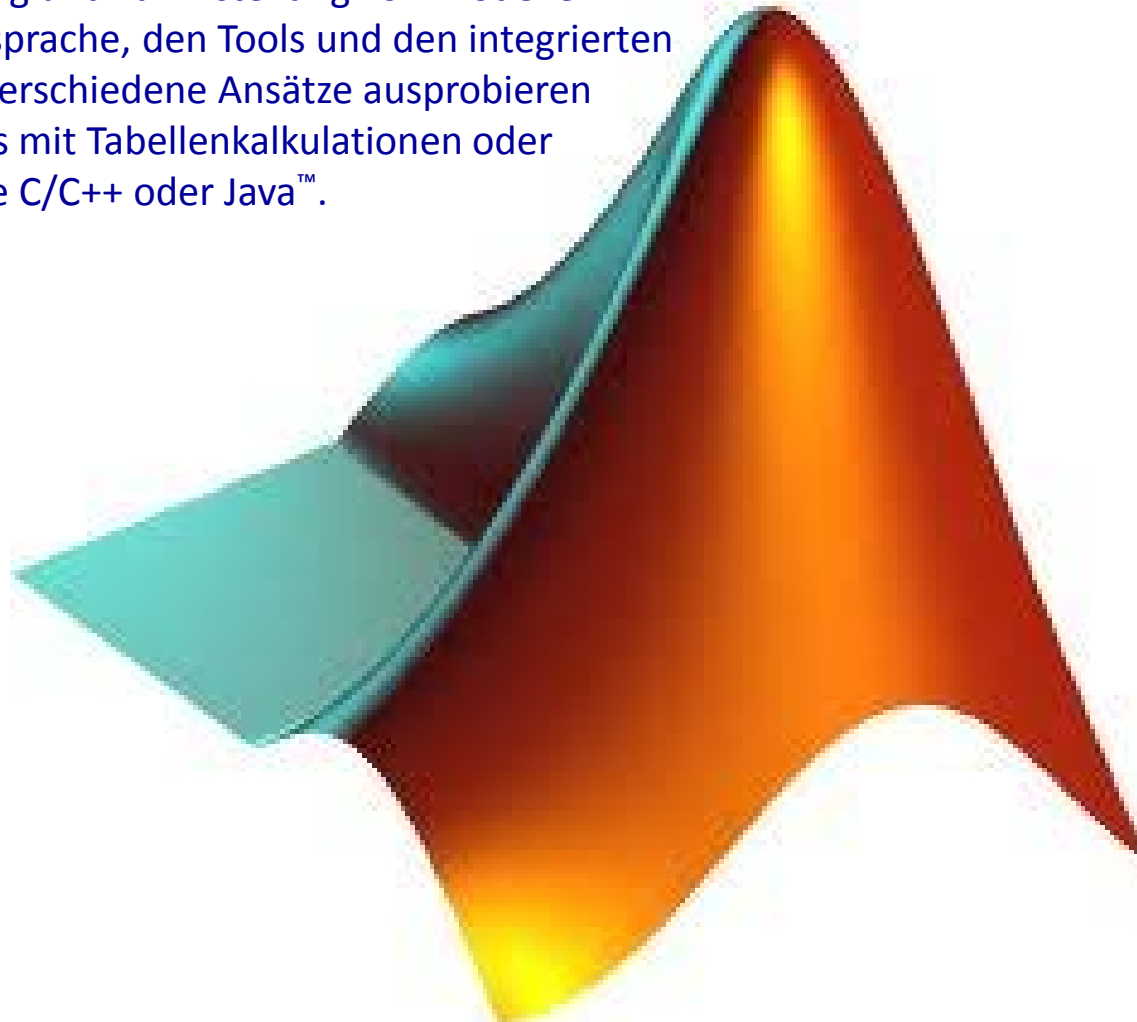


MATLAB eine Einführung

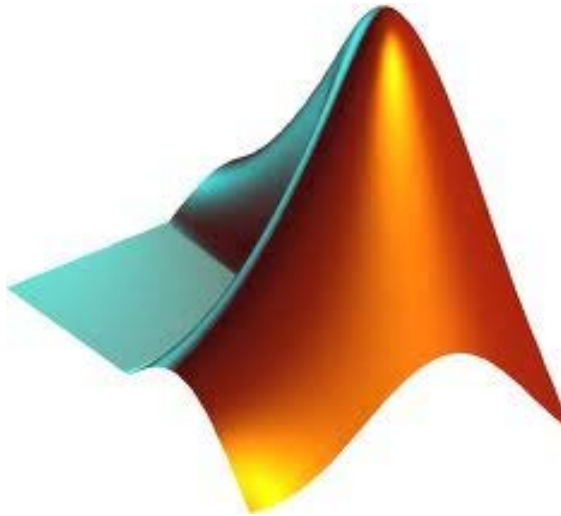
MATLAB® ist eine höhere Programmiersprache und interaktive Umgebung für numerische Berechnungen, Visualisierung und Programmierung. MATLAB dient zur Datenanalyse, Algorithmen-Entwicklung und zur Erstellung von Modellen und Anwendungen. Mit der Programmiersprache, den Tools und den integrierten mathematischen Funktionen können Sie verschiedene Ansätze ausprobieren und schneller zu einer Lösung gelangen als mit Tabellenkalkulationen oder herkömmlichen Programmiersprachen wie C/C++ oder Java™.



