
F1: Introduktion, Matlabrepetition (kap. 1–2)

Gemensam intro

Kursinnehåll

Varför programmera?

Egenskaper hos Matlab

Kommando-, redigerings-, arbetsplats-, tabell-, guide- och hjälpfönster,
kommando-, funktions-, m- och loggfil

Tabeller, storlek, dimension, variabel, identifierare, sökväg

Uttryck, tilldelning, tysta kommandon, radindelning, litteraler

Operatorer, aritmetiska, jämförande, logiska, komponentvisa

Skapa tabell, indexera

Datatyper

Resten av kursen

Interpreterat/kompilerat

Man skriver Matlabkommandon i kommandofönstret och i m-filer (kommandofiler eller funktionsfiler). Kommandona översätts till p-kod (pseudokod). P-kod är maskinberoende. En interpretator läser p-koden och utför en p-kodsinstruktion i taget. P-koden sparas under den aktuella sessionen eller tills minnet blir fullt. Interpretation ger inte maximal effektivitet men maskinberoende. I Matlab finns möjlighet att kompilera, dvs. att översätta till maskinberoende maskininstruktioner. När det är gjort kan programmet utföras maximalt effektivt.

Många klassiska programspråk är kompillerade t.ex. Fortran och C. Java fungerar som Matlab, dvs. använder en sorts p-kod som kan kompileras när den exekveras första gången, för att de snabbare maskininstruktionerna ska användas om programdelen exekveras igen. Perl, Applescript, Javascript m.fl. programspråk interpreteras normalt.

Motivering för programmering

Civilingenjören och teknologen behöver:

- göra snabba kalkyler,
- visa kalkyler som övertygar uppdragsgivare/kund,
- producera hjälpmedel som uppdragsgivare/kund kan använda,
- producera program som ingår i produkter,
- göra program som personliga hjälpmedel
- m.m.

Egenskaper hos Matlab:

- kan användas interaktivt, som kalkylator,
- innehåller många inbyggda funktioner och kan byggas ut med egna kommandofiler och funktioner och extra verktygslådor (toolbox),
- kan relativt lätt skapa diagram och annan grafik,
- är normalt interpreterat

Träna logiskt tänkande och problemlösning.

Att kunna programmera

- Behärska programspråket och programmeringsmiljön.
- Veta hur man hanterar vanliga delproblem.
- Söka fel, testa och ta hänsyn till effektivitet.
- Dokumentera och skriva program så att de är lätta att underhålla.
- Känna till människans förmåga och ta hänsyn till den i designprocessen och i själva programmet.

Repetition av Matlab

Matlab finns för många plattformar, bl.a. Unix, Windows och Mac OS X. Det finns många viktiga fönster. *Kommandofönstret* är centralt. Där skriver man kommandon som utförs direkt. Matlab skriver normalt resultat i detta fönster. Mycket av det som man kan göra i andra fönster, kan man också göra med kommandon i kommandofönstret. Man kan starta kopiering (loggning) av kommandofönstret till en fil med *diary filnamn*. Med *diary off* slutar man spara.

De inbyggda funktionerna och programdelar som man gör själv är normalt m-filer som innehåller Matlabkommandon och har filnamn som slutar på ».m». M-filer kan redigeras med Matlabs redigeringsprogram eller med något annat redigeringsprogram, MSWord, emacs e.d. I Matlabs *redigeringsfönster* har man också tillgång till avlusningsfunktioner för att sätta stoppunkter, stega och inspektera variabler. I *arbetsplatsfönstret* kan man inspektera och i *tabellfönstret* kan man också ändra variabler.

Staffan Romberger, CSC, KTH, 2008-11-13

Tabeller

Matlab står ju för matrix laboratory (matrislab). Med matris menar man ju normalt en tabell med rader och kolumner. Vi kommer att använda termen »tabell» för Matlabs mer generella matriser. Man kan se praktiskt taget allt som Matlab kan hantera som tabeller. En tabell har en storlek (*size*) som är en vektor med antalet komponenter i de olika riktningarna/dimensionerna. Antalet komponenter i storleken är tabellens dimension. En matris är en tabell med dimension 2. Vektorer hanteras som matriser med endast en komponent i den ena dimensionen. En radvektor har en komponent i första dimensionen (en rad) och en kolumnvektor har en komponent i andra dimensionen (en kolumn). Skalärer (tal) hanteras som matriser med storleken [1 1]. Tabeller med dimension 3 eller större finns men används sällan.

