An Introduction to MatLab

Contents

1. Starting MatLab

Possibility 1

Double-click the MatLab shortcut icon on your Windows desktop.
>>cd 'work directory'
>>edit

Possibility 2

explorer -> work directory
create a new m-file
click a 'm-file' with right mouse button
open with, select program matlab.exe,
activate 'open always with this program'

double-click m-file (opens MatLab in the work directory)

>>edit

2. Workspace and m-Files

$>> a = 5$ (stores value 5 in variable a, prints result in workspace)

5

$>> a = 5;$ (stores value 5 in variable a)

$>> a$ (prints value of variable a in workspace)

5

>>

$>> t = 'hello \quad world';$

$>> t$

$'hello \quad world'$

$>> whos$

$variable \quad list$

$>> edit \quad test$ (opens editor with m-file *test*)

$>> test$ (calls function in m-file *test*)

example: our first MatLab program

workspace	*editor*
$>> edit \quad test$	$function \quad test$
	$'hello \quad world'$
	$F5$
$hello \quad world$	

Exercise

Open MatLab

Create a new m-file with name 'first'

Type the program lines $x = 1:20$

$plot(x, \sin(x * pi / 5))$

3. Help

Examples for workspace help	>> help function >> help if >> help for >> help switch >> help whos
Examples for MatLab help menu	Click with left mouse buttom: Menu – Contents – Using MatLab – Mathematics - …
help by	Contents Index Search

4. Vectors and Matrices

Comand	Output	Comment
$>> x(3) = a$	$\rightarrow$ $0 \quad 0 \quad a$	a, b, c, d are numbers (Integer, Real, π,e,i,…
$>> x = a:d:b$	$\rightarrow$ $a \quad a+d \quad a+2d \quad \ldots \quad b\text{-}d \quad b$	
$>> x = a:b$	$\rightarrow$ $a \quad a+1 \quad a+2 \quad \ldots$	
$>> x = [a \quad b \quad c \quad d]$	$\rightarrow$ $a \quad b \quad c \quad d$	$x(3) \rightarrow a+2d$
$>> x = linspace(a,b,N)$	$\rightarrow$ $a \quad a+\frac{b-a}{N-1} \quad \ldots \quad b-\frac{b-a}{N-1} \quad b$	n elements
$>> x = linspace(a,b)$	$\rightarrow$ $a \quad a+\frac{b-a}{99} \quad \ldots \quad b-\frac{b-a}{99} \quad b$	100 elementes

Comand		Output	Comment
$>> A = [1 \quad 2 \quad 3; \quad 4 \quad 5 \quad 6]$	$\rightarrow$	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix};$	$A(2,1) \rightarrow 4$
$>> A = [\underbrace{x}_{\text{Vektor}} ; \underbrace{y}_{\text{Vektor}}]$	$\rightarrow$	$\begin{pmatrix} x \\ y \end{pmatrix}$	$x, y \in R^N \rightarrow A \in R^2 \otimes R^N$
$>> A = [\underbrace{x}_{\text{Vektor}} \quad \underbrace{y}_{\text{Vektor}}]$	$\rightarrow$	$(x \quad y)$	$x \in R^M, y \in R^N \rightarrow A \in R^{M+N}$
$>> A = [x' \quad y']$	$\rightarrow$	$(x' \quad y') = \begin{pmatrix} x \\ y \end{pmatrix}'$	$x, y \in R^N \rightarrow A \in R^N \otimes R^2$

Scalar produkt:

Comand	Comment
$>> x\cdot y' \quad \left(=\sum_n x_n y_n\right)$	$R^N, R^N \to R$
$>> A\cdot x' \quad \left(=\sum_n A_{mn} x_n\right)$	$R^M \otimes R^N, R^N \to R^M$
$>> A\cdot B \quad \left(=\sum_n A_{mn} B_{nk}\right)$	$R^M \otimes R^N, R^N \otimes R^K \to R^M \otimes R^K$

Point produkt:

Comand	Comment	
$>> x.\cdot y$	$=\left(x_n y_n\right)$	$R^N, R^N \to R^N$
$>> A.\cdot B$	$=\left(A_{mn} B_{mn}\right)$	$R^M \otimes R^N, R^M \otimes R^N \to R^M \otimes R^N$

functions:

Comand	Comment	
$>> x.\hat{} 2$	$\left(=\left(x_n^2\right)\right)$	$R^N \to R^N$
$>> \sin(A)$	$\left(=\left(\sin\left(A_{mn}\right)\right)\right)$	$R^M \otimes R^N \to R^M \otimes R^N$

Exercise

a) Which function is evaluated by

$$f = prod(1:N)$$

(>>'help prod' shows the definition and examples for 'prod')

b) Study the results of the commands

$>> N = ones(6)$

$>> for \quad n = 1:6, \quad N(n,1:n) = 1:n; \quad end$

$>> N$

$>> f = prod(N,2)$

$>> plot(N(6,:), f', 'b')$

$>> xlabel('N')$

$>> ylabel('f')$

$>> title('Test \quad 1')$

$>> whos$

c) Use

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 14 \\ 11 \\ 11 \end{pmatrix}$$

ci) Calculate the determinant of A (Menu help: MatLab - Using MatLab – Matrices and linear algebra – Inverses …)

cii) Solve the linear system $Ax = B$ (Menu help: MatLab - Using MatLab – Matrices and linear algebra – Solving linear…)

d) Consider the commands

Comand	Output
>> x=1:5	$x = 1\ 2\ 3\ 4\ 5$
>> y=-2:2	$y = -2$ -1 0 1 2
>> z=y>0	z=0 0 0 1 1
>> x(z)	4 5

Explain the result in the last line!

(z is a logical array, x(z) maps to the elements of x, for which z is 1)

e) Compare the two codes

code 1		code 2
$N = 5e8$		$N = 5e8$
$x = 1:N;$		$x = 1:N;$
$y = 0;$		$y = x \cdot x'$
for n=1:N		
$y = y + x(n) \cdot x(n);$		
end		
y		

Consider specially the run time!

5. Objects

Example

$$>> hnd = text\left(\underbrace{0,0}_{\substack{x \text{ and } y \\ \text{Position}}} , \text{'hello world'} \right)$$

```
>> hnd=text(0,0,'hello world')

hnd =

  101.0002

>> get(hnd)
	Color = [0 0 0]
	EraseMode = normal
	Editing = off
	Extent = [-0.00125786 -0.0176056 0.0792453 0.0316901]
	FontAngle = normal
	FontName = Helvetica
	FontSize = [10]
	FontUnits = points
	FontWeight = normal
	HorizontalAlignment = left
	Position = [0 0 0]
	Rotation = [0]
	String = hello world
	Units = data
	Interpreter = tex
	VerticalAlignment = middle

	BeingDeleted = off
	ButtonDownFcn =
	Children = []
	Clipping = off
	CreateFcn =
	DeleteFcn =
	BusyAction = queue
	HandleVisibility = on
	HitTest = on
	Interruptible = on
	Parent = [100.001]
	Selected = off
	SelectionHighlight = on
	Tag =
	Type = text
	UIContextMenu = []
	UserData = []
	Visible = on
```

Get possibile properties: $>> set\left(hnd, \text{'HorizontalAlignment'}\right)$

[{left} | center | right]

Set possibile properties: $>> set\left(hnd, 'HorizontalAlignment', 'center', 'FontSize', 25\right)$

Exercise

a) Rotate the string 'hello world' by 180 degrees (Rotation is a property of the object 'text'.

b) Type in the command $>> h = plot\left(\begin{bmatrix}-1 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1 & 0\end{bmatrix}\right)$

bi) What different line styles are possible

bii) Change the line style and the line width

biii) change the y-data to [0 1 1.5]

