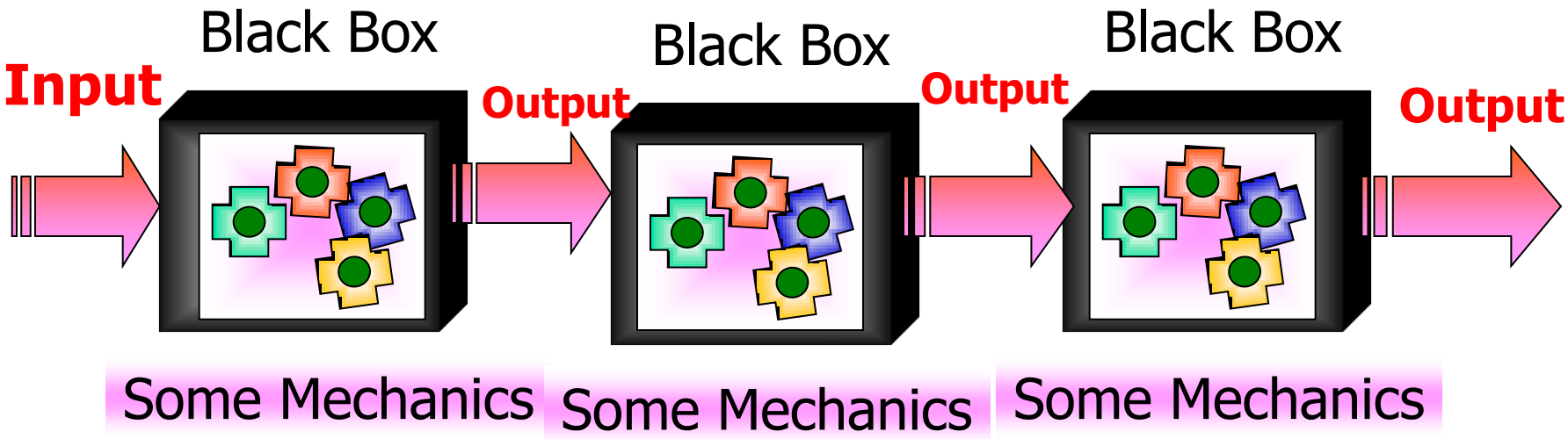
Chapter 5: User-Defined Functions

Yunong Zhang (张雨浓)

