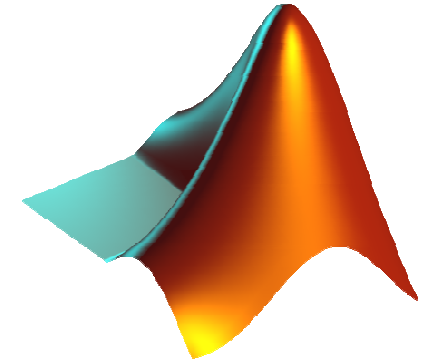




EEE 208 – Programming for EEE

Assist. Prof. Dr. Engin Mendi

Introduction



- **Course Goals:** This course aims to (a) make you familiar with MATLAB programming language and (b) allow you to practice the concepts learned in your Calculus, Linear Algebra, Differential Equations, Physics, Circuits Theory and Electromagnetics courses. Using Simulink to model the electromechanical systems is another goals of this course.
- **Course Objectives:** The objective is to make sure you can visualize and understand the fundamental engineering concepts. Programming in MATLAB will give you an easy and fast computational tool and will be essential in your future courses such as Signals and Systems, Image Processing, Communications and Control, as well as for your final project and graduate-level courses and research projects. Moreover, many data acquisition and development boards use MATLAB and its toolboxes to access the data and programming environment.

Syllabus

Week 1	Introduction to MATLAB, Standard MATLAB Windows, MATLAB as a Calculator, Assignments, Operation with Variables
Week 2	Operations with Arrays and Matrices, Introduction to 2-D Graphics
Week 3	Writing Script files (m-files), conditional statements, Writing Comments
Week 4	Loop statements and program flow, Working with 2-D Graphics (plots, subplots, logarithmic and polar)
Week 5	Writing Functions, Working with graphic handles and 3-D Graphics
Week 6	Review, Class Project, Phasors, s and z-planes in Matlab
Week 7	Data and Data Flow in MATLAB, Data Types (format short, long, bank etc)
Week 8	Midterm Exam
Week 9	Function optimization, working with symbolic math toolbox
Week 10	Differentiation and Integration, Solving Ordinary Differential Equations
Week 11	Regression and Fitting Functions; Statistics Toolbox
Week 12	Signal Processing Toolbox
Week 13	Communication with External Devices
Week 14	Introduction to Simulink, Modeling Electrical and Mechanical Systems



Grading Format

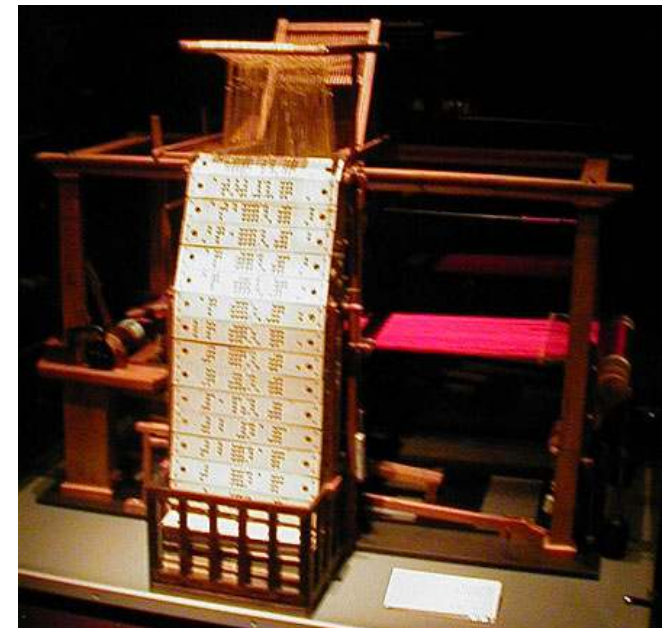
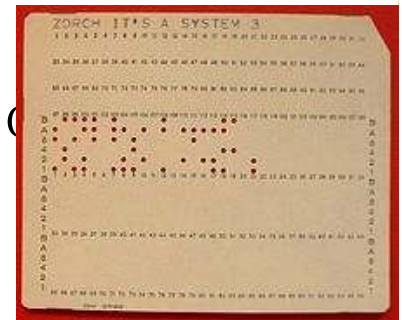
• Attendance	5%
• Quiz	15%
• Homework	15%
• Midterm Exam	30%
• Final Exam	35%

References

- A Practical Introduction to Programming and Problem Solving, S. Attaway, 2nd Ed. Butterworth-Heinemann: 2011.
- Introduction to MATLAB for Engineers, W. J. Palm III, University of Rhode Island. The McGraw-Hill Companies, Inc.: 2010.
- Experiments with MATLAB, C. Moler,
- <http://www.mathworks.com/moler/exm/chapters.html>
- Matlab ile Programlama, D. Dal.

