

Introduction to Scientific and Engineering Computing, BIL108E

INTRODUCTION TO SCIENTIFIC & ENGINEERING COMPUTING BIL 108E, CRN24023

Dr. S. Gökhan Karaman

Technical University of Istanbul

March 01, 2010



TENTATIVE SCHEDULE

Introduction to Scientific and Engineering Computing,			
BIL108E	Week	Date	Topics
	1	Feb. 10	Introduction to Scientific and Engineering Computing
Karaman	2	Feb. 17	Introduction to Program Computing Environment
	3	Feb. 24	Variables, Operations and Simple Plot
	4	Mar. 03	Algorithms and Logic Operators
	5	Mar. 10	Flow Control, Errors and Source of Errors
	6	Mar. 17	Functions
	6	Mar. 20	Exam 1
	7	Mar. 24	Arrays
	8	Mar. 31	Solving of Simple Equations
	9	Apr. 07	Polynomials Examples
	10	Apr. 14	Applications of Curve Fitting
	11	Apr. 21	Applications of Interpolation
	11	Apr. 24	Exam 2
	12	Apr. 28	Applications of Numerical Integration
	13	May 05	Symbolic Mathematics
	14	May 12	Ordinary Differential Equation (ODE) Solutions with Built-in Functions

LECTURE # 4

Introduction to Scientific and Engineering Computing, BIL108E

- 1 NUMERICAL DATA TYPES IN MATLAB
- 2 ERRORS
- **3** LOOPS
- 4 ALGORITHMS
- **5** DATA ANALYSIS



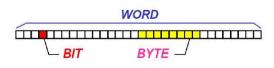
BITS, BYTES AND WORDS

Introduction to Scientific

Engineering

Computing, BIL108E VARIABLES ARE REPRESENTED BY WORDS, COMPOSED OF BYTES, COMPOSED OF BITS

- BIT = elemental circuit, ON (1) / OFF (0)
- **BYTE** = string of **8** BITS
- WORD = string of N BYTES (partially controllable by the programmer)





BITS, BYTES AND WORDS

Introduction to Scientific and Engineering Computing, BIL108E

base 10	conversion	base 2
1	$1 = 2^{0}$	0000 0001
2	$2 = 2^1$	0000 0010
4	$4 = 2^2$	0000 0100
8	$8 = 2^3$	0000 1000
9	$8 + 1 = 2^3 + 2^0$	0000 1001
10	$8 + 2 = 2^3 + 2^1$	0000 1010
27	$16 + 8 + 2 + 1 = 2^4 + 2^3 + 2^1 + 2^0$	0001 1011

one byte = 8 bits



NUMERICAL DATA TYPES IN MATLAB

Introduction to Scientific and Engineering

Computing,

BIL108E

NUMERICAL DATA TYPES IN MATLAB

- int8(-128,127), int16(-32768, 32767), int32(- 2 147 483 648, 2 147 483 647), int64(-9 223 372 036 854 775 808, 9 223 372 036 854 775 807), uint8(0, 255), uint16(0, 65535), uint32(0, 4 294 967 295), uint64(0, 18 446 744 073 709 551 615)
- single(2^{-126} , 3.4×10^{38})
- double(2.2251×10^{-308} , 1.7977×10^{308})

NUMERICAL DATA TYPES IN MATLAB

Introduction to Scientific and Engineering Computing, BIL108E

REAL NUMBERS, FLOATING-POINT NUMBERS

- Real numbers, \mathbb{R} .
- Floating-point numbers, 𝔽.

Only a subset ${\mathbb F}$ of finite dimension ${\mathbb R}$ can be represented.

Any real number x is truncated by the machine as fl(x).



NUMERICAL DATA TYPES IN MATLAB

Introduction to Scientific and Engineering Computing, BIL108E

FLOATING-POINT NUMBERS

- Numeric values with non-zero fractional parts are stored as floating point numbers.
- All floating point values are represented with a normalized scientific notation.

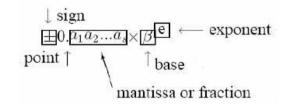


NUMERICAL DATA TYPES IN MATLAB

FLOATING-POINT NUMBERS

Floating-point number representation of a real number $x = (-1)^s \times (0.a_1a_2a_3...a_t) \times \beta^e$

 $a_1
eq 0$





NUMERICAL DATA TYPES IN MATLAB

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLES;

 $12.7887 = 0.127887 \times 10^2$ (base 10) $-0.099 = -0.99 \times 10^{-1}$ (base 10)

DIGITAL STORAGE OF INTEGERS

Introduction to Scientific and Engineering Computing, BIL108E

INTEGERS

DIGITAL STORAGE OF INTEGERS

- Integers can be exactly represented by base 2
- Typical size is 16 bits
- 32 bit and larger integers are available

Note: All standard mathematical calculations in Matlab use floating point numbers.

DIGITAL STORAGE OF NON-INTEGER NUMBERS



BIL108E

Floating point values have fixed number of bits allocated for storage of the mantissa and fixed number of bits allocated for storage of the exponent.

- Two common precisions are provided in numerical computing: single precision and double precision.
- Fixed number of bits are allocated to each number: single precision uses 32 bits per floating point number and double precision uses 64 bits per floating point number



IEEE STANDARD

Introduction to Scientific and Engineering Computing, BIL108E

Total number of bits are split into separate storage for both the mantissa and the exponent.

■ single precision: 1 sign bit, 8 bit exponent, 23 bit mantissa

Single precision	±	exp	mantissa
32 bits	1	8	23

■ double precision: 1 sign bit, 11 bit exponent, 52 bit mantissa

Double precision	±	exp	mantissa	
64 bits	1	11	52	

DIGITAL STORAGE OF NUMBERS

Introduction to Scientific and Engineering Computing, BIL108E

- Limiting the number of bits allocated for storage of the exponent means that there are upper and lower limits on the magnitude of floating point numbers
- Limiting the number of bits allocated for storage of the mantissa means that there is a limit to the precision (number of significant digits) for any floating point number.

ERRORS

Introduction

to Scientific

and

Enginee

Comput BIL10

EXAMPLE:

Δ	MATLAB 7.6.0 (R2008a)	
Eile Edit Debug Desk	top Window Help	
🗋 🖆 👗 🐂 🛅 🤊 (🖭 谢 🗊 🖹 🥝 Current Directory. /media/Transcend/source 🕶 🛄 🔞	
Current Directory 🗰 🗖 🗖	× W Command Window	91 🔲
🔁 🖆 👪 🖥 🔸 👘	New to MATLA8? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Started</u> .	
All Files ∠ T	<pre>vpe >> type ex_04_01</pre>	
	1-file	
	A-file	
	A-file x = 1.e-15;	
	$f_{\rm file}$ ((1 + x) - 1) / x	
	1-file >> ex_04_01	
	1-file	
	A-file ans =	
	1-file	
	1.1102	
	ine line	
	1-file	
≦ex_04_13_4.m N	A-file >>	
1 ex 04 13 5 m N	A_file •	
Command History I++		
ex_04_01		
clc		
type ex_04_	01	
ex_04_01		
ex_04_01		

ERRORS

Introduction to Scientific and Engineering Computing, BIL108E

- **1** PP Physical Problem
- MP Mathematical Problem
- **3** NP Numerical Problem

Each of these steps involve errors.

