DIP 55:148

Introduction to Matlab:
1) Basics

Reinhard R. Beichel
3312 SC
Tel: 335-4597
reinhard-beichel@uiowa.edu
http://www.engineering.uiowa.edu/~rbeichel

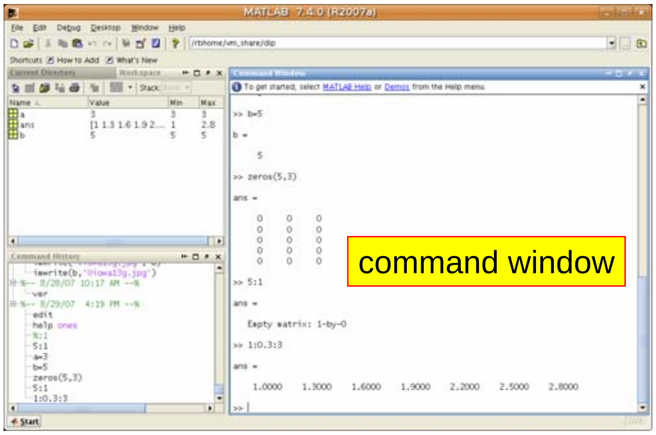
Matlab

- General tool for “technical computing”
- Very powerful toolbox concept
- Usage:
  - Interactively by typing commands
  - Scripts (“m”-files)
  - Functions (parameters, local variables, and return values)
  - MEX-functions (integration of C/C++ code)
- Nearly everything is represented as a Matrix
  - Images are represented as matrices too!

Matlab

- Advantages:
  - – Usually very short programs
  - – Short development time
  - – Powerful commands → rapid prototyping
  - – No declaration of variables (+/- ?)
  - – Built-in editor: *MATLAB> edit*
  - – Free code available on the Internet
  - – Which toolboxes are installed:
    - MATLAB> ver

Matlab GUI

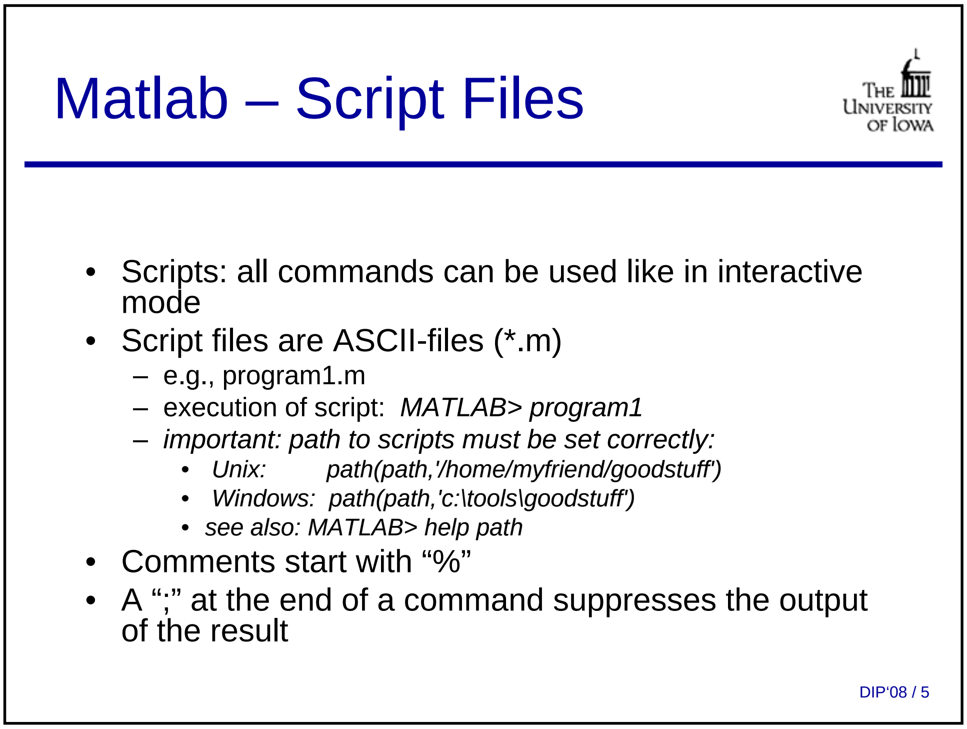

MATLAB 7.4.0 (R2007a)
File Edit Debug Desktop Window Help
/rbhome/vm_share/dip
Shortcuts How to Add What's New
Current Directory
Workspace
Stack:
Command Window
To get started, select MATLAB Help or Demos from the Help menu.
Command History
Start