1GL: First-generation programming languages

- Machine level languages, direct access to the CPU
- Switches were used on the control panel of computers
- Binary instructions (0s and 1s)



2GL: Second-generation programming languages

- Assembly languages
- English words (mnemonics)
- Humans understand it
- An assembler converts it into the machine code

PIC16F87X INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 1fff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 1fff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{\text{TO,PD}}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into standby mode	1	00 0000 0110 0011	$\overline{\text{TO,PD}}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	



3GL: Third-generation programming languages

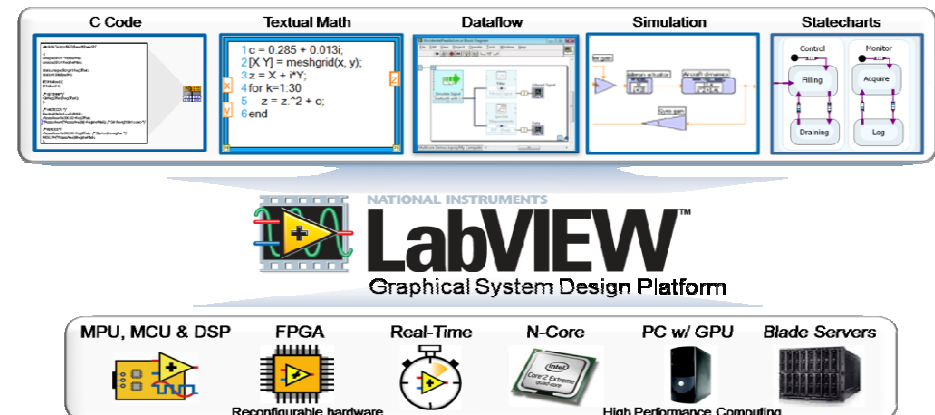
- In the late 1950s
- Independent from computer types
- Closer to human language
- English words used for variables, structures and commands
- Easier to read, write, and modify
- Need a compiler to convert them into the machine language
- Fortran, COBOL, ALGOL
- Basic, Pascal, C, C++ , C#, Java

4GL: Fourth-generation programming languages

- Very high level programming languages
- Developed between 1970s–1990
- Closer to human language and his form of thinking
- Reduce the time, cost and effort of software development
- Database queries, analysis and reporting, data manipulation, GUI creators, mathematical optimization
- LabView, SQL, Mathematica
- MATLAB, SAS, SPSS



MathWorks®



Common Programming Languages

- **FORTRAN** FORmula TRANslator; mathematical operations; mainframe computers. Little text input/output
- **COBOL** The COmmon Business Oriented Language; business data processing applications using English-like statements; handle business data records; limited mathematical capabilities
- **ALGOL** ALGOrithmic Language; a mathematically-oriented language,
- **Pascal** Named after Blaise Pascal, a French philosopher, mathematician, and physicist, A teaching language. Its object was to force the student to correctly learn the techniques and requirements of structured programming. A Pascal program could be compiled on any computer, and the result would run correctly on any other computer, even with a different and incompatible type of processor.
- **BASIC** The Beginner's All-purpose Symbolic Instruction Code; BASIC was the first interpreted language made available for general use. Now Visual Basic in a Windows® environment.
- **FORTH** A fourth-generation language; Both a compiler and an interpreter; handles real-time operations and still allows direct user control and rapid program modifications.
- **Assembly Language** A symbolic representation of the absolute machine code of a particular processor. Used especially for microprocessors and microcontrollers . Each processor has its own specific assembly language, (Intel 80x85 series, Microchip's PIC, AVR, etc)

Common Programming Languages, Cont.

