

MATLAB Tutorial

Andreas Hildebrand, Holger Brandsmeier

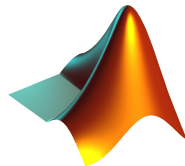
ETH Zürich, Seminar for Applied Mathematics

February 23, 2012

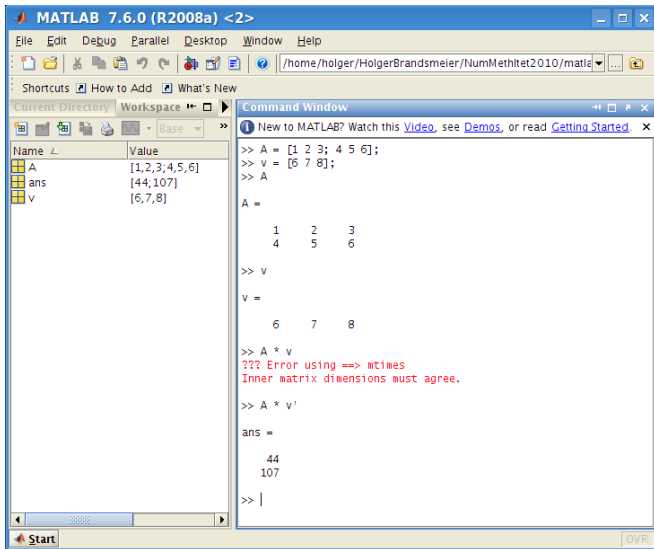


MATLAB

- MATLAB = Matrix Laboratory
- MATLAB auf Poolrechnern vorinstalliert
- MATLAB remote über SSH (Kurswebsite)
- MATLAB auf Heim PC (Kurswebsite)
Support: www.id.ethz.ch → Helpdesk
- + viele vordefinierte Lineare Algebra Befehle
- + einfach zu benutzen
- + stark verbreitet
 - langsamer als z.B. C++
 - Stolperfallen
- Opensource Alternative: Octave



MATLAB Interface



Rechnen mit Zahlen

```
>> a = 7;
>> b = 5;
>> a + b
ans =
    12
>> c = a * b
c =
    35
>> c = a - b;
>> c = a / b, d = a - b,
c =
    1.4000
d =
     2
```

- Intuitive Rechenoperationen
- Ein ';' am Ende der Zeile unterdrückt die Ausgabe
- Default ', ' am Ende der Zeile, Befehlsende uns Ausgabe

Rechnen mit Vektoren / Arrays

```
>> v1 = [1, 2, 3, 4];
>> v2 = [5 6 7 8];
>> v1 + v2
ans =
     6     8    10    12
>> v1 - v2
ans =
    -4    -4    -4    -4
>> v1 .* v2
ans =
     5    12    21    32
>> v1 ./ v2
ans =
 0.2000  0.3333  0.4286  0.5000
```

- anlegen zweier Zeilenvektoren v_1 , v_2 mit 4 Elementen
- Komponenten mit ' ' oder ', ' trennen
- Vektoraddition und Vektorsubtraktion (komponentenweise)
- komponentenweise Multiplikation '.*'
- komponentenweise Division './'

Rechnen mit Vektoren / Arrays

```
>> v1 = [1, 2, 3, 4];  
>> v2 = [5 6 7 8];  
>> v3 = [4; -1; 2; 1]; % Spaltenvektor  
>> v3(2) % Elementzugriff  
ans =  
    -1  
>> v3(2) = 3; % Elementzuweisung  
>> v1 * v3 % Skalarprodukt  
ans =  
    20  
>> v1 * v2' % Skalarprodukt  
ans =  
    70  
>> sqrt(v1) % Komponentenweise sqrt()  
ans = % siehe auch sin, cos, tan, exp  
    1.0000    1.4142    1.7321    2.0000
```

Einfache Fixpunktiteration

$$x_{k+1} = f(x_k) = \sqrt{x_k} + 1,$$

$$x_0 = 1$$

```
>> x = 1
```

```
x =
```

```
1
```

```
>> x = sqrt(x) + 1
```

```
x =
```

```
2
```

```
>> x = sqrt(x) + 1
```

```
x =
```

```
2.4142
```

```
>> x = sqrt(x) + 1
```

```
x =
```

```
2.5538
```

```
>> x = sqrt(x) + 1
```

```
x =
```

```
2.5981
```

