MATLAB TUTORIAL – Advanced Numerical Analysis

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Prof. D. Kressner
M. Steinlechner

Knowledge of MATLAB is **essential** to follow this course. In this document, you will find a quick hands-on tutorial to get you started or to freshen up basic MATLAB commands. **We expect that every student knows these basic commands already and is familiar with MATLAB.** If this is not the case for you, we urge you to catch up on it **as quickly as possible!**

For more in-depth tutorials, we refer to the following collection:

`http://ch.mathworks.com/academia/student_center/tutorials/launchpad.html`

Learn the basic commands of MATLAB by just doing them! Type in the following commands and carefully watch the output.

1.1 Variables, vectors, and matrices

`>a=7`	a is interpreted as a scalar (or 1×1 matrix)
`>b=[1,2,3]`	use of comma: arrange entries in the same row, therefore we have $b \in \mathbb{R}^{1\times 3}$
`> ↑`	press the up arrow; it can save typing!
`>c=[1+2,2,1,4]`	
`>d=[7 7 2]`	spaces have the same meaning as commas
`>e=[7 a 2]`	
`>f=[1;2;3;4]`	semicolon: starts a new row, therefore we have $f \in \mathbb{R}^{4\times 1}$
`>g=f(2)`	accesses the second element of the column vector f
`>E=[1 2 3;2 1 3]`	MATLAB distinguishes between lower and upper cases
`>h=E(1,2)`	accesses the $(1,2)$ element of E
`>E`	
`>F(3,4)=7`	MATLAB currently considers F to be a 3×4 matrix. Void elements are set to zero.
`>F(4,3)=2`	Now we need a fourth row!
`>F(1,2)=3`	Sets the $(1,2)$ element of F to 3.
`>whos`	statistics of used variables
`>clear a b`	deletes variables a and b
`>whos`	
`>clear`	deletes all variables
`>whos`	
`>help clear`	help for the command `clear`; don't read it all now
`>A=[1 2];`	concluding the line with a semicolon suppresses the output
`>A`	
`>format long`	
`>pi`	
`>format short`	
`>pi`	
`>Pi`	Error! Once again, MATLAB is case sensitive.
`>eps`	another pre-defined variable (machine precision)
`>A=eye(3)`	3×3 identity matrix
`>b=[1 2 3]`	
`>B=diag(b)`	diagonal matrix
`>diag(B)`	The meaning of MATLAB commands can be different depending on the input.
`>G=zeros(4)`	
`>H=zeros(2,3)`	
`>ones(3,4)`	Generates a matrix whose all elements are equal to 1.
`>rand(2)`	Generates a random 2×2 matrix.
`>(5:9)`	Vector of integers between 5 and 9.
`>(5:3:12)`	Vector of integers between 5 and 12 with step 3.
`>C=[A B;zeros(3) zeros(3)]`	block matrix
`>C'`	transpose of C (in fact, it is the conjugate transpose of C)
`>for i=1:10`	i runs from 1 to 10. (MATLAB waits for the final `end`)
`  y(i)=2*i`	the index $i = 0$ is not possible here, since vectors have no 0th element!
`  z(i)=3*i;`	
` end`	
`>z`	

Quiz. Construct a matrix

$$D = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}$$

using the MATLAB functions `diag` and `ones`. *Hint:* Type `help diag` to learn how to prescribe diagonals below and above the main diagonal.

1.2 Simple operations

```
>clear
>clc
>A=[1 2 3;2 1 0]
>B=[2 2;1 0;0 1]
>C=[0 1 0;5 1 3]
```

`>size(A)`	Returns number of rows and columns of A in a row vector.
`>[m,n]=size(A)`	One way to access these numbers individually.
`>m=size(A,1)`	Another way.
`>b=[2 1 3]`	
`>length(b)`	Vectors have only one useful dimension, the length.
`>x=b'`	
`>length(x)`	
`>A*B`	matrix multiplication
`>A*C`	Error, dimensions do not fit!
`>A.*C`	Dot '.' denotes elementwise matrix operations `.* ./ .^`.
`>A`	
`>C`	
`>A*C'`	$A \cdot C^T$
`>diag(A*C')`	Returns the diagonal of a matrix.
`>ans`	"answer" — returns the last unnamed output
`>D=A+C`	matrix addition
`>E=A+B`	Error, dimensions do not fit!
`>E=A-B'`	$E = A - B^T$
`>g=A*x`	matrix times vector
`>g=A*b`	Error!
`>A`	
`>b`	
`>B`	
`>f=b*B`	row vector times matrix
`>A=[2 -1;-1 2]`	
`>b=[1; 1]`	
`>x=A\b`	Solves the linear system $Ax = b$.