6. Plots

$>>$help plot

$>> x = linspace(0,2*pi);$

$>> plot\left(x,\sin\left(x\right)\right)$

$>> plot\left(x,\sin\left(x\right),'b:'\right)$

$>> x = \left[0 \quad 1 \quad 1 \quad 0\right];$

$>> y = .5*\left[0 \quad 0 \quad 1 \quad 1\right];$

$>> fill\left(x,y,'y'\right)$ or

$>> Hnd = fill\left(x,y,'y'\right)$ (Hnd is 'handle' for object)

$>> get\left(Hnd\right)$ (shows object properties)

$>> set\left(Hnd,'EdgeColor','r','LineWidth',10\right)$ (changes object properties)

$$>> subplot\left(\underbrace{n}_{rows}, \underbrace{m}_{columns}, \underbrace{k}_{PlotNo}\right)$$

$>> hold \quad on$ (overdraw)

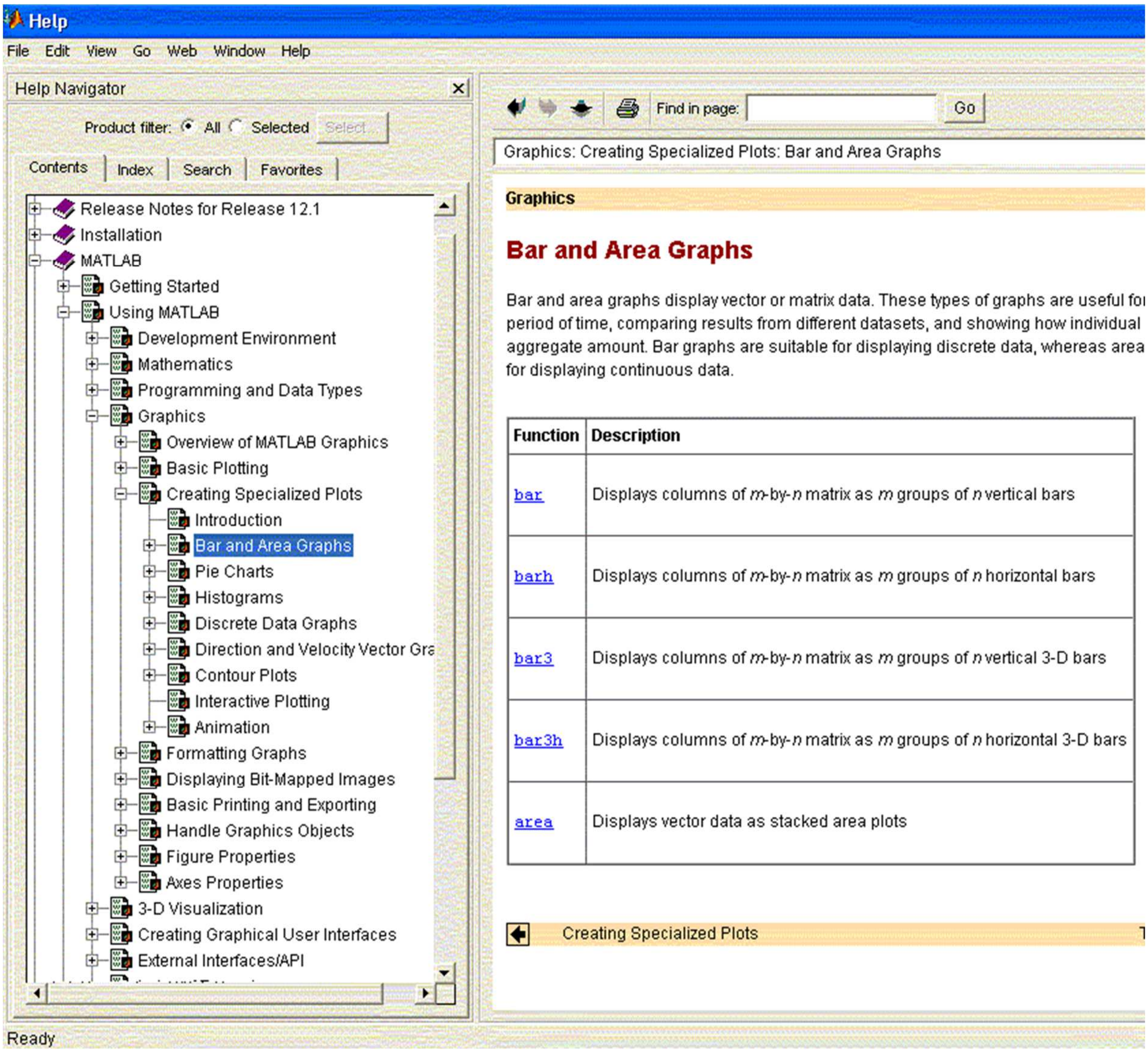
$>> hold \quad off$ (delete old objects when drawing a new one)

$>> try, \quad delete\left(1\right), \quad end$ (delete grafik window 1)

Basic Plotting Commands

MATLAB® provides a variety of functions for displaying vector data as line plots, as well as functions for annotating and printing these graphs. The following table summarizes the functions that produce basic line plots. These functions differ in the way they scale the plot's axes. Each accepts input in the form of vectors or matrices and automatically scales the axes to accommodate the data.

Function	Description
plot	Graph 2-D data with linear scales for both axes
plot3	Graph 3-D data with linear scales for both axes
loglog	Graph with logarithmic scales for both axes
semilogx	Graph with a logarithmic scale for the *x*-axis and a linear scale for the *y*-axis
semilogy	Graph with a logarithmic scale for the *y*-axis and a linear scale for the *x*-axis
plotyy	Graph with *y*-tick labels on the left and right side



Curve fitting

Data:

$$x = \begin{pmatrix} x_1 & x_2 & & x_N \end{pmatrix} \rightarrow y = \begin{pmatrix} y_1 & y_2 & \cdots & y_N \end{pmatrix}$$

fit:

$$Y = a + bX + cX.^2$$

get *a*, *b* and *c* with MatLab:\

$$y' = \underbrace{\begin{pmatrix} 1 & x' & (x.\hat{}\,2)' \end{pmatrix}}_{=M} \underbrace{\begin{pmatrix} a \\ b \\ c \end{pmatrix}}_{=A}$$

$$A = M \setminus y'$$

$$Y = M * A$$

Exercise

Create a new m-file 'Plot1'

Create the time series t=0..10 ,y=exp(at) with N elements;

Reproduce the plots of Plot1 in the appendix

If you don't know how to proceed, look into the program file (after the figure)

Use the desktop help and the help menu to get an understanding of the statements

7. Statistics

Matlab help page: 'Using Matlab – Mathematics – Data analysis and statistics'

Function Summary

A collection of functions provides basic column-oriented data analysis capabilities. These functions are located in the MATLAB `datafun` directory.

This section also gives you some hints about using row and column data, and provides some basic examples. This table lists the functions.

Basic Data Analysis Function Summary

Function	Description
`cumprod`	Cumulative product of elements.
`cumsum`	Cumulative sum of elements.
`cumtrapz`	Cumulative trapezoidal numerical integration.
`diff`	Difference function and approximate derivative.
`max`	Largest component.
`mean`	Average or mean value.
`median`	Median value.
`min`	Smallest component.
`prod`	Product of elements.
`sort`	Sort in ascending order.
`sortrows`	Sort rows in ascending order.
`std`	Standard deviation.
`sum`	Sum of elements.
`trapz`	Trapezoidal numerical integration.