Email: zhynong@mail.sysu.edu.cn

Introduction to Functions

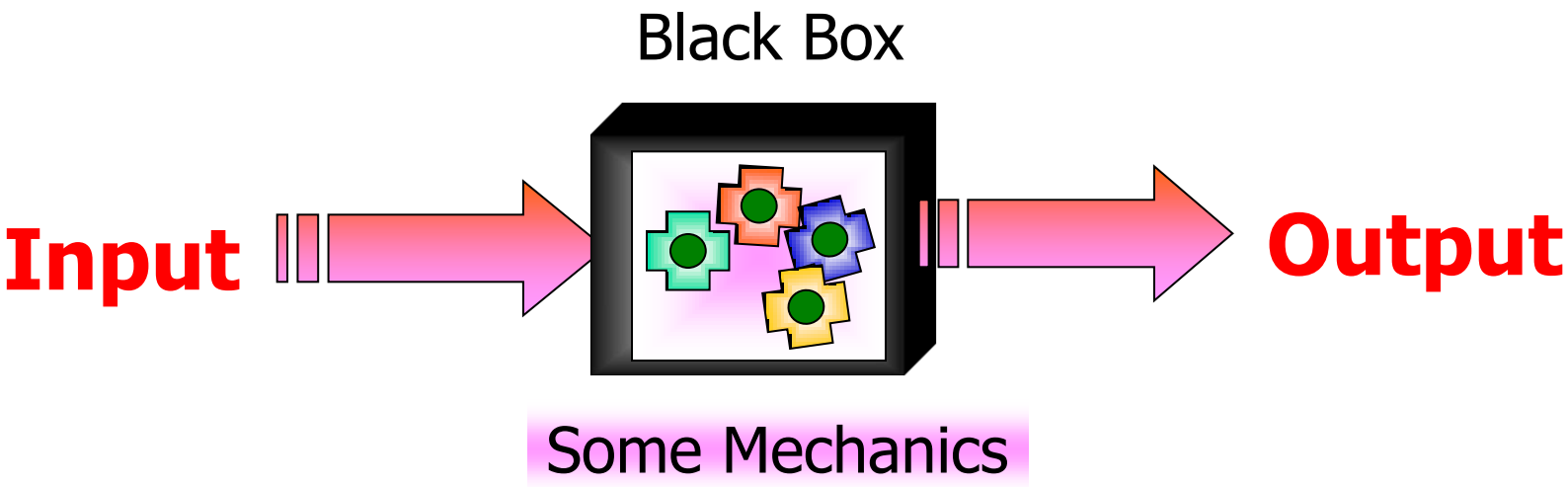


1) In reality, not only serial processing of the above kind, but also parallel processing of many other kinds.

<u>Please give us an example!! (No.1)</u>

2) But, at this point, we just learn this serial-processing paradigm from the reality.

Illustration of Functions



Each box is acted as a separated **function**

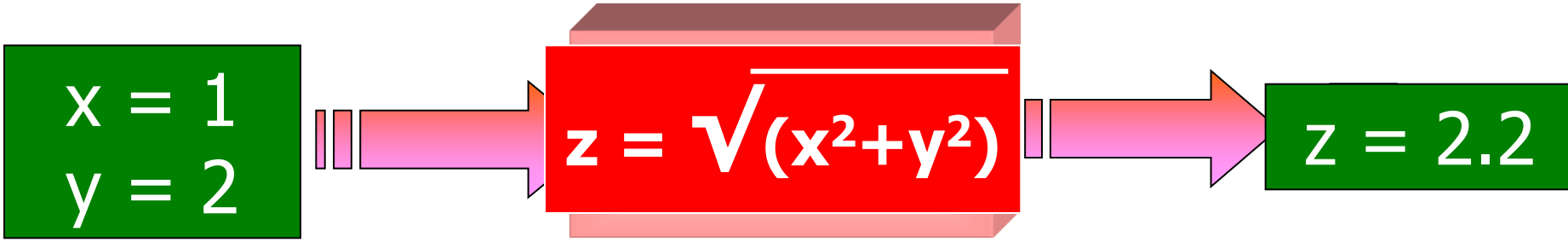
Benefits of using Functions

- Independent debugging and testing of subtasks
- Reusable code
- Isolation from unintended side effects

What function is it?

Square root

What function is it? (Cont.)



- Usually multiple inputs
- Single output (→Multiple outputs)
- More and more complicated

Functions

- Type of functions
  - <u>Built-in functions</u>

    (predefined in MATLAB)
  - <u>User-defined functions</u>

    (create your own function)

Built-in functions (I)

- Exponential functions
  - **exp(x)** = Exponential = e^x
  - **sqrt(x)** = Squart root = $\sqrt{x}$

- Logarithmic functions
  - **log(x)** = Natural log = ln x
  - **log10(x)** = 10-based log = log10(x)

Built-in functions (II)

- Trigonometric functions
  - **sin(x)**
  - **cos(x)**
- Inverse trigonometric
  - **acos(x)** = arccos x = $\cos^{-1} x$
  - **atan(x)** = arctan x = $\tan^{-1} x$
- Hyperbolic functions
  - **cosh(x)**= Hyperbolic cosine = cosh x = $(e^{x}+e^{-x})/2$

Functions

- Type of functions
  - Built-in functions *(predefined in MATLAB)*
  - User-defined functions *(create your own function)*

Function file format

- First line must begin with a **function definition**

```
function [output1,output2,...] = func_name(input1,input2,...)
% H1 comment line
% Other comment lines
Now is the executable code
```

- **input1,input2...: input arguments**
- **output1,output2...: output results**
- **H1 comment line is a line summary of the purpose of the function. It is searched and displayed by the lookfor command.**

Function file format (Cont.)

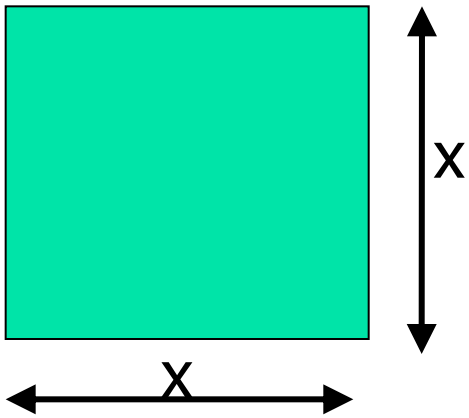
- help command shows the lines from H1 line until the first executable statement lines or the first blank line
- Function name **should** be the same as **.m** file

  function [outputs] = Test1(input1,input2,...)

