

OM MATLABUNDERVISNINGEN I TMV156

INNEHÅLL

1. Undervisning och examination	1
2. Varför Matlab?	1
3. Pedagogiken	2
4. Matlab i kursplanen	2
5. Litteratur och läsanvisningar	3

1. UNDERVISNING OCH EXAMINATION

Matlabundervisningen i denna kurs består av en föreläsning, fyra laborationer samt en inlämningsuppgift. Dessutom kompletteras den av fyra studentledda Matlabtillfällen. Samtliga undervisningstillfällen är klockan 15.15. De studentledda är förlagda till onsdagen den 31/8, fredagen den 2/9, tisdagen den 6/9 samt fredagen den 9/9. Föreläsningen är onsdagen den 7/9 och laborationerna är de efterföljande onsdagarna. På föreläsningen kommer vi ta flera korta raster istället för en kvart klockan fyra.

Examinationen kommer ske med korta flervalsfrågor i slutet på varje laboration samt med en avslutande inlämningsuppgift. För att klara godkänt krävs att man klarat fyra frågor av tio på minst tre av fyra frågetillfällen samt att man blir godkänd på inlämningsuppgiften. För att bli godkänd på kursen måste man bli godkänd på Matlabdelen. Notera att dessa krav mer eller mindre innebär *närvaroplikt vid laborationstillfällena*. Inlämningsuppgiften skall vara klar och inlämnad senast fredagen den 28/10 klockan 15.30, d.v.s. fredag i läsvecka ett i läsperiod två. Den skall läggas i tidskriftssamlaren i fönstret in till Fredrik Lindgrens kontor MV:L2104 på entré-plan i Matematiska vetenskapers byggnad på Chalmers tvärgata 3. Kontoret är beläget i hörnet närmast Olgas trappor men vänt inåt innergården. Elektronisk inlämning kommer bara att accepteras i extrema undantagsfall. Består inlämningen av fler än ett blad skall dessa vara hophäftade. Författarnas (högst två) namn skall stå på varje blad och på första sidan skall även mejladresser och personnummer vara angivna. I år kommer inga försenade rapporter att accepteras. Blir du underkänd kommer du att få retur där du får rätta till bristerna. Dead-line för returen kommer att meddelas när inlämningarna lämnas ut på föreläsning i läsperiod 2.

Frågorna vid laborationstillfällena kommer dels vara enklare frågor som berör det ni gjort under det laborationstillfället, dels svårare frågor på material från de tidigare tillfällena samt en fråga som blickar fram mot nästa tillfälle. Densnare kommer vara formulerad så att den ska gå att lösa utan att ha tittat på nästa laboration. Frågornas syfte är först och främst att vara ett stöd för inlärnigen och för att peka ut svårigheter och subtiliteter. Inlämningsuppgiften kommer likna de som varit tidigare år och sammanfattar det viktigaste från laborationerna. Specificering läggs ut på kurshemsidan efter den sista laborationen.

2. VARFÖR MATLAB?

Anledningarna till att vi kräver att ni lär er Matlab är flera. En är att dess möjlighet att rita grafer och andra objekt kan verka som ett stöd för matematikinlärnigen. Genom att lära dig hantera Matlabs grafikverktyg kan du visualisera funktioner, problem, lösningar, ... och enkelt variera parametrar och se hur dessa påverkar det problem du studerar. En annan anledning är att många matematiska problem har lösningar som är svåra eller omöjliga att hitta med papper och

penna även om man kan visa att sådana existerar. Ofta kan man dock approximera dessa lösningar med hjälp av en bra algoritm. Ni kommer se exempel på när sådana algoritmer är väldigt enkla men kräver en ofantlig mängd beräkningar för att nå fram till ett godtagbart svar. Räkningar som för en människa skulle ta en livstid kan ta några sekunder eller minuter för en dator att genomföra efter att en duktig programmerare skrivit några få rader. Matlab är ett programmeringsspråk som är väldigt lättanvänt. Dessutom finns det en uppsjö program (så kallade funktioner) med färdiga algoritmer – enkla och svåra – som ni kommer att lära er att nyttja. Att vi använder Matlab och inte något annat programspråk beror just på enkelheten, den suveräna grafikhanteringen samt att det finns så mycket färdiga finesser. Dessutom har Matlab en syntax som är väldigt matematisk till sin natur.