ERRORS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE:

Computational Errors

$$f(x) = (x - 1)^7$$

$$f(x) = x^7 - 7x^6 + 21x^5 - 35x^4 + 35x^3 - 21x^2 + 7x - 1$$



ERRORS

Introduction to Scientific and Engineering

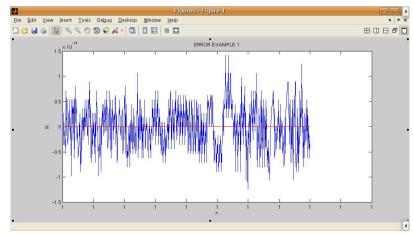
gineering mputing,	EXAMPLE:		
	A	MATLAB 7.6.0 (R2008a)	
BIL108E	Eile Edit Debug Desktop Window	w Help	
,	1 6 × h 6 v e h d	👔 🕜 Current Directory /media/Transcend/source 👻 🖻	
laraman	Current Directory +	Command Window	* 🗆 *
	🖻 🖆 🖪 🗟 🔹	New to MATLA8? Watch this Video, see Demos, or read Getting Started.	
		>> type ex_04_02	
	Mex 03 1.m M-file	s cype ex_04_02	
	🕙 ex_04_0.m M-file		
		x = linspace(1-2e-8, 1+2e-8,801);	
	CALOT OTAIL IN THE	% x_min = 0.99999998	
		% x_max = 1.00000002	
		$Fx = x \cdot \sqrt{7} - 7 \cdot x \cdot \sqrt{6} + 21 \cdot x \cdot \sqrt{5} - 35 \cdot x \cdot \sqrt{4} + 35 \cdot x \cdot \sqrt{3} - 21 \cdot x \cdot \sqrt{2} + 7 \cdot x$	- 1;
	ex_04_10.m M-file	olot(x,fx), title('ERROR EXAMPLE 1')	
	ex_04_10.m~ Editor	klabel('x')	
		/label('fx')	
		>>	
	12 ex_04_13_2.m M-file		
	Mex 04 13 3.m M-file		
	Nex 04 13 4 m M-file		
	Command History + - * ×		
	ex_04_02		
	clc		
	type ex_04_02		
	4 Start		10

Records of the second s

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE:

ERRORS





```
Introduction
to Scientific
and
Engineering
Computing,
BIL108E
```

man

EXAMPLE:

ERRORS

Calculation of pi

$$z_2 = 2, z_{n+1} = 2^{n-1/2} \sqrt{1 - \sqrt{1 - 4^{1-n} z_n^2}}$$

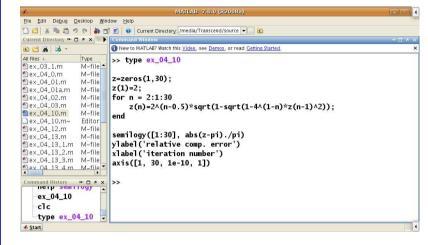
 $n = 2, 3, \dots$



ERRORS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE:

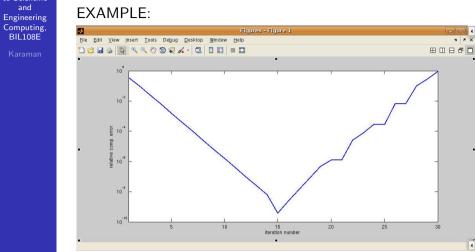


ERRORS

Introduction to Scientific

and

BIL108E





ERRORS

Introduction to Scientific and Engineering Computing, BIL108E

COMPUTATIONAL ERRORS

- ec: computational Error
- **x**: exact solution of mathematical model
- $\hat{\mathbf{x}}$: numerical solution of mathematical model

Absolute Computational Error

$$\mathbf{e}_{\mathbf{c}}^{\mathbf{abs}} = |\mathbf{x} - \mathbf{\hat{x}}|$$

Relative Computational Error

$$\mathbf{e}_{\mathbf{c}}^{\mathrm{rel}} = |\mathbf{x} - \mathbf{\hat{x}}| / |\mathbf{x}|$$



ERRORS



ERRORS RESULTING FROM PROBLEMS

- SYNTAX ERRORS
- LOGIC ERRORS
- ROUNDOFF ERRORS

ERRORS



SYNTAX ERRORS

- Typo errors.
- Incompatible vector sizes.
- Name hiding (try "help command").



ERRORS

Introduction to Scientific and Engineering

Computing,

BIL108E

LOGIC ERRORS

- Try to run the program for some special cases where you know the answer.
- If you don't know any exact answer, use your insight to check whether the answer seems to be of the right order of magnitude.
- Try working through the program by hand to see if you can spot where things start going wrong.

E

Introduction to Scientific and Engineering Computing, BIL108E

ERRORS

ROUNDING ERRORS

- Finite-precision leads round-off in individual calculations
- Effects of round-off accumulate slowly
- The round-off errors are inevitable, solution is to create better algorithms
- Subtracting nearly equal may lead to severe loss of precision



ERRORS

Introduction to Scientific and Engineering

Computing, BIL108E

MACHINE PRECISION

The magnitude of roundoff errors is quantified by machine precision ϵ_{M}

There is a number, ϵ_M such that

 $1+\delta=1$

whenever $\delta < \epsilon_M$ In exact arithmetics, ϵ_M is identically 0.

 $\texttt{eps}=2.2204\times 10^{-16}$ in Matlab



NUMERICAL DATA TYPES IN MATLAB

Introductio to Scientific and Engineering Computing, BIL108E

FLOATING-POINT NUMBERS

Roundoff-Error

 $\frac{|x - fl(x)|}{|x|} \le \frac{1}{2} \epsilon_M$

Ref: Standard for Floating Point Arithmetic P754, IEEE. $\epsilon_M = \beta^{1-t}$, here t is the distance between 1 and its closest floating-point number greater than 1.

In Matlab ϵ_M is obtained through the command eps. Number 0 does not belong to \mathbb{F}



NUMERICAL DATA TYPES IN MATLAB

roduction	
Scientific	
and	
gineering	
omputing,	
311 108E	

FLOATING-POINT NUMBERS

realmin. and realmax

If x is less than x_{min} is treated as 0, UNDERFLOW

If x is greater than x_{max} Inf OVERFLOW

The elements in \mathbb{F} are more dense near x_{min} , and less dense while approaching x_{max} .