Define an own statistic functions

Example: Lineare correlation, r-value

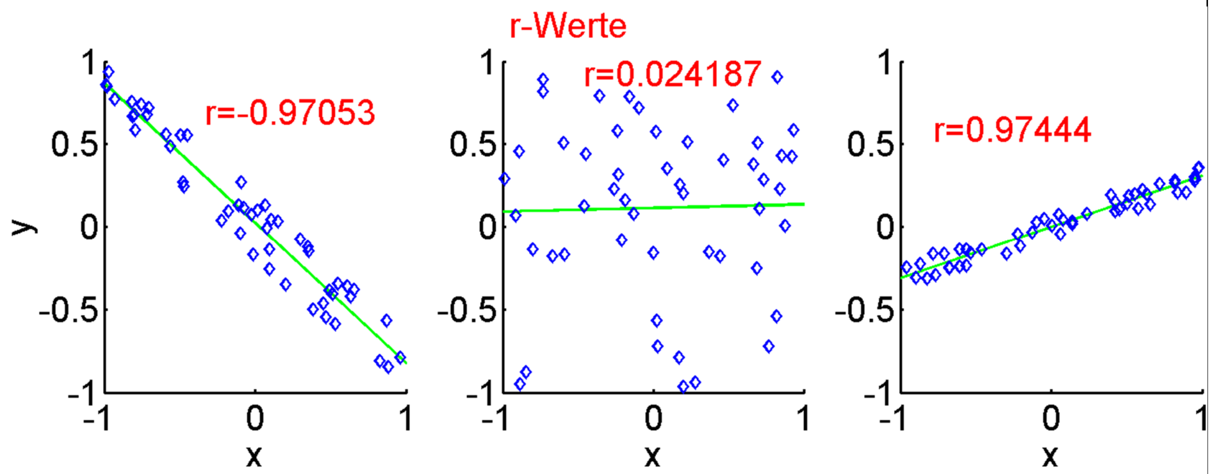
We consider two measured quantities

$$(x_i, y_i) \quad i \in \{1..N_{Dat}\} \tag{7-1}$$

The r-value is defind by

$$r = \frac{\sum_{i\in\{1..N_{Dat}\}} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i\in\{1..N_{Dat}\}} (x_i - \bar{x})^2}\sqrt{\sum_{i\in\{1..N_{Dat}\}} (y_i - \bar{y})^2}} \in [-1,1]. \tag{7-2}$$

It is a measure for the linear correlation between the two measured quantities.



Three examples for r-values. The closer the data pairs (x_i,y_i) lie on a line the closer is the r-value to ±1.

Fig. 7-1

MatLab formulation

$N = 10;$ $x = 1:N;$ $y = rand(1,N);$	(Generte a data)
$x1 = mean(x);$ $y1 = mean(y;)$ $r = sum((x - x1).*(y - y1)) / ...$ $(sqrt(sum((x - x1).\wedge 2)) * ...$ $sqrt(sum((y - y1).\wedge 2)));$	(Define r-value)

Exercise

Create a new m-file 'Stat1'

Reproduce the plots of Stat1 in the appendix

If you don't know how to proceed, look into the program file (after the figure)

Use the desktop help and the help menu to get an understanding of the statements

8. Import and Export of Data

Matlab help: 'Using matlab – Development environment – Importing and exporting -

Table 6-4: ASCII Data Export Function Feature Comparison

Function	Use With	Delimiters	Notes
csvwrite	Numeric data	Only commas	Primarily used with spreadsheet data. See also the binary format spreadsheet export functions
diary	Numeric data or cell array	Only spaces	Can be used for small arrays. Requires editing of data file to remove extraneous text.
dlmwrite	Numeric data	Any character	Easy to use, flexible.
fprintf	Alphabetic and numeric data	Any character	Part of low-level file I/O routines. Most flexible command but most difficult to use. Requires use of `fopen` to obtain file identifier and `fclose` after write.
save	Numeric data	Tabs or spaces	Easy to use; output values are high precision.

Table 6-2: ASCII Data Import Function Feature Comparison

Function	Data Type	Delimiters	Number of Return Values	Notes
csvread	Numeric data	Only commas	One	Primarily used with spreadsheet data. See also the binary format spreadsheet import functions.
dlmread	Numeric data	Any character	One	Flexible and easy to use.
fscanf	Alphabetic and numeric; however, both types returned in a single return variable	Any character	One	Part of low-level file I/O routines. Requires use of `fopen` to obtain file identifier and `fclose` after read.
load	Numeric data	Only spaces	One	Easy to use. Use the functional form of `load` to specify the name of the output variable.
textread	Alphabetic and numeric	Any character	Multiple return values.	Flexible, powerful, and easy to use. Use format string to specify conversions.

Write time series in readable format to file 'TimeSeries.txt'

```
N = 2^8;                                        (Generate time series)
x = 10*linspace(0,2*pi,N);
c = (2*mod(2:N+1,2)-1).*exp(-.5*(1:N));
y = c*sin((1:N)'*x);
---------------------------------------------------------------------
Y = fft(y)                                      (Fast Fourier transformation)
---------------------------------------------------------------------
f_hnd = fopen('TimeSeries.txt','w');            (Out to file)
for n = 1:length(x)
   fprintf(f_hnd,'%10.5f; %10.5f; %10.5f \n',...
       x(n),y(n),Y(n))
end
fclose(f_hnd)
```

Exercise

Write the Data in m-file 'Stat1' into TimeSeries.Txt

Create a new m-file 'ReadSeries.m'

