Usage of MATLAB Report Generator

Eniz Museljic

8. Juli 2018

Inhaltsverzeichnis

Kapitel 1

Introduction

INTRODUCTION

The following MATLAB code did the import of the text of the whole chapter:

```
report.addParagraph('INTRODUCTION')
```

To ensure that Latex can be found by MATLAB the following lines have been added to `startup.m`:

```
path1 = getenv('PATH');
path1 = [path1,':/Library/TeX/texbin'];
setenv('PATH', path1)
```

Kapitel 2

List Environment

LIST

2.1 Itemize

- Text 1
- Text 2
- Text 3
  and some additional text.

Listings can be nested:

- Text 1
- Text 2
  - Text a
  - Text b
  - Text c
- Text 3
  1. Item a
  2. Item b
  3. Item c
- Text 4

2.2 Enumerate

1. Text 1

2. Text 2
3. Text 3

   Further text.

2.3 Description

Def 1: Text 1

And some further description.

Def 2: Text 2

Def 3: Text 3

Further text.

Here the first entry for each item is used to specify the name which is described.

Kapitel 3

Tabular and Table

3.1 A simple example

A simple table from an array with standard settings:

1.00	2.00	3.00
4.00	5.00	6.00

A different format, alignment and labels for colums and rows.

Different settings can be specified using the `set` method:

```
report.set('dataFormat','%i')
```

If one would like to see the Latex output in the console one can use:

```
report.set('dispLatexCode',1)
```

	col1	col2	col3
row1	1	2	3
row2	4	5	6

To reset all table settings please use:

```
report.setDefault('table')
```

To reset Latex output to its default value use:

```
report.setDefault('dispLatexCode')
```

3.2 Another example with booktabs and Not a Number

1.1235	nan	3.12
4.1235	5.123	6.12
7.1235	8.123	9.12
10.1235	11.123	12.12

	col1	col2	col3
row1	1.12	-	3.12
row2	4.12	5.12	6.12
	7.12	8.12	9.12
row4	10.12	11.12	12.12

3.3 MATLAB table as input

Here the MATLAB table is constructed with the following code from `myCodeBlocks.m`:

```
LastName = {'Smith';'Johnson';'Williams';'Jones';'Brown'};
Age = [38;43;38;40;49];
Height = [71;69;64;67;64];
Weight = [176;163;131;133;119];
T = table(Age,Height,Weight,'RowNames',LastName);
```

	Age	Height	Weight
Smith	38	71	176
Johnson	43	69	163
Williams	38	64	131
Jones	40	67	133
Brown	49	64	119

3.4 Using a table instead of tabular

	Smith	Johnson	Williams	Jones	Brown
Age	38	43	38	40	49
Height	71	69	64	67	64
Weight	176	163	131	133	119

Tabelle 3.1: My Caption

Kapitel 4

Images and Figures

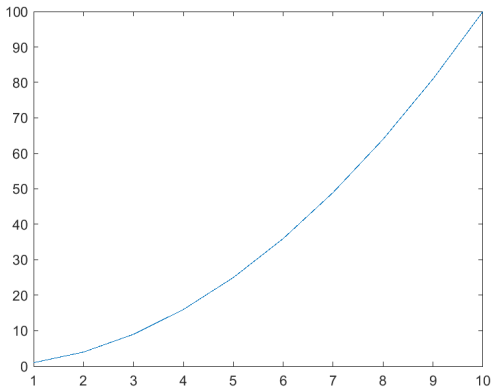
4.1 Code Listing and Evaluation

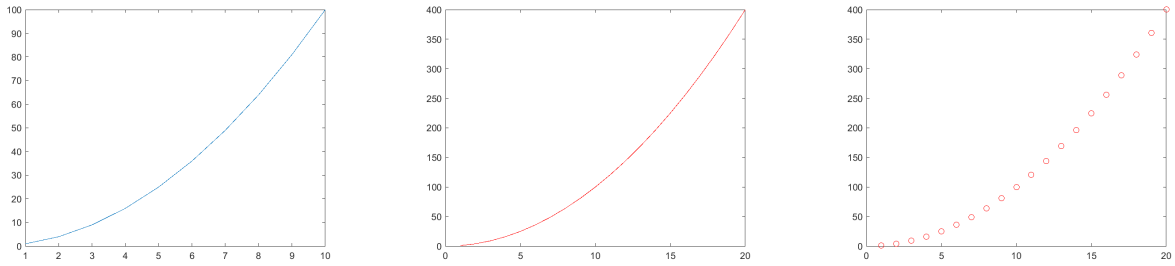
```
x=5; y=3;
z=x+y;
d=5;

v1 = 1:10;
fh(1)=figure;
plot(v1,v1.^2)
v2 = 1:20;
fh(2)=figure;
plot(v2,v2.^2,'r')
v3 = 1:20;
fh(3)=figure;
plot(v3,v3.^2,'ro')
```

