

1 Grundlegendes

Variablendefinitionen werden mittels Gleichheitszeichen umgesetzt, etwa

```
>> x = pi/2
x =
    1.5708e+00
>> sin(x)
ans =
    1
```

Sollen die Ausgaben unterdrückt werden, so wird ein Semikolon ans Ende der Eingabe gesetzt

```
>> x = pi/2;
>> sin(x)
ans =
    1
```

Arbeiten lässt sich mittels des Command-windows und direkten Ein- sowie Ausgaben oder mittels eines *.m-files. In einem *.m-file können viele Befehle aufgelistet werden, welche nacheinander abgearbeitet werden. Ausgeführt wird das *.m-file aus dem Command-window, hier werden auch mögliche Ausgaben bzw. Fehlermeldungen angezeigt. Weiter kann ein *.m-file aber auch als Routine (function) definiert werden, siehe später Abschnitt 4.

2 Vektoren und Matrizen

Definition von 1- und 2D-Feldern mittels eckiger Klammern:

```
>> A = [1 2 3;4 5 6]
A =
    1 2 3
    4 5 6
>> b = [8;9]
b =
    8
    9
>> c = [pi exp(1)]
c =
    3.1416 2.7183
```

Multiplikation von Vektoren und Matrizen mittels *. Stimmen die Dimensionen nicht, so wird gemeckert

```
>> c*b
ans =
    49.5973
>> b*c
ans =
    25.1327 21.7463
    28.2743 24.4645
>> A*b
```

Error using *

Incorrect dimensions for matrix multiplication. Check that the number of columns in the first matrix matches the number of rows in the second matrix. To perform elementwise multiplication, use '.*'.

Transponieren mittels ':

```
>> A'  
ans =  
    1 4  
    2 5  
    3 6  
>> A'*b  
ans =  
    44  
    61  
    78
```

3 Einfache m-files

Werden mehrere Befehle (mglw. noch in Entwicklung befindlich) nacheinander abgearbeitet, so lassen sich diese in einem m-file zusammenfassen. Dieses wird ausgeführt und die darin befindlichen Befehle werden abgearbeitet.

Ein einfaches Beispiel:

```
n = 50;  
h = 1/(n+1);  
  
A = zeros(n,n);  
for i=1:n  
    for j=1:n  
        if i == j  
            A(i,j) = -2;  
        elseif abs(i-j) == 1  
            A(i,j) = 1;  
        end  
    end  
end  
A = A/h^2;  
b = ones(n,1);  
x = A\b;  
  
plot([0:h:1],[0; x; 0])
```

Oder in Kurzform:

```

n = 50;
h = 1/(n+1);

B = 1/h^2*(-2*diag(ones(n,1))+diag(ones(n-1,1),1)+diag(ones(n-1,1),-1));
b = ones(n,1);
x = B\b;

plot([0:h:1],[0; x; 0])

```

4 Funktionen

Wird eine komplexe Prozedur nötig, so bieten sich *.m-files an. Das *.m-file trägt den Namen der Funktion und die erste Zeile lautet

„function RUECKGABE(N)=FUNKTIONSNAME(EINGABEWERT(E))“.

Ein Beispiel:

```

function fac = fakultaet(n)
% Berechnet n!
% Genutzt wird eine rekursive Funktion
if n==1
    fac = 1;
else
    fac = n*fakultaet(n-1);
end

```

Kommentare werden mittels %-Teichen eingefügt. Aufgerufen wird die Prozedur im Command window

```

>> fakultaet(3)
ans =
    6
>> fakultaet(6)
ans =
   720

```

Eine andere Implementierung:

```

function fac = fakultaet2(n)
% Berechnet n!
% Genutzt wird eine for-Schleife

```

```

fac = 1;
for i=2:n
    fac = fac*i;
end

```

If- und for-Schleifen sind damit auch erklärt. Bleiben noch while-Schleifen:

```

function fac = fakultaet3(n)
% Berechnet n!
% Genutzt wird eine while-Schleife
fac = n;
i = n-1;
while i>1
    fac = fac*i;
    i = i-1;
end

```

5 Grafische Darstellungen

Grafische Darstellungen werden mittels Vektoren als Polygonzüge umgesetzt

```

>> x = linspace(0,2*pi,100);
>> plot(x,f(x))

```

Dabei wurde zunächst ein Vektor x definiert, mit 100 äquidistanten Einträgen zwischen 0 und 2π . Geplottet wird der Polygonzug der x -Werte und der Werte $f(x)$. Nützliche Optionen sind

```

>> plot(x,f(x),'-.')
>> plot(x,f(x),'x')
>> plot(x,f(x),'-x')
>> plot(x,f(x),'linewidth',1)

```

Zwei oder mehr grafische Darstellung werden durch Kommata getrennt

```

>> plot(x,f(x),x,cos(x),x,1./(x+1),'linewidth',1)

```

6 Lösung von Gleichungssystemen

Zur Lösung von Gleichungssystemen lässt sich der Backslash nutzen

```
>> A = [1 2 3;4 5 6;7 8 1];
>> b = [1 3 9]';
>> x = A\b
x =
    -0.1667
     1.3333
    -0.5000
```

Lineare Ausgleichprobleme lassen sich ebenfalls mittels Backslash lösen:

```
>> A = [1 2;3 4;5 6]
A =
     1 2
     3 4
     5 6
>> b = [1 3 9]';
>> x = A\b
x =
     3.6667
    -1.6667
>> A*x-b
ans =
    -0.6667
     1.3333
    -0.6667
>> norm(ans,2)
ans =
     1.6330
```