Alla element i normala tabeller har samma typ. Det finns andra sorters tabeller.

Staffan Romberger, CSC, KTH, 2008-11-13

Hjälp

Det finns mycket hjälpinformation om Matlab förutom kursen och läroböcker. Via hjälpmenyn (eller med kommandot *helpbrowser*) når man hjälpfönstret med massor med länkad information. Kommandot *lookfor text* söker efter m-filer med text i första dokumentationskommentarrad. Kommandot *help filnamn* skriver alla dokumentationskommentarer dvs. inledande kommentarrader i m-filen *filnamn.m*. Det är lämpligt att förbereda egna m-filer så att de fungerar bra för *lookfor* och *help*.

Staffan Romberger, CSC, KTH, 2008-11-13

Variabler

En tabell som man vill spara till senare under räkningarna lagras man i en variabel. En variabel har ett namn (identifierare) och man kan se variabeln som en namngiven plats i datorns minne där man kan spara/lagra en tabell.

Identifierare kan innehålla A–Z, a–z, siffror och *_* (understreck) och måste börja med bokstav. Versaler och gemena (stora och små bokstäver) anses olika och identifierare har en maximalt tillåten längd (vanligen 19 tecken). Normalt använder man gemena. För s.k. globala variabler använder man versaler. Det är viktigt att välja bra identifierare.

De variabler som man använder i kommandofiler ingår i arbetsplatsen. Man kan se vilka det är med *who*, få mer information, bl.a. storlek, minnesutrymme och elementtyp, med *whos* och radera med *clear*. Det finns några fördefinierade variabler såsom *ans*, *eps*, *realmax*, *realmin*, *pi*, *Inf*, *NaN*, *i*, *j*. Variabeln *eps* påverkas inte av *clear*.

Staffan Romberger, CSC, KTH, 2008-11-13

Sökväg m.m.

Matlab har en aktuell sökväg. När Matlab ska tolka en identifierare provas tolkningar i denna ordning:

- variabel,
- inbyggd funktion eller inbyggt kommando,
- m-fil längs aktuell sökväg.

Om Matlab verkar ha hängt sig, pröva med C-c.

På vissa plattformar kan man skicka kommandon till OS med **!kommando**.

Det finns särskilda fönster (guidefönster) för att skapa användargränssnitt och hantera grafiska objekt.

Staffan Romberger, CSC, KTH, 2008-11-13

Operatorer

Matlab har många operatorer:

- + , - addition, subtraktion
- * multiplikation
- / division (högerdivision) $10/5$ är 2
- \ vänsterdivision $10\5$ är $5/10$ är 0,5
- ^ exponentiering (upphöjt till) 2^3 är 8

För tabeller finns det elementvisa operationer. För att $A \text{ op } B$ ska vara tillåtet måste A och B ha samma storlek eller åtminstone den ena vara skalär:

- + , - elementvis addition, subtraktion
- .* elementvis multiplikation
- ./ elementvis division
- .\ elementvis vänsterdivision
- .^ elementvis exponentiering

Staffan Romberger, CSC, KTH, 2008-11-13

Uttryck och tilldelning

När man skriver en formel/ett uttryck i kommandofönstret beräknas uttryckets värde, värdet tilldelas variabeln `ans` och variabeln skrivs ut. Man kan lagra uttryckets värde i en annan variabel med ett tilldelningskommando. Man kan undertrycka utskriften genom att skriva `»;»` (semikolon) efter kommandot. Man kan styra hur mycket plats utskrifterna tar med format `compact/format loose`. Man kan skriva mer än ett kommando på en rad med `»;»` (kommatecken) eller `»;»` mellan kommandona. Ett kommando kan fortsättas på nästa rad om man avslutar raden med `»...»` (tre punkter).