- **C** Initially developed by Dennis MacAlistair Ritchie between 1969–1973, in AT&T Bell Labs. First came an experimental language called A, which was improved, corrected, and expanded until it was called B. Simple, versatile, and very powerful. The Unix operating system, which has been adapted to a wide range of platforms and is gaining in popularity, is written in C.
- **C++** When the concepts of *objects* and *object-oriented programming* were being developed, the standard C language didn't have the built-in structures to handle them. However, C was (and is) still highly useful and well worth keeping around, so a sort of extended C language was developed. This language was essentially "C and then some", or C-plus (C+). As the concepts of object-oriented programming continued to develop, C+ had to be upgraded, and became C++.
- **Java/JavaScript** A platform-independent language. Any computer with a Java Runtime Environment can run a Java program. Used for developing dynamic web pages
- **VHDL** VHSIC (very high speed integrated circuits) Hardware Description Language; electronic design of integrated circuits; digital and mixed-signal systems; a behavioral language
- **Python** Another general purpose programming language. Supports object-oriented and functional programming
- **And MATLAB:** A numerical computing environment, now used by millions of people in science, engineering, economics, and finance. Widely used in Academia for research and teaching. Cleve Moler wrote the first version in late 1970s. Later, Jack Little, Seteve Bangert and Moler rewrote MATLAB in C and founded their company MathWorks in 1984.



High-level and Low-level Programming Languages

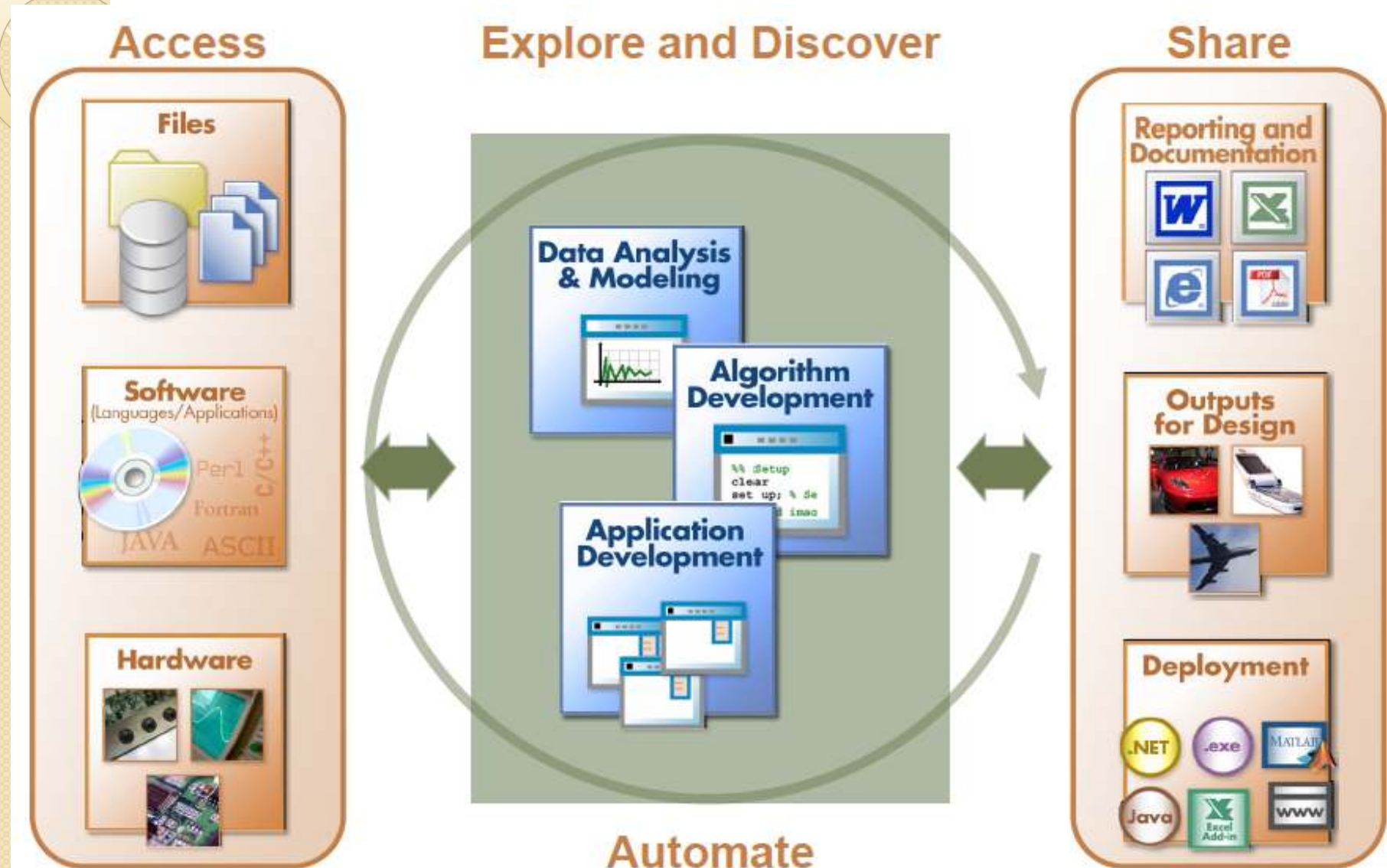
- Low-level languages: A very low-level language would involve a human giving instructions directly to a computer's hardware, using the language and structure of that hardware. Ex: machine code, Assembly.
Requires controlling values of bits, bytes and registers in the memories of the microprocessor/microcontroller directly
- High-level languages: complex, abstract logical tools to write instructions that a special program must translate for the computer. It is generally easier for humans to use these tools. Ex: C, C++ , C#, VHDL, Python, Microsoft Visual Basic, Visual C Using logical statements, loops, classes of variables, functions