Intervalle

```
>> it1 = 1:5           % Anfang:Ende
it1 =
     1     2     3     4     5
>> it2 = 1:2:5         % Anfang:Schrittweite:Ende
it2 =
     1     3     5
>> x = linspace(1, 5, 3)
x =           % Anfang, Ende, Anzahl Punkte
     1     3     5
>> it = [it1, 7]       % Intervalle zusammenfügen
it =
     1     2     3     4     5     7
```


Einfache Fixpunktiteration – for Schleife

$$x_{k+1} = f(x_k) = \sqrt{x_k} + 1, \quad x_0 = 1$$

```
>> x = 1;  
>> for i=1:5  
    x = sqrt(x) + 1  
end
```

```
x =
```

```
    2
```

```
x =
```

```
    2.4142
```

```
x =
```

```
    2.5538
```

```
x =
```

```
    2.5981
```

```
x =
```

```
    2.6118
```

Einfache Fixpunktiteration – eigenes Skript

```
>> format long          % genauere Zahlenausgabe
>> edit fixpunktIt.m    % jetzt Skript editieren
>> fixpunktIt           % Skript aufrufen

x =

    2

x =

    2.414213562373095

x =

    2.553773974030038
```

Datei: fixpunktIt.m:

```
x = 1;
for i=1:5
    x = sqrt(x) + 1
end
```

Einfache Fixpunktiteration – Endlosschleife

```
>> format long           % genauere Zahlenausgabe
>> edit fixpunktIt.m     % jetzt Skript editieren (s.u.)
>> fixpunktIt            % Skript aufrufen
x = 2
x = 2.414213562373095
[...]
```

Zum Abbrechen Strg + C drücken

```
x =
    2.618033988749895
??? Operation terminated by user during ==> fixpunktIt at 4
```

Datei: fixpunktIt.m

```
x = 1;
for i=1:Inf           % <-- Inf ist infinity
    x = sqrt(x) + 1
end
```

If-then-else Anweisung

$$y = \begin{cases} 2 & \text{falls } x > 1 \\ 7 & \text{sonst.} \end{cases}$$

in MATLAB:

```
>> x = 3;  
>> if x > 1           % Bedingung  
    y = 2             % Then-Anweisung  
else  
    y = 7             % Else-Anweisung  
end  
y =  
    2
```

MATLAB Hilfe: doc <Befehl> oder help <Befehl>, z.B.

```
>> doc if
```

Einfache Fixpunktiteration – Abbruchbedingung

Abbrechen, wenn $|x_{k-1} - x_k| < 10^{-6}$

Datei: fixpunktIt.m

```
x = 1;
tol = 1e-6;
for i=1:Inf
    xOld = x;
    x = sqrt(x) + 1;
    if abs(x - xOld) < tol
        break;                % aus der Schleife springen
    end
end
x
```

Aufruf:

```
>> fixpunktIt                % Skript aufrufen
x =
    2.618033829661219
```

Einfache Fixpunktiteration – alle Zwischenwerte

Datei: fixpunktIt.m

```
x(1) = 1; tol = 1e-6;
for i=1:Inf
    x(i+1) = sqrt(x(i)) + 1;
    if abs(x(i+1) - x(i)) < tol
        break;
    end
end
x
```

Aufruf:

```
>> fixpunktIt           % Skript aufrufen
x =
    1.0000    2.0000    2.4142    2.5538    [...]
>> length(x) - 1 % Anzahl Iterationsschritte
ans = 14
```

Funktionspointer (Function Handles)

```
>> f = @sin; % funktioniert auch bei eigenen
>> f(pi) % Funktionen/Skripten
ans =
    1.2246e-16
>> sin(pi)
ans =
    1.2246e-16
>> f = @(x) x^2 + x; % Handle auf "anonyme" Funktion
>> [f(0), f(1), f(2)]
ans =
     0     2     6
>> x=0:2; x.^2+x
ans =
     0     2     6
```