Karl-Heinz Gericke
WS 2013/14



Matlab Einführung



MATLAB kompakt [Taschenbuch, 4.Auflage], Wolfgang Schweizer

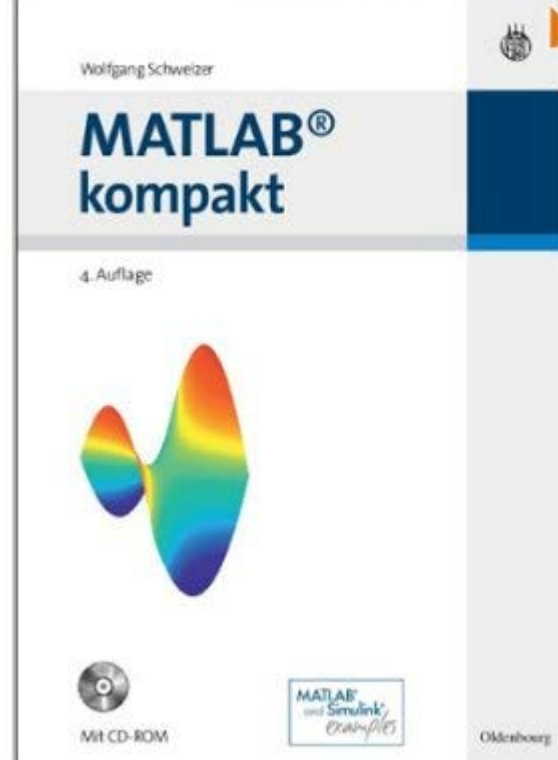
<http://www.pci.tu-bs.de/aggericke/Lehre/index.html>

<http://www.pci.tu-bs.de/aggericke/Matlab/index.html>

Homepage vom Produzenten:

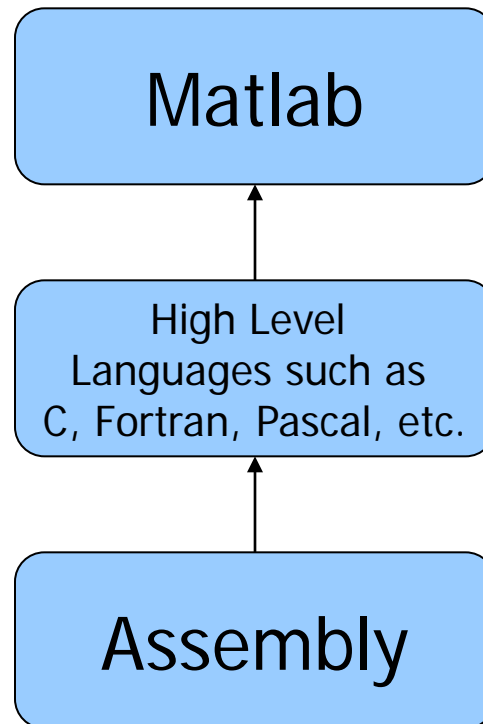
<http://www.mathworks.de/>

<http://www.mathworks.de/de/help/> gibt's auch im Matlab-Programm



Matlab eine Einführung

- Matlab is basically a **high level language** which has many specialized toolboxes for making things easier for us
- How high?