4.2 Image

One of the above images directly here:





4.3 Figure

Two images side by side as a figure with caption:

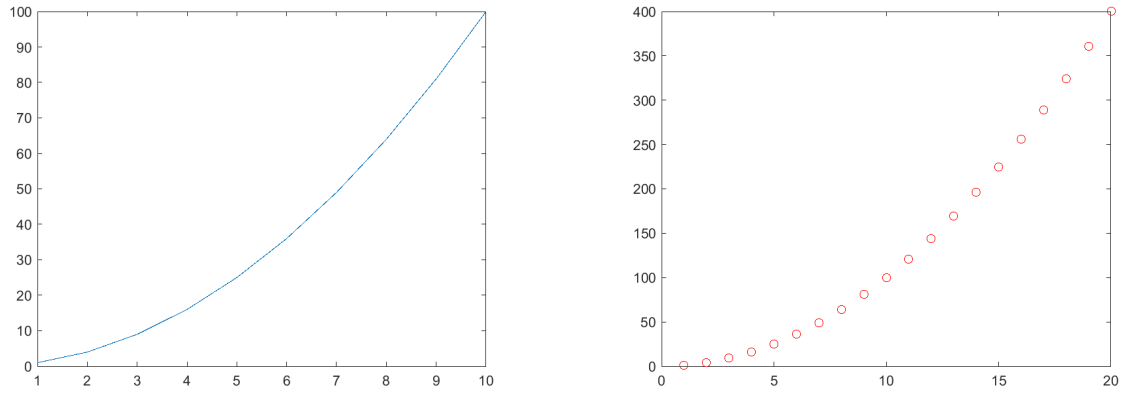


Abbildung 4.1: Two images side by side

All three images, one per line and modified width:

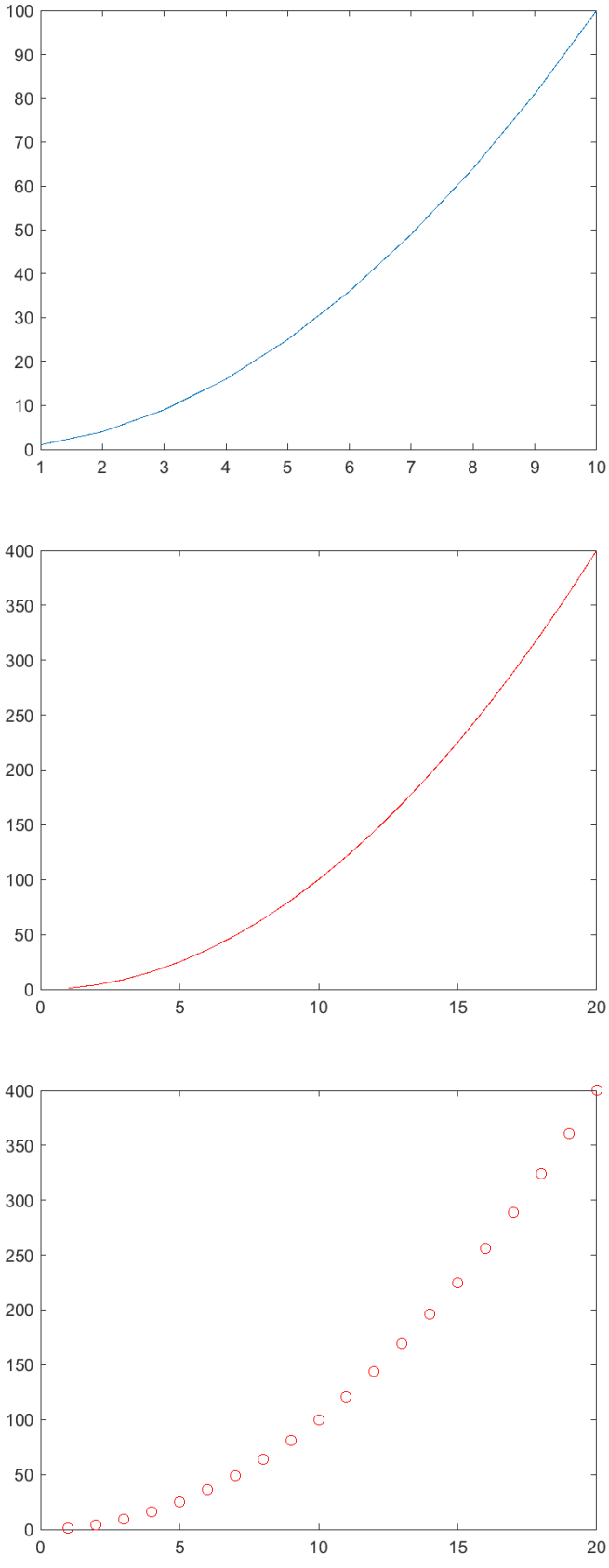


Abbildung 4.2: One image per line

Kapitel 5

Pretty Output

5.1 Listing or Latex

```
M1 = logical(eye(5));
M2 = logical(flipud(eye(5)));
```

And so does it look like as listing

```
M1 =                    M2 =

  1  0  0  0  0           0  0  0  0  1
  0  1  0  0  0           0  0  0  1  0
  0  0  1  0  0           0  0  1  0  0
  0  0  0  1  0           0  1  0  0  0
  0  0  0  0  1           1  0  0  0  0
```

or in Latex mode

$$\mathtt{M1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathtt{M2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