  Should be save as "Test1.m"
- MATLAB is case sensitive!
- Calling a function by typing its name directly in the command window, or by including it in a script or another function

What does ``script" mean?

User-defined functions



Write a function to find the **area of the field**:

Area = x * x

User-defined functions (Cont.)

Area.m

```
function y = Area(x)
% This function is to find the area of a field
% Area of square box is x*x
y = x * x
```

```
>>result = Area(2)
y =
    4
result =
    4
```

User-defined functions (Cont.)

```
>> help Area

This function is to find the area of a field
Area of square box is x*x
```

User-defined functions (Cont.)

>> lookfor Area

Area.m: % This function is to find the area of a field

POLYAREA Area of polygon.

RECTINT Rectangle intersection area.

AREA Filled area plot.

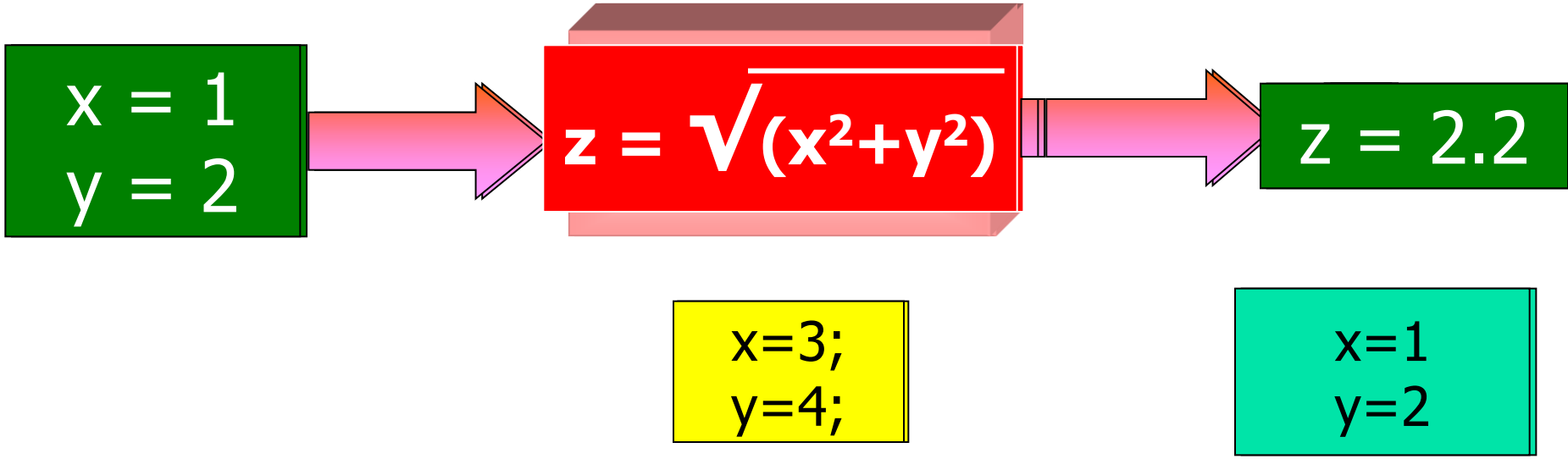
DATAREAD Read formatted data from string or file.

User-defined functions (Cont.)

```
>> lookfor area
Area.m: % This function is to find the area of a field
POLYAREA Area of polygon.
RECTINT Rectangle intersection area.
AREA  Filled area plot.
DATAREAD Read formatted data from string or file.
BWAREA Compute the area of objects in binary image.
BWAREAOPEN Binary area open; remove small objects.
ASSEMA Assembles area integral contributions in a PDE problem.
PDEONAX Checks if current pointer position is inside PDE Toolbox axes area
PDETRIDI Side lengths and areas of triangles.
LAR2RC Convert log area ratios to reflection coefficients.
RC2LAR Convert reflection coefficients to log area ratios.
 UPDATE_ANALYSISFRAME Updates the plotting area and calls appropriate
HDFVDATAREAD read HDF Vdata
 RENDER_ANALYSISAREA Renders the frame where all the analysis plots and
UPDATE_PLOT Update plot area for quantized filters.
>> |
```