Matlab eine Einführung

- **MATLAB Plattform unabhängig**
 - UNIX
 - z.B. Remote via Putty
 - Windows (PC)
 - Mac

Matlab eine Einführung

- **Vorteile von Matlab**

- Interpreter und interaktive Befehlseingabe
 - Schnelles Implementieren von wissenschaftlichen Methoden
 - Compilierbar
- Hilfesystem (stark verbesserungswürdig für Neueinsteiger)

```
>> lookfor 'sum'
```

```
TRACE Sum of diagonal elements.
```

```
CUMSUM Cumulative sum of elements.
```

```
SUM Sum of elements.
```

```
...
```

Matlab eine Einführung

- **Vorteile**

- Interpreter und interaktive Befehlseingabe

- Schnelles Arbeiten zum Entwickeln von Programmen

```
>> help sum
```

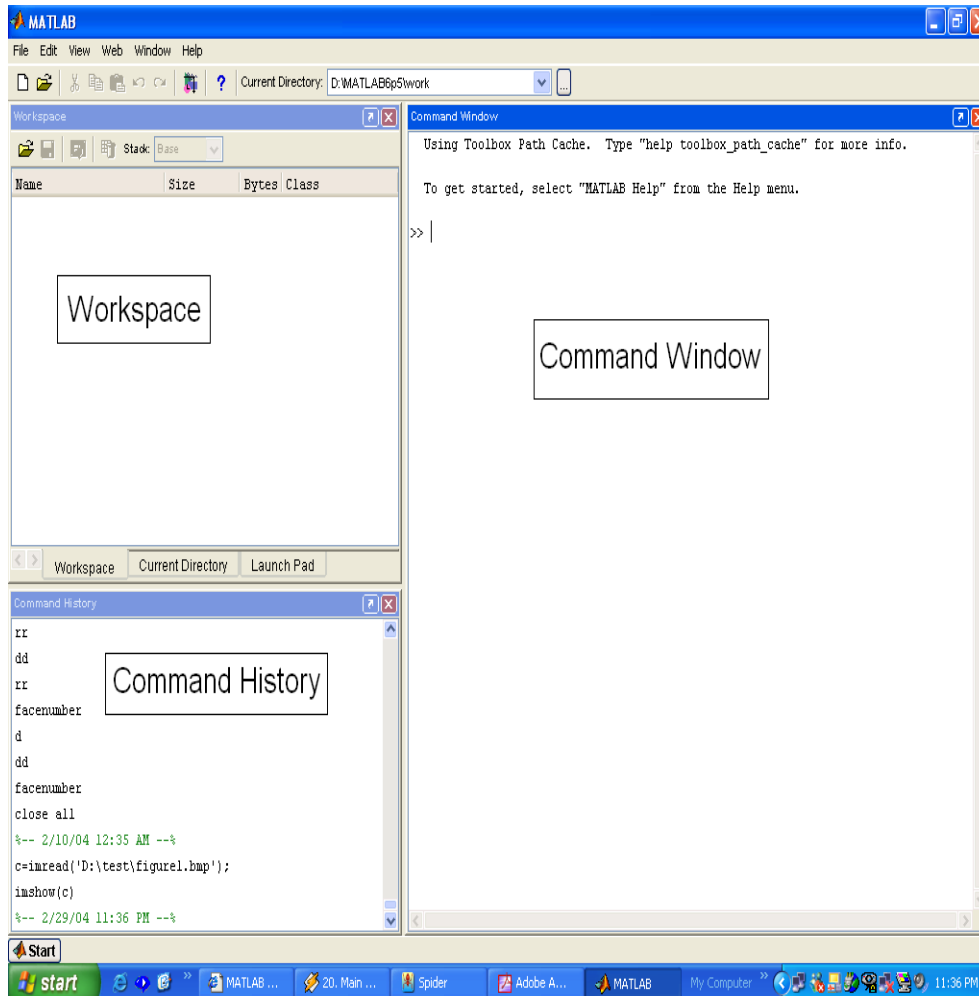
```
SUM Sum of elements.
```

```
S = SUM(X) is the sum of the elements of the  
vector X. If X is a matrix, S is a row  
vector with the sum over each
```

```
...
```