Introduction to Scientific and Engineering Computing, BIL108E

NUMERICAL	DATA	TYPES	IN	MATLAB

EXAMPLE:

Bile Edit Degug Desko Window Help Bile Edit Degug Desko Window Bile Edit Degug Desko Window Command Heroson Bile Edit Degug Desko Window Bile Edit Degug Desko Window Command Heroson Bile Edit Degug Desko Window Bile Edit Degug Desko Window Command Heroson Bile Edit Degug Desko Window	
Current Directon Command Window Carent Directon Command Window Command Window	
Image: Second State Image: Second State Image: Second State Image: Second State Image: Second State Image: Second State Image: Second State Image: State Image: Second State	
Image: Constraint of the second of the se	* 🗆 *
ex.04.0.m M-file ex.04.0.m M-file ex.04.0.m M-file ex.04.0.m M-file ex.04.0.m M-file ex.04.0.m M-file ex.04.10.m M-file ex.04.10.m M-file ex.04.10.m M-file ex.04.11.m M-file ex.04.13.m M-file ex.04.13.4.m M-file ex.04.13.5.m M-file ex.04.13.4.m M-file ex.04.13.4.m M-file ex.04.13.4.m M-file ex.04.13.4.m M-file ex.04.13.4.m M-file ex.04.13.4.m M-file	
ex.04.0.m M-file ex.04.10.m M-file ex.04.10.m M-file ex.04.10.m M-file ex.04.13.m M-file ex.04.13.3.m M-file	
ex.04_01.m M-file ans = ex.04_02.m M-file 2.2204e-16 ex.04_03.m M-file 2.2204e-16 ex.04_01.m M-file >> realmin ex.04_01.3.m M-file >> realmin ex.04_13.2.m M-file >> realmin ex.04_13.2.m M-file >> 2.2251e-308	
ex_04_0.0.m M-file ex_04_0.0.m M-file ex_04_0.0.m M-file ex_04_10.m M-file ex_04_12.m M-file ex_04_13.m M-file ex_04_13.m M-file ex_04_13.2.m M-file ex_04_13.2.m M-file ex_04_13.2.m M-file ex_04_13.2.m M-file ex_04_13.3.4.m M-file ex_04_13.4.m M-file ex_04_13.4.m M-file	
ex. 04. 03.m M-file 2.2204e-16 ex. 04. 12.m M-file x-realmin ex. 04. 13.m M-file x-realmin ex. 04. 13.2.m M-file x-realmin ex. 04. 13.2.m M-file x-realmin ex. 04. 13.2.m M-file x.2251e-308	
ex_04_10.m M-file ex_04_113.m M-file ex_04_13.1.m M-file ex_04_13.2.m M-file ex_04_13.3.m M-file ex_04_13.3.m M-file ex_04_13.4.m M-file ex_04_13.5.m M-file	
ex_04_12.m M-file >> realmin ex_04_13.1.m M-file ans = ex_04_13.2.m M-file 2.2251e-308 ex_04_13.5.m M-file 2.2251e-308	
ex.04.13.m M-file ex.04.13.lm M-file ex.04.13.2.m M-file ex.04.13.2.m M-file ex.04.13.4.m M-file ex.04.13.5.m M-file	
ex_04_13.m M-file ex_04_13.1.m M-file ex_04_13.2.m M-file ex_04_13.3.4.m M-file ex_04_13.5.m M-file ex_04_13.5.m M-file	
ex_04_13_2.m M-file ans ex_04_13_3.m M-file 2.2251e-308 ex_04_13_5.m M-file 2.2251e-308	
ex_04_13_4.m M-file ex_04_13_5.m M-file ex_04_13_5.m M-file ex_04_13_5.m M-file	
ex_04_13_4.m M-file 2.2251e-308	
ans =	
eps ans =	
1.7977e+308	
-realmax 🚽	



ERRORS



Computing, BIL108E

TRUNCATION ERROR

Example:

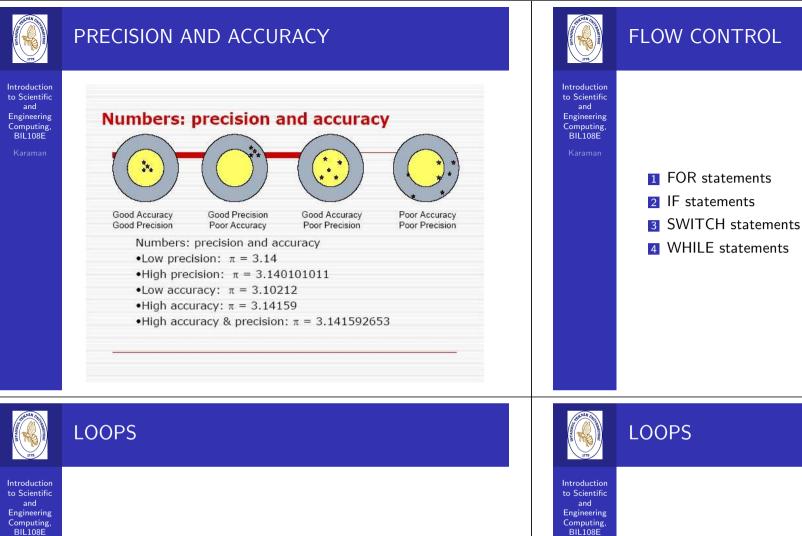
Consider the series for *sinx*

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

For small x, only a few terms are needed to get an accurate aproximation to sinx. The higher order terms are truncated.

 $f_{true} = f_{sum} + truncation \ error$

The size of truncation error depends on x and the number of terms included in f_{sum}



• j: k is a vector with elements j, j + 1, j + 2, ..., k.

that the last element can not exceed k.

j:m:k.

• j: m: k is a vector with elements j, j + m, j + 2m, ... such

index must be a variable. Each time through the loop it will contain the next element of the vector *j* : *k* or

REPETITIVE TASKS

Loops are used for repetitive tasks.

Basic for Construct

The most common form of the loop is;

for index = j : kstatements

end



LOOPS

index = *first* : *increment* : *last*

The number of times that the loop is executed is defined as iteration:

iteration = floor($\frac{last-first}{increment}$) + 1 Here floor(x) is a function, that rounds x down toward $-\infty$

This value is called iteration or trip count



LOOPS

Introduction to Scientific and

Engineering

Computing, BIL108E

- On completion of the for loop the index contains the last value used.
- If the vector j : k or j : m : k is empty, statements are not executed and control passes to the statement following end.
- If the index does appear explicitly in *statements*, the for can often be vectorized. It runs faster.
- It is good programming style to indent (tabulate) the statements inside a for loop.

A TANK CHARTER

Introduction to Scientific and Engineering Computing, BIL108E

for in a single line

for index = j : k, statements, end

or

LOOPS

for index = j : m : k, statements, end

Don't forget the commas.



LOOPS

Introduction to Scientific and Engineering Computing, BIL108E

More general form of the for is

```
for index = v
```

Here v is any vector.

The index moves through each element of the vector.



