

## Vektorer, matriser och mera plot

*I denna laboration skall du lära dig lite mer om möjligheterna att rita ut saker i MatLab, använda inbyggda matrisfunktioner samt skapa ett par funktioner som hanterar matriser (kopplat till algebra).*

### Tidsåtgång

Denna laboration beräknas ta ca 9 timmar i anspråk (6 timmar bokade med assistent).

### Mål

Du ska efter denna laboration känna till

- ännu mer om hur plot fungerar
- andra former av utritningsmöjligheter
- hur man hanterar vektorer och matriser
- diverse inbyggda saker i MatLab

### Uppgift 1 (ca 20 min)

Skriv funktionen `my_zeros` som returnerar en vektor med 20 nollor. Du skall i denna uppgift inte använda dig av den inbyggda `zeros` (givetvis) utan använda dig av upprepningar m.m.

### Uppgift 2 (ca 10 min)

Skapa en vektor som enbart innehåller nollor, en som innehåller enbart ettor samt en som innehåller talen 1, 2, 3, ..., 20. Alla vektorerna skall innehålla 20 heltal. Använd vad du vill i MatLab.

### Uppgift 3 (ca 20 min)

Plotta ut innehållet i en vektor som innehåller talen 1 till 20 på två olika sätt. De data som finns i vektorn skall kunna avläsas i y-led. På x-axeln skall positionerna vara i första fallet 1 till 20 och i andra fallet 1, 1.1, 1.2, ... Detta går att lösa med endast `plot`.

Tips: Man kan skicka två vektorer till `plot`.

### Uppgift 4 (ca 10 min)

Skapa ett program som skriver ut resultatet av ett valfritt antal tärningskast. Användaren skall få ange antalet kast. Vi antar att det alltid är 5 tärningar.

```
>> throw_dices(3)
     1     3     2     5     5
     6     4     3     5     6
     5     6     2     3     1
```

Tips: "`help rand`" (eller kanske något som liknar detta).

## Uppgift 5 (ca 2 tim)

Titta på det testprogram som finns givet på kurshemsidan som heter *sorting\_out\_sorting.m*. Det handlar om hur man kan visualisera sorteringsmetoder. Titta även på programkoden och ta reda på hur allt utförs.

Tips: Saker du inte förstår kan finnas med i den inbyggda hjälpen i MatLab (*help*, *helpwin* eller *doc*).

Din uppgift är att lägga till den sorteringsalgoritm som kallas "Selection Sort" (urvalssortering). Sök på nätet efter beskrivning över hur denna algoritm fungerar..

## Uppgift 6 - kopplad till er algebrakurs (ca 3 tim)

Skriv funktionerna **my\_add** och **my\_mtimes** som adderar respektive multiplicerar två matriser med varandra. Tänk på att de två matrisernas dimensioner antingen måste överensstämja, eller att minst en av dem måste vara en matris av dimension 1x1 (d.v.s. en skalär), för att operationerna skall vara möjliga. Om matrisernas dimensioner gör att det inte går att utföra uppgiften skall ett felmeddelande skrivas ut och exekveringen skall avbrytas. Du skall inte använda dig av inbyggda funktioner för att utföra hela matrisadditionen/-multiplikationen utan skapa funktionerna från grunden.

Addition/multiplikation med en skalär innebär att alla tal i matrisen skall adderas/multipliceras med denna skalär. OBS! skalären kan antingen vara den första eller sista parametern och det skall inte vara någon skillnad i resultat. Du skall INTE införa fler parametrar. Exempel på testfall:

```
my_add([1, 2, 3], [1, 2, 3])      => [2, 4, 6]
my_add([1, 2, 3], 2)             => [3, 4, 5]

my_mtimes([1, 2, 3], [1 ; 2 ; 3]) => 14
my_mtimes(2, [1, 2, 3])          => [2, 4, 6]
my_mtimes([1 ; 2 ; 3], [1, 2, 3]) => [1, 2, 3 ;
                                       2, 4, 6 ;
                                       3, 6, 9]
```

## Uppgift 7 - kopplad till er algebrakurs (ca 2 tim)

Skriv en funktion **my\_det** som beräknar determinanten av en matris. Som indata till funktionen skall enbart matrisen skickas.

Krav 1: Den matris som kommer in i er funktion skall kontrolleras att den är kvadratisk. Du skall inte använda dig av den inbyggda funktionen för att utföra hela determinantberäkningen utan skapa funktionen från grunden.

Krav 2: Det skall gå att lösa olika dimensioner av matriser. Inte bara 3x3-matriser utan även t.ex. 10x10-matriser eller större.

Tips: Det kan vara trevligt att lösa denna uppgift rekursivt då man behöver determinanten för en delmatris (eller flera) för att komma fram till det slutgiltiga resultatet.

## Uppgift 8 - kopplad till er algebrakurs (ca 1 tim)

Titta på och fundera på hur programmet *rotate\_3d\_cube.m* fungerar (ligger på hemsidan). Det som finns i denna uppgift relaterar till rotationer, translationer m.m. som tas upp i algebrakursen. Din uppgift är att sätta dig in i hur programmet fungerar och sen modifiera det så att kuben roterar kring en annan koordinat (t.ex. (1, 2, 3)).

Som extra uppgift (som ni ej behöver redovisa) kan ni modifiera programmet så att kuben flyttar sig runt i en cirkel runt origo samtidigt som den roterar.

2010-11-16