

DIP 55:148

Introduction to Matlab: 1) Basics



Matlab



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

DIP'07 / 2

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

DIP'07 / 3

Matlab – Script Files



- Scripts: all commands can be used – as in interactive mode
- Script files are ASCII-files (*.m)
 - e.g., program1.m
 - execution of script: *MATLAB> program1*
- Comments start with “%”
- A “;” at the end of a command suppresses the output of the result

DIP'07 / 4

THE UNIVERSITY OF IOWA



THE UNIVERSITY OF IOWA

- DIP'07 / 6

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)
```

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

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

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

DIP'07 / 7

Matrix Initialization



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

```
MATLAB> D = 1:5
```

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

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

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

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)
```

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

DIP'07 / 8

Manipulation of Matrix Elements



- Accessing single elements: $A(\text{row}, \text{column}) = \text{value}$
NOTE: index starts with 1!

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

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

- $A(1,:) \rightarrow$ gets first row
- $A(:,1) \rightarrow$ gets first column
- Accessing an undefined element expands the matrix!

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

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

DIP'07 / 9

Manipulation of Matrix Elements



- 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
```

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

DIP'07 / 10

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
```

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

DIP'07 / 11

Program Control Statements



- Conditional Control:
- Loop Control:

```
if logical_expression1  
...  
elseif logical_expression2  
...  
else  
...  
end
```

```
for variable = row_vector  
...  
end
```

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

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

DIP'07 / 12

Operators:



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)

DIP'07 / 13

Matlab Optimized Programming



- **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

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

www.dip.iowa.edu

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

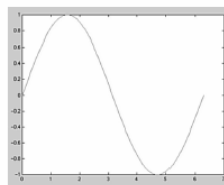
```
MATLAB> A = (1:4).^2           A = ( 1  4  9 16 );
```

DIP'07 / 15

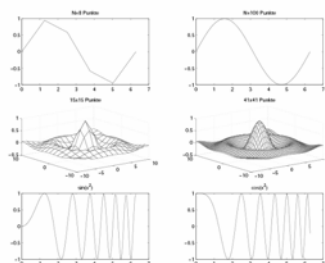
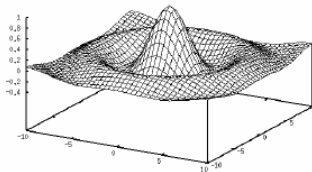
Plotting/Visualization



- 2D Plots



- 3D Plots



...multiple figures in one window

DIP'07 / 16

2D Plots

- X-Y-plots:

- **plot(x,y):**

```
MATLAB> t = 0:0.01:2*pi;
```

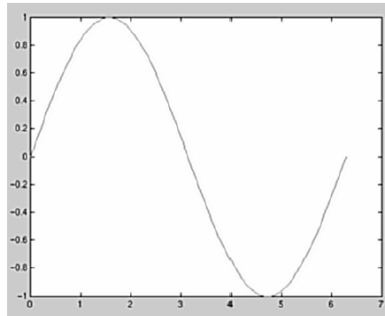
```
MATLAB> y = sin(t);
```

```
MATLAB> plot(t,y);
```

- line style & color:

```
plot(x,y,'format');
```

→ see help



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

DIP'07 / 17

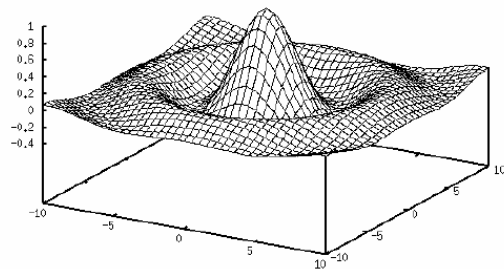
3D Plots

```
MATLAB> X = ...
```

```
MATLAB> Y = ...
```

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

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



DIP'07 / 18

3D Plots - Example



- MATLAB> **x = linspace(-10,10,40);**
→ $x = (-10, \dots, 10)$ % x has 40 elements, same def. for y
- MATLAB> **[X,Y] = meshgrid(x,y);**
→ $X = \begin{pmatrix} -10 & \dots & 10 \\ \vdots & \ddots & \vdots \\ -10 & \dots & 10 \end{pmatrix}; Y = \begin{pmatrix} -10 & \dots & -10 \\ \vdots & \ddots & \vdots \\ 10 & \dots & 10 \end{pmatrix};$
- MATLAB> **R = sqrt(X.^2+Y.^2)+0.1;**
MATLAB> **Z = sin(R)./R;** % sinc-function
MATLAB> **mesh(X,Y,Z)**

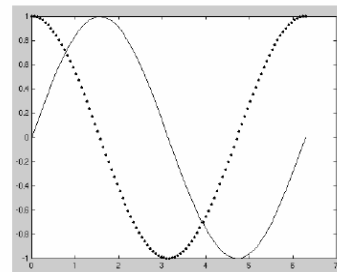
DIP'07 / 19

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-');
```



DIP'07 / 20

Selection of figure windows



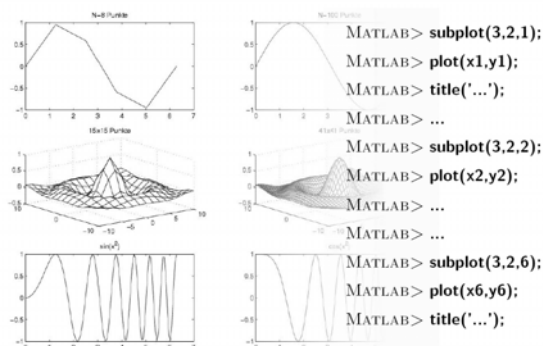
- MATLAB> figure
 - opens a new figure window
- MATLAB> figure(2)
 - makes figure 2 the current figure, forces it to become visible, and raises it above all other figures on the screen.
 - if figure 2 does not exist, a new figure is created

DIP'07 / 21

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.



DIP'07 / 22

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
```

DIP'07 / 23

Solution:



```
>> a = mod(1:8,2)
>> b = abs(a-1)
>> A = b'*b + a'*a

>> imshow(A,[])
```

DIP'07 / 24