exact)
# Find relative error
print("Relative error", abs((exact-result)/exact))
```

The following extended version of the trapezoidal rule allows us to plot the relative error by comparing with the exact result. By increasing to 10^8 points we arrive at a region where numerical errors start to accumulate, as seen in the figure 1.1.

```
from math import log10
import numpy as np
from sympy import Symbol, integrate
import matplotlib.pyplot as plt
# function for the trapezoidal rule
def Trapez(a,b,f,n):
   h = (b-a)/float(n)
   s = 0
   x = a
   for i in range(1,n,1):
       x = x+h
       s = s+ f(x)
   s = 0.5*(f(a)+f(b)) +s
   return h*s
# function to compute pi
def function(x):
    return x*x
# define integration limits
a = 0.0; b = 1.0;
# find result from sympy
# define x as a symbol to be used by sympy
x = Symbol('x')
exact = integrate(function(x), (x, a, b))
# set up the arrays for plotting the relative error
n = np.zeros(9); y = np.zeros(9);
# find the relative error as function of integration points
for i in range(1, 8, 1):
    npts = 10**i
    result = Trapez(a,b,function,npts)
    RelativeError = abs((exact-result)/exact)
    n[i] = log10(npts); y[i] = log10(RelativeError);
plt.plot(n,y, 'ro')
plt.xlabel('n')
plt.ylabel('Relative error')
plt.show()
```

The last example shows the potential of combining numerical algorithms with symbolic calculations, allowing thereby students to validate their algorithms. With concepts like unit testing, one has the possibility to test and verify several or all parts of the code. Validation and verification are then included *naturally*.

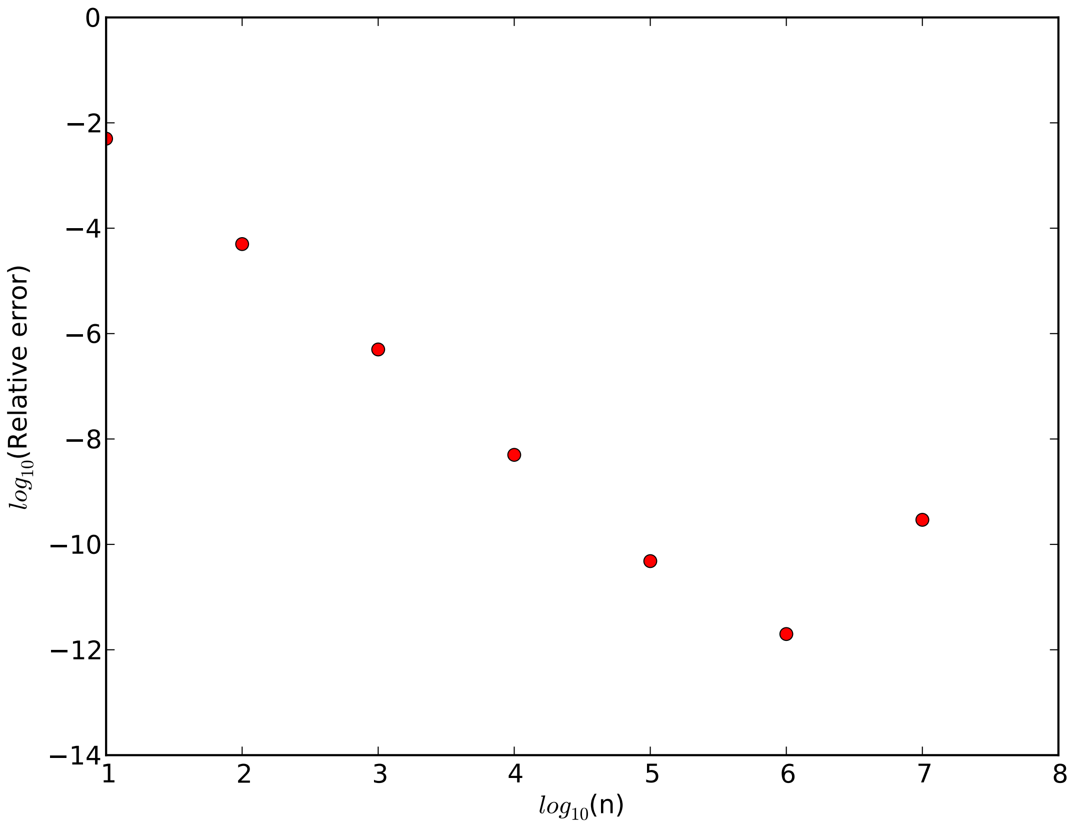

Fig. 1.1 Log-log plot of the relative error as function of the number of integration points. Till approximately $n = 10^6$, the relative error follows the predicted mathematical error of the trapezoidal rule. For higher numbers of integration points, numerical round off errors and loss of numerical precision give an increasing relative error.

The above example allows the student to also test the mathematical error of the algorithm for the trapezoidal rule by changing the number of integration points. The students get trained from day one to think error analysis. Figure 1.1 shows clearly the region where the relative error starts increasing. The mathematical error which follows the trapezoidal rule goes as $O(h^2)$ where h is the chosen numerical step size. It is proportional to the inverse of the number of integration points n, that is $h \propto 1/n$.

Before numerical round-off errors and loss of numerical precision kick in (near $h \sim 10^{-6}$) we see that the relative error in the log-log plot has a slope which follows the mathematical error.

There are several additional benefits here. The general learning outcomes on computing can be included as in for example the following ways. We can easily bake in how to structure a code in terms of functions and modules, or how to read input data flexibly from the command line or how to write unit tests etc. The conventions and techniques outlined here will save students a lot of time when one extends incrementally software over time, from simpler to more complicated problems. In the next subsection we show how algorithms for solving sets of ordinary differential equations and finding eigenvalues can be reused in different courses with minor modifications only.

1.4.2 From Mathematics to Physics

We assume that our students know how to solve and study systems of ordinary differential with initial conditions only. Later in this section we will venture into two-point boundary value problems that can be studied and solved with eigenvalue solvers.

Let us start with initial value problems and ordinary differential equations. Such equations appear in a wealth of physics applications. Typical examples students will encounter are the classical pendulum in a mechanics course, an RLC circuit in the course on electromagnetism, the modeling of the Solar system in an Astrophysics course and many other cases. The essential message is that, with properly scaled equations, students can use essentially the same algorithms to solve these problems, either starting with a simple modified Euler algorithm or a Runge-Kutta class of algorithms or the so-called Verlet class of algorithms, to mention a few.

The idea is that algorithms students develop and use in one course can be reused in other courses. This allows students to make the relevant abstractions discussed above, opening up for a much wider range of applicabilities.

Here we look at two familiar cases from mechanics and electromagnetism, the equations for the classical pendulum and those for an RLC circuit. When properly scaled, these equations are essentially the same. To scale equations, either in terms of dimensionless variables or appropriate variables, is an important aspect which allows the students to see the potential for abstractions and hopefully see how the problems studied in say a mechanics course can be transferred to other fields.

The classical pendulum with damping and external force as it could appear in a mechanics course is given by the following equation of motion for the angle θ as function of time t

$$ml\frac{d^2\theta}{dt^2}+\nu\frac{d\theta}{dt}+mgsin(\theta)=Acos(\omega t),$$

where m is its mass, l the length, ν a damping factor and A the amplitude of an applied external source with frequency ω. The solution of this type of equations (second-order differential equations with given initial conditions) is something the students encounter the first semester thorugh the courses IN1900 and MAT-INF1100 at the University of Oslo. At Michigan State University there is now a compulsory course for physics majors that includes many of these elements. With this background, students are already familiar with the numerical solution and visualization of such equations. If we now move to a course on electromagnetism, we encounter almost the same equation for an RLC circuit, namely

$$L\frac{d^2Q}{dt^2}+\frac{Q}{C}+R\frac{dQ}{dt}=Acos(\omega t),$$

where L is the inductance, R the applied resistance, Q the time-dependent charge and C the capacitance.

Let us consider first the classical pendulum equations with damping and an external force and define the scaled velocity $\hat{v}$ as

$$\frac{d\theta}{d\hat{t}}=\hat{v},$$

where we have defined a dimensionless time variable $\hat{t}$. With the equation for the velocity we can rewrite the second-order differential in terms of two coupled first-order differential equations where the second equation represents the acceleration

$$\frac{d\hat{v}}{d\hat{t}}=Acos(\hat{\omega}\hat{t})-\hat{v}\xi-\sin(\theta).$$

We have scaled the equations with $\omega_0 = \sqrt{g/l}$, $\hat{t} = \omega_0 t$ and $\xi = mg/\omega_0 \nu$. The frequency ω_0 defines a so-called natural frequency defined by the gravitational acceleration g and the length of the pendulum l. The frequency $\hat{\omega} = \omega/\omega_0$. In a similar way, our RLC circuit can now be rewritten in terms of two coupled first-order differential equations,

$$\frac{dQ}{d\hat{t}} = \hat{I},$$

and

$$\frac{d\hat{I}}{d\hat{t}} = Acos(\hat{\omega}\hat{t}) - \hat{I}\xi - Q,$$

with $\omega_0 = 1/\sqrt{LC}$, $\hat{t} = \omega_0 t$ and $\xi = CR\omega_0$. Here we see that the natural frequency is defined in terms of the physical parameters L and C.

The equations are essentially the same, the main differences reside in the different scaling constants and the introduction of a non-linear term for the angle θ in the pendulum equation. The differential solver the students end up writing in the mechanics course (which comes normally before the course on electromagnetism) can then be reused in the electromagnetism course, with a great potential for further abstraction.

Let us now move to another frequently encountered problem in several physics courses, namely that of a two-point boundary value problem. In the examples below we will see again that if the equations are properly scaled, we can reuse the same algorithm for solving different physics problems. Here we will start with the equations for a buckling beam (a case which can be found in a mechanics course or a course on mathematical methods in physics). Thereafter, with a simple change of variables and constants, the same problem can be used to study a quantum mechanical particle confined to move in an infinite potential well. By simply changing the diagonal matrix elements of the discretized differential equation problem, we can study particles that move in a harmonic oscillator potential or other types of quantum-mechanical one-body or selected two-body problems. With slight modifications to the matrix that results from the discretization of a second derivative, we can study Poisson's equation in one dimension, a problem of relevance in electromagnetism.

Let us start with the buckling beam. This is a two-point boundary value problem

$$R\frac{d^2u(x)}{dx^2} = -Fu(x),$$

where $u(x)$ is the vertical displacement, R is a material specific constant, F is the applied force and $x \in [0,L]$ with $u(0) = u(L) = 0$. We scale the equation with $x = \rho L$ and $\rho \in [0,1]$ and get (note that we change from $u(x)$ to $v(\rho)$)

$$\frac{d^2v(\rho)}{dx^2} + Kv(\rho) = 0,$$

which is, when discretized (see below), nothing but a standard eigenvalue problem with $K = FL^2/R$. Here we can assume that either the force F or the material specific rigidity R are unknown. If we replace $R = -\hbar^2/2m$ and $-F = \lambda$, we have the quantum mechanical variant for a particle moving in a well with infinite walls at the endpoints. The way to solve these equations numerically is to discretize the second derivative and the right hand side as

$$-\frac{v_{i+1} - 2v_i + v_{i-i}}{h^2} = \lambda v_i,$$

with $i = 1,2,\ldots,n$. Here h is the step size which is defined by the number of integration (or mesh) points. We need to add to this system the two boundary conditions $v(0) = v_0$ and $v(1) = v_{n+1}$, although they are not needed in the solution of the equations since their values are known. For all integration points $i = 1, 2, \dots, n$ the set of equations to solve result in a so-called tridiagonal Toeplitz matrix (a special case from the discretized second derivative)

$$
\mathbf{A} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & \dots & \dots & \dots & \dots & \dots \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}
$$

and with the corresponding vectors $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$ allows us to rewrite the differential equation as a standard eigenvalue problem

$$
\mathbf{A}\mathbf{v} = \lambda \mathbf{v}.
$$

The tridiagonal Toeplitz matrix has analytical eigenpairs, providing us thereby with an invaluable check on the equations to be solved.

If we stay with quantum mechanical one-body problems (or special interacting two-body problems) adding a potential along the diagonal elements allows us to reuse this problem for many types of physics cases. To see this, let us assume we are interested in the solution of the radial part of Schrödinger's equation for one electron. This equation reads

$$
-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr} - \frac{l(l+1)}{r^2}\right)R(r) + V(r)R(r) = ER(r).
$$

Suppose in our case $V(r)$ is the harmonic oscillator potential $(1/2)kr^2$ with $k = m\omega^2$ and E is the energy of the harmonic oscillator in three dimensions. The oscillator frequency is ω and the energies are

$$
E_{nl} = \hbar\omega\left(2n + l + \frac{3}{2}\right),
$$

with $n = 0, 1, 2, \dots$ and $l = 0, 1, 2, \dots$.

Since we have made a transformation to spherical coordinates it means that $r \in [0, \infty)$. The quantum number l is the orbital momentum of the electron. In order to find analytical solutions for this problem, we would substitute $R(r) = (1/r)u(r)$ (which gives $u(0) = u(\infty) = 0$ and thereby easier boundary conditions) and obtain

$$
-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r) + \left(V(r) + \frac{l(l+1)}{r^2}\frac{\hbar^2}{2m}\right)u(r) = Eu(r).
$$

The boundary conditions are $u(0) = 0$ and $u(\infty) = 0$.

