Tips til masterstudenter i matematikk

Tommy Odland

27. januar 2018

Sammendrag

Dette dokumentet inneholder konkrete tips for masterstudenter i matematikk, og er i hovedsak myntet på studenter i anvendt matematikk. Det er nok nyttig informasjon for andre realfagsstudenter her også.

Introduksjon

Studentlivet inviterer kanskje litt lite til refleksjon og tilbakeblikk. Ofte planlegger man fremover mot neste eksamen når man er ferdig med den forrige. I dette dokumentet ønsker jeg å bryte litt med dette. Jeg ser tilbake på hva jeg skulle ønske jeg visste når jeg begynte på mastergraden, og deler det med deg.

Valg av oppgave

Snakk med mange potensielle veiledere. Selve temaet for oppgaven er nok ikke fullt så viktig som man kanskje skulle tro. Det er minst like avgjørende å ha en veileder som man kan samarbeide med, og at arbeidsoppgavene er litt spennende og meningsfulle.

Er du usikker på hva numerisk analyse, partielle differensiallikninger eller klassisk mekanikk egentlig er? [1] inneholder korte sammendrag av forskjellige fagfelt, skrevet av ledende forskere.

Se på tidligere oppgaver på Bora[1]. Gjennomsnittlig antall sider i `.pdf` filene i matematikkoppgavene er rett over 90. "Stjel" gode idéer fra tidligere oppgaver, f.eks. oppsett og struktur.

[1] Bora: bora.uib.no

Finne referanser

Bruk programvare til å organisere referanser. Zotero[2] henter automatisk metadata[3] fra nettsider som Amazon, realfagsbiblioteket, og andre.

Jeg finner hovedsaklig referanser på søkemotoren til realfagsbiblioteket[4] og på Google. Deretter finner jeg artikkelen/boken på realfagsbiblioteket eller Amazon og lagrer metadata med Zotero. Zotero genererer `.bib` filer, som jeg henter inn i LaTeX. Dette er veldig effektivt.

Når du har en grei idé om hva du skal skrive om, bør du sette av en ettermiddag og finkjemme realfagsbiblioteket etter relevante bøker. Søkemotoren er ikke perfekt, og selv om det tar en stund å gå gjennom noen hyller på biblioteket vil du spare tid på det i lengden.

Skriveprosessen

Bruk LaTeX, du finner god informasjon på Wikibooks[5]. Jeg bruker TeXStudio[6] som editor.

Begynn å skriv tidlig, og noter gjerne viktig informasjon digitalt. Når du skal skrive selve oppgaven er det lurt å planlegge godt. Forsøk å planlegg strukturen på hele oppgaven i detalj før du faktisk skriver, i den grad det lar seg gjøre.

Det er selvfølgelig skrevet bøker om skriving. En god, lettlest bok om det å skrive artikler er [2]. Både [3] og [4] er også verdt å kikke på, kanskje spesielt for matematikkstudenter.

Ta backup hver dag. Jeg brukte Git[7] og student-konto på GitHub[8] med privat repository. Alternativt kan du lagre alle filene på Dropbox[9] eller Google Drive[10].

Figurer til oppgaven

Bruk alltid vektorisert grafikk istedet for pixelgrafikk. Med andre ord, bruk `.pdf` format til alle

[2] Zotero: zotero.org

[3] Med metadata mener jeg informasjon om forfatter, tittel, publikasjonsår, journal, osv.

[4] Realfagsbiblioteket: www.uib.no/ub/76636/realfag

[5] Wikibooks LaTeX: en.wikibooks.org/wiki/LaTeX

[6] TeXStudio: texstudio.sourceforge.net

[7] Git: git-scm.com

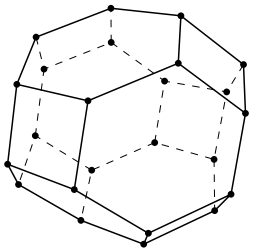
[8] GitHub: github.com

[9] Dropbox: dropbox.com

[10] Google Drive: google.com/drive

figurer, ikke `.jpeg`, `.png` og annet.