Name	Value	Min	Max
a	3	3	3
ans	[1 1.3 1.6 1.9 2....	1	2.8
b	5	5	5

```
    iwrite(b,'Uiowa13g.jpg')
%-- 8/28/07 10:17 AM --%
    ver
%-- 8/29/07  4:19 PM --%
    edit
    help ones
    %:1
    5:1
    a=3
    b=5
    zeros(5,3)
    5:1
    1:0.3:3
```

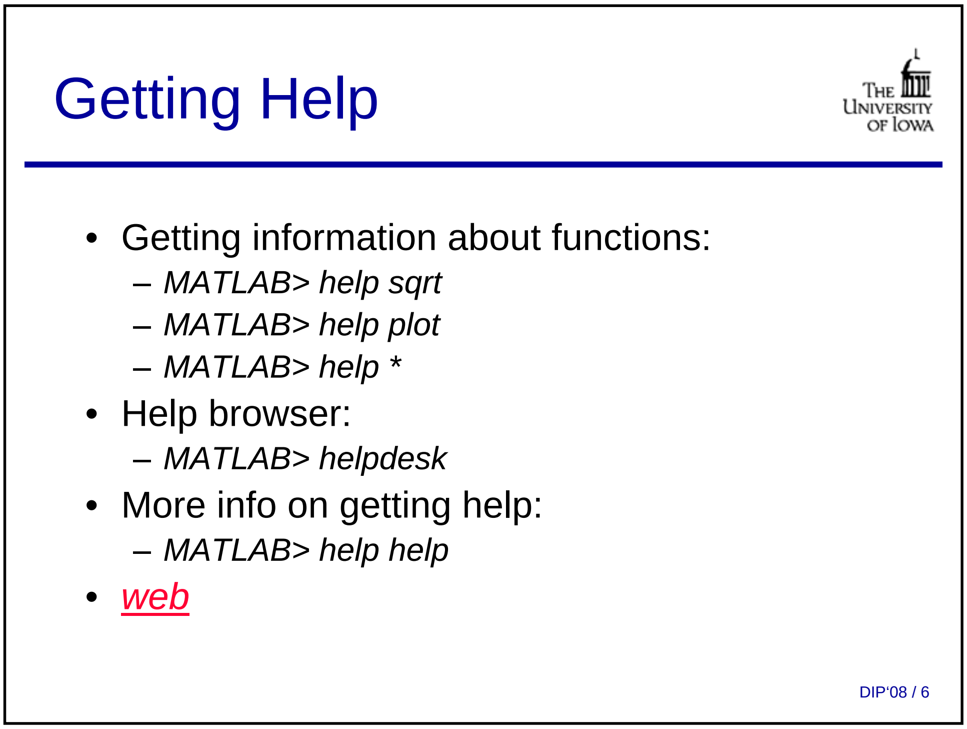
```
>> b=5

b =

     5

>> zeros(5,3)

ans =

     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0

>> 5:1

ans =

   Empty matrix: 1-by-0

>> 1:0.3:3

ans =

    1.0000    1.3000    1.6000    1.9000    2.2000    2.5000    2.8000

>> 
```