In order to scale the equations, we introduce a dimensionless variable $\rho = (1/\alpha)r$ where α is a constant with dimension length and get

$$
-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}v(\rho) + \left(V(\rho) + \frac{l(l+1)}{\rho^2}\frac{\hbar^2}{2m\alpha^2}\right)v(\rho) = Ev(\rho).
$$

Let us choose $l = 0$ for the mere sake of simplicity. Inserting $V(\rho) = (1/2)k\alpha^2\rho^2$ we end up with

$$
-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}v(\rho) + \frac{k}{2}\alpha^2\rho^2 v(\rho) = Ev(\rho).
$$

We multiply thereafter with $2m\alpha^2/\hbar^2$ on both sides and obtain

$$-\frac{d^2}{d\rho^2}v(\rho)+\frac{mk}{\hbar^2}\alpha^4\rho^2 v(\rho)=\frac{2m\alpha^2}{\hbar^2}Ev(\rho).$$

A natural length scale comes out automatically when scaling. To see this, since α is constant we are left to determine, we determine α by requiring that

$$\frac{mk}{\hbar^2}\alpha^4=1.$$

This defines a natural length scale in terms of the various physical constants that determine the equation. The final expression, inserting $k=m\omega^2$ is

$$\alpha=\left(\frac{\hbar}{m\omega}\right)^{1/2}.$$

If we were to replace the harmonic oscillator potential with the attractive Coulomb interaction from the hydrogen atom, the parameter α would equal the Bohr radius a_0. This way students see the general properties of a two-point boundary value problem and can reuse the code they developed for a mechanics course to the subsequent quantum mechanical course.

Defining

$$\lambda=\frac{2m\alpha^2}{\hbar^2}E,$$

we can rewrite Schroedinger's equation as

$$-\frac{d^2}{d\rho^2}v(\rho)+\rho^2 v(\rho)=\lambda v(\rho).$$

This is similar to the equation for a buckling beam, except for the potential term. In three dimensions with our scaling, the eigenvalues for $l=0$ are $\lambda_0=3,\lambda_1=7,\lambda_2=11,\dots.$

If we define first the diagonal matrix element

$$d_i=\frac{2}{h^2}+V_i,$$

and the non-diagonal matrix element

$$e_i=-\frac{1}{h^2},$$

we can rewrite the Schröedinger equation as

$$d_iu_i+e_{i-1}v_{i-1}+e_{i+1}v_{i+1}=\lambda v_i,$$

where v_i is unknown and $i=1,2,\dots,n$. We can reformulate the latter equation as a matrix eigenvalue problem

$$\begin{bmatrix} d_1 & e_1 & 0 & 0 & \dots & 0 & 0 \\ e_1 & d_2 & e_2 & 0 & \dots & 0 & 0 \\ 0 & e_2 & d_3 & e_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & d_{n-1} & e_{n-1} \\ 0 & \dots & \dots & \dots & \dots & e_{n-1} & d_n \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ \dots \\ \dots \\ \dots \\ v_n \end{bmatrix}=\lambda\begin{bmatrix} cv_1 \\ v_2 \\ \dots \\ \dots \\ \dots \\ v_n \end{bmatrix}$$

or if we wish to be more detailed, we can write the tridiagonal matrix as

$$\begin{bmatrix} \frac{2}{h^2}+V_1 & -\frac{1}{h^2} & 0 & 0 & \dots & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2}+V_2 & -\frac{1}{h^2} & 0 & \dots & 0 & 0 \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2}+V_3 & -\frac{1}{h^2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \frac{2}{h^2}+V_{n-1} & -\frac{1}{h^2} \\ 0 & \dots & \dots & \dots & \dots & -\frac{1}{h^2} & \frac{2}{h^2}+V_n \end{bmatrix}.$$

The following Python code sets up the matrix to diagonalize by defining the minimun and maximum values of r with a maximum value of integration points. It plots the eigenfunctions of the three lowest eigenstates.

```
#Program which solves the one-particle Schrodinger equation
#for a potential specified in function
#potential().

from matplotlib import pyplot as plt
import numpy as np
#Function for initialization of parameters
def initialize():
   RMin = 0.0
   RMax = 10.0
   lOrbital = 0
   Dim = 400
   return RMin, RMax, lOrbital, Dim
# Harmonic oscillator potential
def potential(r):
   return 0.5*r*r

#Get the boundary, orbital momentum and number of integration points
RMin, RMax, lOrbital, Dim = initialize()

#Initialize constants
Step  = RMax/(Dim)
DiagConst = 2.0/ (Step*Step)
NondiagConst = -1.0 / (Step*Step)
OrbitalFactor = lOrbital * (lOrbital + 1.0)

#Calculate array of potential values
v = np.zeros(Dim)
r = np.linspace(RMin,RMax,Dim)
for i in range(Dim):
   r[i] = RMin + (i+1) * Step;
   v[i] = potential(r[i]) + OrbitalFactor/(r[i]*r[i]);

#Setting up a tridiagonal matrix and finding eigenvectors and eigenvalues
Matrix = np.zeros((Dim,Dim))
Matrix[0,0] = DiagConst + v[0];
Matrix[0,1] = NondiagConst;
for i in xrange(1,Dim-1):
   Matrix[i,i-1] = NondiagConst;
   Matrix[i,i] = DiagConst + v[i];
   Matrix[i,i+1] = NondiagConst;
Matrix[Dim-1,Dim-2] = NondiagConst;
Matrix[Dim-1,Dim-1] = DiagConst + v[Dim-1];
# diagonalize and obtain eigenvalues, not necessarily sorted
EigValues, EigVectors = np.linalg.eig(Matrix)
# sort eigenvectors and eigenvalues
permute = EigValues.argsort()
EigValues = EigValues[permute]
EigVectors = EigVectors[:,permute]
```

```
# now plot the results for the three lowest lying eigenstates
for i in range(3):
   print(EigValues[i])
FirstEigvector = EigVectors[:,0]
SecondEigvector = EigVectors[:,1]
ThirdEigvector = EigVectors[:,2]
plt.plot(r, FirstEigvector**2 ,'b-',r, SecondEigvector**2 ,'g-',r, ThirdEigvector**2 ,'r-')
plt.axis([0,4.6,0.0, 0.025])
plt.xlabel(r'$r$')
plt.ylabel(r'Radial probability $r^2|R(r)|^2$')
plt.title(r'Radial probability distributions for three lowest-lying states')
plt.savefig('eigenvector.pdf')
plt.show()
```

The last example shows the potential of combining numerical algorithms with analytical results (or eventually symbolic calculations), allowing thereby students to test their physics understanding. One can easily switch to other potentials by simply redefining the potential function. For example, a finite box potential can easily be defined as

```
# Finite depth and range box potential, with strength V and range a
def potential(r):
   if r >= 0.0 and r <= 10.0:
      V = -0.05
   else:
      V =0.0
   return V
```

Thereafter, the students can explore the role of the potential depth and the range of the potential. Analyzing the eigenvectors gives additional information about the spatial degrees of freedom in terms of different potentials. The possibility to visualize the results immediately, as shown in figure 1.2, aids in providing students with a deeper understanding of the relevant physics.

This example contains also many of the computing learning outcomes we discussed above, in addition to those related to the physics of a particular system. We see that, by proper scaling, the students can make further abstractions and explore other physics cases easily where no analytical solutions are known. With unit testing and analytical results they can validate and verify their algorithms.

The above example allows the student to test the mathematical error of the algorithm for the eigenvalue solver by simply changing the number of integration points. Again, as discussed above in connection with the trapezoidal rule, the students get trained to develop an understanding of the error analysis and where things can go wrong. The algorithm can be tailored to any kind of one-particle problem used in quantum mechanics.

A simple rewrite allows for the reuse in linear algebra problems for solution of say Poisson's equation in electromagnetism, or the diffusion equation in one dimension. To see this and how the same matrix can be used in a course in electromagnetism, let us consider Poisson's equation. We assume that the electrostatic potential Φ is generated by a localized charge distribution $\rho(\mathbf{r})$. In three dimensions the pertinent equation reads

$$\nabla^2\Phi = -4\pi\rho(\mathbf{r}).$$

With a spherically symmetric potential Φ and charge distribution $\rho(\mathbf{r})$ and using spherical coordinates, the relevant equation to solve simplifies to a one-dimensional equation in r, namely

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\Phi}{dr}\right) = -4\pi\rho(r),$$

which can be rewritten via a substitution $\Phi(r) = \phi(r)/r$ as

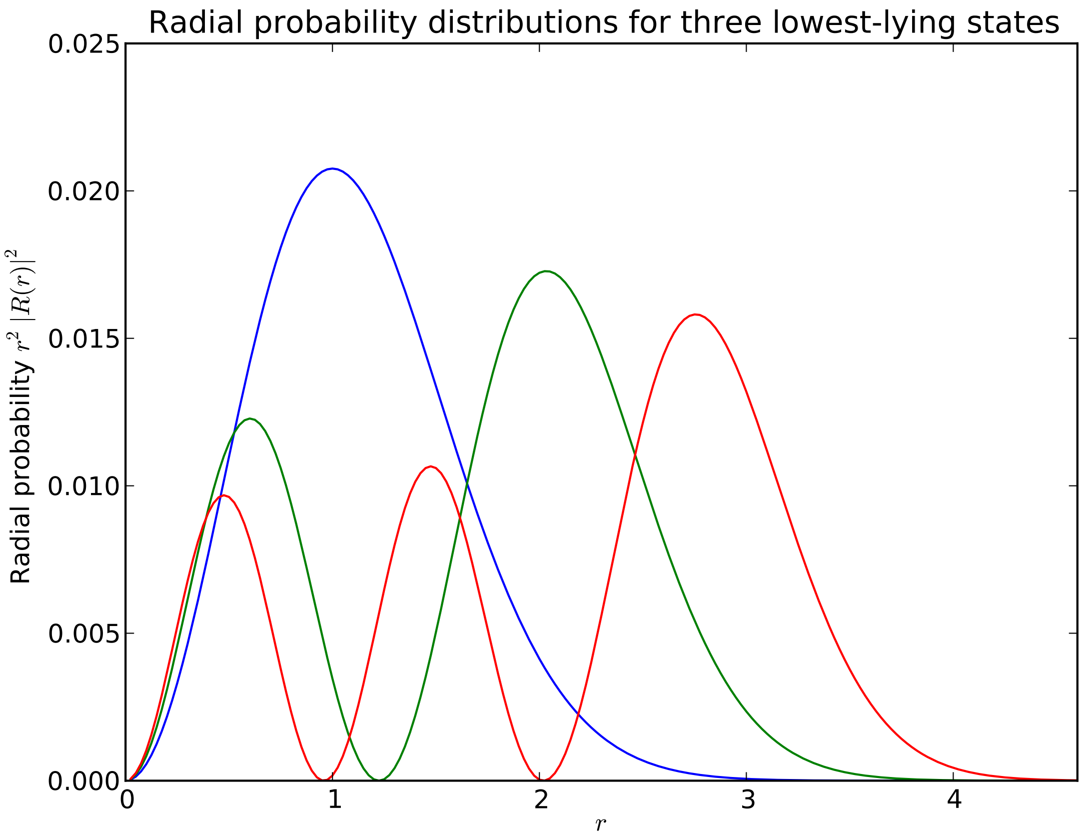

Fig. 1.2 Plot of the eigenfunctions of the three lowest-lying eigenvalues for a harmonic oscillator problem in three dimensions. The students can easily change the type of potential and explore the physics that arises from these potentials.

$$\frac{d^2\phi}{dr^2} = -4\pi r\rho(r).$$

The inhomogeneous term f or source term is given by the charge distribution ρ multiplied by r and the constant -4π.

We can rewrite this equation by letting $\phi \to u$ and $r \to x$. Scaling again the equations and replacing the right hand side with a function $f(x)$, we can rewrite the equation as

$$-u''(x) = f(x).$$

Our scaling gives us again $x \in [0,1]$ and the two-point boundary value problem with $u(0) = u(1) = 0$. With $n+1$ integration points and the step length defined as $h = 1/(n)$ and replacing the continuous function u with its discretized version v, we get the following equation

$$-\frac{v_{i+1} + v_{i-1} - 2v_i}{h^2} = f_i \qquad \text{for } i = 1,\ldots,n,$$

where $f_i = f(x_i)$. Bringing up again the tridiagonal Toeplitz matrix,

$$\mathbf{A} = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & & -1 & 2 & -1 \\ 0 & \dots & & 0 & -1 & 2 \end{bmatrix},$$

our problem becomes now a classical linear algebra problem

$$\mathbf{Av} = \mathbf{f},$$

with the unknown function $\mathbf{v}$. Using standard LU decomposition algorithms [11] (here one can use the so-called Thomas algorithm which reduces the number of floating point operations to $O(n)$) one can easily find the solution to this problem.

These examples demonstrate how one can, with a discretized second derivative, solve physics problems that arise in different undergraduate courses using standard linear algebra and eigenvalue algorithms and ordinary differential equations, allowing thereby teachers to focus on the interesting physics. Many of these problems can easily be linked up with ongoing research. This opens up for many interesting perspectives in physics education. We can bring in at a much earlier stage in our education basic research elements and perhaps even link with ongoing research during the first year of undergraduate studies.

Instead of focusing on tricks and mathematical manipulations to solve the continuous problems for those few case where an analytical solution can be found, the discretization of the continuous problem opens up for studies of many more interesting and realistic problems. However, we have seen that in order to verify and validate our codes, the existence of analytical solutions offer us an invaluable test of our algorithms and programs. The analytical results can either be included explicitely or via symbolic software like Python's Sympy package. Thus, computing stands indeed for solving scientific problems using all possible tools, including symbolic computing, computers and numerical algorithms, numerical experiments (as well as real experiments if possible) and analytical paper and pencil solutions.