Write a code to read the data from TimeSeries.Txt and display them in a diagram

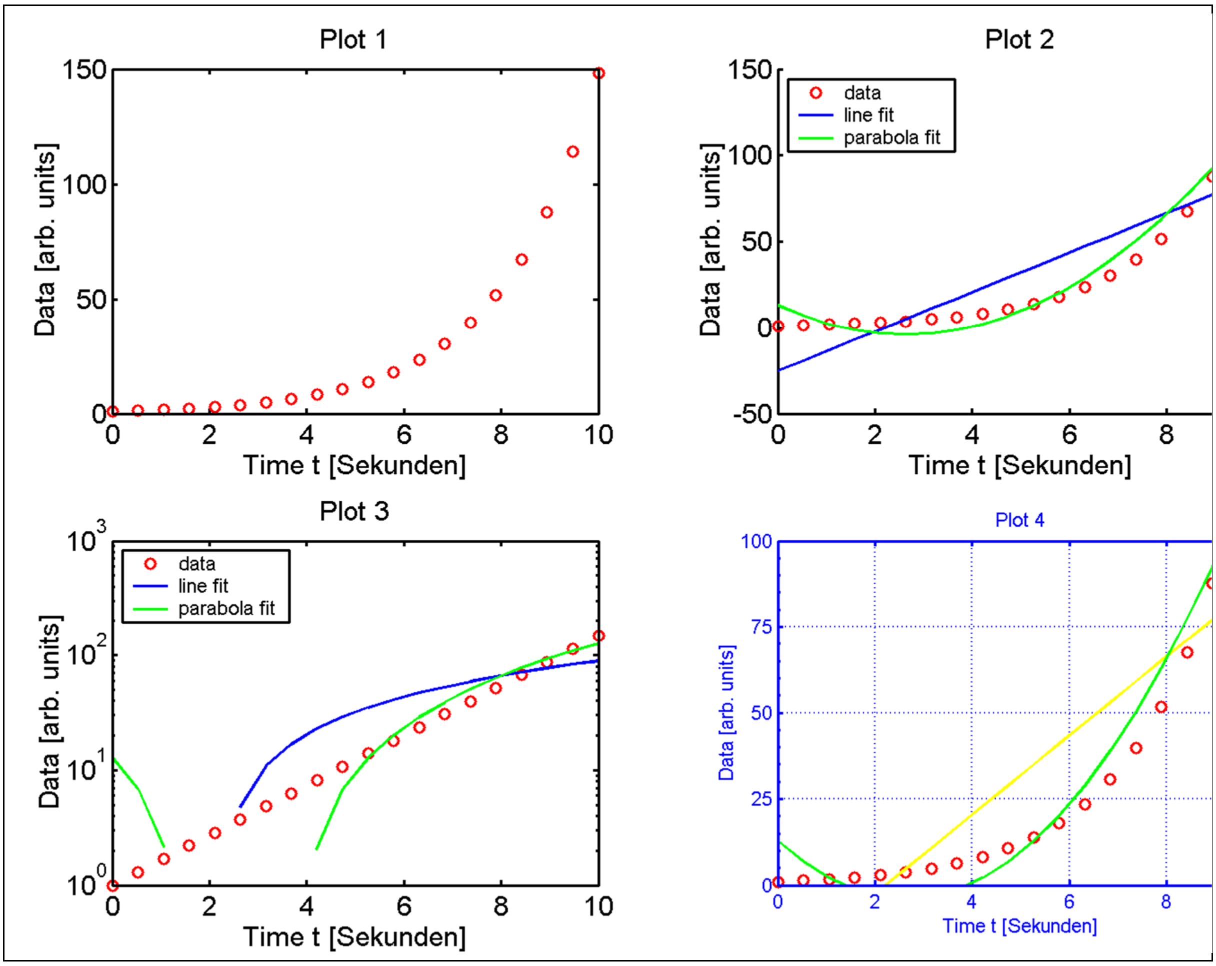
Further exercises

Create new m-files 'Plot2' and 'Plot3'

Reproduce the plots in the appendix

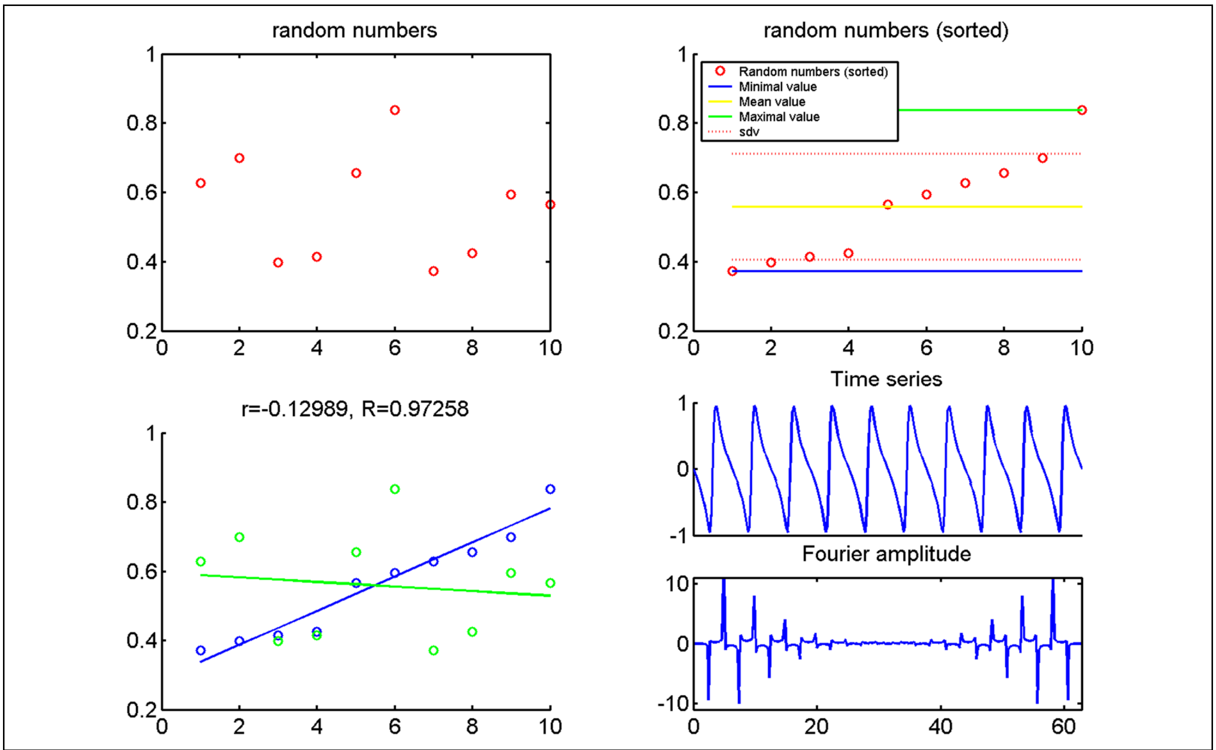
9. Appendix (Plots and Codes)

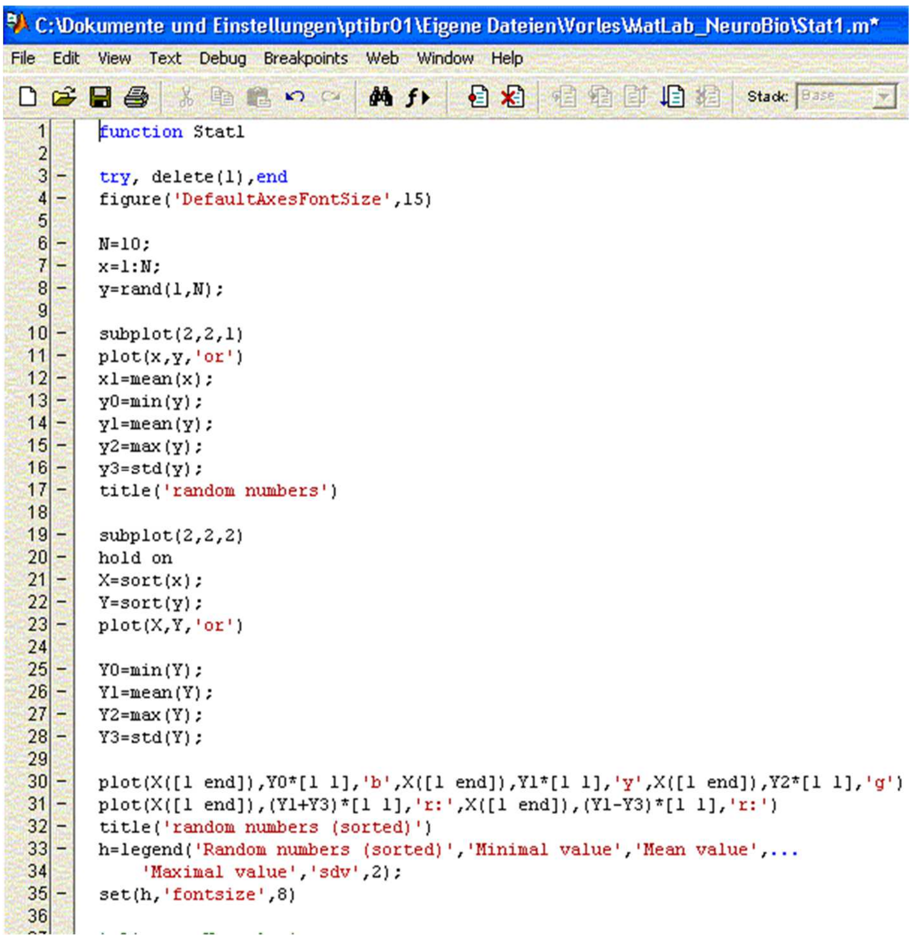
Plot 1



```
function Plot1

try, delete(1), end
figure('DefaultAxesFontSize',15)
%%%%%%%%%%% generate time series
N=20;
t=linspace(0,10,N);
%%%%%%%%%%% 'Data'
a=.5;
y=exp(a*t);
%%%%%%%%%%% 1 show data
subplot(2,2,1)
plot(t,y,'or');
xlabel('Time t [Seconds]')
ylabel('Data [arb. units]')
title('Plot 1')
%%%%%%%%%%% 2  curve fit by line and parabola
subplot(2,2,2)      %data
xlabel('Time t [Seconds]')
ylabel('Data [arb. units]')
title('Plot 2')
hold on
plot(t,y,'ro')
M1=[ones(N,1) t'];   %line
A1=M1\y';
y1=M1*A1;
plot(t,y1,'b')
M2=[ones(N,1) t' (t.^2)'];  %parabola
A2=M2\y';
y2=M2*A2;
plot(t,y2,'g')