Matlab eine Einführung

The MATLAB Environment



- MATLAB window components:

Workspace

- > Displays all the defined variables

Command Window

- > To execute commands in the MATLAB environment

Command History

- > Displays record of the commands used

File Editor Window

- > Define your functions

Matlab eine Einführung

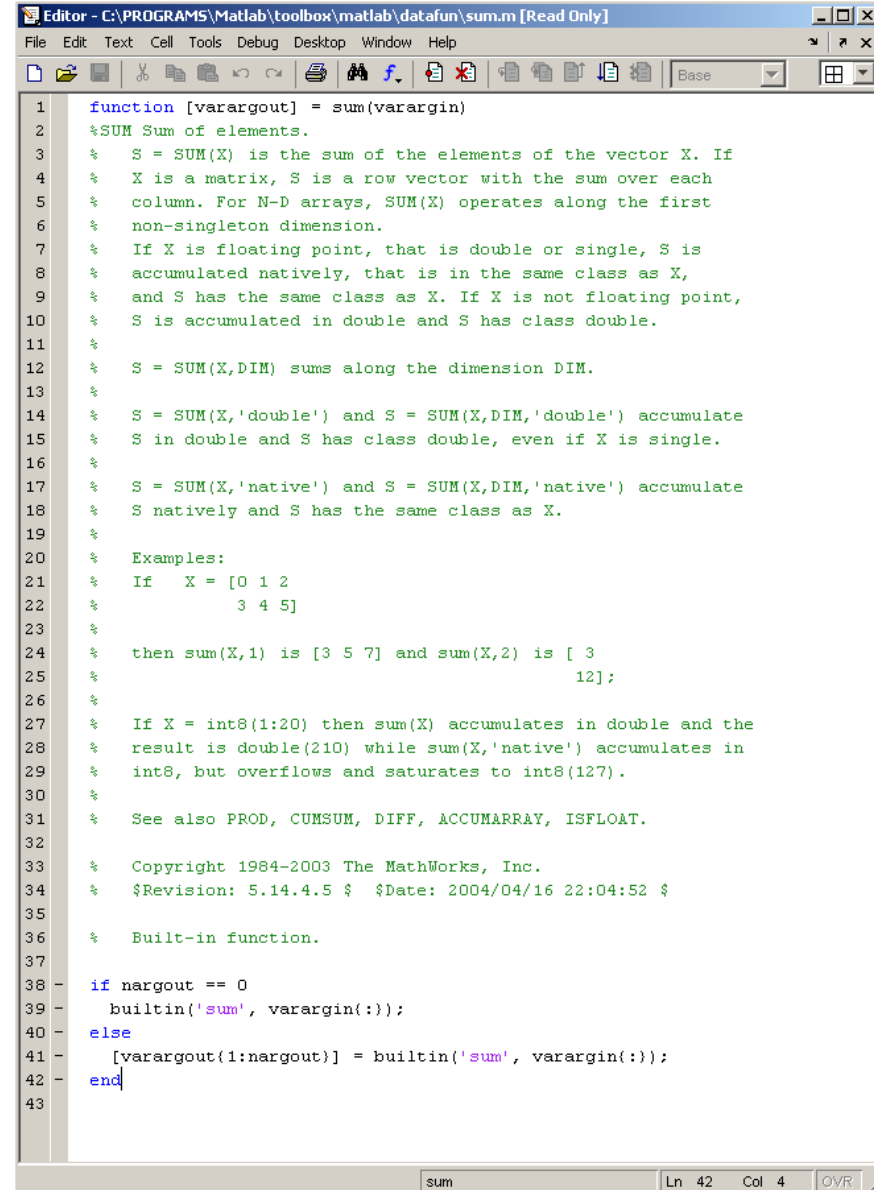
- **Vorteile**

- Modulorientiert / Offen

- Die meisten Matlabskripte sind einsehbar

```
>> edit sum
```

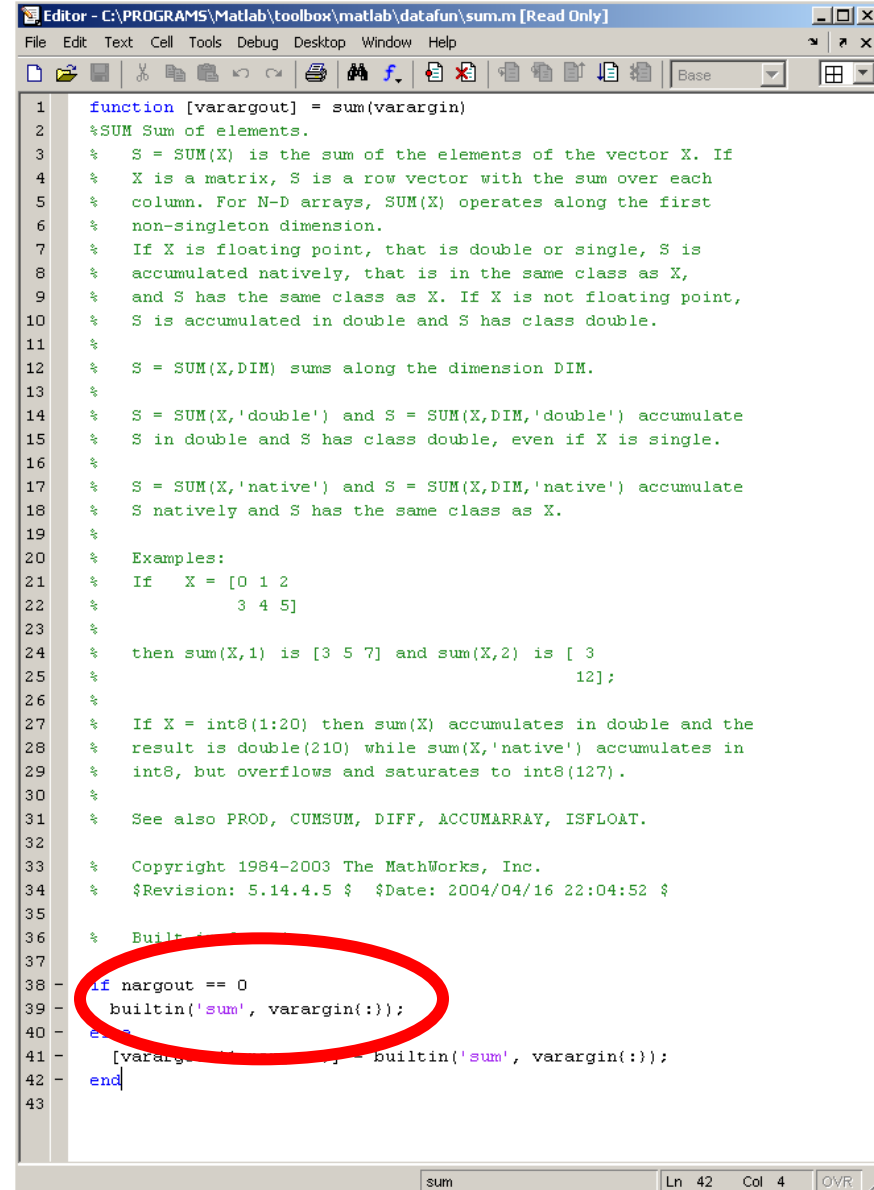
```
>> edit gradient
```



```
Editor - C:\PROGRAMS\Matlab\toolbox\matlab\datafun\sum.m [Read Only]
File Edit Text Cell Tools Debug Desktop Window Help
Base
1 function [varargout] = sum(varargin)
2 %SUM Sum of elements.
3 % S = SUM(X) is the sum of the elements of the vector X. If
4 % X is a matrix, S is a row vector with the sum over each
5 % column. For N-D arrays, SUM(X) operates along the first
6 % non-singleton dimension.
7 % If X is floating point, that is double or single, S is
8 % accumulated natively, that is in the same class as X,
9 % and S has the same class as X. If X is not floating point,