The cases we have presented here represent only a limited set of examples. A longer version of this article, with more examples and details on assessments programs, is under preparation as a textbook [4]. The possible learning outcomes we defined for various physics courses are often based on the above simple discretization. With basic knowledge on how to solve linear algebra problems, eigenvalue porblems and differential equations, topics normally taught in mathematics and computational science courses, we can offer our students a much more challenging and interesting education. Furthermore, we give our students the competencies which are required by future employers, either in the private or the public sector.

1.5 Conclusions and Perspectives

In this contribution, we have outlined some of the basic elements that we feel are necessary to address in order to introduce computing in various undergraduate physics courses. Some of the conclusions we would like to emphasize include a proper definition of computing, the development of learning outcomes that apply to both computational science, mathematics, and physics courses as well as proper assessment programs.

Collaboration across departments is necessary in order to achieve a synchronization between various topics and learning outcomes, as well as an early introduction to programming. Many universities require such courses as part of a physics degree. Coordinating such a programming course with mathematics courses and other science courses results in a better coordination of both learning outcomes and computing skills and abilities. The experiences we have drawn from the University of Oslo and Michigan State University show that an early and compulsory programming course, which includes central scientific elements, is important in order to integrate properly a computational perspective in our physics education.

The benefits are many, in particular it allows us to make our research more visible in early undergraduate physics courses, enhancing research-based teaching with the possibility to focus more on understanding and increased insight. It gives also our candidates the skills and abilities that are requested by society at large, both from the private and the public sectors. With computing, we emphasize a broader and more up-to-date education with a problem-based orientation, often requested by potential employers. Furthermore, our experiences from the both universities indicate that a discussion of computing across disciplines results in an increased impetus for broad cooperation in teaching and a broader focus on university pedagogical topics.

We are now in the process of developing computing learning outcomes with examples for central physics courses. Together with a research based assessment program, we will be able to answer central questions like whether the introduction of computing increases a student's insights and understanding of the underlying physics.

Acknowledgements MHJ's work is supported by U.S. National Science Foundation Grant No. PHY-1404159. MDC's work is supported by U.S. National Science Foundation Grants Nos. DRL-1741575, DUE-1725520, DUE-1524128, DUE-1504786, and DUE-1431776. Both authors acknowledge support from the recently established Center for Computing in Science Education, University of Oslo, Norway.

References

1. Association for Computing Machinery, 2013, `http://pathways.acm.org/executive-summary.html`.
2. Computing in Science Education project, University of Oslo, Norway, `https://www.mn.uio.no/english/about/collaboration/cse/`.
3. Center for Computing in Science Education, University of Oslo, Norway, `http://www.mn.uio.no/ccse/english/`.
4. Marcos Daniel Caballero and Morten Hjorth-Jensen, *Integrating a Computational Perspective in Physics Courses*, in preparation.
5. Partnership for Integration of Computation into Undergraduate Physics, `http://www.compadre.org/picup/`PICUP project.
6. Physics Education Research Group at Michigan State University, `https://perl.natsci.msu.edu/`.
7. Anders Malthe-Sørenssen, *Elementary Mechanics Using Python*, Undergraduate Lecture Notes in Physics (Springer, Berlin, 2015).
8. Arnt Inge Vistnes, *Waves and Motion*, in press, Undergraduate Texts in Physics (Springer, Berlin, 2018).
9. Anders Malthe-Sørenssen and Dag Kristian Dysthe, *Statistical and Thermal Physics*, in press, Undergraduate Texts in Physics (Springer, Berlin, 2018/2019).
10. Symbolic Python, .
11. Gene H. Golub and Charles F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, 1996, Baltimore and London).

Predicting time to graduation at a large enrollment American university

John M. Aiken[1,3] Riccardo De Bin[2] Morten Hjorth-Jensen[1, 3, 4] Marcos D. Caballero[1, 3, 5]

1 University of Oslo Department of Physics, Centre for Computing in Science Education, Blindern, Oslo, Norway
2 University of Oslo Department of Mathematics and Statistics, Blindern, Oslo, Norway
3 Michigan State University Department of Physics and Astronomy, East Lansing, Michigan, USA
4 Michigan State University, National Superconducting Cyclotron Laboratory and Facility for Rare Ion Beams, East Lansing, Michigan, USA
5 CREATE for STEM Institute, Michigan State University, East Lansing, Michigan 48824

* j.m.aiken@fys.uio.no

Abstract

The time it takes a student to graduate with a university degree is mitigated by a variety of factors such as their background, the academic performance at university, and their integration into the social communities of the university they attend. Different universities have different populations, student services, instruction styles, and degree programs, however, they all collect institutional data. This study presents data for 160,933 students attending a large American research university. The data includes performance, enrollment, demographics, and preparation features. Discrete time hazard models for the time-to-graduation are presented in the context of Tinto's Theory of Drop Out. Additionally, a novel machine learning method: gradient boosted trees, is applied and compared to the typical maximum likelihood method. We demonstrate that enrollment factors (such as changing a major) lead to greater increases in model predictive performance of when a student graduates than performance factors (such as grades) or preparation (such as high school GPA).

1 Introduction

University students must meet a number of objectives to obtain degrees and in many cases this can prolong their time at the university [1] or they drop out altogether [2]. During their studies, American students may take on substantial financial obligations when choosing to pursue degrees and extending beyond the "four-year degree" can greatly increase the cost of obtaining that degree [3]. Thus understanding the paths that students take towards degree completion can help faculty and administrators better serve student populations to meet their educational goals. Tinto's Theory of Drop Out [4] has seen large acceptance in its ability to describe the factors that influence a student's path towards degree completion [5, 6, 7, 8, 9]. Tinto theorized that a student's college drop out decision is mediated by two conglomerate features: 1) educational goal commitment, and 2) institutional commitment. He further theorized that these commitments are dynamic. Students begin their university studies with initial commitments that are then mediated by how students participate in the academic and social systems of the university. Tinto suggested that using institutional data such as that collected by university registrars, could quantify these relationships and provide predictive models for determining student's at-risk to drop out or alternatively, graduate.

This paper examines student ultimate success at university. We define this success as obtaining a bachelor's degree. It uses 20 years of institutional data for 160,933 students attending a large enrollment American research university. This paper examines two specific research questions:

1. In the context of Tinto's Theory of Drop Out [4], what factors (such as grades, participation in a major, student background, etc.) contribute to the time it takes to obtain a degree at a large enrollment research university in the United States? How does the contribution of these factors change for longer durations until graduation?

2. How can recent innovations in statistics and machine learning, such as gradient boosting and xgboost, improve educational model performance?

We focus on a comparison of student participation in an academic system, their involvement in the social system, and their initial conditions due to high school training. We demonstrate that enrollment factors (e.g., changing a major) and academic performance are more important to predicting when a student graduates than pre-college experiences (e.g., high school GPA). Additionally, we compare traditional statistical modeling (maximum likelihood estimation) to new techniques from machine learning (gradient boosting [10]) and demonstrate that the machine learning methods are more effective at estimating the function predicting time-to-graduation.

It is important to use an analysis technique that respects the dynamics assumptions of Tinto's Theory of Drop Out. In this paper we use a Discrete Time Hazard Model framework [11]. Discrete time hazard model is useful when the classic Cox regression model assumption that events happen on a continuous interval no longer hold valid [11]. This is true when data has highly discretized time intervals such as semesters enrolled. Traditionally this modeling is done with logistic regression via maximum likelihood estimation [11]. In this paper we compare two models that fit the logistic model including logistic regression (maximum likelihood estimation) and a gradient boosted tree model (xgboost, [10]).

2 Background

In this section Tinto's theory[4] and the use of that theory will be described, the use of discrete time hazard model [11] to predict when students graduate and/or drop out will be described, and the value of a new machine learning method known as gradient boosting will be described [10]. The functional descriptions of the discrete time hazard model and gradient boosting are described in Section 4.

2.1 Tinto's Theory of Drop Out

Tinto's Theory of Drop Out describes a students intent to drop out based on the interplay of two quantities: 1) a student's commitment to education, and 2) a student's commitment to a specific institution. A student's commitment to education is mediated by the initial state of a student entering the university and the dynamics that occur while the student attends university. These dynamics are dictated by a student's participation and acceptance into the social and academic communities that are at a university. A student's commitment to an institution is tempered by many factors such as the educational goals available at an institution (e.g., a technical university degree offerings versus a liberal arts university), family commitment to a university, and social acceptance at the university. While Tinto's theory explicitly attempts to describe why students drop out from college, it is not uncommon that this theory is used to study student graduation from college given that graduation is an alternative outcome to dropping out [9, 12, 1]. In this paper we focus on a student's commitment to education as the framing for features that predict when a student will graduate if they do so.

It is common to characterize Tinto's theory of dropout as a discrete time hazard problem [8, 9, 12, 13, 1]. Tinto's theory posits that a student's educational commitment changes over time as they work toward's a degree and that effects, such as high school GPA, that may be strong in the beginning of a degree program are weak toward's the end of a degree program. Discrete time hazard modeling provides a systematic method for examining the various effects on the probability to graduate over time [11]. To that end, discrete time hazard modeling has been used to examine the dynamic impact of various effects that Tinto predicts effect a student's educational commitment. College GPA has been demonstrated to have a profound but diminishing time varying effect on graduation [8, 14, 1]. Financial aid and money spent on student services has also demonstrated a time varying (diminishing) effect on the probability to graduate [14, 1]. First generation student's are considered high risk for drop out but this effect diminishes over time indicating that first generation students need specific resources other students may not [9]. Non-traditional enrollment factors such as delaying enrollment, working while enrolled, and stopping education for some period of time can all have a negative time varying effect on graduation [12]. In each case these studies showed a statistically significant time varying impact that supports the dynamic claims of Tinto's theory.

A student's preparation has long been known to impact a student's ability to graduate. Preparation is assessed in many ways and can be represented as the experiences a student has in school and also their present innate ability to perform a specific function. Math preparation correlates with both performance in university and graduating [15, 16]. The same is true for physics and english preparation [17, 18, 19]. High school GPA and SAT scores typically account for some but not all of student success at university (GPA, etc.) [20, 21, 22]. Preparation for university often can be experienced differentially as well. Women in STEM (Science, Technology, Engineering, and Mathematics) degree programs report having less access to laboratory experiences in high school, are encouraged towards science by their father's differently, and overall have a different preparation than their male counterparts [23, 19].

A student's demographics can include the student's gender, race, the financial support they can expect from their family, and if a student is the first in their family to attend college. Race has long been shown to be a factor in whether a student graduates or not. Black students are less likely to graduate than fellow white students when adjusted for socio-economic status and academic ability and they are more likely to have unwelcoming experiences in STEM programs while at university [24, 23, 1]. Female students are also likely to have unwelcoming experiences that cause them to switch from STEM programs [23]. However they have been show to be equally likely to graduate or more likely to graduation in comparison to their male counterparts [24, 23]. The financial support both in terms of loans and scholarships and the socio-economic status of a student's family have long been known to be a factor in university graduation with students who have more financial support typically being more likely to graduate [24, 25, 26, 27, 1]. First generation students are less likely to participate in extra-curricular activities at university [28, 29] and are more likely to dropout even when adjusting for race, family income, gender, and preparation [9]. Ultimately, American students from different backgrounds often see different success rates at university due to the different experience they have at the university due to their race, gender, or socio-economic status.

2.2 Gradient boosting

In this paper we use a novel method known as "gradient boosting". Gradient boosting can solve a large number of statistical modeling problems including logistic problems as found in this paper. Logistic problems are a group of statistical problems that assume that a sigmoidal function made from parameters θ and data $\boldsymbol{x}$ (i.e., $[1 + e^{-\theta^T \boldsymbol{x}}]^{-1}$) approximates the probability $P(Y|X)$. In plain language, the model uses input data X to determine how likely the outcome, Y, is to be true. In the context of this paper that would mean the input data X is used to predict whether a student will graduate in the following semester. In education research, the maximum likelihood method has commonly been used to find the solutions to logistic problems. Maximum likelihood solves the logistic regression problem by picking parameters θ that maximize the log likelihood function [30].

Since there is no closed form solution to this likelihood equation the solution is found iteratively via some optimization algorithm [30]. Maximum likelihood is the typical default solution for logistic problems in most statistical packages such as statsmodels [31] in python and glm in R [32].

Gradient boosting produces the solution to the logistic problem in a different way. The log-likelihood is maximized (or, in machine learning terms, the negative log-likelihood is minimized) in small steps: at each step, called boosting iteration, the gradient of the log-likelihood is computed to identify the best direction for the maximization, and it is then fed to a base learner, through which the model is updated. Different choices of base learners (in this paper we will use trees) provide different solutions, guaranteeing maximum flexibility [33]. An early stop, through a tuning parameter which controls the number of iterations, prevents overfitting and provide a better bias-variance trade-off [34]. If the base learner is weak enough in comparison to the signal-to-noise, it can be shown that, at least for a continuous response, boosted models outperform their unboosted versions in term of MSE [35]. Several modifications of the boosting algorithm have been proposed, including stochastic gradient boosting, that uses column and row subsampling to further avoid overfitting and even better deal with the bias-variance trade-off issue [36].

Gradient boosting has been demonstrated to produce better fit models in comparison to traditional methods across a number of domains both in binary classification and in hazard modeling [37, 38, 39]. Boosted logistic models have been demonstrated to be more effective at fitting data than models that use maximum likelihood estimation [40]. The structure of educational data can be highly complex often containing many sub groups that have different effects [41]. It is likely that gradient boosting can provide better parameter estimation for education research questions as well.