h=legend('data','line fit','parabola fit',2);
set(h,'FontSize',10)
%%%%%%%%%%% 3 lograythmic presentation of data
subplot(2,2,3)
%plot(t,y,'ro',t,y1,'b',t,y2,'g')
plot(t,y,'ro',t,y1.*(y1>0),'b',t,y2.*(y2>0),'g')
get(gca)
set(gca,'yscale','log')
xlabel('Time t [Sekunden]')
ylabel('Data [arb. units]')
title('Plot 3')
h=legend('data','line fit','parabola fit',2);
set(h,'FontSize',10)
%set(gca,'ygrid','on')   %grids
%%%%%%%%%%%  manual axis design
subplot(2,2,4)
plot(t,y,'ro',t,y1,'y',t,y2,'g')
xlabel('Time t [Sekunden]','FontSize',10)
ylabel('Data [arb. units]','FontSize',10)
title('Plot 4','Color','b','FontSize',10)
set(gca,'ylim',[0 100])     %limits on y-axis
set(gca,'ytick',0:25:100)   %limits on y-axis
set(gca,'yminortick','on')   %limits on y-axis
set(gca,'xgrid','on','ygrid','on')   %grids
set(gca,'xcolor','b','ycolor','b')   %grids
set(gca,'FontSize',10,'Color',.5*[.95,.95,1])   %grids
%set(gca,'yminorgrid','on')   %grids
```