Lag generelle figurer med IPE[11]. Figur 1 laget jeg selv i IPE.



Figur 1: En figur laget med IPE.

Grafer kan du lage i MATLAB eller i Python-pakken `matplotlib`[12], jeg bruker sistnevnte.

Programmering

Python[13] er et generelt, seriøst programmeringsspråk som er lett å lære og brukes mye til vitenskap. R[14] brukes en del til statistikk og dataanalyse. MATLAB[15] blir også brukt. Jeg anbefaler Python til alle formål, da gjerne Anaconda[16] distribusjonen. Les om Python på hjemmesiden, [5] er også en god referanse.

Dersom du skriver kode i forbindelse med oppgaven bør du vurdere "test-driven development." Det virker tidkrevende, men sparer deg nok tid i lengden.

Dersom programmet ditt går treigt kan det skyldes flere ting. **(1)** Det vanligste er at du bruker loops istedet for å vektorisere. Effektiv vektorisering gir ofte fartsøkning på 10^1–10^3. **(2)** Det er også mulig at du ikke har brukt effektive algoritmer og datastrukturer. Se gjerne [6] for en introduksjon til algoritmer, boka er pensum i INF234. [7] er en klassisk referanse for algoritmer, men kanskje noe mer teoretisk. Aldri gjør ting fra bunnen om du kan unngå det, innebygde algoritmer er alltid bedre enn det du klarer å koke sammen. **(3)** Dersom nettverket eller skriving/lesing til fil går sakte, bør du bruke threading.

[11] IPE: ipe.otfried.org
[12] Matplotlib: matplotlib.org
[13] Python: www.python.org
[14] R: r-project.org
[15] MATLAB: mathworks.com
[16] Anaconda: anaconda.com

Lær deg terminalkommandoer i Linux[17], da kommer du tettere innpå hvordan en datamaskin fungerer. En spennende bok er [8], sammendragene i [9] er nyttige. Eksempelvis lar `grep` kommandoen deg søke effektivt gjennom dokumenter, og er uvurderlig for skriving og programmering.

Jobbsøking etter master

Du må vite hva du er verdt. Ifølge TEKNA[18] bør startlønnen i 2017 ligge på omtrent 522 000 kr. Bedrifter er ute etter litt andre ferdigheter enn de du lærer på universitetet. Les jobbannonser før du er ferdig, selg deg selv. Lær fra jobbannonsene hva som er nyttig å ta med seg.

Lykke til med masteren!

Referanser

[1] Timothy Gowers, June Barrow-Green, and Imre Leader, editors. *The Princeton Companion to Mathematics*. Princeton University Press, Princeton, y first printing edition edition, September 2008.

[2] Bodil Holst. *Scientific Paper Writing - A Survival Guide*. CreateSpace Independent Publishing Platform, Bergen, December 2015.

[3] Nicholas J. Higham. *Handbook of writing for the mathematical sciences*. Society for Industrial and Applied Mathematics, Philadelphia, 2nd ed. edition, 1998.

[4] Steven G. Krantz. *A Primer of mathematical writing: beeing a disquisition on having your ideas recorded, typeset, published read and appreciated*. American Mathematical Society, Providence, R.I, 1997.

[5] Luciano Ramalho. *Fluent Python*. O'Reilly, 1st edition. edition, 2015.

[6] Sanjoy Dasgupta, Christos H. . Papadimitriou, and Umesh Virkumar. Vazirani. *Algorithms*. McGraw Hill, Boston, Mass, 2008.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. The MIT Press, Cambridge, Mass, 3rd edition edition, July 2009.

[8] Jeroen Janssens. *Data Science at the Command Line: Facing the Future with Time-Tested Tools*. O'Reilly Media, Sebastopol, CA, 1 edition edition, October 2014.

[9] Sander van Vugt. *Beginning the Linux Command Line*. Apress, New York, 2nd ed. edition edition, November 2015.

[17] Mac bruker også UNIX shell, så de samme kommandoene er stort sett tilgjengelige.
[18] Tekna: www.tekna.no