In this paper we use the Extreme Gradient Boosting (xgboost) algorithm [10]. Xgboost implements stochastic gradient boosting [42] with column and row-wise subsampling, regularization, decision tree base learners with a custom tree split finding algorithm, and an in-model data imputation system for missing data. The specifics of xgboost are explained in Section 4.

3 Data Set

The data in this study comes from registrar information from a large enrollment American research university [21]. It includes timestamped course grades, demographics, majors declared, degrees awarded, and preparation information for 160,933 students for the years 1992 to 2012. Student's are included in the study if they remain enrolled for at least 5 semesters. The 5 semester mark is chosen because a large number of students(18.3%) remain enrolled for only four or less semesters taking core courses [1]. These students do not graduate the university and likely transfer elsewhere as their institutional commitment may be low [4]. 88.02% students who remain enrolled for at least 5 semesters graduate from this university.

In this study, data is organized by semester into of four categories: demographics, preparation, enrollment, and performance (Fig. 1). Demographics and preparation are considered "static" features. That is, they are not changing over the course of the study. The enrollment and performance features are considered "dynamic". That is, they are cumulative and values can change as students progress in their studies.

The demographics data includes gender, race, cohort year, and the median family income for the zipcode the high school the student attended. The gender category is binary: male or female. The race category uses the IPEDs definitions [43] which is then reduced to a binary category of white/asian or other. This is due to two reasons: 1) the university has a primarily white/asian population, and 2) in this paper we establish model comparison's via out-of-sample prediction accuracy. Thus due to the small minority population, out-of-sample, per semester data may not include all racial groups with sufficient numbers (>100, [44]). The cohort year is the year the student begins their first courses. The median income comes from the 2011 American Community Survey 5-Year Estimate [45]. If the student has a reported high school GPA, then we also know the zip code of the high school the student attended. This zipcode is matched to the 2011 census data zip codes. The census data includes the median incomes for families living in that zipcode. The log of the median income is then recorded per student. If the zip code data is missing the median income is imputed as described in Section 4. This method is not an attempt to provide a course grained estimate of the socio-economic status of the student. Instead, it is a method to obtain a measure of the socio-economic status of the high school the student attended. Student learning can be impacted not only by a student's personal socio-economic status, but also the socio-economic status of the learning community they belong to [46]. We do not have individual level socio-economic data such as parental income. Nor do we have data on financial aid status.

The preparation data includes the per student reported high school GPA, a math placement score, and if amount of Advanced Placement credits if any. The math placement score comes from a 30 item test that the university requires students to take upon admission. This test has been used for the entirety of the study. A higher score will place the student in a higher math course up to calculus 1. High school GPA is also reported for each student. In both the cases of high school GPA and math placement score the data can be missing for several reasons. High school GPA is not required to be reported by applicants to this university, thus high school GPA is not always recorded for each student. If students have transfer courses for higher math courses (e.g., calculus 1) then the student does not need to take a placement test to determine if they can begin in calculus 1. In the case of missing data the math placement score and the high school GPA are imputed as described in Section 4.

The enrollment features are organized per semester. They include whether a student changed their major in that semester, the number of currently enrolled majors, the ratio of credit hours to a full time load (12 credit hours), the total registered semester credit hours, the total cumulative credit hours accrued, the non-major credits the student has registered for in this semester, the major credits hours enrolled per semester, whether the student was enrolled in the previous semester, and the cumulative number of skipped semesters.

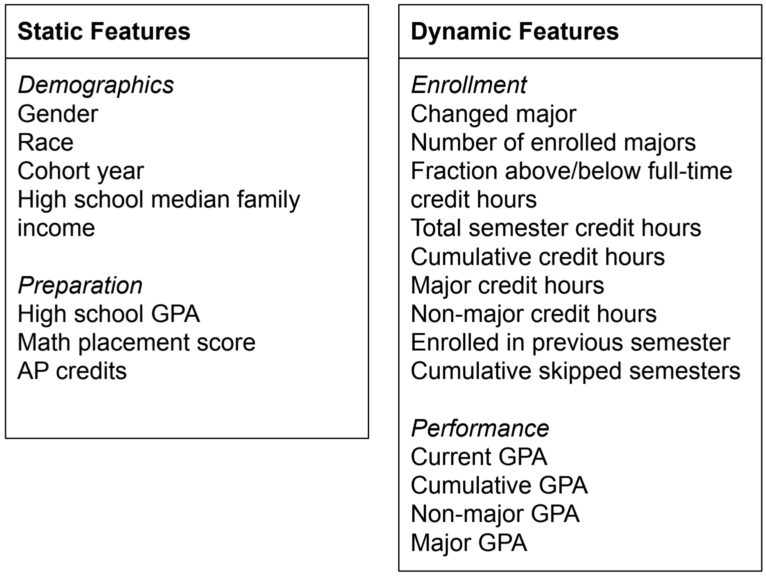

Fig 1. Data model for all models presented in this paper.

The performance features are organized per semester. They include the current semester GPA, the cumulative GPA, the current semester's GPA for courses outside of the student's current major, and the current semester's GPA for courses within the student's current major.

4 Methods

In this paper we have used a discrete time hazard modeling framework for all models that are presented. We begin by comparing the logistic regression model to a gradient boosted (xgboost) [10] model to produce the most predictive model. In this section we will describe the discrete time hazard modeling framework, the logistic regression equation, the xgboost model equation, and the statistics and evaluation methods that we have used to determine the xgboost model is the best predictive model. We will then describe the methods we used to evaluate what features the model uses to produce predictions. The superior xgboost model is then used to describe what factors are most predictive of students graduating in a particular semester.

4.1 Discrete Time Hazard Models

Discrete time hazard models are, essentially, logistic regression (or other classification) models calculated per time step for time changing data. They control for time to event predictions when the times to events are simultaneous or explicitly discrete and there is new data being collected over the course of the study period [11]. Traditional hazard analysis such as Cox regression is not capable of doing this type of analysis due to the duration being measured being a common value amongst many students [11]. This is due to the likelihood function of a continuous event duration model assumes independent and unique durations (i.e., the time it takes to graduate). When these durations are not unique, they can lead to overfitting. In this study the time unit used is a semester [1]. Thus the models attempt to predict whether a student will graduate in the immediate following semester. If a student drops out from the university, they

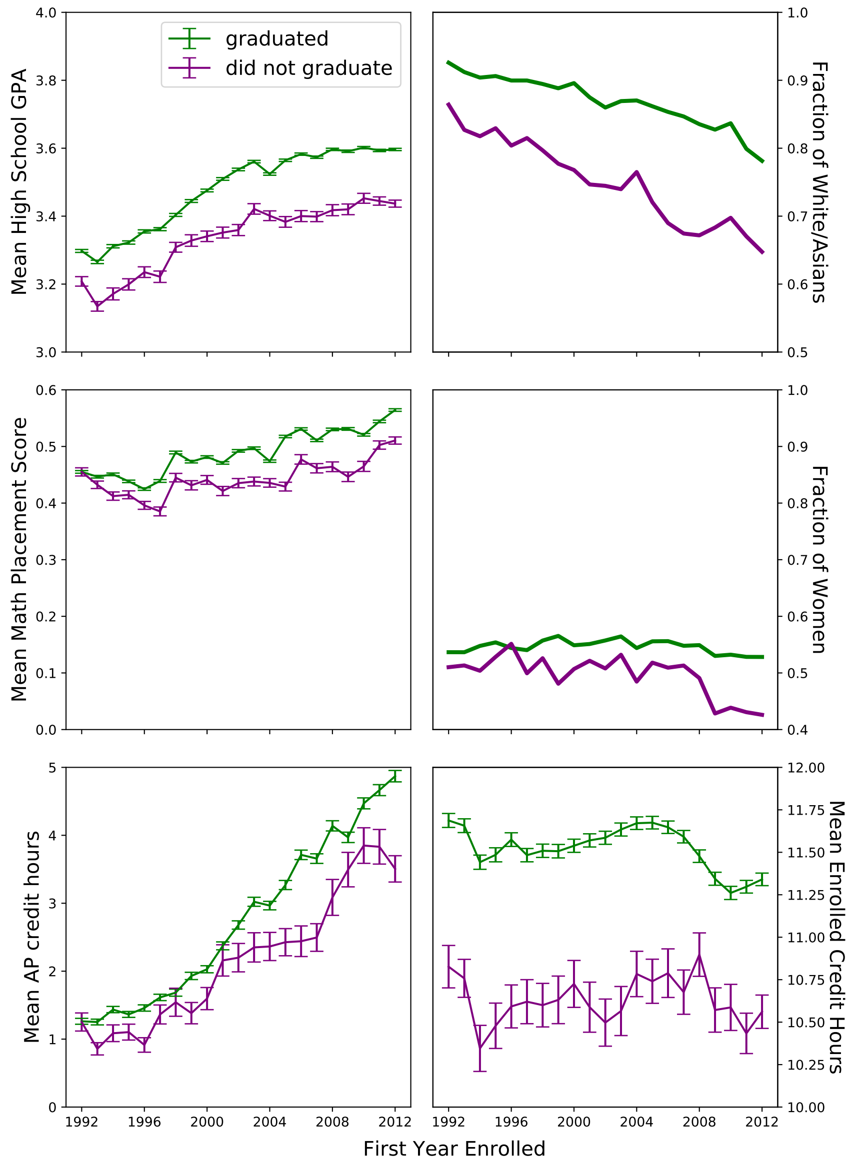

Fig 2. A comparison of a subset of features for students who graduate versus those who do not based on first year enrolled. Since 1992 the student population has become more diverse racially, students have had increasing high school GPAs and math placement scores, and the time it takes students who graduate to graduate has been decreasing. Students who graduate typically take more credit hours than those who don't, typically are better prepared as measured by high school GPA and math placement score, and are less diverse than the total university population.

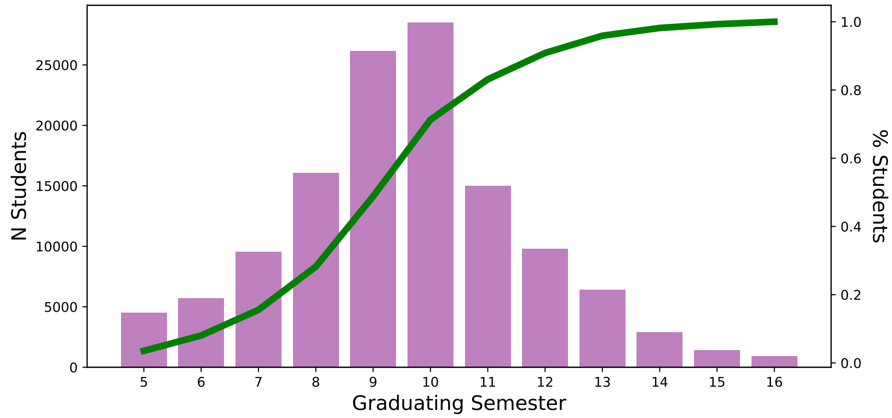

Fig 3. Students typically graduate within the window of 8 to 12 semesters. The bars represent the number of students across all cohort years who graduate in the following semester. The green line represents the cumulative fraction of students who have graduated.

are not included in following semester data set.

4.2 Logistic Regression

The logistic regression equation used in this paper is as follows:

$$\log \frac{y(t)}{1-y(t)} = \beta_0(t) + \boldsymbol{\beta}_S^T \boldsymbol{X}_S + \boldsymbol{\beta}_D(t)^T \boldsymbol{X}_D(t) \tag{1}$$

Where $y(t)$ is the likelihood to graduate in the following semester and $\boldsymbol{\beta}_S$ and $\boldsymbol{X}_S$ are static features indicated by subtext S (Fig. 1). The dynamic terms (indicated by subtext D) are calculated per semester t from semester 5 until semester 16 [1]. ϵ is then the irreducible random error. This is effectively an iteratively calculated multinomial logistic regression model [1]. The model is fit using the maximum likelihood estimation method. No regularization is used.

The logistic regression model is built using the statsmodels library in python [31].

4.3 Gradient Boosted Trees

Xgboost is an implementation of a stochastic gradient boosting machine [47, 42, 10] that can also be used to attempt to solve the logistic equation. Gradient boosted machines can be seen as models that are iteratively fit on the current residuals starting from the null model. The additive collection of these models produce the output. Xgboost uses decision trees as its base learners.

The gradient boosted model is thus:

$$\log \frac{y(t)}{1-y(t)} = F_0 + \sum_m^M h_m(\boldsymbol{X}(t)) F_m(\boldsymbol{X}(t)) \tag{2}$$

Where m is the iteration index, $h_m(X)$ is the previous iterations residual model and $F_m(X)$ is the current iteration's model fit to previous iteration's residuals. The total number of iterations is set by M.

A gradient boosted machine can use any learner for the iterative procedure. In this paper we use the decision tree learner as implemented in Xgboost [10]. Decision trees are models that use a tree like structure to fit data and produce regressions and classifications. For each "leaf" node in a tree, a specific decision is made that separates data into two diverging paths. Each node represents a single variable. Categorical variables (e.g., gender) are simply split by the category. Continuous variables (e.g., semester GPA) are split by a decision boundary (e.g., GPA > 3.0). This boundary is typically determined through iteratively fitting the model to find the best boundary.