LOOPS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLES:

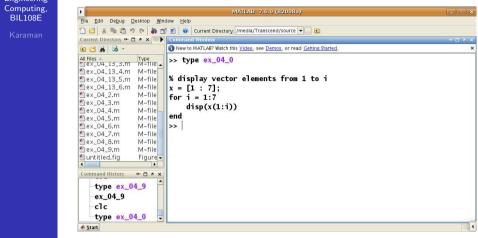
% display vector elements from 1 to i x = [1 : 7];for i = 1:7 disp(x(1:i))

end



LOOPS

Introduction to Scientific and EXAMPLE: Engineering



to Scientific and Engineering Computing, BIL108E

Introduction

LOOPS

A					MATLAB	7.6.0 (R	2008a)			- 0
<u>File Edit Debug</u>	Desktop W	(indow <u>H</u>	elp							
06 8 6	2 1 2	21 E	Curre	nt Director	ry: /media/	Transcend/:	source 👻 .	🙃		
Current Directory ++	W × S □	Comm	and Wind	low						н 🗆 Х.
🖻 🖆 🖪 😼 •		Nev	to MATL	AB? Watch	this <u>Video</u> .	see Demos	, or read <u>G</u>	etting Starte	d.	
All Files ∠ ex_03_1.m ex_04_0.m ex_04_01.m ex_04_02.m	Type M-file M-file M-file M-file	>> 6	ex_04_ 1	.0						
<pre></pre>	M-file M-file M-file M-file	_	1	2	3					
▲ ex_04_13_1.m ▲ ex_04_13_2.m	M-file M-file		1	2	3	4	-			
●ex_04_13_3.m ●ex_04_13_4.m ●ex_04_13_5.m	M-file	-	1	2	3 3	4 4	5	6		
Command History	4 < 5 🗆 ++		1	2	3	4	5	6	7	
realmin realmax clc ex_04_0		>> •	I	2	3	4	2	0	7	
📣 <u>S</u> tart										0



Engineering

LOOPS

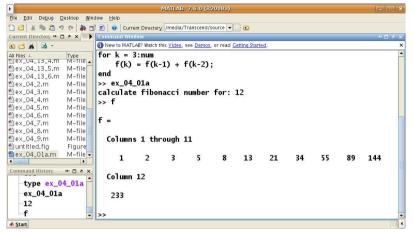
Introduction to Scientific

EXAMPLE:

Computing,	MATLAB: 7.6.0 (R2008a)	_ 0 X
BIL108E	File Edit Debug Desktop Window Help	
	🗋 😂 🐇 🍡 🖻 🤊 (*) 🐞 🗊 🖻 🛛 Current Directory. /media/Transcend/source 🖣 🛄 😰	
Karaman		** 🗆 * ×
	😢 🚔 🗟 🔸 🚺 New to MATLA8? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Starled</u> .	×
	AllFiles / Type ex_04_01a	
	1 ex_04_13_5.m M-file % ***********************************	
	🖺 ex_04_2.m M-file % This program calculates	
	ex_04_3.m M-file % Fibonacci number for a given number.	
	© ex_04_4.m M-file % 25.02.2010	
	↑ ex_04_5.m M-file % ***********************************	
	1 ex_04_7.m M−file %	
	<pre>ex_04_8.m M-file num = input('calculate fibonacci number for: ');</pre>	
	f(2) = 2;	
	for k = 3:num	
	Command History $+ \Box + x$ $f(k) = f(k-1) + f(k-2);$	
	type ex_04_01a ex_04_01a clc type ex_04_01a •	
	▲ Start	0



Introduction to Scientific and Engineering Computing, BIL108E





Introduction to Scientific EXAMPLES: and Engineering Computing,

Vectorize if possible

BIL108E

Evaluate the expression given below without the formula for the sum.



clock function

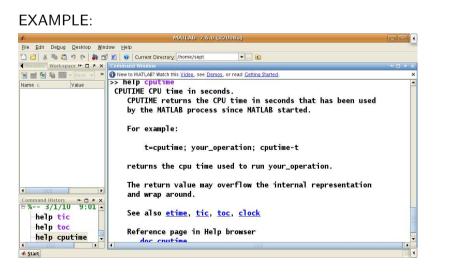
returns a six element vector.

etime function

returns the time in seconds between its two arguments.

LOOPS

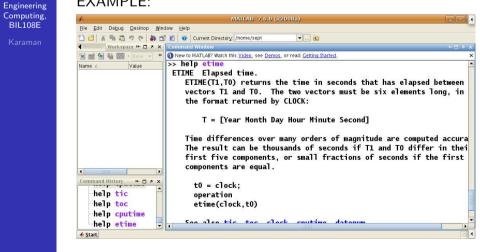
Introduction to Scientific and Engineering Computing, BIL108E



LOOPS

Introduction to Scientific and

EXAMPLE:





LOOPS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLES

with for loop

t0 = clock;s = 0;for n = 1 : 100000s = s + n;end etime(clock, t0)



LOOPS

Introduction to Scientific

EXAMPLE:

outing,	MATLAB 7.6.0 (R2008a)	
108E Elle Edit Debug Desktop	<u>Window</u> Help	
aman 🗋 😂 👗 🛅 🤊 😋	🐌 🗊 🗐 🥹 Current Directory /media/Transcend/source 👻 🛄 致	
Current Directory 🏎 🗖 🤻 🗙		91 🗖 🖉
🔁 🗃 🖬 🗟 🔸	New to MATLAB? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Started</u> .	
All Files 4 Type	>> type ex_04_2	
ex_04_13_4.m M−t 1 ex_04_13_5.m M−t		
Mex_04_13_6.m M-f		
1 ex_04_2.m M-f		
🖸 ex_04_3.m M-f		
🖺 ex_04_4.m M-f	ile	
🔮 ex_04_5.m M-f	The second se	
🖺 ex_04_6.m M-f	iie iii	
Mex_04_7.m M−f Mex_04_8.m M−f		
ex_04_8.m M−f ex_04_9.m M−f		
Suntitled.fig Figu	179	
	ile ans =	
Command History 🗰 🗖	a x 0.0057	
ex_04_10		
Construction and the second second second second	>>	
clc		
type ex_04_2	773	
ex042		
📣 Start		0



Introduction to Scientific Engineering Computing, BIL108E

EXAMPLES

with vectorization

t0 = clock;n = 1 : 100000;s = sum(n);etime(clock, t0)



Engineering

Computing BIL108E

LOOPS

Introduction to Scientific

EXAMPLE:

•	MATLAB 7.6.0 (R2008a)	
Eile Edit Debug D	esktop <u>Window</u> Help	
1 6 8 1 6 9	🕴 🏷 🗊 😰 🖉 Current Directory: /media/Transcend/source 👻 🛄 💽	
Current Directory 🗰 🗖	Command Window	** 🗖
🖻 🖆 🖪 🗟 •	New to MATLAB? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Started</u> .	
All Files ∠ ■ ex_04_13_4.m ■ ex_04_13_5.m	Type MITTRE >> type ex_04_3 MITTRE	
ex_04_13_5.m	M-file t0 = clock;	
1 ex_04_2.m	M-file n = 1 : 100000:	
🖺 ex_04_3.m	M-file s = sum(n);	
🖺 ex_04_4.m	M-tile stime(clock +0)	
1 ex_04_5.m	M-ITIE	
ex_04_6.m	M-file >> ex_04_3	
1 ex_04_7.m 1 ex_04_8.m	M-file ans =	
1 ex_04_0.m	M-file	
● untitled.fig ● ex_04_01a.m	Figure M−file → →	
Command History	<pre><< x s □ ++</pre>	
ex_04_2		
Charles and a second		
clc		
type ex_04	4_3	
ex_04_3		
📣 Start		



LOOPS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLES

LOOPS

tic and toc functions

monitor the time to interpret MATLAB statements Evaluate;