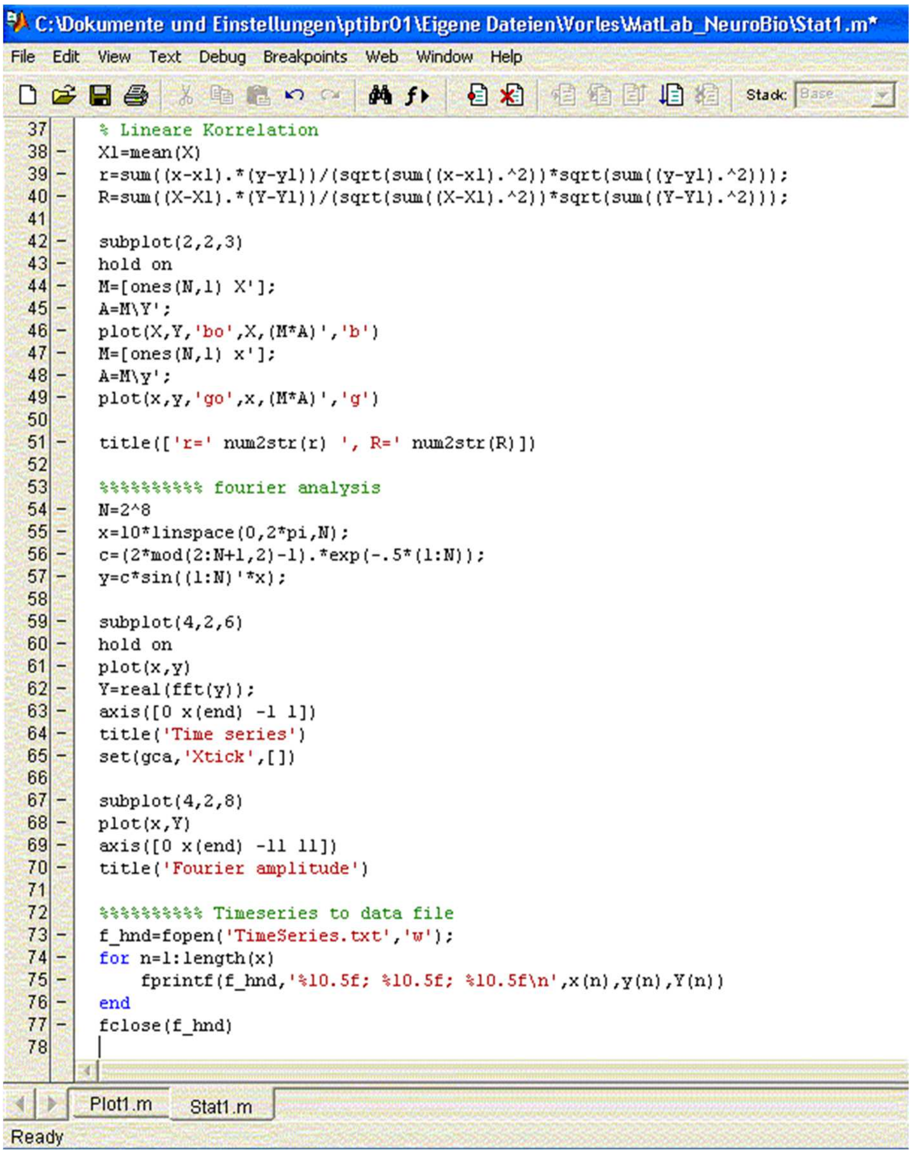
Statistics 1





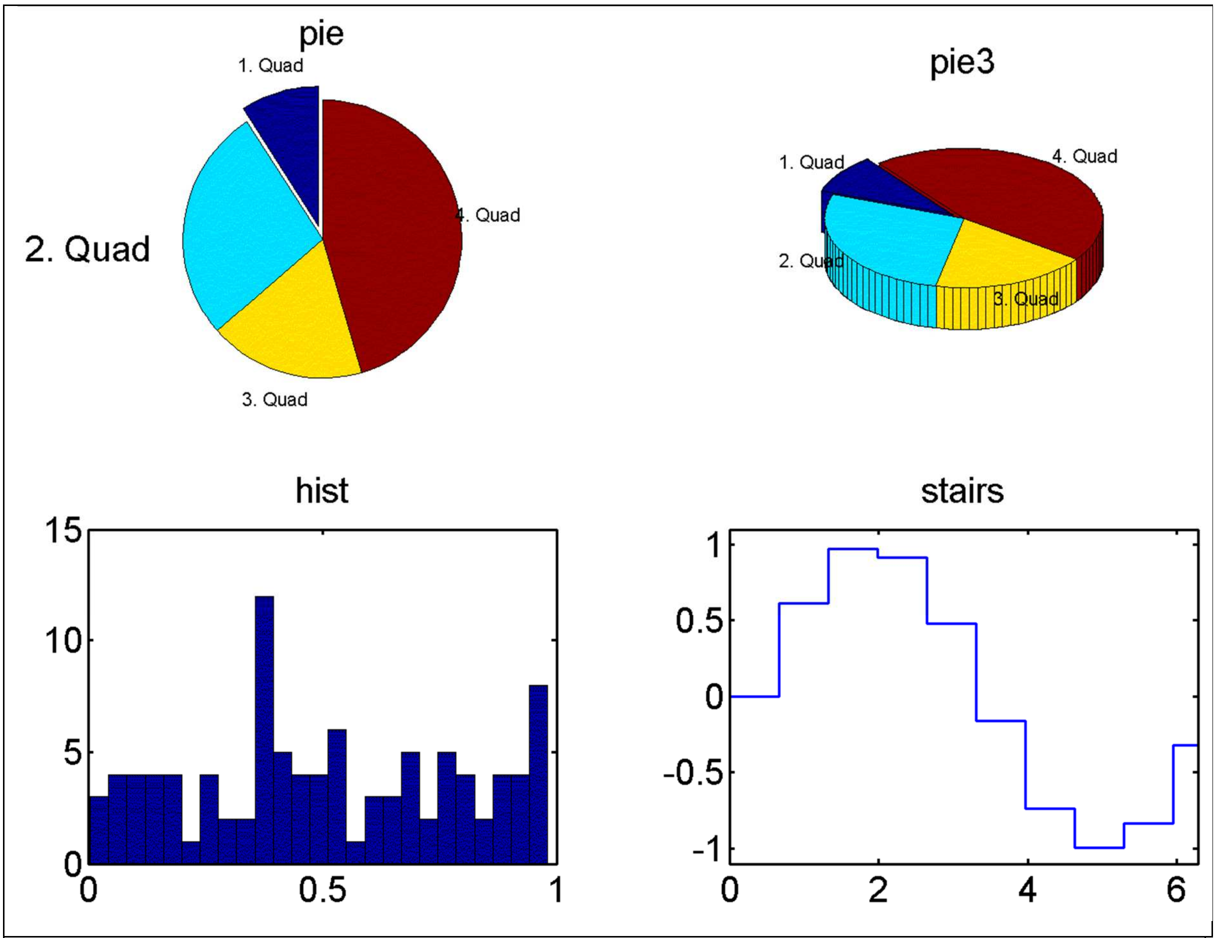

```
function Stat1

try, delete(1),end
figure('DefaultAxesFontSize',15)

N=10;
x=1:N;
y=rand(1,N);

subplot(2,2,1)
plot(x,y,'or')
x1=mean(x);
y0=min(y);
y1=mean(y);
y2=max(y);
y3=std(y);
title('random numbers')

subplot(2,2,2)
hold on
X=sort(x);
Y=sort(y);
plot(X,Y,'or')

Y0=min(Y);
Y1=mean(Y);
Y2=max(Y);
Y3=std(Y);

plot(X([1 end]),Y0*[1 1],'b',X([1 end]),Y1*[1 1],'y',X([1 end]),Y2*[1 1],'g')
plot(X([1 end]),(Y1+Y3)*[1 1],'r:',X([1 end]),(Y1-Y3)*[1 1],'r:')
title('random numbers (sorted)')
h=legend('Random numbers (sorted)','Minimal value','Mean value',...
    'Maximal value','sdv',2);
set(h,'fontsize',8)
```

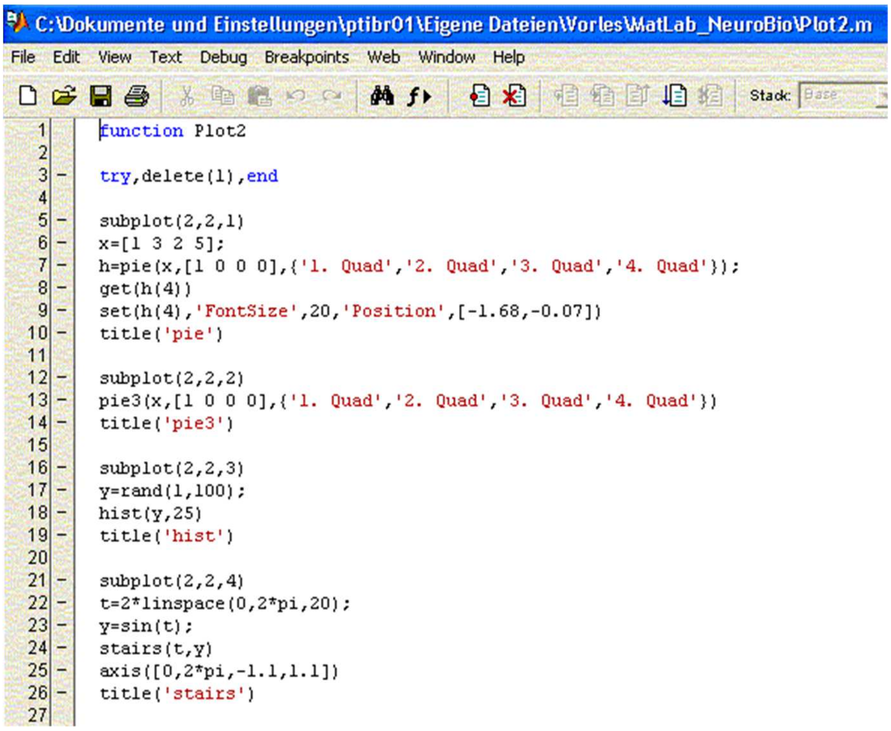
Fortsetzung des Programms




```
% Lineare Korrelation
X1=mean(X)
r=sum((x-x1).*(y-y1))/(sqrt(sum((x-x1).^2))*sqrt(sum((y-y1).^2)));
R=sum((X-X1).*(Y-Y1))/(sqrt(sum((X-X1).^2))*sqrt(sum((Y-Y1).^2)));

subplot(2,2,3)
hold on
M=[ones(N,1) X'];
A=M\Y';
plot(X,Y,'bo',X,(M*A)','b')
M=[ones(N,1) x'];
A=M\y';
plot(x,y,'go',x,(M*A)','g')

title(['r=' num2str(r) ', R=' num2str(R)])

%%%%%%%%%% fourier analysis
N=2^8
x=10*linspace(0,2*pi,N);
c=(2*mod(2:N+1,2)-1).*exp(-.5*(1:N));
y=c*sin((1:N)'*x);

subplot(4,2,6)
hold on
plot(x,y)
Y=real(fft(y));
axis([0 x(end) -1 1])
title('Time series')
set(gca,'Xtick',[])

subplot(4,2,8)
plot(x,Y)
axis([0 x(end) -11 11])
title('Fourier amplitude')

%%%%%%%%%% Timeseries to data file
f_hnd=fopen('TimeSeries.txt','w');
for n=1:length(x)
    fprintf(f_hnd,'%10.5f; %10.5f; %10.5f\n',x(n),y(n),Y(n))
end
fclose(f_hnd)
```