In Xgboost, each decision tree is trained from a randomly sampled set of rows and columns. Each tree is grown up to a maximum depth using a leaf growing algorithm that estimates whether an additional leaf will produce a better or worse tree [10]. Thus there is no separation between static and dynamic features as in the logistic regression model for each semester. All xgboost models are built using the xgboost library in python [10].

Xgboost has a built in algorithm to deal with missing data [10]. When missing, data is imputed for each tree by selecting the most common decision path for a node. Because a feature may appear in many trees in the model, the decision boundary can be very different per tree in comparison to an mean imputation scheme.

Xgboost models have hyperparameters that define how the model is to be constructed prior to the model being trained. These hyperparameters govern two classes of model design: 1) how the boosting functions, and 2) how the trees are grown and structured. In the case of boosting, the most common hyperparameter used is the number of total boosting learners. For the tree structure this includes the maximum depth a tree can grow to and the fraction of variable and row sub-selection. Typically these hyperparameters are determined by grid search when the total combination of hyperparameters is below 1000 combinations [48, 49].

Because the learners in the xgboost model are decision trees and not linear models, the xgboost model does not report typical coefficient values like in a traditional logistic regression. Instead they report feature "importances" known as "gain". Each time a variable is used in a tree, the tree is built optimally by splitting in the optimum location. The increase in accuracy due to this split is the gain. The feature importance for a specific variable reported then is the average gain across all the instances that the variable is used in in the model. Xgboost uses a custom split finding algorithm that compares the utility of growing a tree to its maximum depth based on increase in accuracy [10].

4.4 Model Evaluation

The goal of this study is to produce a predictive model of when student's graduate. We use a number of techniques to verify the veracity of the model and increase its accuracy (Fig. 4). These include: splitting the data into a training/testing sets, imputing missing data, and picking custom thresholds for the predicted probability of graduation. We also limit the over-estimation of a variable's impact on when a student graduates by weighting explanation methods with the F_1-score [30]. The following section will explain these techniques in detail.

The data processing and model evaluation follows the procedure shown in the flowchart in Fig. 4. Data for the discrete time hazard model models are:

1. queried from the pathways database [21]
2. Continuous values such as high school GPA and semester GPA are rescaled using the z-score. The means and standard deviations are determined by either the starting year cohort (high school GPA) or the timestamp for the enrolled semester (semester GPA). The median income is scaled using the logarithm.
3. Data is split into training (50%) and testing (50%) sets. Since the discrete time hazard models are evaluated per semester data is split by student and not by

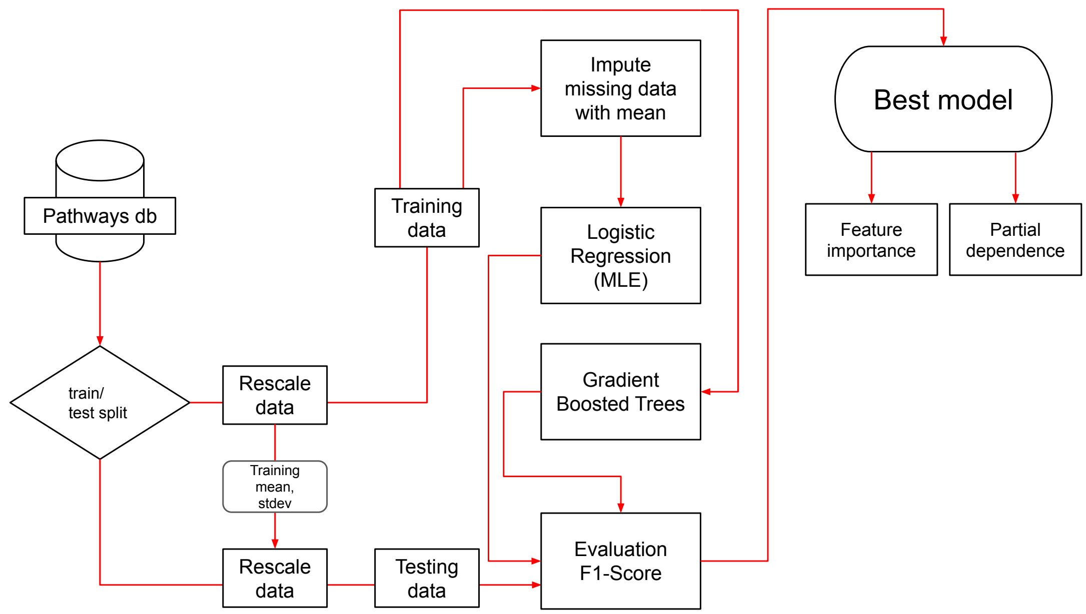

Fig 4. Model evaluation flowchart. Data is split into testing and training sets and is evaluated for two separate models: logistic regression solved by maximum likelihood and gradient boosted trees (xgboost). Missing data for the xgboost model is not imputed from the mean and instead uses the built in imputation engine within xgboost [10].

semester. Thus a student in training data in semester 5 is also in training data for subsequent semesters.

4. Missing data that is input into the logistic regression model is imputed from the means calculated during the rescaling step. Missing data for the xgboost model is not imputed prior to fitting the model since the xgboost algorithm has an imputation engine built in.

5. Each model is trained and evaluated using the F_1 score. In-sample F_1 scores are used to determine the best threshold for splitting the predicted probability distributions for classification (i.e., does the student gradute in the next semester).

6. Models using the selected threshold are then evaluated using out-of-sample test data to compute the F_1 score.

Several data handling procedures were used to prevent model overfitting, data leakage, and poor predictive performance due to unbalanced class issues. To evaluate the predictive capability of each model, data in this paper is separated into training and test sets for each semester that the discrete time hazard model model is applied. Separating data into partitioned training and testing sets is important to prevent a too optimistic evaluation of the model error[30]. Without having hold out data to evaluate the model performance, we have no way of knowing how a model will predict the outcome of data it has not seen before [30, 50, 44]. Using hold-out data is atypical in recommendations within education research for assessing predictive ability [51, 52] and is not used typically in modern papers using discrete time hazard analysis (e.g., [1]).

Data leakage could occur between two semesters if, for example, a student in semester 6 is in a training set and the same student is in a test set in semester 7 [44].

This is due to some of the features being cumulative thus they would carry forward information from the previous training period into the next test period. Thus, students are identified prior to model training as belonging to the training or testing data set. Students in the test data set are only given to the model to evaluate model predictive performance.

Students do not always have an entry in the database for their high school GPA or their math placement score. This can be due to a variety of reasons such as the high school GPA was not reported, the reported high school GPA was self reported and potentially untrustworthy, and the math placement score was not given to the student because they transferred math credits from another institution. To prevent errors associated with removing data [53], this data is imputed in one of two ways. For the logistic model data is imputed from the mean for the student's cohort year. For the xgboost model data is imputed within the model. Xgboost will actively impute missing data within each tree learner based on the most likely branch for each decision node (see Section 4 and [10] for more details).

The predictive ability of models in this paper is estimated using the F_1-score and the recall [54]. The F_1-score is the harmonic mean of the precision and recall and is calculated with the following equation:

$$F_1\text{-score} = 2\frac{\mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}} \tag{3}$$

$$Pr = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{4}$$

$$Re = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{5}$$

The F_1 score is over the range 0 to 1 with 1 being perfect performance and 0 being random performance. Precision is defined as the ratio of true positives to the sum of true positives and false positives. That is, it is the ratio of true predicted graduations per semester to the sum of the true predicted graduations and those predicted to graduate but actually do not. Precision is over the range 0 to 1 with 1 being perfect performance and 0 being random performance. Recall is defined as the ratio of true positives to the sum of true positives and false negatives. That is, it is the ration of the number of students predicted to graduate to the total number of people who actually graduated per semester. Recall is over the range 0 to 1 with 1 being perfect performance and 0 being random performance.

We use the F_1-score over other statistics (such as the area under the reciever operating curve (AUC), [30]) because it balances the trade off between high precision (which includes falsely labeling students as graduated) and high recall (which includes falsely labeling students as not graduating). We use the Recall score in the cases of understanding model performance for sub-groups (such as under-represented minorities) because it gives a more direct interpretation of how accurately the model selects the graduating case.

Students are enrolled for many semesters however they only graduate in a single semester. Thus the majority class in each semester is that a student will *not* graduate in the following semester. This unbalanced class issue can lead to under-fitting of the model specifically such that models simply predict the majority class (in this case not graduating) far too often [55]. We attempted to use an over-sampling technique [56] to address this issue however this did not increase model performance. Instead we choose custom thresholds for the model probability distributions for predicting when a student graduated. The custom thresholds are picked using the F_1 scores calculated from the training data [57]. The best threshold is calculated via a grid-search between a threshold of 0 to 1 using increments of 0.01. This still, typically, does not increase F_1 scores.

Beyond predicting when a student will graduate we are interested in the factors that explain why a student was predicted to graduate (or not graduate). A typical logistic regression model using maximum likelihood produces coefficients that represent the magnitude and direction a particular feature given the other variables. For gradient boosted trees these values are calculated differently. Xgboost uses instead the average gain in prediction accuracy across all instances a feature is used in each tree learner. This is commonly called the "feature importance". Additionally we can estimate how much a feature, depending on individual values, contributes to the predicted probability of graduating. In this case we use a method called "partial dependence" that was originally developed to use with gradient boosted models [47]. Both the xgboost model and the logistic regression model have a different level of confidence in the prediction per semester (Fig. 5). Due to this variable confidence our confidence in feature importances and partial dependences varies as well. In order to account for this confidence we have weighted the feature importances and partial dependences with the out-of-sample F_1 score. This is explained below.

We estimate the effects of model features on prediction using the using the weighted mean of the feature importances. For all 11 semesters the feature importance is weighted by the overall F_1-score for the semester following this equation:

$$\boldsymbol{\beta}_{weighted} = \sum_{i=0}^{N=11} F_1^T \boldsymbol{\beta}_i \tag{6}$$

This then allows for the cumulative weighted features in Fig. 7 and the weighted features in Fig. 8.

We also estimate the effects of model features on prediction using the partial dependence [47]. A partial dependence plot represents the average model output for a single variable (S) across the entire feature space (C) in the context of all other features [47]. This means that for a specific feature S, the exact values of S are first, fixed for all rows in the data set for each unique value of S. Then the model is calculated per value. The average contribution of a specific unique value to the overall predicted probability of graduation is then considered the partial dependence. Partial dependence is then assumed to be a continuous function over the entire feature space of S. Partial dependence is estimated using the following equation as:

$$\hat{f}_{x_S}(x_s) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(x_S, x_C^i) \tag{7}$$

The partial dependence is calculated for each semester and is weighted by the F_1-score for that semester (Fig. 9, 10) in a similar fashion to equation 6.

Partial dependence allows the researcher to see the direct contribution to the predicted probability of a given variable value. Because variables often have many values, they can give fluctuating contributions to the predicted probability. In linear models, it is assumed that variables produce linear contributions to the partial dependence [47, 30] and thus coefficients of a linear model represent unit increases [30, 58]. For nonlinear models such as gradient boosted trees, this assumption is relaxed and variables can produce nonlinear contributions to the predicted probability.

5 Results

In this study we have two broad research questions: 1) exploring the effectiveness of a gradient boosted logistic model in comparison to traditionally solved logistic models, and 2) quantifying the components of Tinto's Theory of Drop Out in the context of who does or does not graduate at this university during the study period. In this section we will first describe the effectiveness of the gradient boosted model in comparison to the traditional maximum likelihood model. Then we will describe the results that are drawn from the gradient boosted model in the context of Tinto's theory.

5.1 Gradient boosting

The models in this study attempt to predict whether a student will graduate or not during the observation period of being enrolled for 5-16 semesters. The xgboost model is generally more effective than the logistic regression model except in the last semesters (15-16) studied as assessed by the out-of-sample F_1-score (Fig. 5). Xgboost is particularly more effective in highly imbalanced case of students graduating (approximately 5% per semester (Fig. 3)) within ≤8 semesters (Fig. 5). In the more balanced case (semesters 8-14 have approximately 20% of the students eligible to graduate do so (Fig. 3)), Xgboost still performs better than the logistic regression model (Fig. 5). In the final semesters (semesters 15-16) logistic regression outperforms xgboost slightly.

Xgboost is more effective at correctly predicting the graduating semester of female students as assessed by the out-of-sample recall for all semesters except for semester 16 (Fig. 6). Xgboost is also more effective at correctly predicting the graduating semester of under represented minority students for all semesters (Fig. 6). Additionally, Xgboost is able to handle missing data cases better than the logistic regression model for all semesters (Fig. 6). Due to the effectiveness of xgboost over logistic regression, the remaining results will focus on the xgboost model.

5.2 Effects on Time-to-Graduation

Tinto's Theory of Drop Out posits two overarching quantities as governing student choice to drop out or continue to graduation: 1) educational goal commitment, and 2) institutional commitment. Students initially start university with these commitments. These commitments are then mediated by a student's participating and integration into the academic system and the social system of a given institution. Generally speaking (Fig. 2), graduating students take more credit hours per semester ($N(grad) = 15.1 \pm 0.01$, $N(no-grad) = 13.17 \pm 0.05$), enroll for more semesters ($N(grad) = 10.57 \pm 0.01$, $N(no-grad) = 8.79 \pm 0.05$), are more likely to be white or asian ($N(grad) = 86.7\%$, $N(no-grad) = 75.3\%$), and more likely to be female ($N(grad) = 54.2\%$, $N(no-grad) = 48.9\%$).