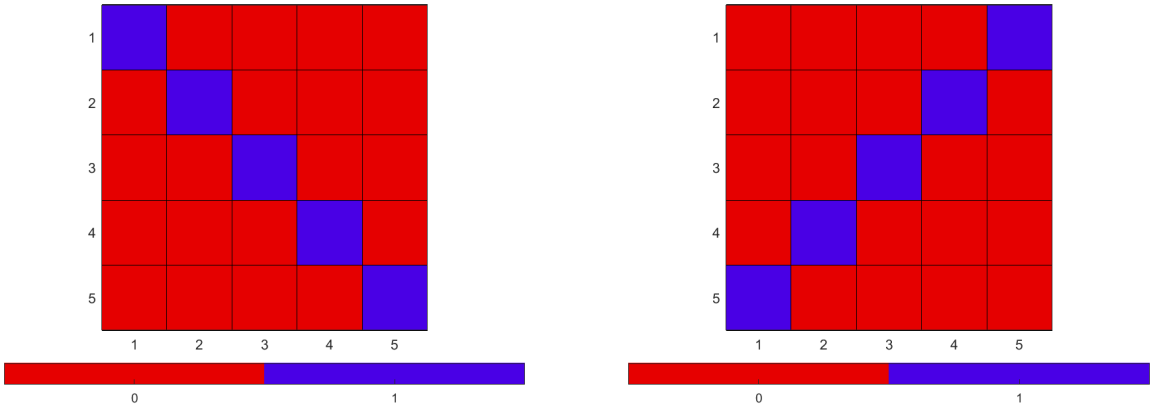


Abbildung 5.1: My arrays separate

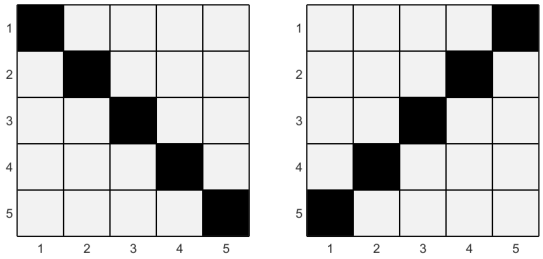


Abbildung 5.2: My arrays together

5.2 Graphics output for arrays

5.2.1 Two images in color with colorbar

5.2.2 One image in bw without colorbar

5.2.3 Another example

```
R1 = rand(100);
```

And this is the output in form of a listing

```
R1 =

  0.1371  0.1273  0.7990  0.3933   >   0.6934
  0.2922  0.0157  0.4417  0.1813   >   0.3792
  0.0073  0.7531  0.4925  0.2094   >   0.7139
  0.9327  0.5050  0.0295  0.5937   >   0.1277
  0.6093  0.7221  0.5726  0.5945   >   0.0776
  0.8918  0.7611  0.5725  0.1725   >   0.5682
  0.4208  0.2689  0.7417  0.9330   >   0.8953
  0.4002  0.1936  0.0766  0.0562   >   0.6109
  0.5539  0.0641  0.7872  0.2017   >   0.8061
    |       |       |       |      >     |
  0.1799  0.2964  0.3201  0.6773   >   0.2174
```

or again with latex syntax

$$\texttt{R1} = \begin{bmatrix} 0.1371 & 0.1273 & 0.7990 & 0.3933 & \cdots & 0.6934 \\ 0.2922 & 0.0157 & 0.4417 & 0.1813 & \cdots & 0.3792 \\ 0.0073 & 0.7531 & 0.4925 & 0.2094 & \cdots & 0.7139 \\ 0.9327 & 0.5050 & 0.0295 & 0.5937 & \cdots & 0.1277 \\ 0.6093 & 0.7221 & 0.5726 & 0.5945 & \cdots & 0.0776 \\ 0.8918 & 0.7611 & 0.5725 & 0.1725 & \cdots & 0.5682 \\ 0.4208 & 0.2689 & 0.7417 & 0.9330 & \cdots & 0.8953 \\ 0.4002 & 0.1936 & 0.0766 & 0.0562 & \cdots & 0.6109 \\ 0.5539 & 0.0641 & 0.7872 & 0.2017 & \cdots & 0.8061 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.1799 & 0.2964 & 0.3201 & 0.6773 & \cdots & 0.2174 \end{bmatrix}$$

One can also display it as an image.

