

**OBS! Snabbinsatt Matlab-intro
vissa fönsterhanteringsdetaljer kan vara fel men
gör gärna Matlab-uppgifterna.**

DN1240, Numeriska metoder för OPEN1.

Laboration 0 – del 1-3 (frivilliga delar)

Del 1-3 (dvs uppg 1.1-1.17) behöver inte redovisas

Introduktion till UNIX och MATLAB

Del 1: UNIX och kontoadministration

Uppgift 1.1

Ni bör jobba två och två med laborationerna. Dels för att det blir mindre jobb per person men framför allt därför att man lär sig bäst genom att diskutera med varandra. De två som jobbat ihop bör redovisa tillsammans men båda skall då kunna alltihop som om han/hon gjort det ensam. Man får jobba ensam om man vill.

Sätt er vid en terminal och sedan loggar ni in på ett av era konton. Det spelar ingen roll vilket. Den vane unix-datorhanteraren kan nu hoppa direkt till uppgift 1.11, dvs inledande Matlab. Den som vill öva lite Unix och fönsterhantering fortsätter här.

När du loggar in för allra första gången får du upp ett tomt skrivbord med diverse knappar/rullgardiner i över- och/eller underkant. Jag brukar ha tre fönster när jag jobbar. Ett terminalfönster, ett Emacs-fönster och ett Matlab-fönster. Emacs-fönstret använder du till att skapa och skriva texter och program, terminal-fönstret använder du till att ge kommandon och köra program och Matlab-fönstret till att köra Matlab-program och styra Matlab-grafik-fönster.

Om du vill ha ett nytt terminalfönster kan du öppna ett genom att klicka klicka på rullgardinen *Applications* (uppe, till vänster) välja *Accesories* och sedan välja *Terminal*. Om du vill ha ett Emacs-fönster kan du öppna ett på samma sätt eller genom att skriva `emacs &` i terminalfönstret. (OBS! Jag rekommenderar ETT Emacs-fönster när man jobbar! INTE FLERA!).

Du kan flytta ett fönster vart du vill genom att dra i det i den övre kanten (dvs håll nere vänster musknapp medan du flyttar musen). Storleken kan ändras genom att klicka och dra (med vänster knapp) i fönstrets hörn.

Uppgift 1.2

Några användbara Unix-kommandon i terminal-fönstret som du kan prova nu är `pwd` som visar vilken katalog du befinner dig i och `ls` som visar vilka filer som finns i den katalog du befinner dig i (`pwd` är kort för *print working directory* och `ls` är kort för *list*).

Kommandot `ls /info/numo12` (eller `ls /info/numo12/`) (Obs! NUMO12 med små bokstäver) visar vilka filer som finns i kurskatalogen (kurskatalogen kallas också kursbiblioteket), `ls /info/numo12/mfiler` som visar vilka filer som finns i katalogen *mfiler* som finns i kurskatalogen (mfiler är kort för Matlabfiler=Matlabprogram), `finger xxx` som visar info om den användare som heter *xxx*, (ofta är användarnamnet samma som Epost-adressen).

Uppgift 1.3

Skriv i terminalfönstret `xeyes` och tryck sedan retur. Du har nu startat programmet *Xeyes* och det ritar ut ett par ögon som tittar på din markör, oavsett var på skärmen den befinner sig. Du kan flytta ögonen genom att klicka med musens vänstra knapp på den övre tjocka ramen och så länge du håller knappen nedtryckt flyttas ögonen med. Storleken ändras genom att du klickar en gång på minustecknet i terminal-fönstrets övre vänstra hörn och sedan väljer "Size". Storleken ändras sedan när man drar i ögon-fönstrets sidor.

Försöker du nu skriva kommandon i terminal-fönstret, prova tex `ls`, så händer inget. Detta beror på att terminal-fönstret jobbar med *Xeyes*-programmet. Ställ markören i terminal-fönstret och tryck CTRL-C, (dvs håll ned *Control*-tangentsen och tryck till C-tangentsen en gång). Nu försvinner ögonen eftersom vi stoppar programmet som kördes i terminal-fönstret. Om du nu skriver något i terminal-fönstret så utförs det snällt. CTRL-C är standard-kommandot för att tvärt avbryta en program-körning.