In this study, we characterize student's initial commitments as being mediated via the following quantities: high school GPA, math placement score, the number of AP credit hours the student possesses, the median income of the high school the student attended, gender, and race. Preparation is about half as important in predicting when a student graduates in comparison to enrollment factors (Fig. 7). For early graduating students (≤8 semesters), both math placement scores and high school GPAs were important in predicting that a student will graduate (Fig. 8). While students are likely to have more AP credits on average if they graduate in 8-10 semesters ($AP_{8-10} = 4.06 \pm 0.03, AP_{other} = 2.26 \pm 0.01$), having AP credits was not an important indicator for predicting graduation during any semester in comparison to other features (Fig. 7, 8). While white and asian students and female students are more likely to graduate (Fig., 2), demographics are not major indicators for predicting when a student graduates for any semester (Fig. 7, 8). This is also true for the median incomes of the zipcode the students went to high school.

In this study academic integration is represented by a combination of features including the cumulative credit hours, cumulative mean grade; and the per-semester

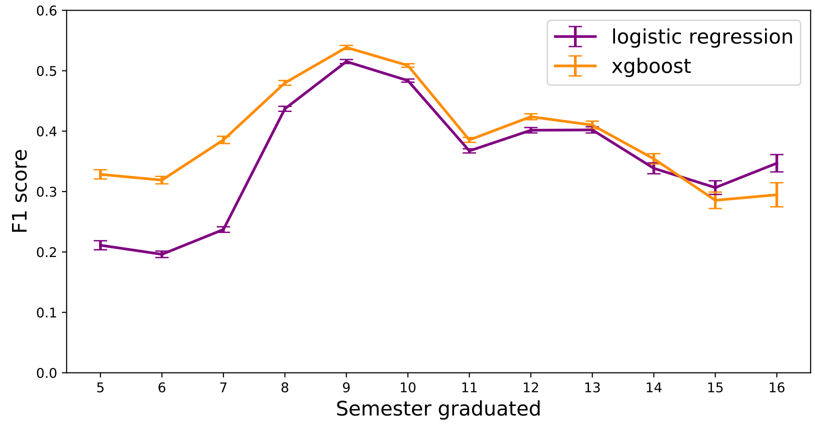

Fig 5. Out-of-sample F_1 scores for all models per semester. Xgboost clearly increases the precision and recall tradeoff for the bulk of semesters. Error bars are the bootstrapped standard deviation of the F_1 score.

fraction above or below full-time credit hour, mean grade, non-major GPA. A student's enrollment and performance are the most predictive features when predicting the semester when a student graduates (Fig. 7, 8). A student's cumulative credit hours is most important to predicting whether a student graduates "on-time" (after 8 semesters) or not and is overall important for prediction. The cumulative credit hours is less important for prediction the further a student's credits are away from 120 total credit hours (Fig. 9). A student's average grade is most important for predicting that students will graduate within 10 semesters and is overall important for predicting when a student will graduate (Fig. 7, 8). Having above average grades is important for predicting students who graduate in 9-10 semesters however this is less important for other semesters (Fig. 10).

In this study we use per semester major GPA, major credit hours, cumulative number of majors, and whether a student changed major in that semester or not to represent their social integration into the departments and learning communities associated with their chosen degree program. Student's performance within their major was not as an important indicator for predicting graduation in comparison to overall grade and cumulative credit hours (Fig. 7). Changing a major has increased importance for predicting students who graduate later (Fig. 8).

6 Discussion

This paper presents two complimentary results: 1) gradient boosting is a useful tool for predicting when students graduate in comparison to traditional statistical algorithms, 2) students who actively integrate into their academic and social communities is the primary effect that predicts when a student graduates thus following Tinto's Theory. This section will discuss why the xgboost model outperforms logistic regression, the implications that come with the results of the model, and compare it to other studies that have predicted time-to-graduation using Tinto's Theory of Drop Out and hazard modeling more broadly.

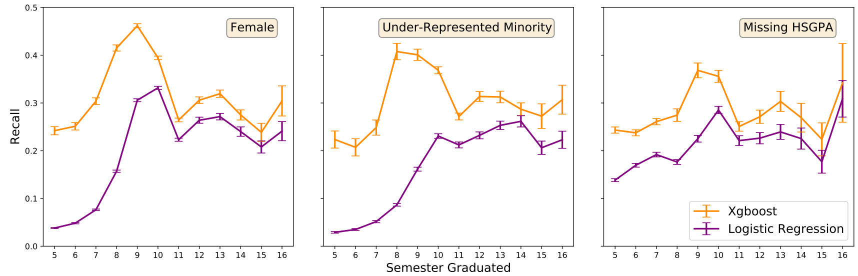

Fig 6. Recall scores for sub-populations within the data. The Xgboost model consistently labels women, under represented minorities, and students with missing data better than logistic regression. In Fig. 5, there is a large disparity between the logistic regression model and the xgboost model for semesters 5-7. This may be due to the imputation method used with the logistic regression model and the correlation between graduating and missing data during those semesters. However, the gaps in model performance for women, minorities, and missing data in later semesters with no correlation are likely not due to imputation and demonstrate a strong preference for using the xgboost model over the logistic regression model.

6.1 Why does gradient boosting produce better fitted models than maximum likelihood estimation?

In almost every case, the xgboost model predicts graduation better than the logistic regression model. This can be for several reasons. First, the xgboost model fitting procedure is a slower iterative procedure (1000 iterations) than the maximum likelihood estimation (typically 5-6 iterations) that the logistic regression model uses. Thus, in each iteration the xgboost model can focus on the local neighborhood of feature details within the training data that the logistic regression model may miss [59].

Second, the xgboost model uses a custom imputation engine [10]. Whenever data is missing and an learner is using a feature with missing data, the missing data is imputed to be the most likely choice in the decision tree. Because there are many learners, they can account for different local patterns in the data. The data for the logistic regression model is imputed from the mean of the student's starting year cohort. This mean imputation likely loses information that the xgboost model is able to attend to. It could be that a more sophisticated imputation model will increase the logistic regression model performance. Typically if data missingness correlates with the outcome variable, then data should be imputed to prevent bias due to the missing data [60]. In our case, data missingness for both high school GPA and math placement score correlates with early graduation rates (see supplemental Fig. 11). Thus the logistic regression model performance would likely increase for semester's 5-7 with a more sophisticated imputation model. However this would not explain the substantial improvements xgboost makes over the logistic model for women and under represented minorities in later semesters.

Third, xgboost penalizes leafs within the tree learners that are fit on few examples from the training data [10]. Additionally, xgboost weights on class labels as well. This is especially useful in the highly unbalanced case of predicting when a student graduates. Because the weight of training data from students who do not graduate is tuned via a grid search, this prevents the xgboost model from overfitting on the majority class simply due to having more representative samples.

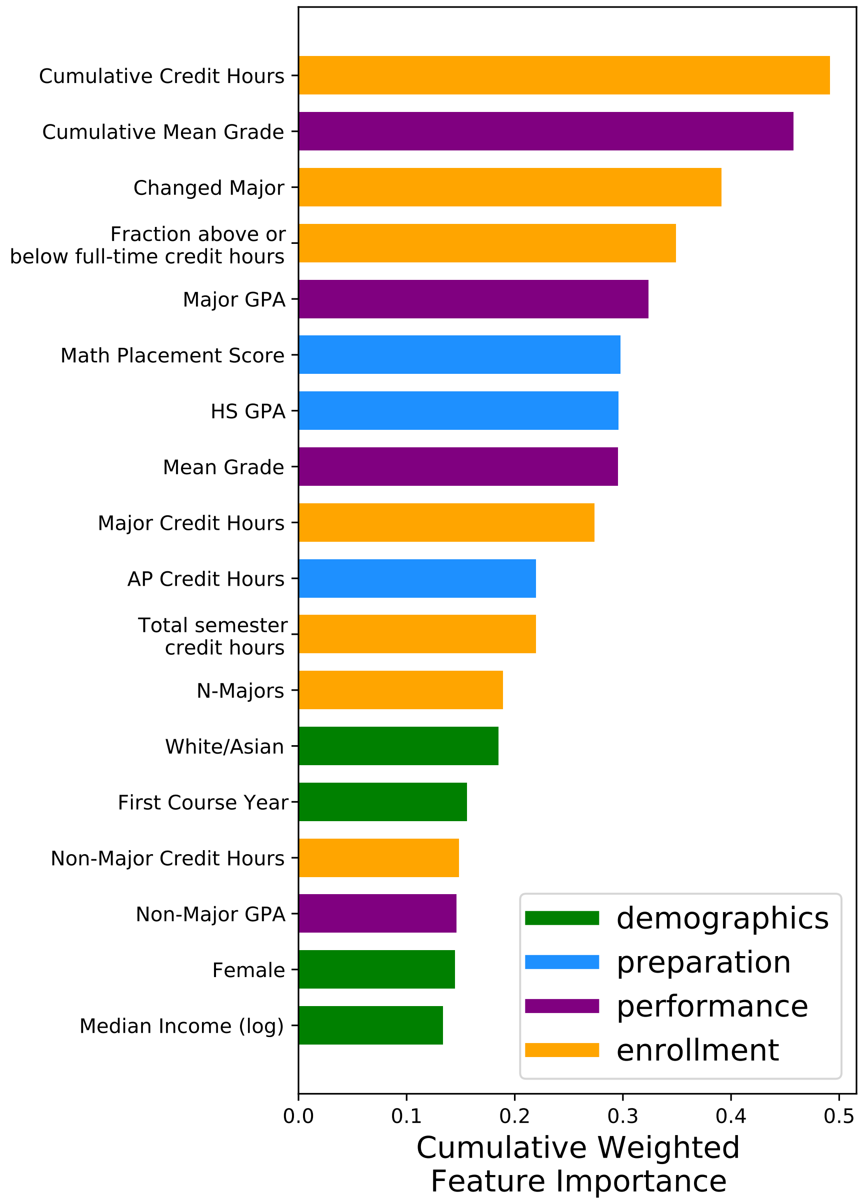

Fig 7. Average feature importance for all graduating semesters for the xgboost model. Feature importances are weighted by the F_1 scores per semester. Enrollment factors and the cumulative average grade are more likely to predict when a student graduates than other factors.

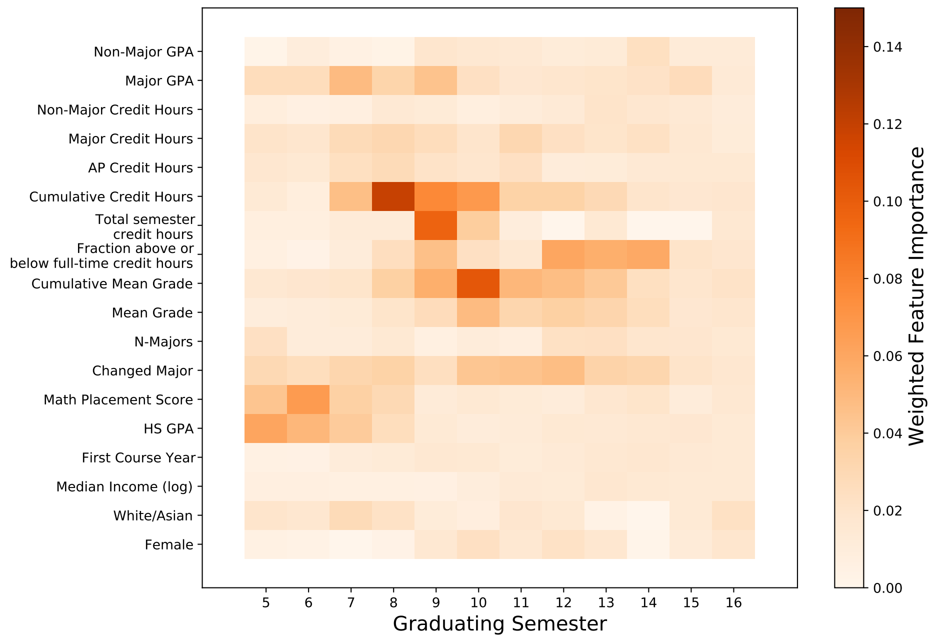

Fig 8. Xgboost feature importances for predicting if a student will graduate in the next semester. Feature importances are weighted by the F_1 score calculated from test data. The row-wise sum of these features would produce Fig. 7. By far the most important feature for graduating "on time" (within 8 semesters) is having enough credit hours. Student preparation is important for graduating "early" (<8 semesters). Students who change majors are more likely to graduate later and thus this feature becomes more important in later graduating semesters.

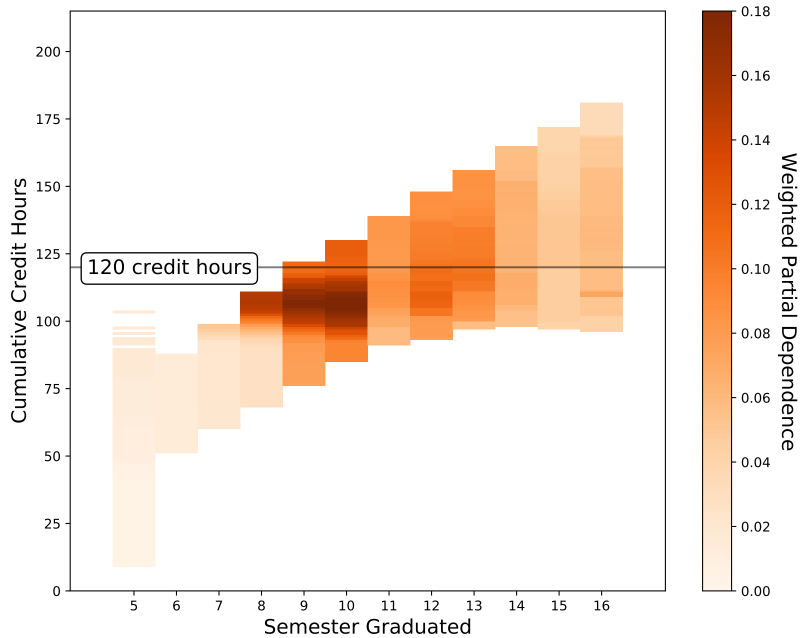

Fig 9. The partial dependence per semester for the cumulative credit hours a student has obtained. The stronger the partial dependence, the more contribution that value of cumulative credit hours has on the predicted probability that a student will graduate in the given semester. The partial dependence has been weighted by the per semester F_1 score. Having a total number of credit hours close to 120 is highly predictive of graduating if students graduate between 8 and 10 semesters of enrollment. Outside of this window the impact of the total number of credit hours diminishes. This is likely due to students having additional credit hours that do not count towards a degree such as when they change their major.