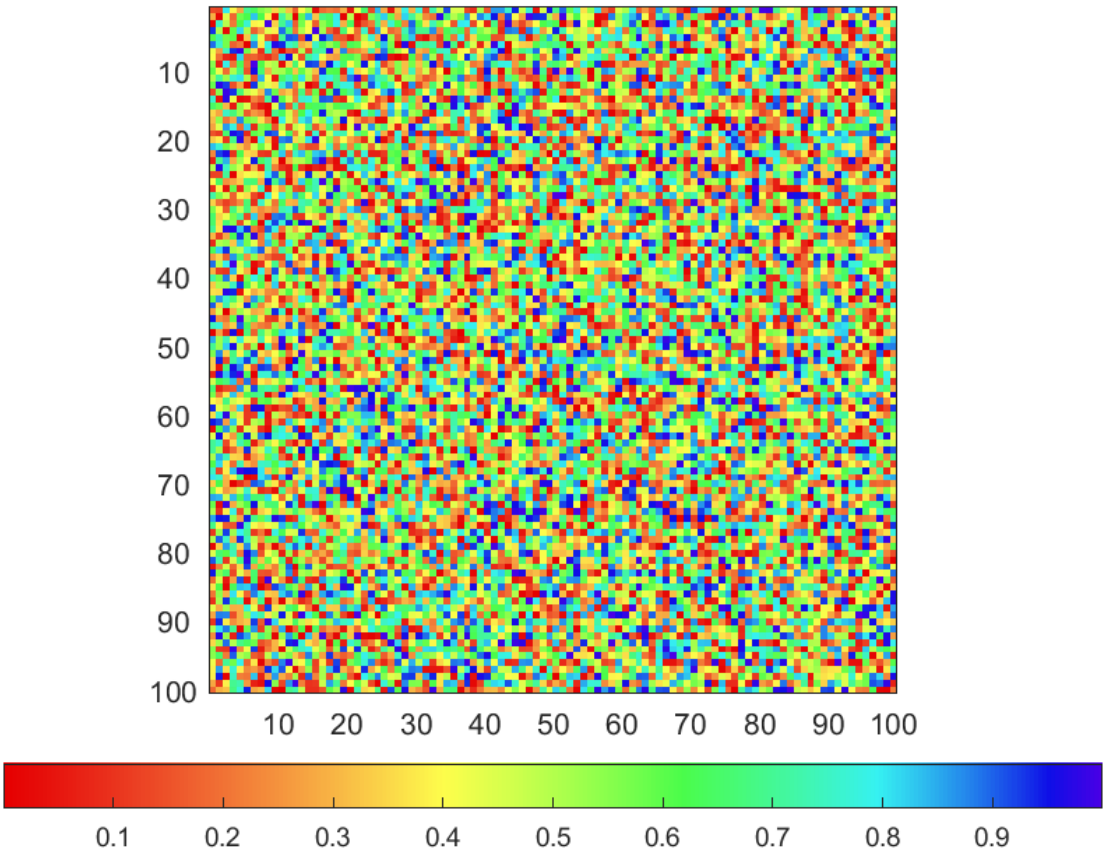


Abbildung 5.3: Some random numbers

Kapitel 6

Function description

6.1 A simple example

Some text before. Write the following function with specified in- and output.

```
function [r1,r2] = func(i1,i2,i3)
```

InOut	Name	Type	Description
Input	`i1`	`double`	2-D array
Input	`i2`	`double`	Scalar
Input	`i3`	`double`	Scalar
Output	`r1`	`double`	Result 1
Output	`r2`	`double`	Result 2

Some text after.

6.2 A complete example

Schreiben sie die folgende Funktion mit dem in der Tabelle spezifizierten In- und Output.

```
function [L,R,r] = myTestFunc(M)
```

InOut	Name	Type	Description
Input	`M`	`double`	2-dimensionale Matrix; quadratisch; alle Werte ungleich Null
Output	`L`	`logical`	Stern in der gleichen Größe wie `M` (siehe Beispiel)
Output	`R`	`double`	gleich wie `M` aber mit Nullen wo `L false` ist (siehe Beispiel)
Output	`r`	`double`	Skalar; Summe aller Werte in `M` wo `L true` ist

Die folgenden Beispiele erläutern die Aufgabe.

Beispiel 1 - 6×6**-Array**

```
M1 = reshape(1:36,6,6)+5;
[L1,R1,r1] = myTestFunc(M1);
```

```
M1 =

   6  12  18  24  30  36
   7  13  19  25  31  37
   8  14  20  26  32  38
   9  15  21  27  33  39
  10  16  22  28  34  40
  11  17  23  29  35  41

R1 =

   6   0   0   0   0  36
   0  13   0   0  31   0
   0   0  20  26   0   0
   0   0  21  27   0   0
   0  16   0   0  34   0
  11   0   0   0   0  41

L1 =

   1   0   0   0   0   1
   0   1   0   0   1   0
   0   0   1   1   0   0
   0   0   1   1   0   0
   0   1   0   0   1   0
   1   0   0   0   0   1

r1 =

   282
```

Beispiel 2 - 5×5**-Array**

```
M2 = reshape(1:25,5,5)+5;
[L2,R2,r2] = myTestFunc(M2);
```

```
M2 =

   6  11  16  21  26
   7  12  17  22  27
   8  13  18  23  28
   9  14  19  24  29
  10  15  20  25  30

R2 =

   6   0   0   0  26
   0  12   0  22   0
   0   0  18   0   0
   0  14   0  24   0
  10   0   0   0  30

L2 =

   1   0   0   0   1
   0   1   0   1   0
   0   0   1   0   0
   0   1   0   1   0
   1   0   0   0   1

r2 =

   162
```

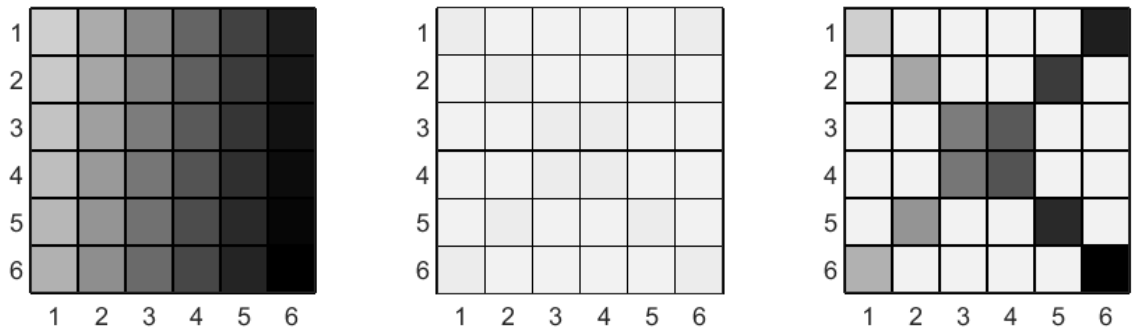


Abbildung 6.1: M1 (Links), L1 (Mitte), R1 (Rechts)