command window



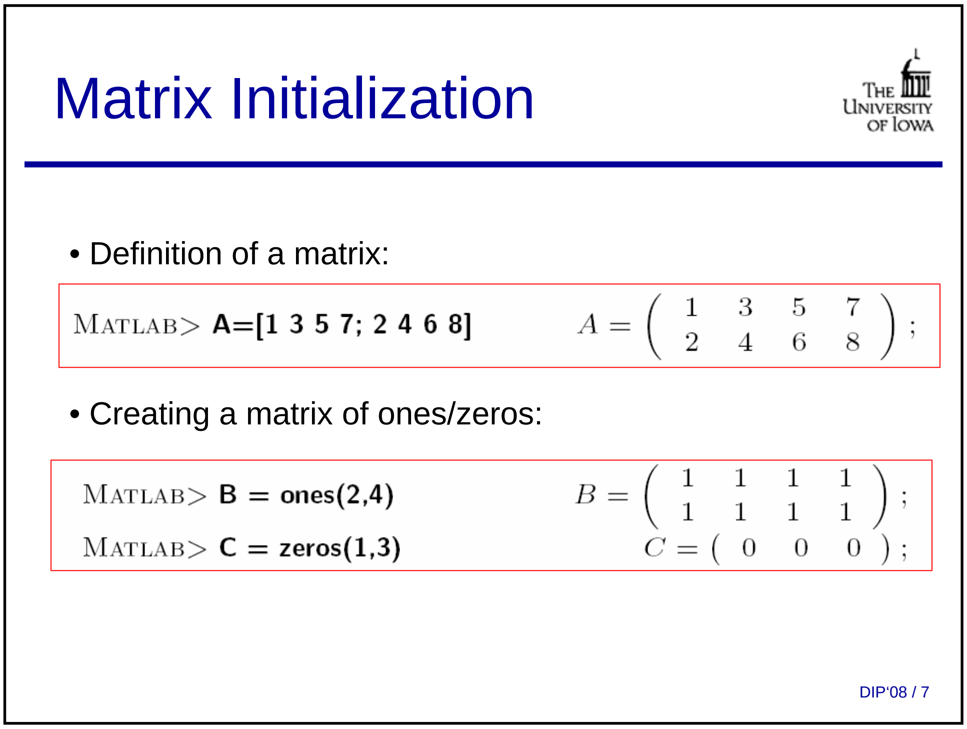
Matlab – Script Files

- Scripts: all commands can be used like in interactive mode
- Script files are ASCII-files (*.m)
  - e.g., program1.m
  - execution of script: *MATLAB> program1*
  - *important: path to scripts must be set correctly:*
    - *Unix: path(path,'/home/myfriend/goodstuff')*
    - *Windows: path(path,'c:\tools\goodstuff')*
    - *see also: MATLAB> help path*
- Comments start with “%”
- A “;” at the end of a command suppresses the output of the result



Getting Help

- Getting information about functions:
  - *MATLAB> help sqrt*
  - *MATLAB> help plot*
  - MATLAB> help *
- Help browser:
  - *MATLAB> helpdesk*
- More info on getting help:
  - *MATLAB> help help*
- *web*



Matrix Initialization

- Definition of a matrix:

MATLAB> **A=[1 3 5 7; 2 4 6 8]**

$$A = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix};$$

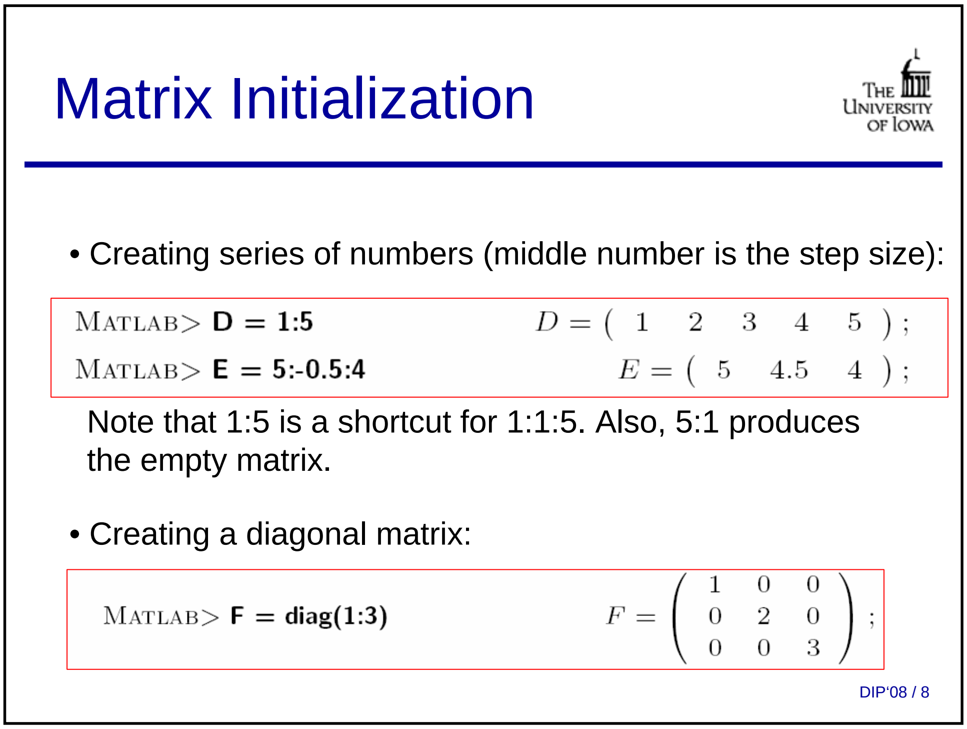
- Creating a matrix of ones/zeros:

MATLAB> **B = ones(2,4)**

MATLAB> **C = zeros(1,3)**

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix};$$

$$C = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix};$$



Matrix Initialization

- Creating series of numbers (middle number is the step size):

MATLAB> **D = 1:5**

MATLAB> **E = 5:-0.5:4**

$$D = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix};$$

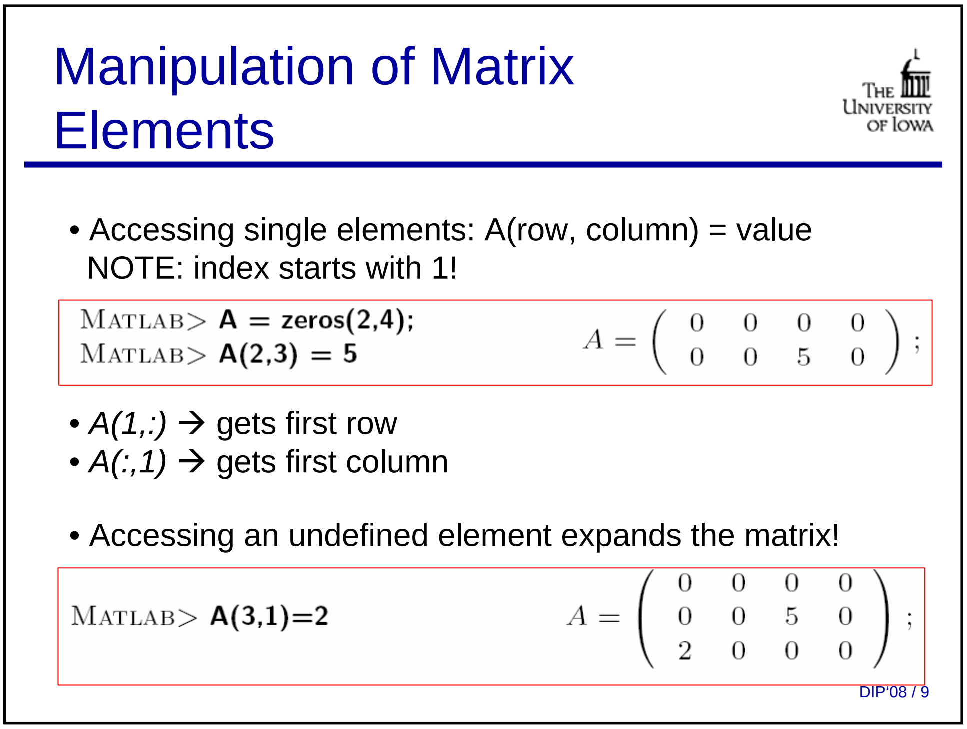
$$E = \begin{pmatrix} 5 & 4.5 & 4 \end{pmatrix};$$

Note that 1:5 is a shortcut for 1:1:5. Also, 5:1 produces the empty matrix.

- Creating a diagonal matrix:

MATLAB> **F = diag(1:3)**

$$F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix};$$



Manipulation of Matrix Elements

THE UNIVERSITY OF IOWA

- Accessing single elements: A(row, column) = value
  NOTE: index starts with 1!

```
MATLAB> A = zeros(2,4);
MATLAB> A(2,3) = 5
```