Uttryck byggs upp, ungefär som matematikens formler, av tallitteraler, parenteser, operatorer, [] (hakparenteser), , (kommatecken), ; (semikolon) (för att sätta ihop tabeller) och funktionsanrop.

Litteral är beteckning för ett värde. Litteraler kallas ibland för konstanter, men i programmeringssammanhang är en konstant en variabel vars värde inte kan ändras, när den har fått sitt första värde (write once).

Staffan Romberger, CSC, KTH, 2008-11-13

```
A = [1 2;3 4]; B = [2 4;8 16]; C = 5;
A+B
ans = 3 6
11 20
A./B
ans = 0.5 0.5
0.3750 0.25
C.\B
ans = 0.4 0.8
1.6 3.2
```

Staffan Romberger, CSC, KTH, 2008-11-13

Icke elementvisa operatörer

När den ena operanden är skalär fungerar dessa operatörer elementvis.

$C = A * B$ matrismultiplikation, A och B har högst dimension 2 och sista dimensionens storlek i A är lika första dimensionens storlek i B

$C = A / B$ matrisdivision, lösning av linjärt ekvationssystem

$F = E \setminus D$ $C = A * \text{inv}(B)$ resp. $F = \text{inv}(E) * D$

$C = A'$ komplexkonjugering, $c_{ij} = \text{conj}(a_{ji})$

$C = A.'$ transponat, $c_{ij} = a_{ji}$

$C = A^p$ A är kvadratisk, p är skalär $C = A * \dots * A$ (p ggr)

Logiska operationer

Värdet 0 betraktas som falskt (false), alla andra värden betraktas som sant (true). Resultatet sant lagras som 1.

< elementvis mindre än
<= elementvis mindre än eller lika med
> elementvis större än
>= elementvis större än eller lika med
== elementvis lika med
~= elementvis skilt från
& elementvis och
| elementvis eller
~ elementvis icke
&& genvägsoch
|| genvägseller

Prioritet

1 ^, .^, ', .'
2 +, -, ~ (med en operand, unära)
3 *, .*, /, ./, \, .\
4 +, - (binära)
5 :
6 ==, ~=, <, <=, >, >=
7 &
8 |
9 &&
10 ||

Ordningen kan förbigås med parenteser. Operatörer med samma prioritet utförs från vänster till höger, utom de unära som utförs från höger till vänster.

```
v = ~17
ans = 0
a = [1];
b = 1>3 & a(2);
b = 1>3 && a(2);
c = 1>3 & 6==10+--7 | 5/2/2\10
```

Skapa tabell

I en tabelliterall räknar man, mellan »[» och »]», upp elementen med »;» eller radbyte mellan raderna och »» eller blanktecken mellan kolumnerna. Om tabellen ska ha dimension 3 eller mer skapar man ett lager i taget.

Man kan också sammanfoga befintliga tabeller:

```
A = [1 2;3 4]; B = [5 6;7 8]; x = [9 10];
y = [11;12]; z = [13 14];
```

Tre sätt att skapa xnew som [9 10 0 15]:

- `xnew = x; xnew(3) = 0; xnew(4) = 15;`
- `xnew = [x 0 15];`
- `temp = [0 15]; xnew = [x temp];`

Indexering, omformning

Man kan nå enstaka element eller subtabeller med indexering. Elementen lagras i följd motsvarande att tidiga index varierar fortare än senare index.

Man kan skriva ut elementen i denna ordning med `A(1:numel(A))` eller `A(:)`. Man kan ange en subtabell som vänsterled i en tilldelning.

```
A = reshape(1:9,3,3)
A = [1 4 7
     2 5 8
     3 6 9]
A(:)'
ans = [1 2 3 4 5 6 7 8 9]
```

Skapa tabell forts.