Ytterligare en anledning till valet av Matlab är att det används så mycket både inom forskning och i näringslivet. Av den anledningen dyker Matlab upp i nära nog varje kurs under ingenjörsutbildningen. Det finns alltså all anledning att se till att lära sig grunderna redan nu!

3. PEDAGOGIKEN

Att lära sig programmera i Matlab på fyra laborationstillfällen är svårt. Det kommer krävas arbete även på andra tider för att ro detta i hamn. Det här är måhända en första kontakt med problembaserad inläring med ett väldigt stort ansvar för att skaffa och bearbeta kunskapen själv. Du kommer i mångt och mycket att tvingas lära dig saker samtidigt som du examineras. Du kommer få i uppgift att lösa problem som ingen innan har redogjort för hur de ska lösas. Detta är helt i linje med hur man jobbar inom såväl avancerad ingenjörskonst som forskning. Matlabundervisningen syftar alltså inte bara till att lära ut Matlab utan även till att träna teknologen för en framtida ingenjör- och/eller forskarroll. Det kräver alltså bland annat att du letar information själv i litteratur, på internet, i Matlabs dokumentation samt genom att fråga kamrater och handledarna i datorlaboratoriet. Det kräver också att du hela tiden stannar upp och frågar dig själv om du förstår vad du gör och – om svaret är nej – tar ansvar för att du når förståelse. Matlabkunskaperna kommer byggas upp under dina olika kurser på grundutbildningen och det kommer kosta mer än det smakar att skjuta upp inlärandet av grunderna. Det kommer med all säkerhet komma tillfällen då du kommer känna stor frustration och det gäller i det läget att hålla huvudet kallt. Även det är en god egenskap hos en ingenjör.

4. MATLAB I KURSPLANEN

I kursplanen står följande att läsa om Matlab och numeriska metoder. Matlab är ju ett språk med vars hjälp vi kan implementera numeriska metoder.

Ur lärandemålen:

- ”Studenten skall kunna rita grafer och lösa ekvationer såväl för hand som med MATLAB. Studenten skall också kunna lösa linjära ekvationssystem för hand och med MATLAB. ”
- ”[S]tudenten skall kunna använda MATLAB för att lösa ekvationer numeriskt och för grafritning.”
- ”[S]tudenten skall kunna använda MATLAB för att lösa ekvationssystem numeriskt.”

Ur beskrivningen av kursens innehåll:

- ”Tillämpningar på derivata: Max-min-problem, numerisk ekvationslösning, Newton-Raphson och andra iterationer. Matlabbtillämpningar. ”

Dessutom bör man tolka varje påstående i stil med att studenten ska kunna *lösa* någon typ av problem eller *beräkna* något som att studenten även ska kunna göra detta numeriskt.

En egen implementering av Newton-Raphsons metod med hjälp av en så kallad `for`-loop och tillämpning av denna på olika problem blir slutstationen på denna kurs. Ett sådant program kommer visa sig vara ganska kort och enkelt men det kommer kräva att studenten, d.v.s. du, klarar av att hantera en hel del olika begrepp innan du förstår hur detta skall skrivas och hur det fungerar. De olika undervisningstillfällena syftar till att visa vägen dit – och till att lära ut en hel del annat matnyttigt på vägen: grafritning inte minst.