 $\sum_{n=1}^{100000} \frac{1}{n^2}$



Introducti

to Scientif

Computin

BIL108E

and Engineerir

LOOPS

LOOPS

A	MATLAB 7.6.0 (R2008a)	
	2esktop <u>Window</u> Help	
	Command Window	
🖻 🖆 🖪 🗟 🔹	New to MATLAB? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Started</u> .	
All Files 🗠	Type >> help tic	
😂 .adobe	Folder TIC Start a stopwatch timer.	
🗀 .audacity-data	Folder TTC and TOC functions work together to measure alansed ti	0
audacity1.3	Folder	
avidemux		
cache compiz	Folder the elapsed time. The sequence of commands:	
confia		
.cryptonit	Folder	
.cups	Folder	
idbus	Folder	
🗀 .elisa	Folder	
🗀 .eqe	Folder measures the amount of time MATLAB takes to complete the	ne
euler	Folder or more operations specified here by "operations" and dis	
N HICKNESS N		,
Command History		
help cont		

Introduction to Scientific and Engineering Computing, BIL108E

DILIUGE

EXAMPLE: MATLAB 7.6.0 (R2008a) File Edit Debug Desktop Window Help • ... 🗈 🎦 🗃 😹 ங 🛅 🤊 🥲 🖓 🗊 😰 🛛 Current Directory: /home/sept Current Directory 🏎 🖬 🔻 🐨 🕨 New to MATLAB? Watch this <u>Video</u>, see <u>Demos</u>, or read <u>Getting Started</u>. 🖻 🖆 🖪 🖥 🔹 • All Files L Type Folder See also <u>toc</u>, <u>cputime</u>. adobe 🗀 .audacity-data Folder Reference page in Help browser 🗀 .audacity1.3 -... Folder Folder doc tic 🗅 .avidemux Folder a.cache Folder 🗀 .compiz >> help toc 🗅 .config Folder TOC Read the stopwatch timer. 🗀 .cryptonit Folder TIC and TOC functions work together to measure elapsed time. ⊃.cups Folder TOC, by itself, displays the elapsed time (in seconds) since TIC 🗀 .dbus Folder 🖨 .elisa Folder was used. t = TOC; saves the elapsed time in t as a double scalar. 🗅 .eqe Folder euler Folder See also tic, cputime. Command History + - * × Reference page in Help browser %-- 3/1/10 9:01 doc toc help tic help toc **▲** <u>S</u>tart



Introduction

to Scientific

Engineering

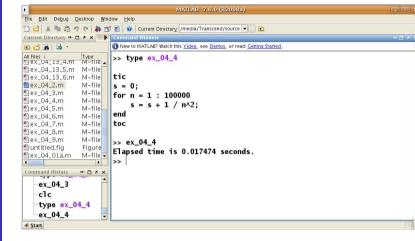
Computing, BIL108E

EXAMPLES with for loop tic s = 0; for n = 1 : 100000 s = s + 1 / n^2; end toc



Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE:





Introduc

to Scient

Engineer

Computi

BIL108

and

LOOPS

on fic	
ng Ig, E	
	EXAMPLES
	with vectorization
	tic

n = 1 : 100000; $s = sum(1 . / n.^2);$ toc

LOOPS

EXAMPLE:

Introduction to Scientific and Engineering Computing, BIL108E

MATLAB 7.6.0 (R20 Elle Edit Debug Desktop Window Help 🛅 🖆 🐇 🍡 🛱 🤊 🥲 🗿 🗊 😰 🥹 Current Directory /media/Transcend/source 💌 ... 💽 Current Directory 🗰 🖬 🛪 🛪 🗤 💽 🖸 🖻 🖆 🐻 🗟 🔸 1 New to MATLAB? Watch this <u>Video</u>, see <u>Demos</u>, or read <u>Getting Started</u>. All Files 4 Type >> type ex_04_5 Mex 04 13 5.m M-file 13 ex_04_13_6.m M-file tic 1 ex_04_2.m M-file n = 1 : 100000; ex_04_3.m M-file s = sum(1 ./ n.^2); ex_04_4.m M-file toc ex_04_5.m M-file 🖺 ex_04_6.m M-file ex_04_7.m M-file >> ex_04_5 🖺 ex_04_8.m M-file Elapsed time is 0.024687 seconds. 🖺 ex_04_9.m M-file >> 🖞 untitled.fig Figure 🖺 ex_04_01a.m M-file 💌 • Command History 😐 🖬 🛪 🗙 ex_04_4 clc type ex_04_5 ex_04_5 📣 Start



CONDITIONAL STATEMENTS

Introduction to Scientific and

Engineering

Computing,

BIL108E

if statements

Relational Operators

Usage;

if condition statement, end

- condition is usually a logical expression
- if *condition* is **true** *statement* is executed but if *condition* is **false**, nothing happens.
- Condition may be a vector or a matrix, in which case it is true only if all of its elements are nonzero. A single zero element in a vector or matrix renders it false.



LOGICAL EXPRESSIONS

Logical operators are used to combine logical expressions (with "and" or "or"), or to change a logical value with "not"

Operators:

& AND, | OR, ~ NOT.

INF	TUY	OUTPUT			
A	В	A&B	AB	~A	~В
false	false	false	false	true	true
false	true	false	true	true	false
true	false	false	true	false	true
true	true	true	true	false	false



CONDITIONAL STATEMENTS

Introduction to Scientific and Engineering Computing, BIL108E Karaman EXAMPLE

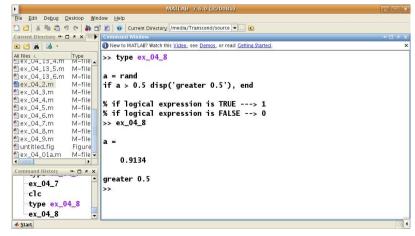
a = rand if a > 0.5 disp('greater 0.5'), end

% if logical expression is TRUE ---> 1 % if logical expression is FALSE --> 0

Introduction to Scientific and Engineering Computing, BIL108E

CONDITIONAL STATEMENTS

EXAMPLE:



CONDITIONAL STATEMENTS

Introduction to Scientific and

Engineering

Computing, BIL108E IF-ELSE

if condition

blockofstatementsA

else

blockofstatementsB

end

- blockofstatementsA or blockofstatementsB represents one or more statements.
- If condition is true blockofstatementsA is executed and if false blockofstatementsB is executed.
- else is optional.



Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE

a = rand if a > 0.5 disp('a is greater than 0.5') else disp('a is less than 0.5')

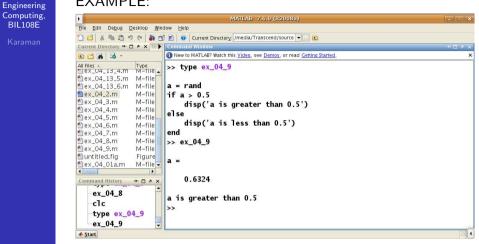
end



ERRORS

Introduction to Scientific and

EXAMPLE:



CONDITIONAL STATEMENTS

Introduction to Scientific and Engineering Computing, BIL108E

elseif

if condition1

statementA elseif condition2 statementB elseif condition3 statementC

•••

else

statementE

end

This is called **elseif** ladder.