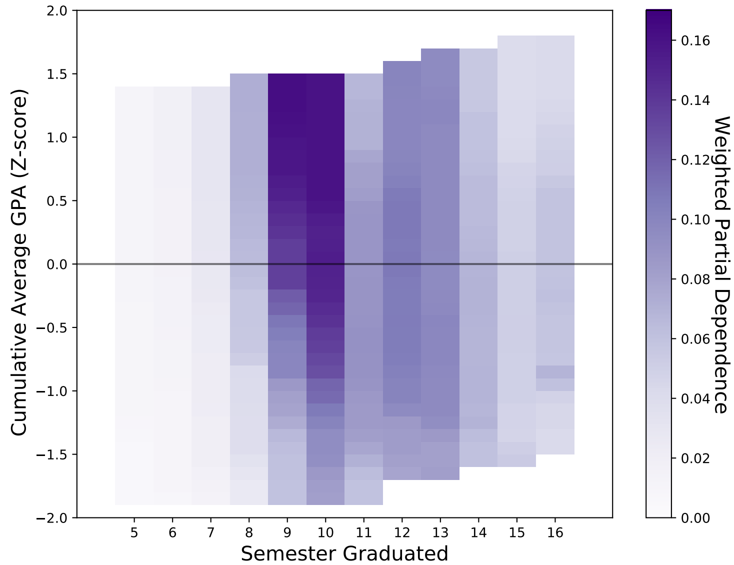

Fig 10. The partial dependence per semester for the cumulative average GPA a student has obtained. The stronger the partial dependence, the more contribution that the cumulative average GPA has on the predicted probability that a student will graduate in the given semester. The partial dependence has been weighted by the per semester F_1 score. Having a far above average GPA is a major contribution to graduating if a student graduates between semesters 9 and 10. This is likely due to these students never failing a course and having a high commitment to their chosen major.

6.2 Effects on Time-to-Graduation

Tinto's theory suggests that students have an initial level of intent to graduate from an institution upon entry [4, 2]. This intent is the combination of a students family background (e.g., socio-economic status), individual attributes (e.g., academic ability, race), and pre-college experiences (e.g., high school GPA). This intent is then tempered by the student's at-college experience such as social integration [61], financial support [9], and academic performance.

6.2.1 Effects on the initial conditions of educational commitment

Student's backgrounds can set up a wide variety of contributions to their initial educational and institutional commitments. In this study we assess these initial conditions using a student's high school GPA, math placement score, the number of AP credit hours the student possesses, the median income of the high school the student attended, gender, and race. This university uses a math placement test to determine a student's incoming math ability to place them in the appropriate math course. This test was designed at the university and has been used for the entire study period. This test was not designed with psychometrics in mind. Thus, while the test is representative of some measurement of math ability, it is unclear how much the test is representative of math ability. However, we see that a student's math placement score is the most important attribute that predicts when a student will graduate that is not a performance or enrollment variable 8). This could be similar to [13] evidence that the starting math course is very important to staying in STEM. In [13], students taking lower level math courses at college had increased likelihood of leaving the STEM major they were enrolled in for a different major. In our case, students who take remedial math courses simply have more courses to take and thus must remain at the university longer. Given that students in our study who come from higher income communities are able to remain at university longer before graduating, this indicates that remedial mathematics should be examined more in terms of the cost for a student. It should also be noted that in this study we are predicting all students time to graduation even those who are not pursuing quantitative degrees. The result that math ability has a dominant effect in the context of other demographic and preparation variables is consistent with the literature which claims that math ability has an sizeable effect on student performance at university [16].

A student's high school GPA is similarly important to a student's math ability in predicting when they graduate (Fig. 8,7). This is consistent with some literature that finds that students with higher high school GPAs are more likely to graduate within 4 years [9]. [8] found in a similar study that high school GPA had little to no effect on predicting when a student drops out from university. In both this paper and [8] the study uses data from university's that serve primarily students who come from the state the university resides in. Thus this may indicate that the effect of high school GPA on college success is geographically dependent on the quality of high school preparation for university.

Perhaps counter-intuitively having AP credits is not as important as a student's high school GPA or math ability (Fig. 7). AP credits directly count for college credits. Students with more AP credit have fewer credit hours necessary to graduate. In this study, having AP credits is an indicator that a student will graduate in four years (8-9 semesters). However it is not a strong indicator that a student will graduate early in comparison to a student's math placement score and high school GPA. [8] found that having transfer credits had no effect on on students dropping out. This was also true in a study of physics students who change their majors [21]. It could be that AP credits are too random in whether they count for credit or not. A student may choose to take a course anyways they have AP credit for if its in a sequence (e.g., introductory physics) because they may feel ill prepared for the second semester course. It could also be that some students who take AP courses do so with the intent to take a minor or dual major. In this case, the AP courses "free" up more time at university to be able to graduate in four years.

Financial aid is one of the top reasons students leave the university without a degree [9]. In this study we do not know if a student has access to financial aid or not. However, we do include the median income for families that live in the zip code of the high school the student attended according to the 2011 census American Community Survey 5- year estimate [45]. This is a rough estimate of the socio-economic status of the high school that the student attends. A course grained measurement like this does not capture all of the nuances of individual students financial support and in our case, we see that this feature was less important in comparison to performance and enrollment features. In our study, we find that coming from higher income high school's is an indicator that a student will take longer to graduate. [46] found that the socio-economic status of a student's peers had an approximately equal effect as a student's individual socio-economic status on high school student exit examination scores. While this effect is small, it may indicate that students from higher income regions have access to more resources and thus can spend more time in college before entering the work force. This is a similar result to [9] which found that students who come from low income households are likely to graduate within 4 years in comparison to students from higher income households.

6.2.2 Dynamic contributions to a student's academic integration

A student's academic integration is defined by Tinto [4] as being a combination of performance in university and their intellectual development. In this study we characterize performance through both cumulative GPA measurements and per semester GPA measurements. Further, we split this into major and non-major GPAs.

Performance as measured by GPA has had a demonstrated primary effect on graduating [8, 14, 1]. [8] found that there was a decaying impact on drop out due to GPA. The longer a student was enrolled, the less their GPA was likely to be a strongly influencing factor on dropping out. In our study we find somewhat different results, namely students with above average cumulative GPAs are more likely to graduate on time (Fig. 10). However this effect diminishes with time. It could be that this peak effect is due to two reasons: 1) high performing students are more likely to graduate on time [1], and 2) failing a single course significantly sets back both the time to graduate and a student's cumulative GPA.

There is an interplay between the cumulative credit hours and the cumulative mean grade. In semester 8, the single most predictive feature is cumulative credit hours (Fig. 8). However by semester 10, cumulative credit hours has exchanged the highest rank with the cumulative mean grade. This has a couple implications. First, the decaying impact of grades as noted by [8] in this case begins later at semester 10. Second, it may be that to graduate within 8 semesters (4 years), there are very few paths other than a strictly laid out course schedule with no deviations and no failing grades. Whereas graduating within 5 years allows more leeway for students to take additional courses that could be due to, for example, receiving a minor.

Second, there is a transition across graduating semesters of what is important (Fig. 8). Early graduation is predicated more by a student's initial conditions than anything else. By semester 8, the overwhelming effect on successfully predicting students graduating in this semester is their cumulative credit hours. Past semester 8, there is a strong combination of features that are predictive of graduating. This suggests that while Tinto's theory indicates that there is a dynamic contribution of on-campus interactions to student educational and institutional commitments, these dynamic contributions may not matter that much for students with very strong backgrounds who wish to graduate early.

6.2.3 Dynamic contributions to a student's social integration

A student's social integration is defined by Tinto [4] as being a combination of peer group interactions and faculty interactions. In this study we use a course grained measurement of peer group and faculty interactions by measuring the total per semester credit hours a student registers for in their major. Additionally, we note when a student changes their major. Changing a major is a distinct situation where a student will leave their peer and faculty group for a new group and thus may indicate low social integration.

In this study students who attend the university and take a recommended load of courses and pass each course were likely to graduate in 8-10 semesters. When students altered from this path (e.g., a student changes their major) they increased the amount of time it took to graduate or did not graduate during the study period. [13] provides, at least for STEM students, a complementary explanation as to why a student may graduate later. This university has a very large enrollment (typically >50000 students), a strong research program, and a strong greek life. In each case these may contribute to social integrations given there is more opportunity at this university than some others for meeting new people, participating in research, or participating in social events.

While not being the largest effect, students who take higher amounts of major credit hours were more likely to graduate in semesters 8 and 9 (Fig. 8). This is sensible given's Tinto's theory since taking more major credit hours both works toward's graduation and is an indication of a strong integration into a peer and faculty group. Within the literature, STEM students who took fewer STEM courses in the first year, were enrolled in less challenging math courses in the first year, and performed poorly in their STEM courses in comparison to non-STEM courses were highly likely to switch majors. Tinto would describe these students as having a reduced educational commitment due to not integrating into the social community and/or lack in the academic engagement of their chosen degree program. In our study we find similar conclusions, higher performance and the number of major credit hours is a more likely indicator of graduating in four years and this effect diminishes over time after the 4 year (8-9 semesters) mark (Fig. 8). Thus, it may be likely that performance in a major and the frequency of major courses taken may be an indicator of lower social and academic engagement described by Tinto [4].

In this study we have highlighted that changing a major can have a profound effect on the time it takes to graduate. Changing a major impacts a student's time-to-graduation and is predictive of students who graduate later (Fig. 8). A student who changes their major could do so due to low academic integration [13]. However this low academic integration could represent low social integration as well [23]. Student's who perform poorly may ask themselves if they "belong" in a major. In many cases STEM students who demonstrate high academic performance change their major due to reasons associated with social and personal interactions at university [23, 13]. This especially affects women and under represented minorities [23]. In this study changing a major does not show a strong correlation with a student's race or gender ($\rho_{race} = -0.07, \rho_{gender} = -0.03$). In many cases, the experiences that lead to changing a major are more nuanced than a single variable can contain thus while race and gender are a strong component to major change as reported in qualitative interviews [23], this may not be captured in binary variables. Ultimately, changing majors has been a poorly investigated and deserves more investigation. Future work will look at using Tinto's theory as a framework for investigating student's changing their major.

6.2.4 Limitations on this Study

Given that a perfect F_1-score is 1 and in this paper we report F_1-scores around 0.5 at the maximum, it is likely that some of these effects such as financial aid or social integration have large effects on when a student graduates. It is common in some studies, (e.g., [1]) to use individual heterogeneity models or "frailty" models to assess unmeasured contributions to graduation [62]. In our case, we attempted initially to use gamma distributed frailty terms [11, 63]. These proved to provide no increase in model performance thus were removed from subsequent analysis. Future work will consider what contributes to the unobserved heterogeneity (e.g., direct measurements of peer social engagement such as social network centrality).

Additionally, this paper has focused on a student's educational commitment (per [4]) as opposed to their institutional commitment. Much of this is mediated by the fact that 88% of students in the study do in fact graduate, there may be a subset of students who leave simply due to the fact they become disillusioned with the institution and want to pursue a degree elsewhere. Thus there is likely other factors that are not observed that effects when a student a graduates. Many of these factors, such as life events such as family hardship, are necessarily hard to observe for an entire university population. There are also likely factors that are particular to this university that are not common or not impacting at other institutions. Future work will investigate the effect of student social network belonging and the amount of financial aid available to the students.

7 Conclusion

This paper has presented a discrete time hazard model predicting when a student will graduate. It uses a novel method to calculate the logistic regression called gradient boosting which has been shown to provide better fits than traditional maximum likelihood. While using more sophisticated imputation with the logistic regression model might have increased performance, the ease of use of xgboosts built-in imputation engine provides a strong advantage in it's use to researchers. Given the utility of gradient boosting in this setting, especially in providing better predictions for under-served populations, this paper recommends that this method be more prevalent in the education research community. Additionally, other methods should be examined such as artificial neural networks [64].

Additionally, this paper used the partial dependence method to examine two of the model variables. Partial dependence tells us the contributions to the predicted probability of the model for the entire feature space. This method allows us to examine the entirety of continuous variables such as GPA instead of reducing them to a single value associated with a model coefficient. This method too should see much broader use in the education research community.

This paper follows Tinto's theory of drop out that predicts social integration and college participation are more likely to impact a student's commitment to graduation than second order effects such as preparation. Future work will connect student academic performance, participation in course work, with social network metrics and financial aid information. Future work will also examine the specific effects that remedial math have on student retention in a major, their time to graduation, and if participating in remedial math courses lowers the likelihood to graduate.