10 % S is accumulated in double and S has class double.
11 %
12 % S = SUM(X,DIM) sums along the dimension DIM.
13 %
14 % S = SUM(X,'double') and S = SUM(X,DIM,'double') accumulate
15 % S in double and S has class double, even if X is single.
16 %
17 % S = SUM(X,'native') and S = SUM(X,DIM,'native') accumulate
18 % S natively and S has the same class as X.
19 %
20 % Examples:
21 % If X = [0 1 2
22 %         3 4 5]
23 %
24 % then sum(X,1) is [3 5 7] and sum(X,2) is [ 3
25 %                                           12];
26 %
27 % If X = int8(1:20) then sum(X) accumulates in double and the
28 % result is double(210) while sum(X,'native') accumulates in
29 % int8, but overflows and saturates to int8(127).
30 %
31 % See also PROD, CUMSUM, DIFF, ACCUMARRAY, ISFLOAT.
32 %
33 % Copyright 1984-2003 The MathWorks, Inc.
34 % $Revision: 5.14.4.5 $ $Date: 2004/04/16 22:04:52 $
35 %
36 % Built-in function.
37
38 if nargin == 0
39     builtin('sum', varargin{:});
40 else
41     [varargout{1:nargout}] = builtin('sum', varargin{:});
42 end
43
sum Ln 42 Col 4 OVR
```