CONDITIONAL STATEMENTS

Introduction to Scientific

and

Engineering

Computing,

BIL108E

elseif

- 1 *condition*1 is tested. If it is true, *statementA* are executed; MATLAB then moves to the next statement after end.
- 2 If *condition*1 is false, MATLAB checks *condition*2. If it is true, *statementB* are executed, followed by the statement after end.
- 3 In this way, all conditions are tested until a true one is found. As soon as a true condition is found, no further elseifs are examined andMATLAB jumps off the ladder.
- 4 If none of the conditions is true, *statementE* after else are executed.



Introduction to Scientific and Engineering Computing, BIL108E

elseif cont'd.

- **5** Arrange the logic so that not more than one of the conditions is true.
- 6 There can be any number of elseifs, but at most one else.
- **7** elseif must be written as one word.
- 8 It is good programming style to indent each group of statements as shown.



CONDITIONAL STATEMENTS

roduction	
Scientific	
and	
gineering	
mputing,	
BIL108E	
TEIOOE	

NESTED ifs

An if construct can contain further ifs.

This is called **NESTING**.

else belongs to the most recent ifs.



CONDITIONAL STATEMENTS

Introduction to Scientific and Engineering Computing, BIL108E

switch STATEMENT

switch value
 case val1
 statement1
 case val2
 statement2
 case [val3 val4 val5]
 statement3

otherweise *statementN*

end

. . .



Introductio

to Scientifi

Engineering

Computing BIL108E

and

CONDITIONAL STATEMENTS

EXAMPLE:
val = 3;
switch val
case 1
disp('one')
case 2
disp('two')
case 3
<pre>disp('three')</pre>
otherwise
disp('not a number between 1-3')
end



MATLAB 7.6.0 (R2008a)

disp('not a number between 1-3')

» New to MATLAB? Watch this Video, see Demos, or read Getting Started

· ... 🗈

Introduction to Scientific and Engineering Computing BIL108E

EXAMPLE:

Name

III i

Hya

File Edit View Graphics Debug Desktop Window Help

Value

3

nmand History 🗰 🗖 🛪 🗙

type ex_04_04

clc

LOOPS

₫ Start

2001

1.0000e-0

<1x2001

<1×2001 0

🎦 😂 👗 ங 🛅 🤊 🥲 🐞 💅 🖹 🕘 Current Directory, /media/Transcend

Command Window

val = 3:

switch val

end

>>

case 1

case 2

case 3

otherwise

disp('one')

disp('two')

disp('three')

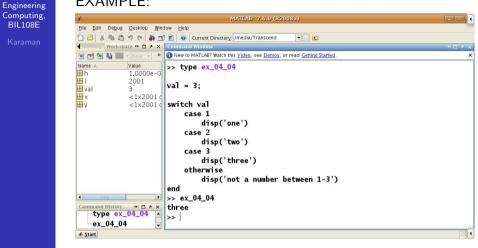
>> type ex_04_04

Introduction

and

CONDITIONAL STATEMENTS

to Scientific EXAMPLE: Engineering





Introduction to Scientific and Engineering Computing, BIL108E

WHILE LOOP

While loops are most often used when an iteration is repeated until some termination criterion is met.

Usage;

while *expression*

block of statements

end

The block of statements is executed as long as expression is true.



Introduction

to Scientific and Engineering

Computing,

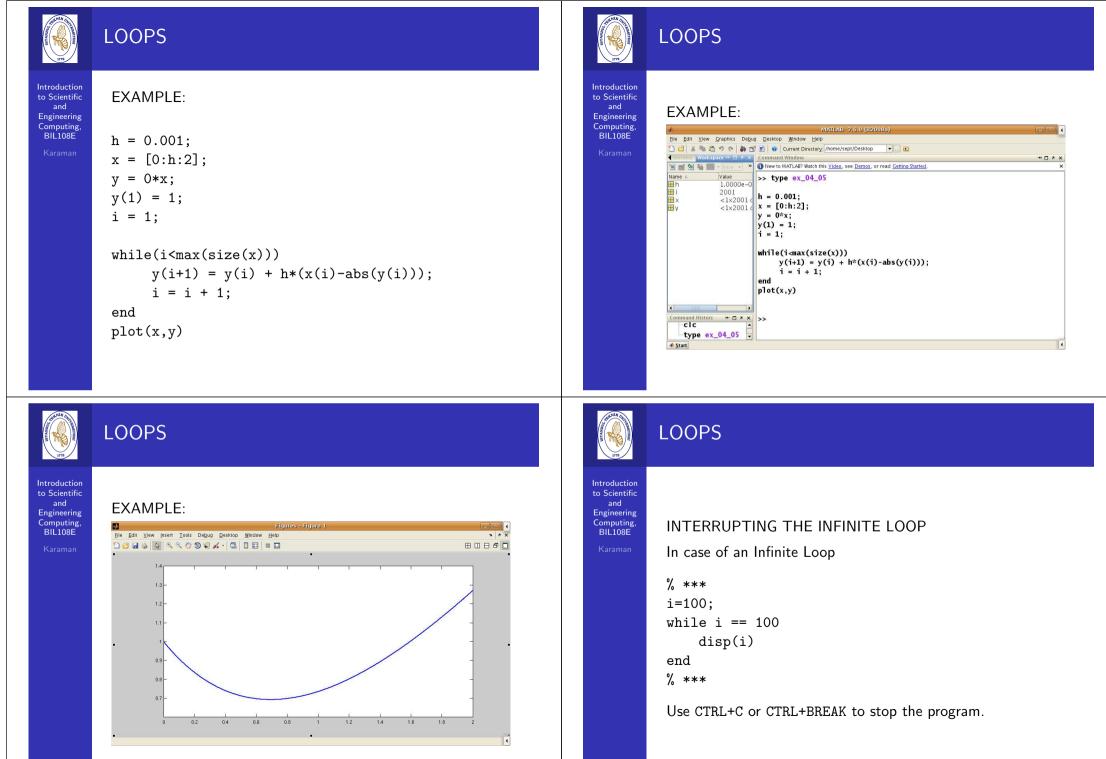
BIL108E

WHILE LOOP

LOOPS

To execute a while-end loop properly;

- The conditional expression in the while command must include at least one variable:
- The variables in the conditional expression must have been assigned when MATLAB executes the while command for the first time;
- At least one of the variables in the conditional execution must be assigned a new value in the commands that are between the while and the end. Otherwise once the looping starts it will never stop since the conditional expression will remain true.