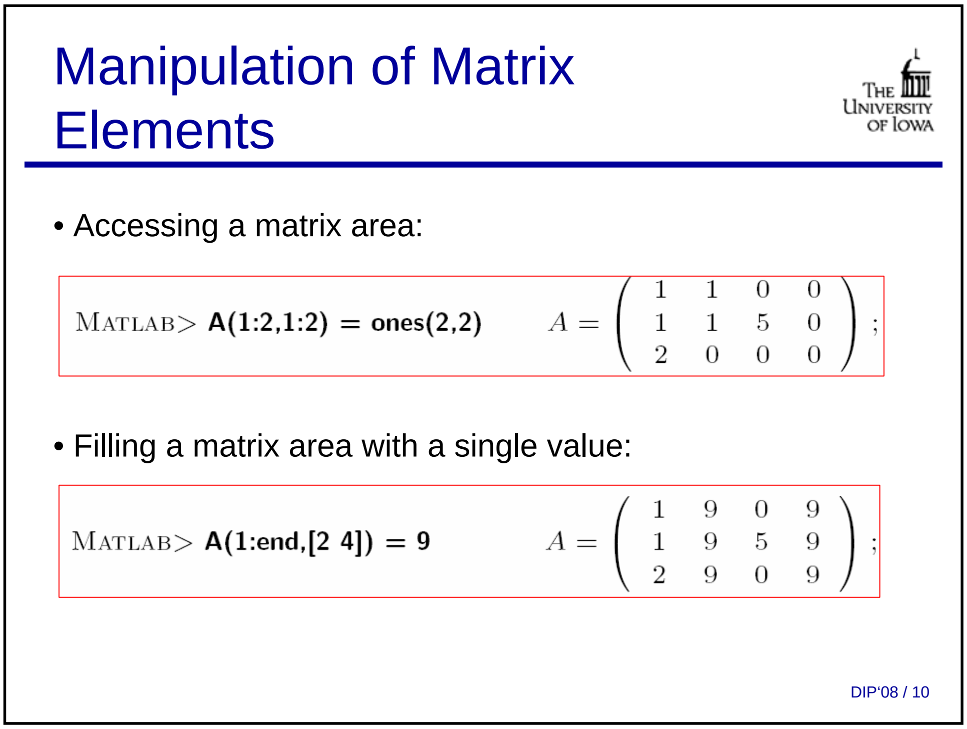
$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 \end{pmatrix};$$

- *A(1,:)* → gets first row
- *A(:,1)* → gets first column

- Accessing an undefined element expands the matrix!

```
MATLAB> A(3,1)=2
```

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix};$$



Manipulation of Matrix Elements

THE UNIVERSITY OF IOWA

- Accessing a matrix area:

```
MATLAB> A(1:2,1:2) = ones(2,2)
```

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 5 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix};$$

- Filling a matrix area with a single value:

```
MATLAB> A(1:end,[2 4]) = 9
```

$$A = \begin{pmatrix} 1 & 9 & 0 & 9 \\ 1 & 9 & 5 & 9 \\ 2 & 9 & 0 & 9 \end{pmatrix};$$

Vectors

- Vector == 1D Matrix
- column vector != row vector
- Creating a transposed vector/matrix: *MATLAB> A'*
- column vector == (row vector)'

```
MATLAB> A = 1:3
MATLAB> B = A'
```

$$A = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix};\, B = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix};$$

```
MATLAB> C = 1:5;
MATLAB> C(3) = 100
```

$$C = \begin{pmatrix} 1 & 2 & 100 & 4 & 5 \end{pmatrix};$$

Program Control Statements

- Conditional Control:

```
if logical_expression1
  …
elseif logical expression2
  …
else
  …
end
```

- Loop Control:

```
for variable = row vector
  …
end
```

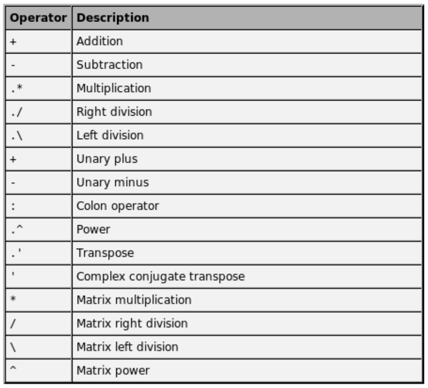
```
for
x=1:10
   x
end
```

```
a = 0;
while ( a < 10 )
   a = a + 1
end
```

Operators:

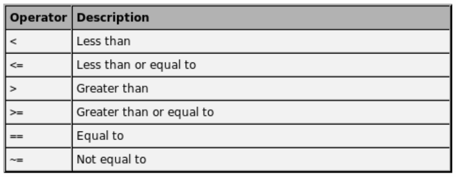
THE UNIVERSITY OF IOWA

Arithmetic Operators



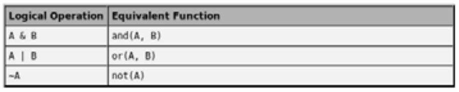
Operator	Description
+	Addition
-	Subtraction
.*	Multiplication
./	Right division
.\	Left division
+	Unary plus
-	Unary minus
:	Colon operator
.^	Power
.'	Transpose
'	Complex conjugate transpose
*	Matrix multiplication
/	Matrix right division
\	Matrix left division
^	Matrix power

Relational Operators



Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Logical Operators



Logical Operation	Equivalent Function
A & B	and(A, B)
A | B	or(A, B)
~A	not(A)

Matlab Optimized Programming

THE UNIVERSITY OF IOWA

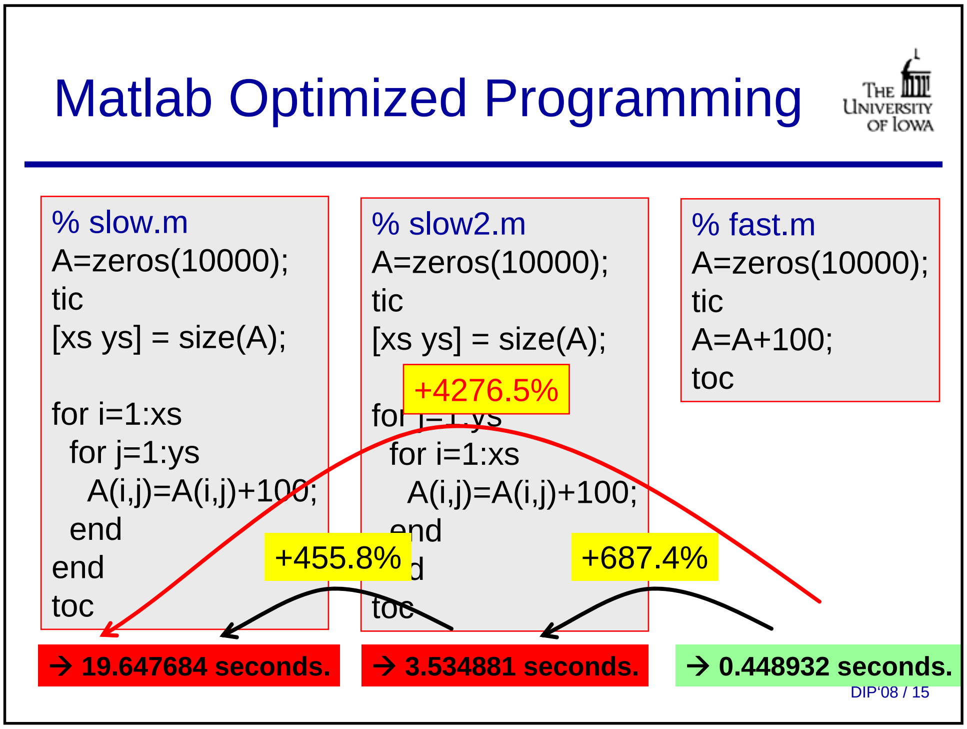
- Loops are VERY inefficient!!!
  - – Avoid loops if possible!
- Almost all functions accept scalars and vectors/matrices as input
  - → operation will be applied to each element of vector/matrix

use:

```
x = 1:10;
y = sqrt(x)
```

Instead:

```
y=[];
for x =1:10
        y = [y sqrt(x)]
end
```



Matlab Optimized Programming

```
% slow.m
A=zeros(10000);
tic
[xs ys] = size(A);

for i=1:xs
  for j=1:ys
    A(i,j)=A(i,j)+100;
  end
end
toc
```

```
% slow2.m
A=zeros(10000);
tic
[xs ys] = size(A);

for j=1:ys
  for i=1:xs
    A(i,j)=A(i,j)+100;
  end
end
toc
```

```
% fast.m
A=zeros(10000);
tic
A=A+100;
toc
```