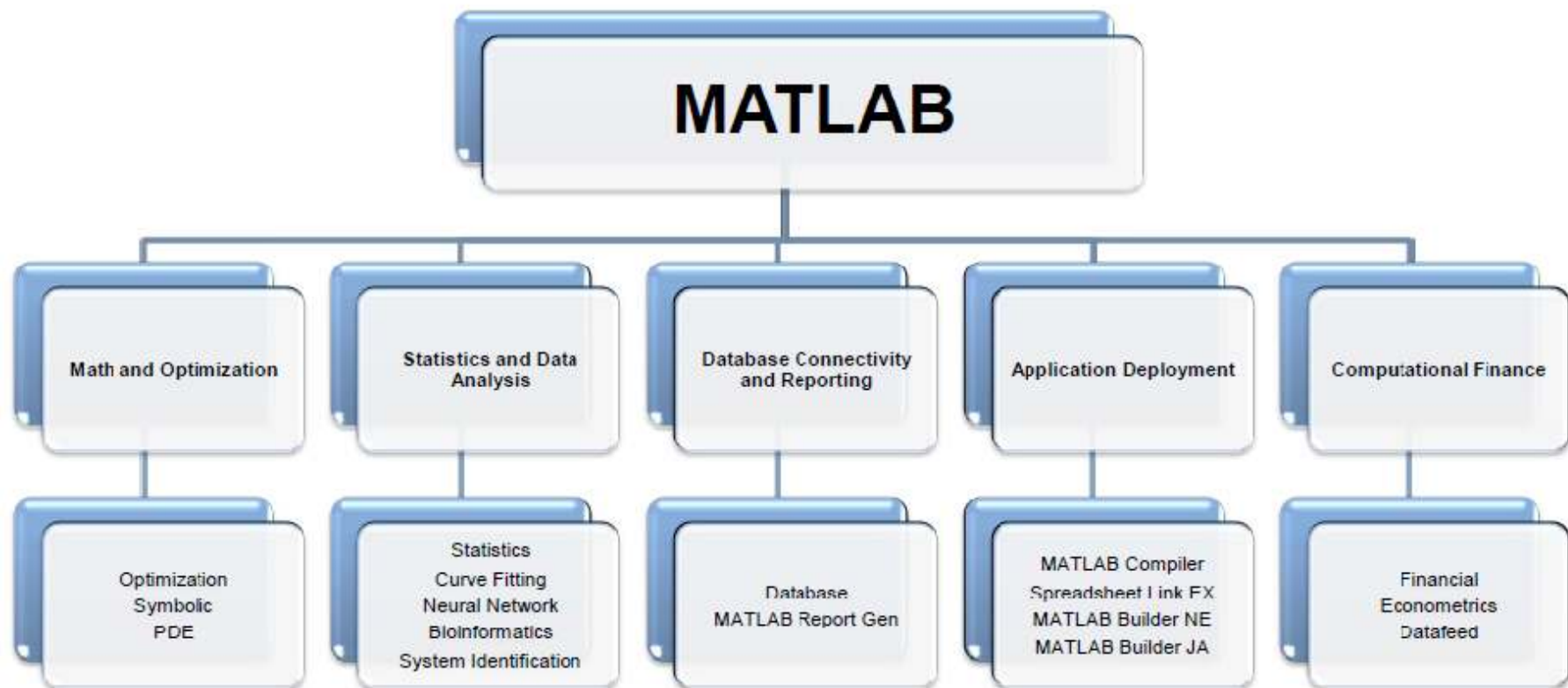
MATLAB: The Language for Technical Computing