LOOPS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE: 🕢 🐔 Applications Places System 🚳 🖓 👰

f MATLAB 7,600 (#2006a) File Edit Debug Desktop Window Help File Edit Debug Desktop Window File Edit Debug Desktop Window Current Directory *** * Command Window *** Command Window *** Command Window *** *** *** *** *** *** *** **** *** *** *** **** *** ***** *** ***** *** ***** *** ***** *** ****** *** ******** *** ************************************
Image: Second Second History Image: Second History Image: Second Second History Image: Second History
Command Window ************************************
Image: Section 2 and Sectio
All Files / Type Dex. 0.3.1:m M-File Dex. 0.4.0:m M-File Dex. 0.4.1:m
• (ex. 03.1.m M-file • (ex. 04.0.m M-file • (ex. 04.0.m M-file • (ex. 04.0.m M-file • (ex. 04.1.3.m M-file • (ex. 04.13.m M-file) • (ex. 04.13.m M-file)
•••••••••••••••••••••••••••••
^m ex_04_03.m M-file ^m ex_04_12.m M-file ^m ex_04_13.m M-file ^m ex_04_13.m M-file ^m ex_04_13.m M-file ^m ex_04_13.m M-file ^m ex_04_13.m M-file ^m ex_04_13.m M-file ^m ex_04_13.2.m >> Command History H = # * x M M M >>
•••••••••••••••••••••••••••••
ex_04_12.m M-File ex_04_13.im M-File ex_04_13.im M-File ex_04_13.im M-File ex_04_13.im M-File ex_04_13.im M-File ix_x >> certain ix_x
th ex_04_13.m th ex_04_13.1.m th ex_04_13.2.m th ex_04_13.3.m th e
• ex_04_13_1m • ex_04_13_2m • ex_04_13_m • ex_04_10_10_10_10_10_10_10_10_10_10_10_10_10_
Command History H = # * *
Operation A 13 3.m M-file >> Command History Im Im Clear Im
clear
_ex_04_12
-clc
type ex_04_12 -
4 Start
🖣 🕱 😻 Bfj Bu 🔮 Gmail 🔺 MATL 📷 sept 🚱 Help 🔚 Transc 🕎 Figure 🚰 Deskt 🕝 Editor 🚺 🙀



Introd

to Scie

Compu

BIL10

an Enginee

LOOPS

Applications Place		
4 1	MATLAB 7.6.0 (R2008a)	- 0
File Edit Debug Desktop	Window Help	
Current Directory + C 7 ×	🕼 🗊 😰 🥝 Current Directory, /media/Transcend/source 💌 🔞	-11 [1]
E Content Directory - C + A	New to MATLAB? Watch this Video, see Demos, or read Getting Started.	
All Files 4 Type	Green to mini bits: Hater hits Thee, Sto perine, of Feed Section States.	
ex_03_1.m M-T ex_04_0.m M-T ex_04_01.m M-T ex_04_02.m M-T ex_04_02.m M-T ex_04_02.m M-T ex_04_02.m M-T ex_04_02.m M-T ex_04_01.m M-T ex_04_13.m M-T ex_04_13.m M-T ex_04_13.m M-T ex_04_13.m M-T ex_04_13.m M-T	100 100 100 100	
Command History F Type ex_U4_12 -ex_04_12 -clear ex_04_12	<pre>x 100 ??? Operation terminated by user during ==> ex_04_12 at 4 >>></pre>	

Introduction to Scientific and Engineering Computing, BIL108E

LOOPS

BREAK, CONTINUE AND RETURN

- The break and return statements provide an alternative way to exit from the flow of the program.
- continue passes control to the next iteration of for or while loop and skips any remaining statements in the body of the loop.
- **break** is used to escape from an enclosing while or for loop. Execution continues at the end of the enclosing loop construct.
- **return** is used to force an exit from a function. This can have the effect of escaping from a loop. Any statement following the loop that are in the function body are skipped (Next week "Functions").

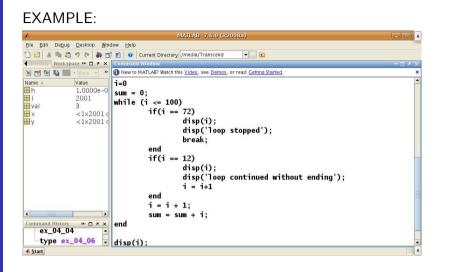


LOOPS

```
EXAMPLE:
Introduction
to Scientific
  and
Engineering
           i=0
Computing,
BIL108E
           sum = 0;
           while (i <= 100)
           if(i == 72)
           disp(i);
           disp('loop stopped');
           break:
           end
           if(i == 12)
           disp(i);
           disp('loop continued without ending');
           i = i+1
           end
           i = i + 1;
           sum = sum + i;
           end
```



Introduction to Scientific and Engineering Computing BIL108E





LOOPS

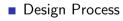
Introduction to Scientific and EXAMPLE: Engineering Computing, MATLAB 7.6.0 (R2008a BIL108E File Edit Debug Desktop Window Help 🎦 😂 🐇 ங 🛅 🤊 🥲 🐞 💅 🖹 🛛 🕢 Current Directory: /media/Transcend • ... 🗈 Workspace 🖛 🗆 🛪 🗙 » >> ex 04 06 🐚 📬 🐜 🔤 - Base Name Value 1.0000e-0 i = 🖽 i 🖽 sum 72 2615 0 🗄 val 3 X <1x2001 12 <1×2001 loop continued without ending i = 13 72 loop stopped type ex_04_06 72 ex_04_06 A Start



PROGRAM DESIGN AND ALGORITHM

Introduction to Scientific and Engineering Computing, BIL108E





Structure Plan

To design a successful program you need to understand a problem thoroughly and break it down into its most fundamental logical stages.

In other words, you have to develop a systematic procedure or an algorithm for solving it.



PROGRAM DESIGN



- The program must be readable and hence clearly understandable.
- It is useful to decompose the main program into subprograms that do specific parts of it.
- Add comments and references so that you know exactly what was done and for what purpose.



Introductio to Scientific and Engineering Computing BIL108E

1 Problem analysis.

- **2** Problem statement. Develop a detailed statement of the mathematical problem to be solved with a computer program.
- **3** Processing scheme. Define the inputs required and the outputs to be produced by the program.
- 4 Algorithm. Design the step-by-step procedure in a top-down process that decomposes the overall problem into subordinate problems.



Er

Co

BII 108F

DESIGN PROCESS cont'd.

oduction	
Scientific	
and	
gineering	
nnuting	

- 5 Program algorithm. Translate or convert the algorithm into a computer language.
- 6 Evaluation. Test all of the options and conduct a validation study of the program. For example, compare results with other programs.
- 7 Application. Solve the problems, the program was designed to solve. If the program is well designed and useful, it can be saved in your working directory

Introduction to Scientific and Engineering Computing, BIL108E

DESIGN PROCESS

EXAMPLE

A function M-file is a script file designed to handle a particular task that may be activated (invoked) whenever needed.



DESIGN PROCESS

Introduction to Scientific and Engineering

Computing, BIL108E

EXAMPLE TRAJECTORY

A ball is thrown with an initial angle of θ and initial velocity of v_0 .

Given

velocity and theta angle

Find

- Projectile Flight Path
- Projectile speed vs. angle



to Scientific and Engineering Computing BIL108E

EXAMPLE

- 1 $v_{0,x} = v_0 \times cos(\theta_0)$
- **2** $v_{0,v} = v_0 \times sin(\theta_0)$
- 3 $x(t) = v_{0,x} \times t$ (horizontal distance from origin as a function of t)
- 4 $y(t) = v_{0,v} \times t 0.5 \times gt^2$ (vertical distance from origin as a function of t)
- **5** $y(x) = \frac{v_{0,y}}{v_{0,x}} \times x 0.5 \times g \frac{x^2}{v_{0,y}^2}$ (with using # 3 and # 4 vertical location of the point is a function of x horizontal distance)



to S

Eng Con

DESIGN PROCESS

duction	
cientific	
and	
ineering	
puting,	
108F	

EXAMPLE

%

% The projectile problem with zero air resistance % in a gravitational field with constant g. %

% Written by ####### 01.03.2010

% Written by D. T. Valentine September 2006 % Revised by D. T. Valentine November 2008 % An eight-step structure plan applied in MATLAB: %

DESIGN PROCESS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE cont'd.

```
% 1. Definition of the input variables.
%
% Gravity in m/s**2
g = 9.81;
disp('*** INPUT DATA FOR PROJECTILE PROBLEM ***');
vo = input('Launch speed in m/s: ');
theta = input('Launch angle in degrees: ');
% Convert degrees to radians
theta = pi*theta/180;
```



DESIGN PROCESS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE cont'd.