Man kan lägga en ny rad eller en ny kolumn till A:

- `Anew = [A;z];`
- `Anew = [A;[13 14]];`
- `Anew = [A y];`
- `Anew = [A [11;12]];`

Indexering forts.

```
A(1:4:9) = 1;
A([3 5 7]) = [1 2 3]
A = [1 4 3
     2 2 8
     1 6 1]
A([1 3],[1 3]) = 0
A = [0 4 0
     2 2 8
     0 6 0]
```

Skapa speciella tabeller

`b:s:e` en radvektor med elementen b $b+s$ $b+2*s$ $b+3*s$... $b+n*s$ så att n är det största heltal som gör $b+n*s \leq e$

```
linspace(a,b,n)    a:(b-a)/(n-1):b
linspace(0,12,6)  [0 2.4 4.8 7.2 9.6 12]
linspace(a,b)     linspace(a,b,100)
ones(n), matris med storlek [n n] resp. [m n ...p] med ettor
ones(m, n, ... , p)
zeros(n), matris med storlek [n n] resp. [m n ...p] med nollor
zeros(m, n, ... , p)
eye(n), matris med storlek [n n] resp. [m n] med ettor i
eye(m, n)         diagonalen och nollor f.ö.
```

Staffan Romberger, CSC, KTH, 2008-11-13

Polynom

Matlab lagrar polynom som en koefficientvektor med högstgradskoefficienten först. Med $p(x)=x^4+5x^3-2x^2+7x-11$ kan vi beräkna:

```
p = [1 5 -2 7 -11]; polyval(p,1)
p = 0
val = polyval(p,[2 3 5 7 9])
val =    51    208   1224  4056 10096
```

Det finns många andra funktioner för att skapa speciella tabeller.

Staffan Romberger, CSC, KTH, 2008-11-13

Eval

Formler kan lagras som text. En aritmetisk följd $a, a+s, a+2*s, \dots, a+(n-1)*s$ kan vi summera med

```
f = 1:99; summa = sum(f)
summa = 4950
```

Summan är ju $(a+a+(n-1)*s)*n/2$ så vi lagrar formeln i variabeln `str`.

```
str = '(2*a+(n-1)*s)*n/2';
a = 1; s = 1; n = 99;
eval(str)
ans = 4950
```

Staffan Romberger, CSC, KTH, 2008-11-13

Datatyper

En datatyp kan vi i Matlab se som ett sätt att lagra data/tolka en variabels värde. I många programspråk anger programmeraren för varje variabel vilken sorts data (datatyp) den ska innehålla. I Matlab kan en variabel ha olika datatyp vid olika tillfällen.

Data (värden) lagras som bitföljder i datorns minne. Det finns ju oändligt många tal. Egentligen skulle man därför reservera oändligt mycket plats för varje variabel. Istället har man olika datatyper, med olika minnesutrymme, för olika behov.

```
int8    heltal    8 bitar    -128..127
int16   heltal    16 bitar    -32 768..32 767
int32   heltal    32 bitar    -2 147 483 648..2 147 483 647
single  flyttal   32 bitar    ung. 3,4E-38..3,4E38 med 7 siffror
double  flyttal   64 bitar    ung. 1,7E-308..1,7E308 med 15 siffror
char    tecken   16 bitar    teckenkod (räcker till Unicode)
```

Staffan Romberger, CSC, KTH, 2008-11-13

Datatyper forts.

`uint8` heltal utan tecken 8 bitar 0..255
`uint16` heltal utan tecken 16 bitar 0..65 535
`uint32` heltal utan tecken 32 bitar 0..4 294 967 295
`logical` inte egen datatyp

Numeriska data lagras normalt som `double`. Man kan själv välja datatyp.

```
d1 = 32; d2 = uint8(32); whos d1 d2
```

```
Name Size Bytes Class  
d1      1x1      8 double array  
d2      1x1      1 uint8 array
```

Resten av kursen

Programmeringsdelen med 3 laborationer. Villkorssatser, slingor, egna funktioner, diagram och grafiskt användargränssnitt ska vi behandla.

Numerikdelen handlar om metoder för att lösa ingenjörens beräkningsproblem.

Kursen avslutas med ett projekt som ger möjlighet att fördjupa delarna i kombination och få betygshöjning.

Titta ofta på kursens webbplats.