- High-level language of technical computing
- Development environment for engineers, scientists
- Interactive tools for design, problem solving
- Mathematical function libraries
- Graphics and data visualization tools
- Custom Graphical User Interfaces (GUIs)
- External Interfaces: C, C++ , Fortran, Java, COM, Excel, .NET

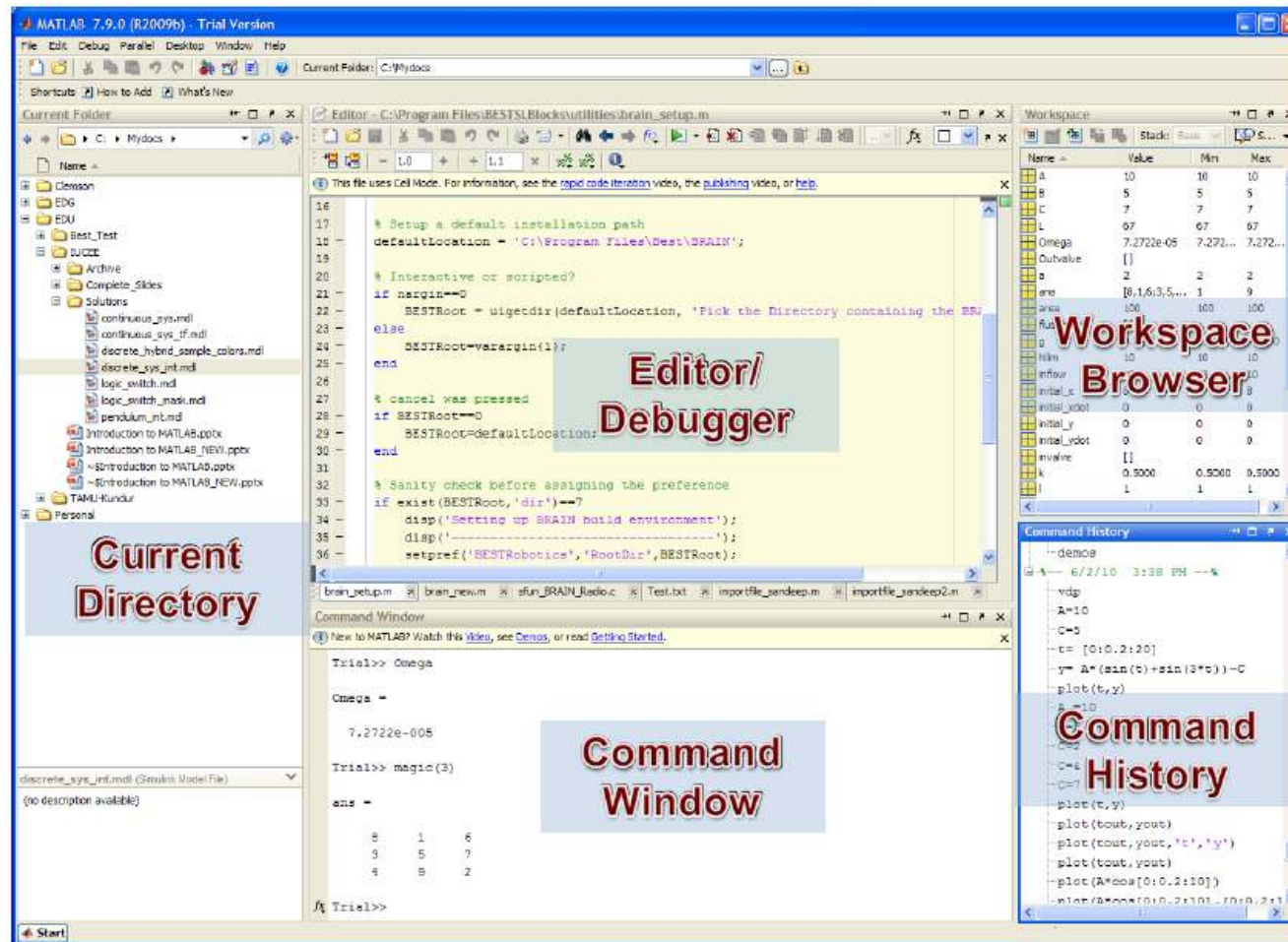
Technical Computing Workflow



MATLAB Product Family



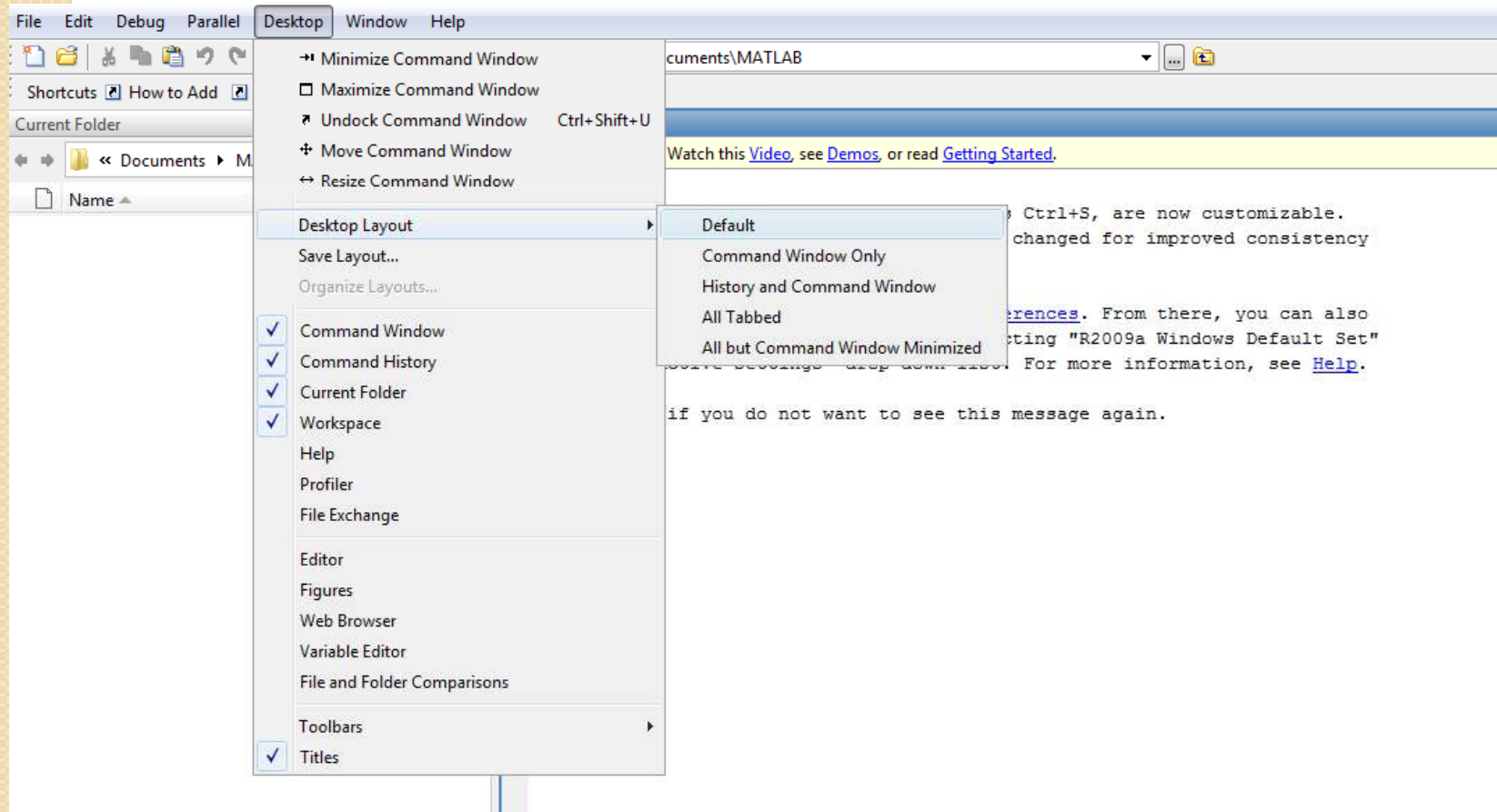
Matlab Desktop



Hint: You can change the current or default categories from the toolbar on the top of windows