% 2. Calculate the range and duration of the flight. % txmax = (2*v0/g) * sin(theta);xmax = txmax * v0 * cos(theta);



Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE cont'd.

% 3. Calculate the sequence of time % steps to compute trajectory. % dt = txmax/100;t = 0:dt:txmax;% % 4. Compute the trajectory. % x = (v0 * cos(theta)) .* t; $y = (v0 * sin(theta)) .* t - (g/2) .* t.^2;$



Introduction

to Scientific

and

Engineering

Computing,

BIL108E

DESIGN PROCESS

```
EXAMPLE cont'd.
% 5. Compute the speed and angular
     direction of the projectile.
%
% Note that vx = dx/dt, vy = dy/dt.
%
vx = v0 * cos(theta):
vy = v0 * sin(theta) - g .* t;
v = sqrt(vx.*vx + vy.*vy);
th = (180/pi) .* atan2(vy,vx);
%
% 6. Compute the time, horizontal
%
     distance at maximum altitude.
%
tymax = (v0/g) * sin(theta);
xymax = xmax/2;
ymax = (v0/2) * tymax * sin(theta);
```

DESIGN PROCESS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE cont'd.

% 7. Display ouput.

%

```
disp([' Range in m = ',num2str(xmax), ...
' Duration in s = ', num2str(txmax)])
disp(' ')
disp([' Maximum altitude in m = ',num2str(ymax), ...
' Arrival in s = ', num2str(tymax)])
plot(x,y,'k',xmax,y(size(t)),'o',xmax/2,ymax,'o')
title([' Projectile flight path, vo =',num2str(vo), ...
' th =', num2str(180*th/pi)])
xlabel(' x '), ylabel(' y ') % Plot of Figure 1.
figure % Creates a new figure.
plot(v,th,'r')
title(' Projectile speed vs. angle ')
xlabel(' V '), ylabel(' \theta ') % Plot of Figure 2.
%
% 8. Stop.
```

```
%
```



DESIGN PROCESS

Introduction to Scientific

and

BIL108E

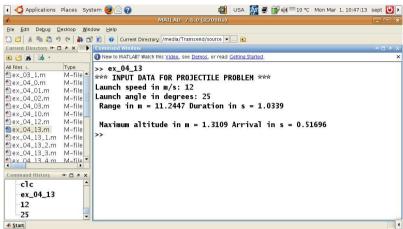
EXAMPLE:

Engineering Computing, 🛄 USA 🌆 🚍 💕 🕪 🔲 10 °C Mon Mar 1, 10:47:04 sept 🕑 🕨 📢 📢 Applications Places System 🥶 🖓 File Edit Debug Desktop Window Help 🚹 😂 🐇 🐂 🚔 🤊 🥐 🐞 🛒 🖹 🥝 Current Directory: /media/Transcend/source 💌 ... 🔞 Current Directory ** 🖬 * × ₩ 🕨 Comm New to MATLAB? Watch this <u>Video</u>, see <u>Demos</u>, or read <u>Gettino Started</u>. 🗈 🖆 🖪 🗟 🔹 All Files ∠ Type >> ex_04_13 1ex_03_1.m M-file *** INPUT DATA FOR PROJECTILE PROBLEM *** M-file Mex 04 0.m Launch speed in m/s: 12 ex_04_01.m M-file Launch angle in degrees: 25 Mex 04 02.m M-file ex_04_03.m M-file 🖺 ex_04_10.m M-file 🖞 ex_04_12.m M-file ex_04_13.m M-file ex_04_13_1.m M-file ex_04_13_2.m M-file ex_04_13_3.m M-file 1ex 04 13 4 m M-file Command History 🗰 🖬 🕷 type ex_04_12 clc ex_04_13 12 A Start Waiting for input



Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE:





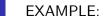
and

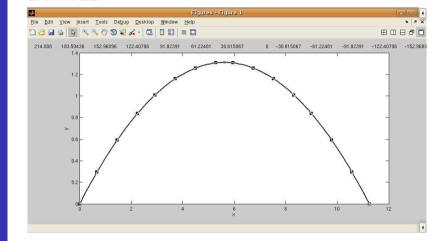
Engineering Computing,

BIL108E

DESIGN PROCESS

Introduction to Scientific

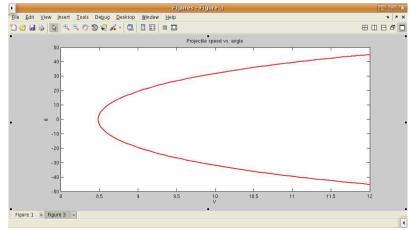




DESIGN PROCESS

Introduction to Scientific and Engineering Computing, BIL108E

EXAMPLE:





DATA ANALYSIS FUNCTIONS

Introduction to Scientific and Engineering

Computing, BIL108E

DATA ANALYSIS FUNCTIONS

- max(x) Determines the largest value in x.
- min(x) Determines the smallest value in x.
- sum(x) Determines the sum of the elements in x.
- prod(x) Determines the product of the elements in x.
- sort(x) Returns a vector with the values of x in ascending order.



DATA ANALYSIS FUNCTIONS

Introduction to Scientific and Engineering Computing, BIL108E

MEAN AND MEDIAN

mean(x) Computes the mean(average value) of the elements of the vector x.

$$\bar{x} = \frac{\sum_{k=1}^{N} x_k}{\sum_{k=1}^{N} x_k}$$

where $\sum_{k=1}^{N} x_k = x_1 + x_2 + \ldots + x_N$

median(x) Determines the median value of the elements in the vector x.



DATA ANALYSIS FUNTIONS

Introduction to Scientific

and Engineering Computing, BIL108E

VARIANCE AND STANDARD DEVIATION

- var(x) Computes the variation of the values in x.
- std(x) Computes the standard deviation of the values in x.
- The standard deviation is defined as the square root of the variance.

$$\sigma^{2} = \frac{\sum_{k=1}^{N} (x_{k} - \bar{x})^{2}}{(N-1)}$$

References

Introduction to Scientific and Engineering Computing, BIL108E

References for Week 4

- Alfio Quarteroni, Fausto Saleri, Scientific Computing with Matlab and Octave, Wissenschaftliches Rechnen mit Matlab, , Springer Verlag, 2006.
- 2 Cleve Moler, Numerical Computing with Matlab, Mathworks, 2008.
- Brian Hahn, Daniel T.Valentine, Essential Matlab for Engineers and Scientists, Elsevier, 2010.