Skriv nu i terminalfönstret `xeyes &` och tryck sedan retur, observera "&"-tecknet! Du har nu återigen startat programmet *Xeyes* men det har startats i ett eget fönster, som en separat process, och du kan denna gång ge nya kommandon i terminal-fönstret fastän ögonen är i gång.

När ni så småningom jobbar med Matlab så rekommenderar jag att ni har (minst) tre fönster öppna: Terminal-, Emacs- och Matlab-fönster, mer om detta senare.

Uppgift 1.4

Nu skall den av er vars konto ni använder **just nu** fixa info-sidor och ange labbkompis. Info-sidorna fås med `course join numo12` (dvs NUMO12 med små bokstäver) Kommandot gör att du får snabbänkar till kursens WWW-sidor och att eventuella kursmeddelanden visas när du loggar in.

Labbkompisen fixas med kommandot `course labdir numo12 xxx` där du bytt ut *xxx* mot din labbkompis användarnamn. Detta skapar en katalog som heter *numo12* och som båda kan jobba med.

När detta är gjort loggar du ut (genom att klicka på EXIT i frontpanelen).

Uppgift 1.5

Nu skall den andra av er logga in. Fönstren dyker upp på samma sätt. Den nya skall ge samma två kommandon som i punkten ovan: `course join numo12` och `course labdir numo12 xxx` där du bytt ut *xxx* mot din labbkompis användarnamn. (Alltså om *aaa* och *bbb* skall jobba ihop, så skriver *aaa* kommandot `course labdir numo12 bbb` och *bbb* skriver kommandot `course labdir numo12 aaa`).

Nu jobbar ni vidare på "den andras" konto men genom kommandona ovan spelar det ingen roll, alla filer ni gör på katalogen *numo12* kan nås på samma sätt från båda era konton.

Observera att inget av detta innebär anmälan till kursen! Ni måste anmäla er till kursen, tex genom att skriva upp er på listan på första förelsnigen.

Uppgift 1.6

När ni loggar in hamnar ni alltid på er "hemkatalog" (som heter `~xxx`), för att gå ned till numme-katalogen skriver ni `cd numo12` Prova att ge kommandot `pwd` efteråt så ser ni att ni flyttats.

Det kan vara praktiskt att direkt skapa en ny katalog *Lab1* för att enkelt hålla redan på alla filer till *Lab1*. Detta gör ni med `mkdir Lab1` och sedan kan ni gå dit med `cd Lab1`. Prova gärna `ls` och `pwd` igen för att se att katalogen skapats och att ni flyttats rätt (syns också i promptern). (`cd` är kort för *change directory* och `mkdir` är kort för *make directory*.)

Uppgift 1.7

Prova nu att kopiera en m-fil från kursbiblioteket. Unix-kommandot for kopiering är `cp` (*copy*). Titta bland m-filerna på kursbiblioteket (`ls /info/numo12/mfiler`), där finns en fil som heter *tomat.m*. Kopiera över

den till er (med `cp /info/numo12/mfiler/tomat.m tomat.m` eller `cp /info/numo12/mfiler/tomat.m .` (OBS punkten!) vilket gör att er kopia får samma namn som originalet). Kolla med `ls` att kopieringen fungerade.

På kursbiblioteket ligger också en m-fil som heter *banan.m*. Med kommandot `cp /info/numo12/mfiler/banan.m gurka.m` skapar ni en kopia av filen *banan.m* men er kopia heter *gurka.m*.

Alla Matlab-program måste ha avslutning ".m" Ni kan spara en fil under vilket namn ni vill, men skall ni kunna köra den i Matlab måste den sluta med "punkt-m". Den får heller inte börja med en siffra, se mer i läroboken.

Uppgift 1.8

Titta på några av filerna i terminal-fönstret med kommandot `more` (tex `more tomat.m`).