Einfache Fixpunktiteration – Funktionspointer

Datei: fixpunktlt.m

```
f = @(x) sqrt(x) + 1;    % <-----  
x(1) = 1; tol = 1e-6;  
for i=1:Inf  
    x(i+1) = f(x(i));    % <-----  
    if abs(x(i+1) - x(i)) < tol  
        break;  
    end  
end  
x
```

MATLAB Hilfe: `doc function_handle`

MATLAB Funktionen

$$f(x) = ax^2 + bx + c$$

$$f'(x) = 2ax + b$$

Datei: myfunc.m

```
function [f, df] = myfunc(x, a, b, c)
    f = a * x.^2 + b * x + c;    % <-- x.^2 komponentenweises
    df = 2*a*x + b;             %      quadrieren
end                             % optional
```

MATLAB Hilfe: doc function

Aufruf:

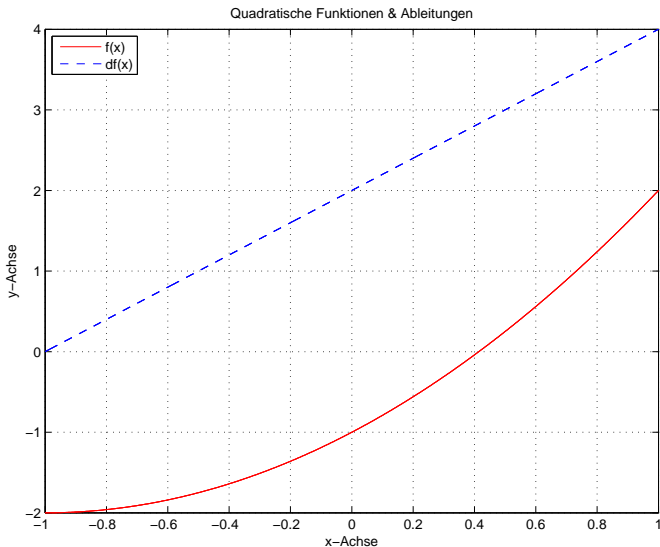
```
>> [f, df] = myfunc(x, 1, 2, -1);
```

MATLAB Plots

```
>> x = linspace(-1, 1, 500);  
>> [f, df] = myfunc(x, 1, 2, -1);  
>> plot(x, f, '-r')           % '-' durchgezogene Linie,  
                               % 'r' rot  
>> hold on;                   % darueberzeichnen aktivieren  
>> plot(x, df, '--b')         % '--' gestrichelte Linie,  
                               % 'b' blau  
>> legend('f(x)', 'df(x)', 'Location', 'NorthWest')  
>> grid on  
>> xlabel('x-Achse')  
>> ylabel('y-Achse')  
>> title('Quadratische Funktionen & Ableitungen')  
>> print('-dpdf', 'myfunc.pdf'); % als pdf speichern
```

MATLAB Hilfe: doc plot

MATLAB Plots – Ergebnis



Einfache Fixpunktiteration – MATLAB Funktion

Datei: fixpunktIt.m

```
function x = fixpunktIt(f, x0, tol)
x(1) = x0;
for i=1:Inf
    x(i+1) = f(x(i));
    if abs(x(i+1) - x(i)) < tol
        break;
    end
end
```

Aufruf:

```
>> x0 = 1; tol = 1e-6; f = @(x) sqrt(x) + 1;
>> x = fixpunktIt(f, x0, tol);
>> x
x =
    1.0000    2.0000    2.4142    [...]
```

Weiterführendes

- MATLAB-Tutorial von der Website ausführen (insbesondere Matrizen Teil)
- Die MATLAB-Hilfe ist dein Freund
- Komplizierte Programme immer in kleinen Schritten schreiben
- Zum Debuggen Funktionen in Skripte umwandeln oder Debugger benutzen:

```
x0 = 1;  
tol = 1e-6;  
f = @(x) sqrt(x) + 1;  
%function x=fixpunktIt(f, x0, tol) % <---  
x(1) = x0;  
for i=1:Inf  
[...]
```