6.3 Same example with information from help text of function

Schreiben sie die folgende Funktion mit dem in der Tabelle spezifizierten In- und Output.

```
function [L,R,r] = myTestFunc(M)
```

InOut	Name	Type	Description
Input	`M`	`double`	2-dimensionale Matrix; quadratisch; alle Werte ungleich Null
Output	`L`	`logical`	Stern in der gleichen Größe wie `M` (siehe Beispiel)
Output	`R`	`double`	gleich wie `M` aber mit Nullen wo `L` `false` ist (siehe Beispiel)
Output	`r`	`double`	Skalar; Summe aller Werte in `M` wo `L` `true` ist

Die folgenden Beispiele erläutern die Aufgabe.

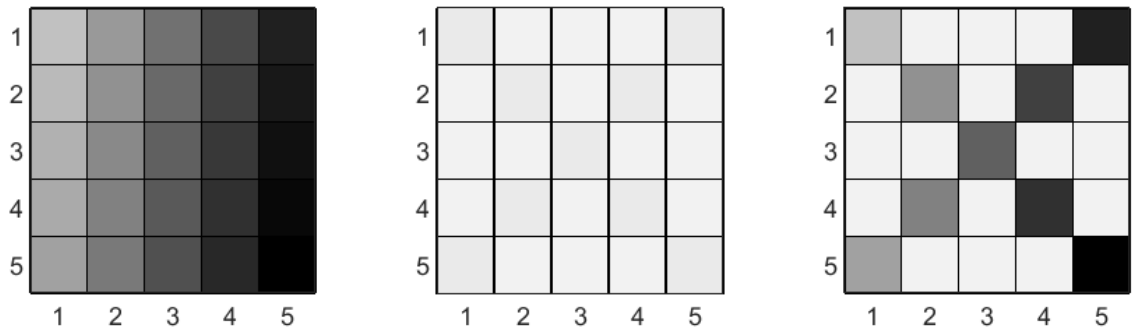


Abbildung 6.2: M2 (Links), L2 (Mitte), R2 (Rechts)

Beispiel 1 - 6×6**-Array**

```
M1 = reshape(1:36,6,6)+5;
[L1,R1,r1] = myTestFunc(M1);
```

```
M1 =

   6  12  18  24  30  36
   7  13  19  25  31  37
   8  14  20  26  32  38
   9  15  21  27  33  39
  10  16  22  28  34  40
  11  17  23  29  35  41
```

```
L1 =

  1  0  0  0  0  1
  0  1  0  0  1  0
  0  0  1  1  0  0
  0  0  1  1  0  0
  0  1  0  0  1  0
  1  0  0  0  0  1
```

```
R1 =                          r1 =

  6   0   0   0   0 36          282
  0  13   0   0  31   0
  0   0  20  26   0   0
  0   0  21  27   0   0
  0  16   0   0  34   0
 11   0   0   0   0 41
```

Beispiel 2 - 5×5**-Array**

```
M2 = reshape(1:25,5,5)+5;
[L2,R2,r2] = myTestFunc(M2);
```

```
M2 =                          L2 =

  6  11  16  21  26            1  0  0  0  1
  7  12  17  22  27            0  1  0  1  0
  8  13  18  23  28            0  0  1  0  0
  9  14  19  24  29            0  1  0  1  0
 10  15  20  25  30            1  0  0  0  1

R2 =                          r2 =

  6   0   0   0  26            162
  0  12   0  22   0
  0   0  18   0   0
  0  14   0  24   0
 10   0   0   0  30
```

Kapitel 7

Some data processing

7.1 My Results

Kapitel 8

Final processing and viewing

Some commands to export, generate and view the results:

```
report.export   % export latex file
report.generate % process latex to pdf
report.view     % viewer
close all    % close all figures
```