Uppgift 1.9

Man kan starta Matlab på många olika sätt, välj tex bland (jag föredrar alternativ b eller c):

- Starta Matlab genom att i *Applications* välja *Programs* och bland dem *Matlab*. Matlab startar nu (det tar en stund) och du kommer att befinna dig, precis som vid inloggningen, på din hemkatalog. Du måste nu gå ned till rätt katalog med kommandot `cd` (om du följt råden ovan: `cd numo12` följt av `cd Lab1`).
- Starta Matlab genom att i terminal-fönstret ge kommandot `matlab` & Fördelen med detta sätt är att vi automatiskt i Matlab är kvar på den katalog vi var på innan vi startade Matlab. (Glömde du et-tecknet efter Matlab har du nu ett dött terminalfönster, men du kan alltid öppna ett nytt terminal-fönster.
- Starta Matlab genom att i terminal-fönstret ge kommandot `matlab -nojvm` & Fördelen och nackdelen med detta är att man slipper en massa fönster. Det går fortare att starta och köra men man får mindre mängd information på skärmen. (Bra om man sitter med låg bandbredd.)

Ge nu kommandot `ls` eller `dir` så skall du se att filerna *tomat.m* och *gurka.m* ligger där. Vill du **titta på** programmet *tomat* skriver du `type tomat.m` i Matlab-fönstret. Vill du **köra** programmet *tomat* skriver du `tomat` (alltså utan ".m")(i *Matlab Command Window*). Det bör då komma upp ett grafik-fönster med en välkänd frukt i. (Den eventuellt konstiga färgen över hela skärmen försvinner när du klickar i Matlab-fönstret igen). Du kan på vanligt sätt flytta grafik-fönstret dit du vill.

Uppgift 1.10

Flytta dig till Emacs-fönstret (om du saknar ett, läs uppgift 1.1 igen). Hämta fram filen *tomat.m*. Det gör du genom att i menyn *File* välja "Open" och ange filnamnet `~/numo12/Lab1/tomat.m` Välj i menyn *File* "Save as..." och spara den under namnet *citron.m* Eftersom vi inte angav någon katalog blir det samma som Emacs hade nyss, dvs *Lab1* under *numo12* Gå tillbaka till Matlab-fönstret. Kör det nya programmet *citron*, det blir fortfarande en röd frukt!? Gå tillbaka till Emacs-fönstret. På rad 6 i *citron*-programmet står det vilken färg man vill fylla cirkeln med. Byt ut röd ('r'=red) mot gul ('y'=yellow) så att det blir `fill(x,y,'y')` och spara filen. Gå tillbaka till Matlab-fönstret. Kör programmet *citron* och jämför med programmet *tomat*. Skillnad? Du har just gjort ett eget Matlab-program!

Notera att man kan titta på innehållet i en fil i alla tre fönstrena, bara kommandona är olika. Vill man däremot ändra i filen skall man vara i Emacs-fönstret och vill man köra ett Matlab-program skall man vara i Matlab-fönstret. Att skriva ut programmet på papper är ett kommando som berör dator-systemnätet och bör därför ske från **terminal-fönstret**: `print citron.m` (Det **går** att skriva ut program även från Matlab-fönstret, men med ett enda litet skrivfel så kan hela programmet ohjälpligt raderas och rekommenderas därför ej. Kommandot `print citron.m` skrivet i Matlab-fönstret **raderar hela filen**!) Hur man skriver ut grafik-fönstret kommer vi till lite senare! (Man kan skriva ut Matlab-koden direkt från Emacs-fönstret också.)

Del 2: Inledande Matlab

Skumma igenom det gröna Matlab-häftets avsnitt 1-6 & 15 (sidorna 1-16 och 27-35) och gör följande uppgifter

Uppgift 1.11

Skriv ut på skärmen en tabell med två kolumner över funktionen $f(x) = \frac{x}{20} - 2 - 3e^{-x}$ för $x = 0, 0.5, 1, \dots, 5$. Den vänstra kolumnen skall vara x -värdena, den högra motsvarande f -värden. Gör sedan samma sak med funktionen $g(x) = \frac{x^3}{20} - 2 - x^3 e^{-x}$.

Uppgift 1.12

Rita grafen för funktionen $g(x) = \frac{x^3}{20} - 2 - x^3 e^{-x}$ för x på intervallet $(0, 5)$. Välj lämpligt tabellsteg och rubriker och axelbeteckningar så att grafen blir snygg. När den är riktigt fin skriver ni ut grafen på skrivaren. Se till att ni har med ert namn i rubriken så att ni känner igen den när den kommer ut på papper.