+4276.5%

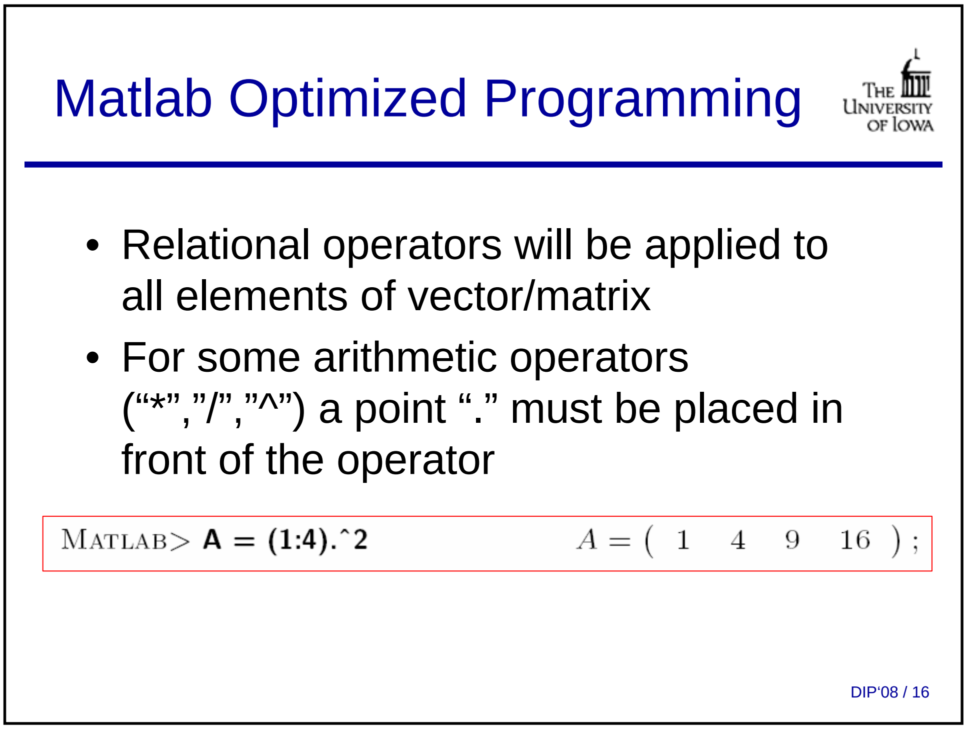
+455.8%

+687.4%

→ 19.647684 seconds.

→ 3.534881 seconds.

→ 0.448932 seconds.



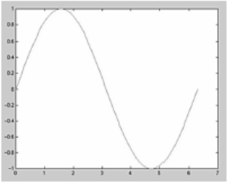
Matlab Optimized Programming

- Relational operators will be applied to all elements of vector/matrix
- For some arithmetic operators (“*”,”/”,”^”) a point “.” must be placed in front of the operator

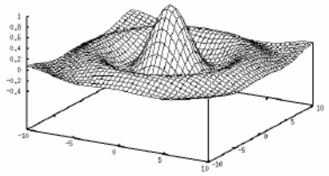
MATLAB> **A = (1:4).^2** $A = (\ 1 \quad 4 \quad 9 \quad 16\ )\ ;$

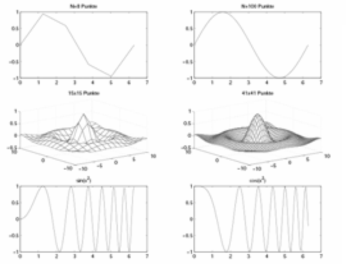
Plotting/Visualization

- 2D Plots



- 3D Plots



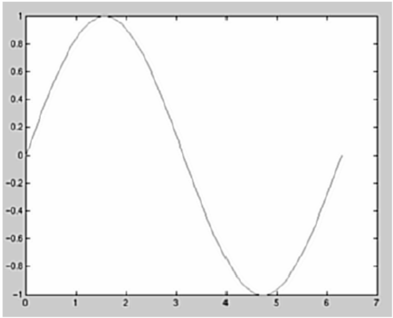


…multiple figures in one window

2D Plots

- X-Y-plots:
  - **plot(x,y)**:

```
MATLAB> t = 0:0.01:2*pi;
MATLAB> y = sin(t);
MATLAB> plot(t,y);
```