8 Acknowledgements

This project was supported by the Michigan State University College of Natural Sciences including the STEM Gateway Fellowship and the Lappan-Phillips Foundation, the Association of American Universities, and the Norwegian Agency for Quality Assurance in Education (NOKUT), which supports the Center for Computing in Science Education. This project has also received support from the INTPART project of the Research Council of Norway (Grant No. 288125). Morten Hjorth-Jensen is supported by the U.S. National Science Foundation (Grant No. PHY-1404159).

9 Supplemental

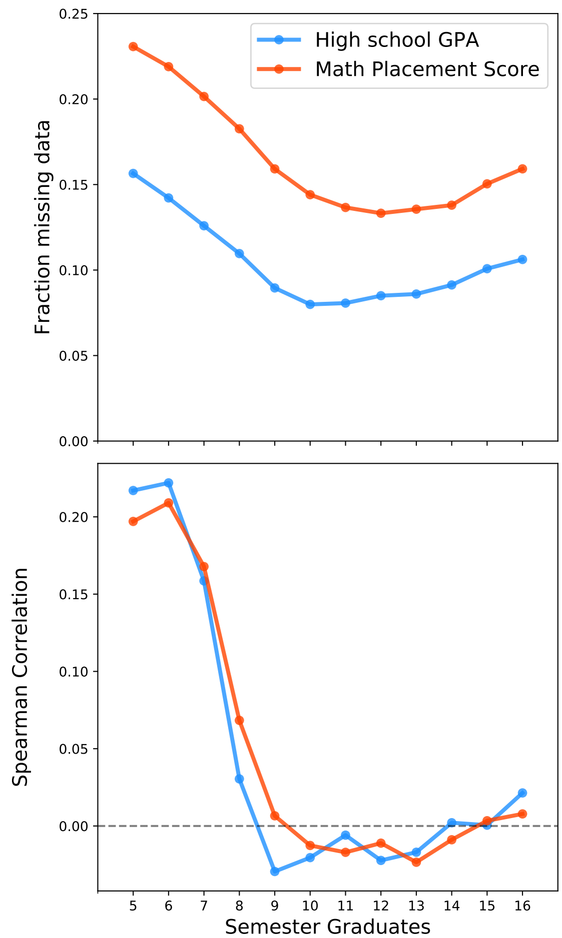

Fig 11. The fraction of missing data and the Spearman rank correlation between whether data is missing and the outcome variable of graduating the following semester. There is a small correlation in the early semesters (5-7) between whether students are graduating and if they have missing data or not.

References

[1] Hongtao Yue and Xuanning Fu. "Rethinking graduation and time to degree: A fresh perspective". In: *Research in Higher Education* 58.2 (2017), pp. 184–213.

[2] John M Braxton, Jeffrey F Milem, and Anna Shaw Sullivan. "The influence of active learning on the college student departure process: Toward a revision of Tinto's theory". In: *The journal of higher education* 71.5 (2000), pp. 569–590.

[3] Jaison R Abel, Richard Deitz, and Yaqin Su. "Are recent college graduates finding good jobs?" In: *Current issues in economics and finance* 20.1 (2014).

[4] Vincent Tinto. "Dropout from higher education: A theoretical synthesis of recent research". In: *Review of educational research* 45.1 (1975), pp. 89–125.

[5] Ernest T Pascarella and Patrick T Terenzini. "Predicting voluntary freshman year persistence/withdrawal behavior in a residential university: A path analytic validation of Tinto's model." In: *Journal of educational psychology* 75.2 (1983), p. 215.

[6] Amaury Nora, LC Attinasi Jr, and Andrew Matonak. "Testing qualitative indicators of precollege factors in Tinto's attrition model: A community college student population". In: *The Review of Higher Education* 13.3 (1990), pp. 337–355.

[7] Alberto F Cabrera, Amaury Nora, and Maria B Castaneda. "College persistence: Structural equations modeling test of an integrated model of student retention". In: *The journal of higher education* 64.2 (1993), pp. 123–139.

[8] Stephen L DesJardins, Dennis A Ahlburg, and Brian P McCall. "An event history model of student departure". In: *Economics of education review* 18.3 (1999), pp. 375–390.

[9] Terry T Ishitani. "A longitudinal approach to assessing attrition behavior among first-generation students: Time-varying effects of pre-college characteristics". In: *Research in higher education* 44.4 (2003), pp. 433–449.

[10] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM. 2016, pp. 785–794.

[11] Kazuo Yamaguchi. *Event history analysis*. Vol. 28. Sage, 1991.

[12] Marc A Scott and Benjamin B Kennedy. "Pitfalls in pathways: Some perspectives on competing risks event history analysis in education research". In: *Journal of Educational and Behavioral Statistics* 30.4 (2005), pp. 413–442.

[13] Xianglei Chen. *STEM Attrition: College Students' Paths into and out of STEM Fields. Statistical Analysis Report. NCES 2014-001.* Tech. rep. 2013. URL: https://nces.ed.gov/pubs2014/2014001rev.pdf.

[14] Rong Chen. "Institutional characteristics and college student dropout risks: A multilevel event history analysis". In: *Research in Higher education* 53.5 (2012), pp. 487–505.

[15] Jerry Trusty and Spencer G Niles. "High-school math courses and completion of the bachelor's degree". In: *Professional School Counseling* (2003), pp. 99–107.

[16] Matthew N Gaertner et al. "Preparing students for college and careers: The causal role of algebra II". In: *Research in Higher Education* 55.2 (2014), pp. 143–165.

[17] Betty Bamberg. "Composition instruction does make a difference: A comparison of the high school preparation of college freshmen in regular and remedial English classes". In: *Research in the Teaching of English* 12.1 (1978), pp. 47–59.

[18] Philip M Sadler and Robert H Tai. "Success in introductory college physics: The role of high school preparation". In: *Sci. Educ.* 85.2 (2001), pp. 111–136. DOI: 10.1002/1098-237X(200103)85:2<111::AID-SCE20>3.0.CO;2-O.

[19] Zahra Hazari, Robert H Tai, and Philip M Sadler. "Gender differences in introductory university physics performance: The influence of high school physics preparation and affective factors". In: *Science Education* 91.6 (2007), pp. 847–876.

[20] Rebecca Zwick and Jeffrey C Sklar. "Predicting college grades and degree completion using high school grades and SAT scores: The role of student ethnicity and first language". In: *American Educational Research Journal* 42.3 (2005), pp. 439–464.

[21] John M Aiken, Rachel Henderson, and Marcos D Caballero. “Modeling student pathways in a physics bachelor’s degree program”. In: *Physical Review Physics Education Research* 15.1 (2019), p. 010128.

[22] Cabot Zabriskie et al. “Using machine learning to predict physics course outcomes”. In: *Physical Review Physics Education Research* 15.2 (2019), p. 020120.

[23] Elaine Seymour. *Talking about leaving: Why undergraduates leave the sciences.* Westview Press, 2000. DOI: 10.1086/495437.

[24] Karl L Alexander et al. “Social background, academic resources, and college graduation: Recent evidence from the National Longitudinal Survey”. In: *American Journal of Education* 90.4 (1982), pp. 315–333.

[25] William H Sewell and Vimal P Shah. “Socioeconomic status, intelligence, and the attainment of higher education”. In: *Sociology of education* (1967), pp. 1–23.

[26] William Richard Smith, Julie Edmister, and Kathleen Sullivan. “Factors influencing graduation rates at Mississippi’s public universities”. In: *College and University* 76.3 (2001), p. 11.

[27] Heather T Rowan-Kenyon. “Predictors of delayed college enrollment and the impact of socioeconomic status”. In: *The Journal of Higher Education* 78.2 (2007), pp. 188–214.

[28] Susan P Choy. *Students whose parents did not go to college: Postsecondary access, persistence, and attainment.* 2001.

[29] Ernest T Pascarella et al. “First-generation college students: Additional evidence on college experiences and outcomes”. In: *The Journal of Higher Education* 75.3 (2004), pp. 249–284.

[30] Trevor Hastie et al. *The elements of statistical learning: data mining, inference and prediction.* Vol. 27. 2. Springer, 2005, pp. 83–85.

[31] Skipper Seabold and Josef Perktold. “statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference.* 2010.

[32] R Core Team. *R: A Language and Environment for Statistical Computing.* ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: http://www.R-project.org/.

[33] Peter Bühlmann, Torsten Hothorn, et al. “Boosting algorithms: Regularization, prediction and model fitting”. In: *Statistical Science* 22.4 (2007), pp. 477–505.

[34] Andreas Mayr et al. “Extending statistical boosting”. In: *Methods of information in medicine* 53.06 (2014), pp. 428–435.

[35] Peter Bühlmann and Bin Yu. “Boosting with the L 2 loss: regression and classification”. In: *Journal of the American Statistical Association* 98.462 (2003), pp. 324–339.

[36] Yuk Lai Suen, Prem Melville, and Raymond J Mooney. “Combining bias and variance reduction techniques for regression trees”. In: *European Conference on Machine Learning.* Springer. 2005, pp. 741–749.

[37] L Lombardo et al. “Binary logistic regression versus stochastic gradient boosted decision trees in assessing landslide susceptibility for multiple-occurring landslide events: application to the 2009 storm event in Messina (Sicily, southern Italy)”. In: *Natural Hazards* 79.3 (2015), pp. 1621–1648.

[38] Qian Xu et al. “PDC-SGB: Prediction of effective drug combinations using a stochastic gradient boosting algorithm”. In: *Journal of theoretical biology* 417 (2017), pp. 1–7.

[39] Xiaolei Ma et al. “Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.9 (2017), pp. 2303–2310.

[40] Fortunato S De Menezes et al. “Data classification with binary response through the Boosting algorithm and logistic regression”. In: *Expert Systems with Applications* 69 (2017), pp. 62–73.

[41] Anthony S Bryk and Stephen W Raudenbush. “Toward a more appropriate conceptualization of research on school effects: A three-level hierarchical linear model”. In: *Multilevel analysis of educational data*. Elsevier, 1989, pp. 159–204.

[42] Jerome H Friedman. “Stochastic gradient boosting”. In: *Computational statistics & data analysis* 38.4 (2002), pp. 367–378.

[43] *IPEDs definitions*. https://nces.ed.gov/ipeds/report-your-data/race-ethnicity-reporting-changes. Accessed: 10-23-2018.

[44] Russell A Poldrack, Grace Huckins, and Gael Varoquaux. “Establishment of Best Practices for Evidence for Prediction: A Review”. In: *JAMA psychiatry* (2019).

[45] *American Community Survey Tables: 2007 – 2011 B19013A*. URL: https://factfinder.census.gov/bkmk/table/1.0/en/ACS/11_5YR/S1903/0100000US.86000.

[46] Stephen J Caldas and Carl Bankston. “Effect of school population socioeconomic status on individual academic achievement”. In: *The Journal of Educational Research* 90.5 (1997), pp. 269–277.

[47] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[48] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. “O’Reilly Media, Inc.”, 2017. URL: https://dl.acm.org/citation.cfm?id=3153997.

[49] James Bergstra and Yoshua Bengio. “Random search for hyper-parameter optimization”. In: *Journal of machine learning research* 13.Feb (2012), pp. 281–305.

[50] Jake M. Hofman, Amit Sharma, and Duncan J. Watts. “Prediction and explanation in social systems”. In: *Science* 355.6324 (2017), pp. 486–488. ISSN: 0036-8075. DOI: 10.1126/science.aal3856. eprint: https://science.sciencemag.org/content/355/6324/486.full.pdf. URL: https://science.sciencemag.org/content/355/6324/486.

[51] Stephen W Raudenbush and Anthony S Bryk. *Hierarchical linear models: Applications and data analysis methods*. Vol. 1. Sage, 2002.

[52] Ben Van Dusen and Jayson Nissen. “Modernizing use of regression models in physics education research: A review of hierarchical linear modeling”. In: *Physical Review Physics Education Research* 15.2 (2019), p. 020108. DOI: 10.1103/PhysRevPhysEducRes.15.020108.

[53] Donald B Rubin. “Inference and missing data”. In: *Biometrika* 63.3 (1976), pp. 581–592.

[54] Cyril Goutte and Eric Gaussier. “A probabilistic interpretation of precision, recall and F-score, with implication for evaluation”. In: *European Conference on Information Retrieval*. Springer. 2005, pp. 345–359.

[55] Gary M Weiss, Kate McCarthy, and Bibi Zabar. “Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?” In: *Dmin* 7.35-41 (2007), p. 24.

[56] Nitesh V Chawla et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[57] Isabelle Guyon and André Elisseeff. “An introduction to feature extraction”. In: *Feature extraction*. Springer, 2006, pp. 1–25. DOI: 10.1007/978-3-540-35488-8_1.

[58] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013.

[59] Didrik Nielsen. “Tree boosting with xgboost-why does xgboost win” every” machine learning competition?” MA thesis. NTNU, 2016.

[60] Jonathan AC Sterne et al. “Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls”. In: *Bmj* 338 (2009), b2393.

[61] Adrianna Kezar. “Higher education change and social networks: A review of research”. In: *The Journal of Higher Education* 85.1 (2014), pp. 91–125.

[62] James W Vaupel, Kenneth G Manton, and Eric Stallard. “The impact of heterogeneity in individual frailty on the dynamics of mortality”. In: *Demography* 16.3 (1979), pp. 439–454.

[63] Philip Hougaard. “Frailty models for survival data”. In: *Lifetime data analysis* 1.3 (1995), pp. 255–273.

[64] Julie M Byers González and Stephen L DesJardins. “Artificial neural networks: A new approach to predicting application behavior”. In: *Research in Higher Education* 43.2 (2002), pp. 235–258.