Quiz. Construct two random vectors x and y of length 10.

- Construct the vector z such that $z_i = x_i y_i$ for $i = 1, \ldots, 10$.
- Compute the sum of the elements of z using the MATLAB function `sum`. Notice that the result equals the scalar product of the vectors x and y. Compute the scalar product of the vectors x and y using matrix multiplication. Compare the results.

1.3 Matrix manipulations

```
>clear
>A=[1 2;3 4]
>A(3,2)=7                Adds a third row!
```

`>A(1:2,2)`	`(1:2,2)`: 1st to 2nd element of column 2
`>A(3,1:2)`	`(3,1:2)`: 1st to 2nd element of row 3
`>B(3:4,3)=[5;6]`	MATLAB creates a matrix B with the 3rd and 4th elements of column 3 being 5 and 6, respectively. All other entries are zero.
`>C(4:5,4)=A(1:2,2)`	
`>B(:,3)`	3rd column of B
`>d=C(3,:)`	3rd row of C
`>E=[1 2 3;4 5 6;7 8 9;10 11 12]`	
`>E(1:2:4,3)`	`(1:2:4,3)`: Picks every other element in column 3 from 1 to 4.
`>F=[1 2 3; 4 5 6; 7 8 9; 10 11 12; 13 14 15]`	
`>F=reshape(F,3,5)`	Reshapes the matrix F to be 3×5.
`>G=F(1:2:3,1:2:5)`	Picks every other element in columns 1, 3, 5 from 1 to 3.
`>G=F(1:2:end,1:2:end)`	Alternative notation if you forgot the dimensions.
`>G=F([1 3],[2 3 1])`	Picks a submatrix by referencing the row/column indices directly.
`>H=[1 3;9 11]`	
`>inv(H)`	Matrix inverse.
`>rank(H)`	Calculates rank of the matrix H.
`>det(H)`	Calculates determinant of the matrix H.
`>norm(H)`	Calculates 2-norm of the matrix H.
`>eig(H)`	Calculates eigenvalues of the matrix H.

Quiz. Construct the matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

without setting each element by hand. *Hint:* Use the MATLAB function `eye` and permute the matrix.

1.4 Subprograms and m-files

Before proceeding with this exercise change your current working directory to the one where you will save your m-files. A convenient way to do this is to select the folder in the dropbox menu at the top of the MATLAB window.
For the following subprogram you have to create files with the name of the function appended by ".m", for example "`test1.m`". (In principle, you may use any editor to this. However, MATLAB comes with a fairly nice built-in editor that also offers debugging.)

```
function y=test1(A)
% This is my first m file.          % You can add comments behind %.
% It is called test1.m and
% computes the sum of all elements
% of a matrix A.
[m,n]=size(A);
y=0;                                % Init
for i=1:m                           % i runs from 1 to m
  for j=1:n                         % j runs from 1 to n
    y=y+A(i,j);
  end
end                                 Save under test1.m.
```

```
>clear
>help test1
>A=[1 2;3 4]
>s=test1(A)
```

An m-file can be either a *function* (like `test1.m`) or just a list of commands, in which

case it is called a MATLAB *script*. For example, we can save the above commands into another m-file.

```
clear
help test1
A=[1 2;3 4]
s=test1(A)
```

Save under `test2.m`.

```
>test2          % execute all commands in test2.m
```

Quiz. Create a MATLAB function `func(x,y)` which, given two vectors x and y of equal length, returns a vector z defined as

$$z_1 = x_1 y_n,\ z_2 = x_2 y_{n-1},\ \ldots,\ z_n = x_n y_1,$$

where n is the length of x and y. *Hint:* Use the MATLAB function `fliplr` and elementwise vector operations.