  - line style & color:
    plot(x,y,'*format*');
    → see help

- see also: xlabel('…'), ylabel('…'), title('…'), …

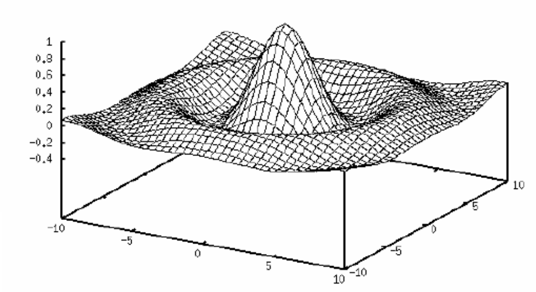
3D Plots

MATLAB> **X = ...**

MATLAB> **Y = ...**

MATLAB> **Z = f(X,Y)**

MATLAB> **mesh(X,Y,Z);**



3D Plots - Example

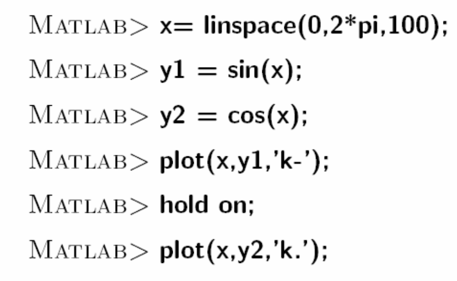
- MATLAB> **x = linspace(-10,10,40);**
  $\rightarrow x = (-10, \ldots, 10)$ % x has 40 elements, same def. for y

- MATLAB> **[X,Y] = meshgrid(x,y);**
  $\rightarrow X = \begin{pmatrix} -10 & \ldots & 10 \\ \vdots & \ddots & \vdots \\ -10 & \ldots & 10 \end{pmatrix}; Y = \begin{pmatrix} -10 & \ldots & -10 \\ \vdots & \ddots & \vdots \\ 10 & \ldots & 10 \end{pmatrix};$

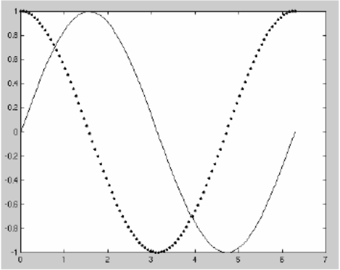
- MATLAB> **R = sqrt(X.^2+Y.^2)+0.1;**
  MATLAB> **Z = sin(R)./R;** % sinc-function
  MATLAB> **mesh(X,Y,Z)**

Multiple plots in one window

- HOLD ON holds the current plot and all axis properties so that subsequent graphing commands add to the existing graph.
- HOLD OFF returns to the default mode



```
MATLAB> x= linspace(0,2*pi,100);
MATLAB> y1 = sin(x);
MATLAB> y2 = cos(x);
MATLAB> plot(x,y1,'k-');
MATLAB> hold on;
MATLAB> plot(x,y2,'k.');
```

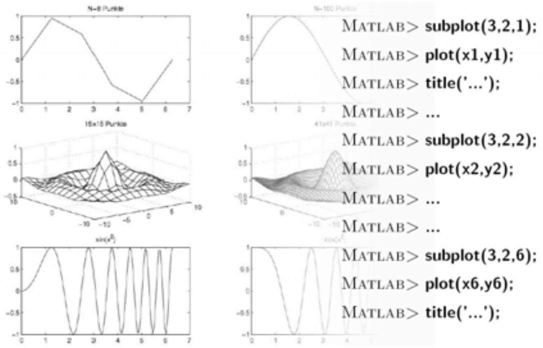


Selection of figure windows

- MATLAB> figure
  - opens a new figure window
- MATLAB> figure(2)
  - makes figure 2 the current figure, forces it to become visible, and raises it above all other figures on the screen.
  - if figure 2 does not exist, a new figure is created

Multiple figures in one window

- subplot(rows, columns, index) → index specifies the active window
- The axes are counted along the top row of the Figure window, then the second row, etc.



```
MATLAB> subplot(3,2,1);
MATLAB> plot(x1,y1);
MATLAB> title('...');
MATLAB> ...
MATLAB> subplot(3,2,2);
MATLAB> plot(x2,y2);
MATLAB> ...
MATLAB> ...
MATLAB> subplot(3,2,6);
MATLAB> plot(x6,y6);
MATLAB> title('...');
```

Task:

- Create a “chequerboard” matrix (8x8)
  - Commands “mod” and “abs” might help

```
A =

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