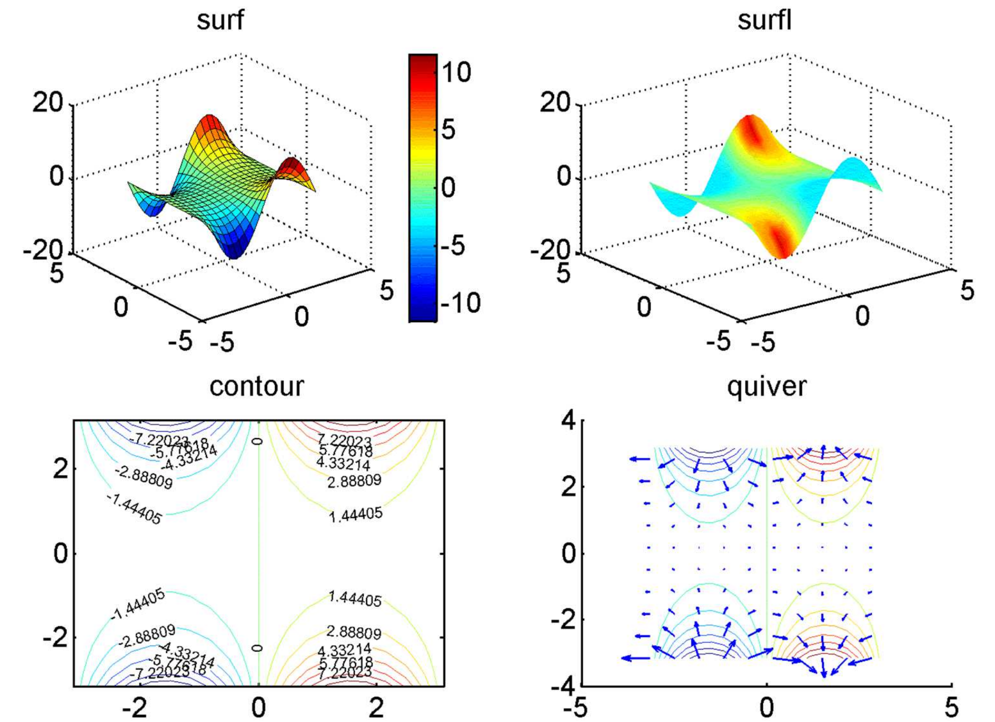
Plot 2

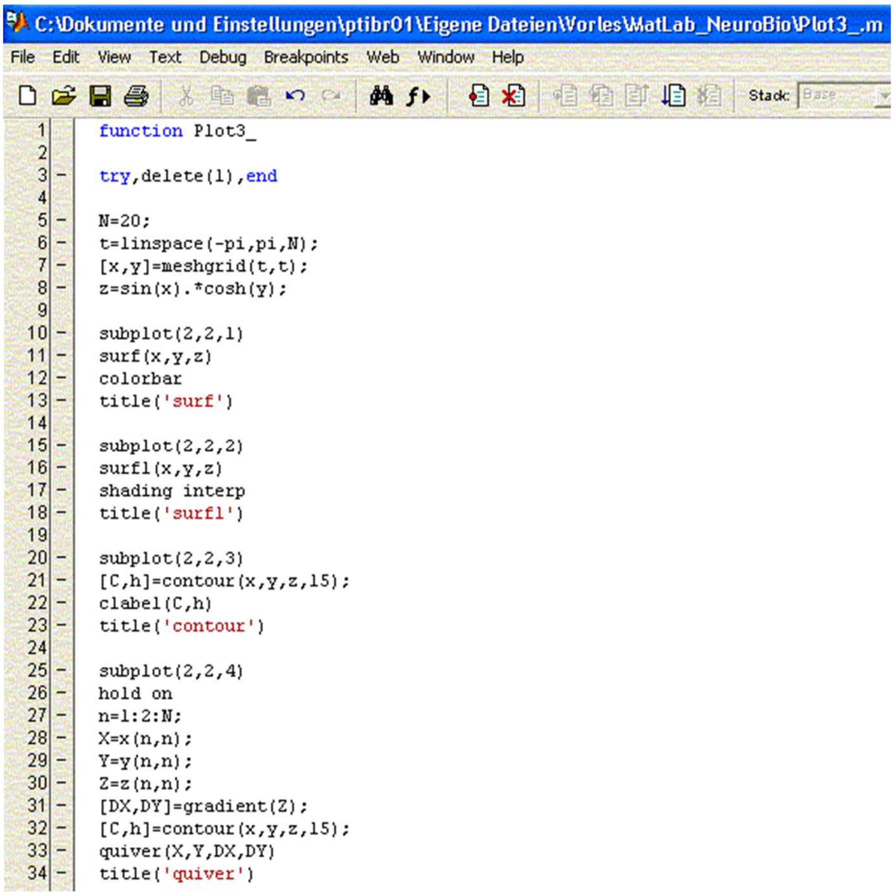





```
function Plot2

try,delete(1),end

subplot(2,2,1)
x=[1 3 2 5];
h=pie(x,[1 0 0 0],{'1. Quad','2. Quad','3. Quad','4. Quad'});
get(h(4))
set(h(4),'FontSize',20,'Position',[-1.68,-0.07])
title('pie')

subplot(2,2,2)
pie3(x,[1 0 0 0],{'1. Quad','2. Quad','3. Quad','4. Quad'})
title('pie3')

subplot(2,2,3)
y=rand(1,100);
hist(y,25)
title('hist')

subplot(2,2,4)
t=2*linspace(0,2*pi,20);
y=sin(t);
stairs(t,y)
axis([0,2*pi,-1.1,1.1])
title('stairs')
```

Plot 3






```
function Plot3_

try,delete(1),end

N=20;
t=linspace(-pi,pi,N);
[x,y]=meshgrid(t,t);
z=sin(x).*cosh(y);

subplot(2,2,1)
surf(x,y,z)
colorbar
title('surf')

subplot(2,2,2)
surfl(x,y,z)
shading interp
title('surfl')

subplot(2,2,3)
[C,h]=contour(x,y,z,15);
clabel(C,h)
title('contour')

subplot(2,2,4)
hold on
n=1:2:N;
X=x(n,n);
Y=y(n,n);
Z=z(n,n);
[DX,DY]=gradient(Z);
[C,h]=contour(x,y,z,15);
quiver(X,Y,DX,DY)
title('quiver')
```

10. Addition

Exercise (Fit of arbitrary functions)

Create a 'random exponential function' by

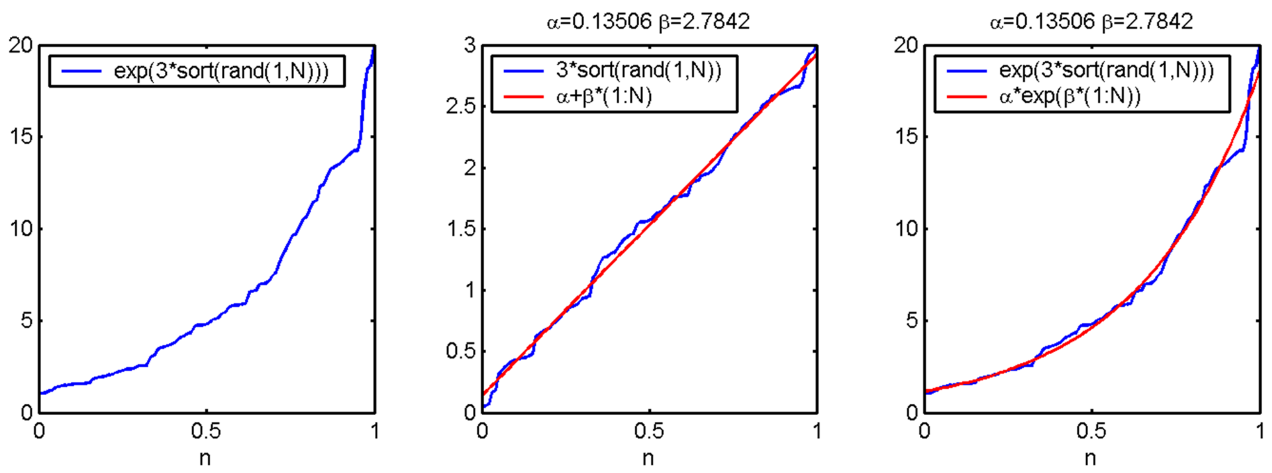
$$y = \exp\left(3 \cdot \mathrm{sort}\left(\mathrm{ran}\left(1:N\right)\right)\right)$$