1.5 Loops and logical operators

The following commands are concerned with creating loops. For the sake of this exercise, you will directly work within the command window. In practice, especially for more complex loops, this becomes quickly cumbersome. The use of loops is a strong indicator that you may want to work with a script rather than typing in the command window!

```
>a=1;b=1;c=10;
>for i=a:b:c          Iterates over the numbers a, a+b, a+2b, ... in the interval [a,c].
i                     Writes the current iterate to the standard output.
end;                  Denotes the end of the for-loop.
>a=2;b=3;c=17;
>for i=a:b:c, i, end;
>if (a < 3)
Conditional instruction if. If true, evaluates the following instructions.
a=a*a                 Since a = 2 before if statement, new value of a is 4.
end;                  Denotes the end of the if statement.
>i=a;
>while( i<= c)
Starts while-loop. Looping over the following set of instructions as long as the condition is satisfied.
i                     Writes the current iterate to the standard output.
i=i+b;                Proceeds to the next iterate.
end;                  Denotes the end of the while-loop.
>(a<b)                Logical operator. Equal to 1 if a < b, 0 otherwise.
>(a>b)                Logical operator. Equal to 1 if a > b, 0 otherwise.
>(a<b) & (c<b)        Logical operator. Equal to 1 if a < b and b < c, 0 otherwise.
>(a<b) | (c<b)        Logical operator. Equal to 1 if a < b or b < c, 0 otherwise.
>!(a<b)               Logical operator negation. Equal to 1 if a ≥ b, 0 otherwise.
>(a==b)               Logical operator. Equal to 1 if a = b, 0 otherwise.
```

Quiz. The matrix product of two matrices $A \in \mathbb{R}^{n\times k}$, $B \in \mathbb{R}^{k\times m}$ is the matrix $AB \in \mathbb{R}^{n\times m}$ with the entries

$$(AB)_{ij} = \sum_{\ell=1}^{k} (A)_{i\ell}(B)_{\ell j}.$$

Write a MATLAB function `matmult(A,B)` which, given two matrices A and B of compatible size, returns their matrix product AB. *Hint*: You need 3 nested

`for` loops in your implementation. Measure the time `matmult` consumes for matrix multiplication of two random matrices of size 1024×1024 using `tic` and `toc`. Compare with the time consumed by the built-in MATLAB matrix multiplication $*$.

1.6 Functions

`>f=@(x) log(x) - 1;`	Defines symbolic function $f(x) = \ln(x) - 1$.
`>f(1)`	Evaluates function f with argument 1.
`>f=@(x,p) log(x) - p;`	Defines multivariate symbolic function $f(x, p) = \ln(x) - p$.
`>f(1,1)`	Evaluates function f with arguments $x = 1, p = 1$.
`>g=@(x,a) x^3 + a*x;`	Defines function $g(x, a) = x^3 + ax$.
`>x = 2; a = 3;`	
`>g(x,a)`	Evaluates at the provided values of the variables x and a.
`>h=@(x) exp(x);`	Defines symbolic function $h(x) = e^x$.
`>h(1)`	Constant e (base of natural logarithm).

Quiz. Compute the expression

$$\left(\int_{-5}^{5} e^{-x^2} dx \right)^2$$

using the MATLAB function `quad`. See `help quad`.

1.7 Graphics

```
>x=(1:10)/10; y=x.^2; z=sqrt(x);
>plot(x,y)
>plot(x,y,'r')
>plot(x,z,'k')
>clf
>plot(x,y,'r')
>hold on
>plot(x,z,'k')
>plot(x,2*z,'bo')
>plot(x,y+z,'g*')
>legend('y','z','2z','y+z')
>hold off
>plot(x,y-z,'k+')
>help plot
>title('My plot')
>xlabel('x axis')
>ylabel('y axis')
>axis([0,2,-5,5])
>axis square
>grid
```

Turn on "Tools/Edit Plot" and right click to change properties of the plots. For inclusion of plots in presentations and papers it is often a good idea to increase the font sizes and thicken the lines.

You can save a plot by choosing "File/Save As..." in the menu. There are different file formats to choose from.

Quiz. Plot the polynomial

$$p(x) = x^4 + 10x^3 - 372x^2 - 2714x + 20995.$$

on the interval $[-20, 20]$. Can you guess the roots of the polynomial by looking at its plot? Verify your guessed roots x_i by checking if $p(x_i) \stackrel{?}{=} 0$.