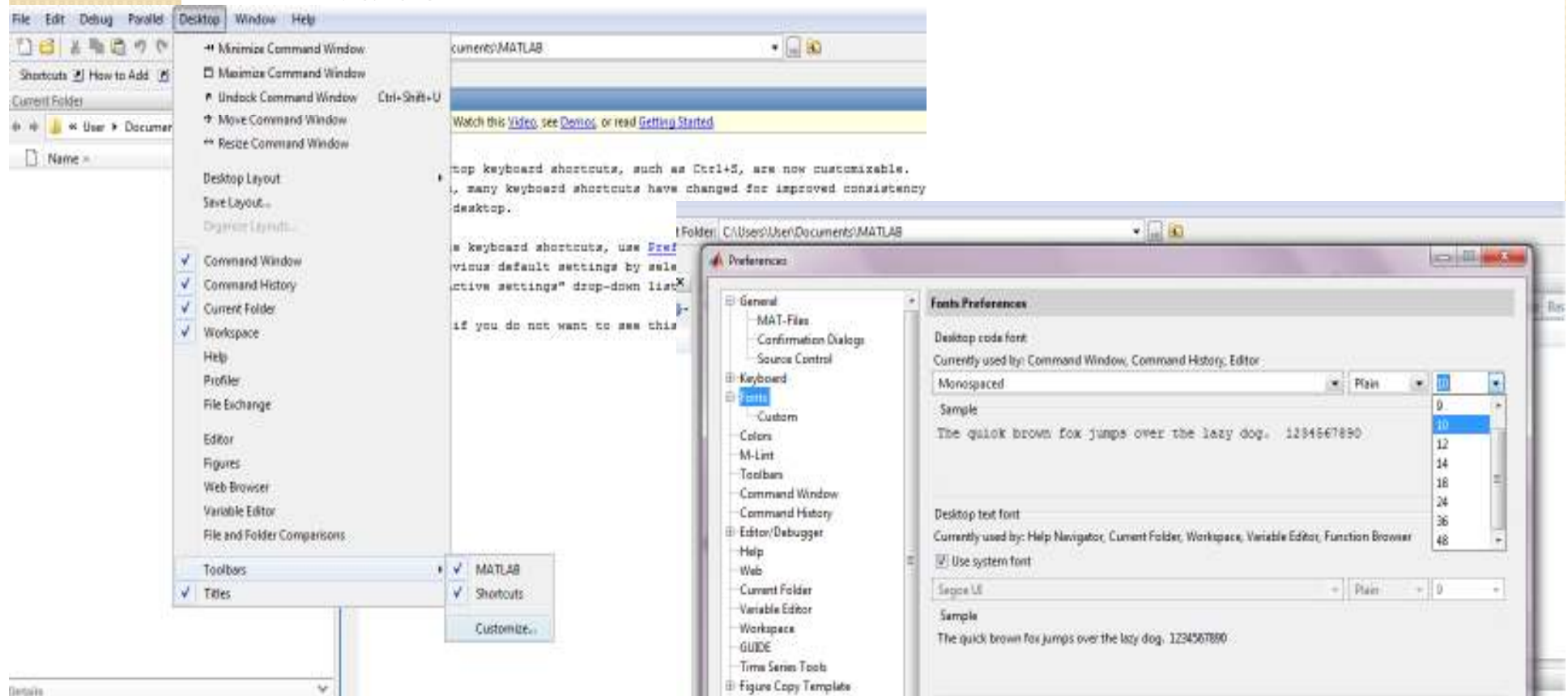
How to choose the Default Desktop Layout

- From Desktop menu, go to Desktop Layout and click on Default. Play around with other options!



How to Change the Font in the Command Window?