Matlab eine Einführung

- Vorteile
 - Erweiterbar
 - C oder FORTRAN Code durch mex Bibliotheken

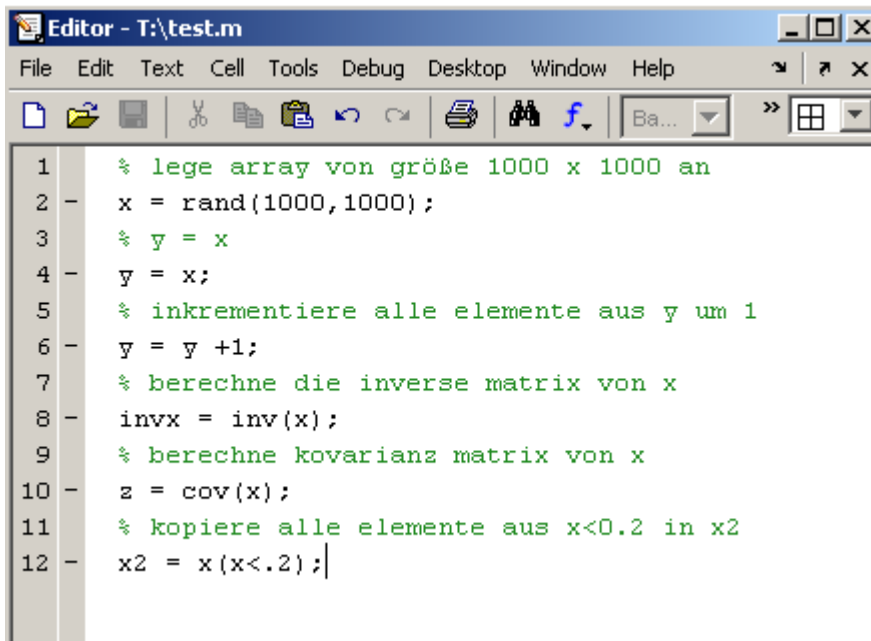


```
Editor - C:\PROGRAMS\Matlab\toolbox\matlab\datafun\sum.m [Read Only]
File Edit Text Cell Tools Debug Desktop Window Help
Base

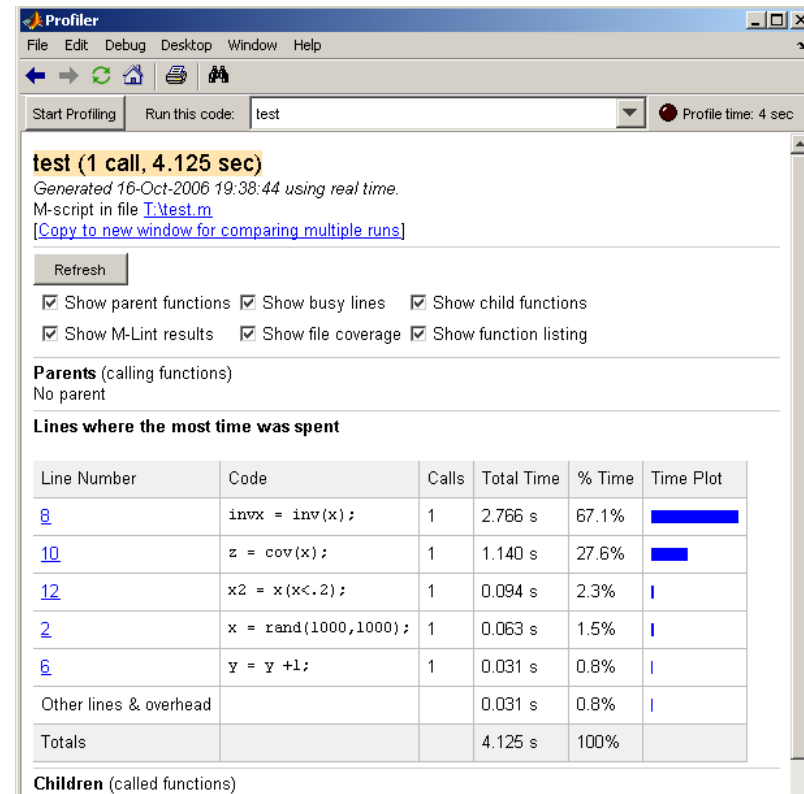
1 function [varargout] = sum(varargin)
2 %SUM Sum of elements.
3 % S = SUM(X) is the sum of the elements of the vector X. If
4 % X is a matrix, S is a row vector with the sum over each
5 % column. For N-D arrays, SUM(X) operates along the first
6 % non-singleton dimension.
7 % If X is floating point, that is double or single, S is
8 % accumulated natively, that is in the same class as X,
9 % and S has the same class as X. If X is not floating point,
10 % S is accumulated in double and S has class double.
11 %
12 % S = SUM(X,DIM) sums along the dimension DIM.
13 %
14 % S = SUM(X,'double') and S = SUM(X,DIM,'double') accumulate
15 % S in double and S has class double, even if X is single.
16 %
17 % S = SUM(X,'native') and S = SUM(X,DIM,'native') accumulate
18 % S natively and S has the same class as X.
19 %
20 % Examples:
21 % If X = [0 1 2
22 %         3 4 5]
23 %
24 % then sum(X,1) is [3 5 7] and sum(X,2) is [ 3
25 %                                           12];
26 %
27 % If X = int8(1:20) then sum(X) accumulates in double and the