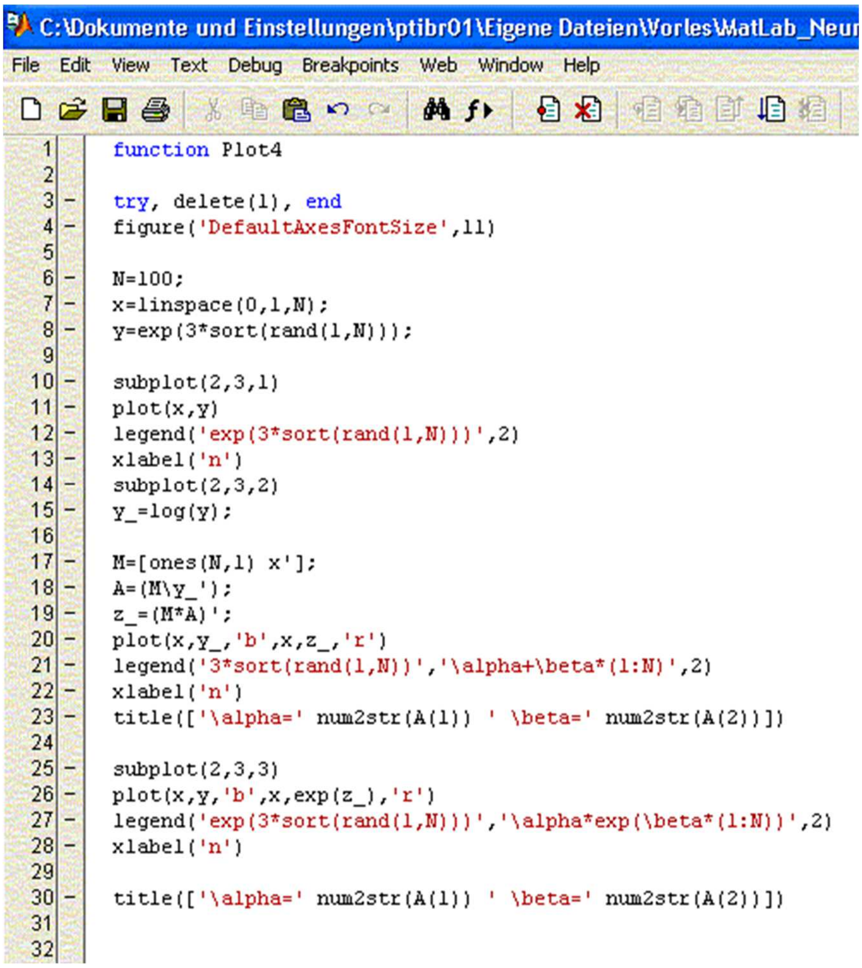
Approximate this function by

$$z = \alpha e^{\beta x}$$

Tipp:

Approximate $\tilde{y}(x) = \log\left(y(x)\right)$ by $\tilde{z}(x) = \alpha + \beta x$. Its an easy way to get α and β. The you can display $z = \alpha e^{\beta x}$.






```
function Plot4

try, delete(1), end
figure('DefaultAxesFontSize',11)

N=100;
x=linspace(0,1,N);
y=exp(3*sort(rand(1,N)));

subplot(2,3,1)
plot(x,y)
legend('exp(3*sort(rand(1,N)))',2)
xlabel('n')
subplot(2,3,2)
y_=log(y);

M=[ones(N,1) x'];
A=(M\y_');
z_=(M*A)';
plot(x,y_,'b',x,z_,'r')
legend('3*sort(rand(1,N))','\alpha+\beta*(1:N)',2)
xlabel('n')
title(['\alpha=' num2str(A(1)) ' \beta=' num2str(A(2))])

subplot(2,3,3)
plot(x,y,'b',x,exp(z_),'r')
legend('exp(3*sort(rand(1,N)))','\alpha*exp(\beta*(1:N))',2)
xlabel('n')

title(['\alpha=' num2str(A(1)) ' \beta=' num2str(A(2))])
```

11. Physics: Solution of the time dependent Schrödinger-Equation

Equation and boundary conditions

$$-\frac{1}{i}\partial_t\psi = \underbrace{-\frac{1}{2}\partial_x^2}_{H}\psi,$$

$$\psi(0,t) = \psi(\pi,t) = 0 \quad (\text{Kasten})$$

Solution

$$-\frac{1}{i}\partial_t\psi = H\psi \quad \Rightarrow \quad \psi(x,t) = e^{-iHt}\psi(x,0)$$

Basis

$$u_n(x) = \sqrt{\frac{2}{\pi}}\sin(nx) \quad (\text{erfüllt Randbedingungen})$$

$$\psi(x,t) = \sum_{n=1}^{N} c_n(t)u_n(x)$$

$$-\frac{\hbar}{i}\partial_t u_m = H_{mn}u_m, \quad \text{mit } H_{mn} = \langle u_m | H | u_n \rangle$$

$$c_m(t) = e^{-iH_{mn}t}c_n(0) \quad \text{oder}$$

$$\vec{c}(t) = e^{-i\hat{H}t}\vec{c}(0)$$

Summary:

$$u_n(x) = \sqrt{\frac{2}{\pi}}\sin(nx)$$

$$H_{mn} = \int_0^{\pi} u_m H u_n dx$$

$$\vec{c}(t) = e^{-i\hat{H}t}\vec{c}(0)$$

$$\psi(t) = \sum_{n=1}^{N} c_n(t)u_n(x)$$

Start function:

$$\psi^{(0)} = \mathrm{e}^{-\left(a\left(x-\frac{\pi}{2}\right)\right)^2} e^{ik^{(x)}x}$$

$$c(0) = \int_0^{\pi} \psi^{(0)} u_n dx$$

Formulation with vectors and matrices

$$x = linspace(0, \pi, N)$$

$$dx = x(2) - x(1)$$

$$f^{(M)} = 1 : M$$

$$u = \left(u_{mx_n}\right) = \left(\sqrt{\tfrac{2}{\pi}} \sin\left(f_m^{(M)} x(n)\right)\right) = \sqrt{\tfrac{2}{\pi}} \sin\left(f^{(M)'} \cdot x\right)$$

$$D^{(2)} = \left(D_{nm}^{(2)} = -m^2\right) = -ones(N,1) \cdot f^{(M)}$$

$$(H_{mn}) = \int_0^\pi u_m \left(-n^2\right) u_n dx = u_m \cdot \left(D^{(2)} .\cdot u_n^{'}\right) dx$$

$$\psi^{(0)} = \left(\psi_m^{(0)}\right) = e^{-\left(\left(x-\frac{\pi}{2}\right)/B_0\right)^2} .\cdot e^{ikx}$$

$$c(0) = u\psi^{(0)'} dx = (u_{nm})\left(\psi_{m1}^{(0)'}\right) dx \in R^N \otimes R^1 \qquad \left(\simeq \int u_n(x) \psi^{(0)}(x) dx\right)$$

MatLab-Program

```
function Schroedinger

%Loesung der zeitabhaengigen Schroedinger-Gleichung in einem 1d-Kasten

try, delete(1),end

N=200  %Punkte des Raumgitters
M=30   %Dimension des Hilbert-Raumes

k_x=20   %Impuls in x-Richtung
B0=.5    %Breite der Anfangsverteilung
if 1
    T=.16      %Maximale Zeit
    N_T=500 %Zeitschritte
else
    T=3.14      %Maximale Zeit
    N_T=2000 %Zeitschritte
end

%Ortsgitter
x=linspace(0,pi,N);
dx=x(2)-x(1);

%Basisfuntionen
f=1:M;
a=f'*x;
u=sqrt(2/pi)*sin(a);

%Hamilton-Operator
D2=-ones(N,1)*f.^2;
H=u*(D2.*u')*dx;

%Startfunktion (auch grafisch)
psi0=exp(-((x-pi/2)/B0).^2).*exp(i*k_x*x);
c0=u*psi0'*dx;
psi0=c0'*u;
PHnd=plot(x,conj(psi0).*psi0);
set(PHnd,'EraseMode','xor')
axis([0 pi 0 1.5])

%Zeitentwicklung
t=linspace(0,T,N_T);
for n=1:length(t)
    c=expm(-i*H*t(n))*c0;
    psi=c'*u;
    set(PHnd,'ydata',conj(psi).*psi)
    drawnow
end
```