- From Desktop menu on the toolbar, go to Toolbars, and click on Customize ...
- From Preferences, go to Fonts and set it to the desired value



How to get Help?

- Matlab has extensive help documents that you can use instead of any textbook
- Type `>>help` and the function's name (`>> help sin`, `help log`, `help cosd`)
- Start -> Help
- <http://www.mathworks.com/help/matlab/index.html>
- Getting help about toolboxes:
<http://www.mathworks.com/help/>
- Ask questions and find answers in Matlab Central
<http://www.mathworks.com/matlabcentral/>

MATLAB as a Calculator

- MATLAB began as a matrix calculator
- MATLAB = **M**atrix **L**aboratory
 - Type expressions at the `>>`, and press return
 - Result is computed, and displayed as `ans`
 - Use numbers
 - algebraic operators: `+`, `*`, `/`, `-`, `()`, `^`
 - logical operators: `&`, `|`
 - exponential and logarithmic functions: `exp`, `log`, `log10`, `log2`, `pow2`, `sqrt`
 - trigonometric functions: `sin`, `cos`, `tan`, `sec`, `asin`, `acos`, `atan`, `asec` (work with radians) and `sind`, `cosd`, `tand`, `asind`, `acosd`, `atand` (work with degrees)
 - Use `abs`, `round`, etc
 - hyperbolic functions: `sinh`, `cosh`, `tanh`, `asinh`, `acosh`, `atanh`

For a complete list: `Help-> MATLAB-> functions-> mathematics-> elementary math`

Algebraic Operators and Precedence (Priority)

- Operators
 - + Addition
 - - Subtraction
 - * Multiplication
 - / Division
 - \ Left division (described in Linear Algebra in the MATLAB documentation)
 - ^ Power
 - ' Complex conjugate transpose
 - () Specify evaluation order
- All the basic algebraic rules apply
- Precedence rules in expressions
 - Left-to-right within a precedence group
 - Precedence groups are (highest first)
 - Parenthesis
 - Power (^)
 - Multiplication and division (*, /)
 - Addition and subtraction (+, -)

Complex Numbers and Functions

When you start Matlab, two variables already exist, and are equal to $\sqrt{-1}$. They are i and j . It's common to overwrite them without even realizing, but you can always create the number with the expression `sqrt(-1)`

```
>> i
>> sqrt(-1)
>> j
>> 4 + 6j
>> 4 + 6*j
>> C = 1 - 2i;
>> real(C)
>> imag(C)
>> abs(C)
>> angle(C)*180/pi
```

Constants/Reserved Numbers

- $\pi = 3.141592\dots$
- $i = \sqrt{-1}$
- $j = \sqrt{-1}$
- `eps`: Floating-point relative precision 2^{-52}
- `realmin`: Smallest floating-point number 2^{-1022}
- `realmax`: Largest floating-point number $(2 - \epsilon)2^{1023}$
- `Inf`
- `NaN`

Expressions

- Legal expressions
 - `>> 4`
 - `>> 5 + pi`
 - `>> 6*sqrt(2)^4-12`
 - `>> 6 * sqrt(2.0) ^ 4 - 12`
 - `>> sin(pi/3)^2 + cos(pi/3)^2`
 - `>> exp(0)`
 - `>> 5*(cos(pi/6)+j*sin(pi/6))`
 - `>> 5*exp(j*pi/6)`
- Illegal expressions
 - `>> 2 4`
 - `>> (2,4)`
- Try `0/0`, `0*0`, `0^0`, `1/0`, `1/inf`, `1^inf`, `inf/inf`, ...
- What other expressions can you think of?
- Get familiar with Error messages

MATLAB Functions and Commands

info	i	rand
demo	j	rng
help	inf	clock
lookfor	NaN	randint
quit	exp	randi
namelengthmax	single	randn
who	double	linspace
whos	int8	zeros
clear	int16	ones
format	int32	length
sin	int64	size
abs	uint8	numel
fix	uint16	end
floor	uint32	reshape
ceil	uint64	fliplr
round	intmin	flipud
rem	intmax	rot90
sign	char	repmat
pi	logical	

Useful Commands

- Try each command and look up its help file
 - `ver`
 - `help`
 - `exit`
 - `quit`
 - `clear all`
 - `clean all`
 - `clc`
 - `clg`
 - `whos`