28 % result is double(210) while sum(X,'native') accumulates in
29 % int8, but overflows and saturates to int8(127).
30 %
31 % See also PROD, CUMSUM, DIFF, ACCUMARRAY, ISFLOAT.
32 %
33 % Copyright 1984-2003 The MathWorks, Inc.
34 % $Revision: 5.14.4.5 $ $Date: 2004/04/16 22:04:52 $
35 %
36 % Built-in function
37
38 if nargin == 0
39     builtin('sum', varargin{:});
40 end
41 [varargout{1:nargout}] = builtin('sum', varargin{:});
42 end
43
```

Matlab eine Einführung

- Vorteile
 - Profiler ab Version 7
 - Hilft beim Auffinden von ressourcenverbrauchenden Programmteilen



```
1 % lege array von gröÙe 1000 x 1000 an
2 x = rand(1000,1000);
3 % y = x
4 y = x;
5 % inkrementiere alle elemente aus y um 1
6 y = y +1;
7 % berechne die inverse matrix von x
8 invx = inv(x);
9 % berechne kovarianz matrix von x
10 z = cov(x);
11 % kopiere alle elemente aus x<0.2 in x2
12 x2 = x(x<.2);
```



Profiler

File Edit Debug Desktop Window Help

Start Profiling Run this code: test Profile time: 4 sec

test (1 call, 4.125 sec)
Generated 16-Oct-2006 19:38:44 using real time.
M-script in file [T:\test.m](#)
[\[Copy to new window for comparing multiple runs\]](#)

Refresh

☒ Show parent functions ☒ Show busy lines ☒ Show child functions
☒ Show M-Lint results ☒ Show file coverage ☒ Show function listing