Function Workspace

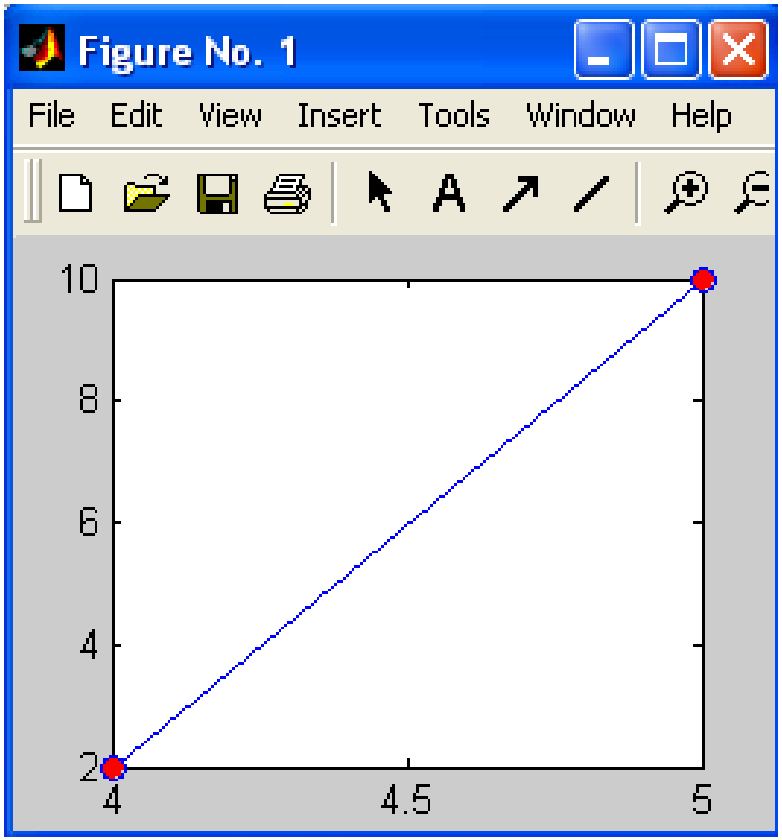
Black box



cf. system identification

Function Workspace--Example

- Calculate the distance between points A(ax,ay) and B(bx,by) in Cartesian coordinates



ax=4
ay=2
bx=5
by=10

What does it mean by ``Cartesian coordinates''?

Function Workspace - Example (Cont.)

Function Workspace-Example (Cont.)

C:\MATLAB6p5\work\matlab_programming\p190.m

File Edit View Text Debug Breakpoints Web Window Help

```
%script file: p190.m is used to calculate the distance
%between two points.
% points: A(ax, ay), B(bx, by)

ax=4;
ay=2;
bx=5;
by=10;

result=distance(ax, ay, bx, by);
fprintf('The distance between A and B is %f\n',result);
```

p190.m distance.m

script Ln 9 Col 1

Function Workspace-Example (Cont.)

C:\MATLAB6p5\work\matlab_programming\...

File Edit View Text Debug Breakpoints Web Window Help

```
1  function dis=distance(x1,y1,x2,y2)
2  %Calculate the distance between two points
3  %where A(x1,y1) and B(x2,y2)
4  dis=sqrt((x2-x1)^2+(y2-y1)^2);
```

p190.m distance.m

distance Ln 2 Col 27

Function Workspace-Example (Cont.)

Workspace

Stack: p190

Name	Size	Bytes	Class
ax	1x1	8	double array
ay	1x1	8	double array
bx	1x1	8	double array
by	1x1	8	double array

Workspace

Stack: distance

Name	Size	Bytes	Class
dis	1x1	8	double array
x1	1x1	8	double array
x2	1x1	8	double array
y1	1x1	8	double array
y2	1x1	8	double array

Workspace

Name	Size
ax	1x1
ay	1x1
bx	1x1
by	1x1
result	1x1

ax=4
ay=2
bx=5
by=10

$$dis = \sqrt{(x2-x1)^2+(y2-y1)^2}$$

result=8.06

script file:
p190.m

function distance.m

script file:
p190.m

MATLAB

File Edit View Web Window Help

Desktop Layout
Undock Workspace
Command Window
Command History
Current Directory
Workspace
Launch Pad
Help
Workspace View Options
Current Directory Filter