Uppgift 1.13

Givet följande två matriser A och B samt vektorn x :

$$A = \begin{pmatrix} 2 & 1 \\ 4 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix} \quad x = \begin{pmatrix} 7 \\ 9 \end{pmatrix}$$

Skapa dessa variabler i Matlab och bilda sedan $C = AB$, $D = BA$, $F = A * B$, $G = B * A$, $z = Ax$, $p = z^T z$, $E = A^T A$ samt $q = x^T E x$. Vilka av de nya variablerna är matriser, vektorer respektive skalärer? Är p och q lika? Är C och D lika? Är C och F lika? Är F och G lika? Varför?

Uppgift 1.14

Använd Matlab för att lösa det linjära ekvationssystemet

$$\begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 4 & 16 & 64 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ -2 \\ 14 \end{pmatrix}$$

Del 3: Lite Matlab-program och grafik

I följande uppgifter kommer du att rita flera grafik-bilder på frukter. Skriv helst inte ut dem på papper! De drar väldigt mycket färg!

Uppgift 1.15

Med dina nya kunskaper i Matlab, fixa till din citron så att den inte blir så kantig utan rund och fin.

Uppgift 1.16

Skriv i Matlab-fönstret kommandona `clf`, `clear` Det första rensar grafik-fönstret, det andra rensar Matlabs variabel-minne. Varje gång när man börjar med en ny uppgift bör man ge dessa två kommandon för att få en fräsch start. Skriv `x,y` och Matlab klagar och säger att variablerna x och y är odefinierade. Kör programmet `gurka.m` och ge därefter ånyo kommandot `x,y` Nu finns variablerna och deras värden skrivs ut på skärmen. (Med kommandot `who` kan du se vilka variabler som finns.)

- Hur många element har vektorn x ?
- Varför finns inte värdet 1.2 med i vektorn?
- Vad händer om man tar bort `hold on`? (görs lättast genom att man (i Emacs) kommenterar bort raden genom att sätta ett %-tecken först på raden).
- Vad händer om man också kommenterar bort den sista raden, den som börjar med `"fill([1,1 ...])"`?
- Ditt program heter ju `gurka.m`. Gör om det så att det ritar en gurka! (Vilka färger som finns kan du se i Matlab-häftet eller genom att i Matlab-fönstret skriva `help plot`).
- Vad händer om du byter ut `fill` mot `plot`?

Uppgift 1.17

Skriv i Matlab-fönstret kommandona `clf`, `clear` Tag en kopia av `banan`-programmet från kursbiblioteket (i vilket fönster gör du det?)

- Kör programmet `banan.m` med kommandot `banan` Kör därefter programmet `tomat.m`? med kommandot `tomat` Vad händer?
- Kör programmen i direkt följd genom att på samma rad skriva `banan, tomat` Vad syns på skärmen när allt är klart? (dvs du har åter fått `>>` först på raden i Matlab-fönstret.)
- Tag bort kommandot `clf` ur `tomat`-programmet. Kör dem nu efter varandra med `banan, tomat` Vad blir slutresultatet? Varför?
- Vad händer om du i stället kör dem i omvänd ordning med `tomat, banan`? Varför?

Den sista uppgiften visade förhoppningvis varför man ogärna har `clear` och `clf` inuti sina program. Man bör i stället göra dessa kommandon innan man startar programmet. Dvs om jag vill köra `banan`-programmet så skriver jag normalt på en rad `clear, clf, banan` Vill jag köra båda programmen skriver jag `clear, clf, banan, hold on, tomat` eller `clear, clf, tomat, hold on, banan` och utan `clf` inne i programmen spelar ordningen ingen roll för bilden. När jag vill köra om programmen använder jag uppåt-pilen för att få tillbaka föregående Matlab-kommando utan att behöva skriva om det.

Slutligen, om att starta en web-browser: I rullgardinen *Applications* under *Internet* kan du välja vad du vill starta, tex *Firefox Web Browser*. Den kan du också starta genom att i ett terminalfönster skriva `firefox &`

Nu är ni definitivt förberedda att jobba vidare med MÖ-uppgifter och Lab1!

/---NC---