Parents (calling functions)
No parent

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
8	invx = inv(x);	1	2.766 s	67.1%	<div></div>
10	z = cov(x);	1	1.140 s	27.6%	<div></div>
12	x2 = x(x<.2);	1	0.094 s	2.3%	<div></div>
2	x = rand(1000,1000);	1	0.063 s	1.5%	<div></div>
6	y = y +1;	1	0.031 s	0.8%	<div></div>
Other lines & overhead			0.031 s	0.8%	<div></div>
Totals			4.125 s	100%	

Children (called functions)

Matlab eine Einführung

- **Vorteile**

- Umfangreiche Softwarepakete vorhanden

- z.B.Toolboxes

- Image Processing Toolbox
 - Curve Fitting Toolbox
 - Statistics Toolbox
 - Financial Toolbox
 - Neural Network Toolbox
 -



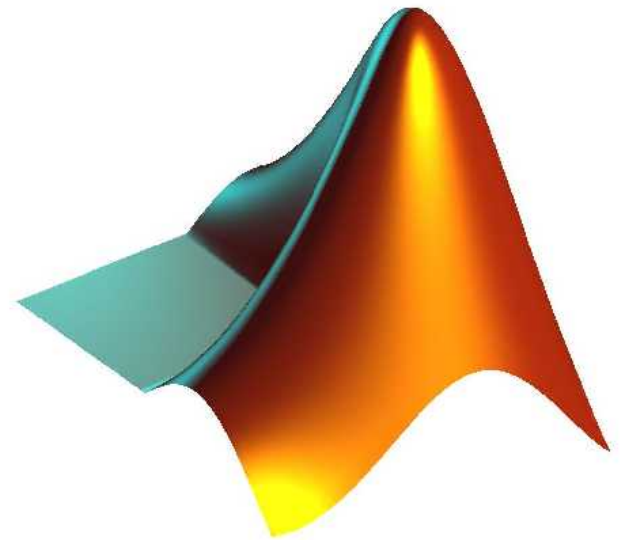
<http://www.mathworks.de/matlabcentral/fileexchange>

- www.mathworks.com/matlabcentral

- Hier findet ihr alles was ihr sonst nicht findet... :)

MATLAB Demos

- Demonstrations are invaluable since they give an indication of the MATLAB capabilities.
- A comprehensive set are available by typing the command `>> demo` in MATLAB prompt.



Matlab eine Einführung

- **Besonderheiten**

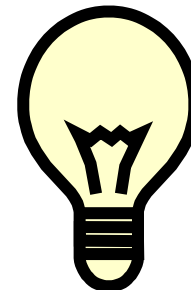
- JAVA liegt häufig unter Matlab
- Effiziente Programme durch Vektorisierung

```
>> y=sqrt(1:5);
```

statt

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

„Umdenken“



Matlab eine Einführung

- **Besonderheiten**

- Fast **Alles** ist erlaubt!

- Keine Variablendeklaration nötig
 - Variablenneubelegungen (Typänderung)
 - Funktionen mit gleichen Ein- und Ausgabeparametern

```
>> x=2  
>> x='super'
```

```
function x = myfun(x)  
    ...  
    x = x*2;
```

Matlab eine Einführung

- **Besonderheiten**

- Fast **Alles** ist erlaubt!

- Sogar Build-In Matlab Funktionen können als Variablen genutzt werden

=> **ACHTUNG** tötet eingebaute Funktionen!!!

```
>> sum = 1:5;
>> sum([0 0 1 1 0])
      3 4          % intuitiv erwartet 0+0+1+1+0=2
>> who           % zeigt alle Variablen im Speicher an
ans sum
>> tmp = sum      % Sichern der Variablen sum
>> clear sum      % Löschen der Variablen sum
>> sum([0 0 1 1 0]) % richtig!
      2
>> sum(tmp)       % richtig! (tmp ist [1,2,3,4,5])
      15
```

Matlab eine Einführung

- Handwerkzeug zum starten

- Arbeitsverzeichnis

```
>> pwd
```

```
>> ls
```

```
>> cd projekt1
```

- Variablen im Speicher

```
>> who
```

- Variablen löschen/speichern

```
>> clear y
```

```
>> save dateiname
```

```
>> load dateiname
```

- History

Cursor

- Editor

```
>> edit dateiname
```