5. LITTERATUR OCH LÄSANVISNINGAR

Vi kommer att hänvisa till boken *Physical Modeling in Matlab* av Allen B. Downey som finns att ladda ned gratis från sidan <http://greenteapress.com/matlab/> samt Vilhelms kompendium *Matlab – en kort handledning* som finns att ladda ned på kurshemsidan. Vi refererar till den förra texten som ABD och till den senare som VA efter författarnas respektive initialer. Dessutom kommer korta sammanfattningar över matlabs plotfunktioner och en liten text om begrepp i programmering att göras tillgängliga. För den som vill ha en komplett lärobok rekommenderas i första hand *Matlab for engineers* av Holly More som presenterar saker i en för oss passande ordning samt har relevanta exempel från teknikvetenskaper. Föredrar man en svensk bok så finns *Matlabberäkningar inom teknisk och naturvetenskap* av Per Jönsson som i och för sig är en bra bok men introducerar matrisoperationer och texthantering på ett i våra ögon för tidigt stadium.

Här följer rekommenderad läsning inför föreläsningen och de lärarledda datorövningarna.

- 7/9 På föreläsningen kommer mycket att gå igenom. Läs Vilhelms kompendium. Viktiga begrepp är variabler, variabelers räckvidd (globala, lokala), operatorer, elementvisa operationer, funktioner, funktionsanrop, anonyma funktioner, funktionshandtag, logiska operatorer, villkorssats, if-else, for-loop . . .
- 14/9 Laboration 1. Läs avsnitt 1.1-1.10, 2.6, 4.5-4.9, 4.11 och 12.1-12.2 i ABD. Studera också alla avsnitt till och med avsnitt 11 i VA. Du kan hoppa över avsnitt 8.
- 21/9 Laboration 2. Läs avsnitt 1.9-1.10, hela kapitel 2, 3.1-3.3, 3.5 och för den sista delen av laborationen, kapitel 6 i ABD. Anonyma funktioner är ett viktigt begrepp i laborationen och ett utmärkt hjälpmedel. Tyvärr innehåller litteraturen vi använder inget bra om detta. Det kan dock vara värt att studera kapitel 5 i ABD samt avsnitt 12 och 15 i VA. För att först vitsen med det hela är även avsnitt 13 och 14 i VA viktiga. Även avsnitt 8, 11 och 17 i VA bör studeras liksom länk från hemsidan om grafitning.
- 28/9 Laboration 3. I den här laborationen behandlar vi något som tycks höra till det svåraste i den grundläggande programmeringsläran, åtminstone när man lär sig programmera med Matlab: funktionsanrop. Studera kapitel 5 i ABD och avsnitt 12-17 i VA mycket noggrant. Tänk på hur funktioner hjälper till att skapa skyddade lokala variabler och vad som är inargument/invariabler respektive utargument/utvariabler och vilken roll dessa spelar när en funktion ska ingå i en större programstruktur och anropas av andra funktioner. Gå inte ifrån den här laborationen utan att känna att du behärskar detta, det är fullkomligt nödvändigt för att kunna göra mer avancerade saker i Matlab, och det kommer vi göra redan nästa laboration och i alla kurser framöver. Läs också kapitel 6 i ABD. För att förstå programmet `sinfaf.m` måste du förstå `if-else`-satser och hur dessa testas påstående med hjälp av realtions-operatorer och logiska operatorer. Det kan du läsa om i avsnitt 4.2-4.4. Mer om detta kommer också att publiceras i ett litet programmeringskompendium på hemsidan.
- 5/10 Laboration 4. I den här laborationen implementeras Newton-Raphsons metod. Läs om den i *Calculus – A Complete Course* av Robert A. Adams. Du hittar den lätt genom att slå efter Newton's method i registret. Förstå hur den bygger på att iterativt linjärisera den ekvation man vill lösa runt en gissad punkt och sedan lösa den linjära ekvationen varvid man får en ny punkt att linjärisera runt o.s.v. För att kunna implementera en bra lösare så måste vi förstå hur en `for`-loop fungerar. Läs därför hela kapitel 3 i ADB och studera särskilt noggrant avsnitt 3.4 samt 3.6-3.10. För att göra en *riktigt* bra lösare bör man istället för en `for`-loop använda en `while`-loop. Du kan läsa om detta i det programmeringskompendium som kommer publiceras på hemsidan (överkurs, men `while`-loopen återkommer i envariabelanalysen). Dessutom är det i det här läget bra att studera de delar av kapitel 4 som du inte redan läst.