Current Directory: C:\MATLAB6p1\work

Current Dir

C:\MATL

es	st Modified	Description
thtime1.m	九月-2007 10:52 上午	plot(eigA)
thtime2.m	九月-2007 11:06 上午	plot(eigA)
.tif	二月-2008 02:42 下午	
opts	六月-2004 09:26 下午	

Workspace

Stack: Base

Name	Size	Bytes	Class
ans	1x1	8	double array

Launch Pad Workspace

Command Window

```
ans =

    36

??? Input argum

Error in ==> C:
On line 4  ==> 
>> Area(6)

y =

    36


ans =

    36

>>
```

MATLAB

File Edit View Web Window Help

C:\MATLAB6p1\work\Area.m

File Edit View Text Debug Breakpoints Web Window Help

Step	F10
Step In	F11
Step Out	Shift+F11
Continue	F5
Go Until Cursor	
Exit Debug Mode	

```
1  function y
2  % This fun
3  % Area of
4  x
5  y = x * x
```

Ready

Command Window

```
To get started

>> Area(6)
K>>

x =

     6


y =

    36

K>>
```

Workspace

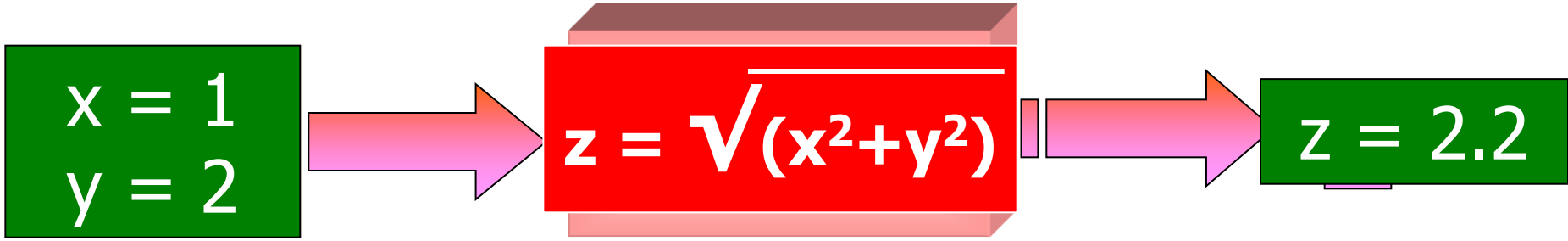
Stack: Area

Name	Size	Bytes	Class
x	1x1	8	double array
y	1x1	8	double array

Launch Pad Workspace

Variable Passing in Matlab

- Communicating with functions using a pass-by-value scheme




```
x=3;
y=4;
```

```
x=1
y=2
```

Variable Locality and Passing

p190.m

```
a=2;
b=[6 4];
fprintf('Before sample: a=%d,b=%f %f\n',a,b);
a=b(1)+2*a;
b=a.*b;
out=a+b(1);
fprintf('After sample: a=%d,b=%f %f\n',a,b);
```

```
>> p190
Before sample: a=2,b=6.000000 4.000000
After sample: a=10,b=60.000000 40.000000
```

Variable Passing and Locality (Cont.)

p191.m

```
a=2;
b=[6 4];
fprintf('Before sample: a=%d,b=%f %f\n',a,b);
out=sample(a,b);
fprintf('After sample: a=%d,b=%f %f\n',a,b);
```

sample.m

```
function out=sample(a,b)
fprintf('In sample: a=%d,b=%f %f\n',a,b);
a=b(1)+2*a;
b=a.*b;
out=a+b(1);
fprintf('In sample: a=%d,b=%f %f\n',a,b);
```

Variable Passing and Locality (Cont.)

```
>> p191
Before sample: a=2,b=6.000000 4.000000
In sample: a=2,b=6.000000 4.000000
In sample: a=10,b=60.000000 40.000000
After sample: a=2,b=6.000000 4.000000
```

Sincere Thanks!

- Using this group of PPTs, please read

- [1] Yunong Zhang, Weimu Ma, Xiao-Dong Li, Hong-Zhou Tan, Ke Chen, MATLAB Simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs, Neurocomputing 72 (2009) 1679-1687

- [2] Yunong Zhang, Chenfu Yi, Weimu Ma, Simulation and verification of Zhang neural network for online time-varying matrix inversion, Simulation Modelling Practice and Theory 17 (2009) 1603-1617