CDS 130 Computing for Scientists

Topics:

MATLAB

1. **Variables**

1) Scalars, vectors, matrices.. can be assigned to a variable
2) Assignment
3) rules to name a variable:
A valid vairable name starts with a letter, followed by letters, digits, or underscores.
You cannot define variables with the same name as Matlab key words such as for, end, if...
Invalid variable names: 6x; end; n!; exp; ...
4) built-in variables
pi; i;
(note that the built-in variables can be reassinged with new values, e.g., pi = 4.0;)

Example: How to swap the values of two variables?

Matlabe code:

```
clear all; clc;
 a = 3.0;
 b = 4.5;
% To exchange the values of a and b, use a temporary variable to hold the value of the first variable.
 c = a;
  a = b;
  b = c;
```

2. **Built-in math functions**

Math operation order of precedence:

P. E.M.D. S.
()
^
*, /, \,
+, -

trigonomitircal functions:
sin(x), cos(x), tan(x), asin(x), acos(x), sind(x), cosd(x), tand(x)

tansendental functions:
log10(x), log(x), exp(x):

powers:
3.0^4;
sqrt(4.0);

other functions: (please use matlab to find the answeers to the following functions).

```
abs (-5);
round(3.6)
ceil (4.1);
floor(4.0);
mod(4,5);      % finding the remainder of  the interger division 4 devided by 5
rand(5);
int16(3.6);
int8(156);
uint8(345);
uint16(-1);
char(128);
```

3. **Anonymous functions**. e.g., to create an anoymous function in matlab such that

$$f(x,y)=\sin^2(x)+\cos^2(y)$$

matlab code:

```
f=@(x,y)    (sin(x))^2 + (cos(y))^2
```

4. **User-defined functions.**

Remember that the user-defined function must be saved with the same funtion name as the file name in the working directory.

For instance, to create $f(x,y)=\sin^2(x)+\cos^2(y)$
first, create (and save) a matlab script named 'my_func.m',
second, in my_func.m, the script reads:

```
func f=my_func(x,y)
 f  = (sin(x))^2 + (cos(y))^2
end
```

Having created this user-defined function my_func.m, you can use it just like other built-in math functions, such as:

```
a = my_func(3,4);
b = my_func(4,5);
...
```

5. **input and display**

```
>> a = input('A variable is provided here = " );   % matlab will wait for the user to provide a value after this command is executed.

>> display (a);  % the content of  a will be displayed
```

6. **Create vectors : Row vector and Column vector**

method 1: A(5) = 1;

method 2: A = [2,3,4,5]; or A=[2; 3; 4; 5;]

For row vectors, you can use either commas or spaces to separate the elements

method 3: A = 2: -1: -6; (colon notation).

If the increment is 1, the short-handed notation is A = 2:5; (i.e., A=[2, 3, 4, 5])

method 4: form a new vector from previously defined vectors

```
>> A = [1, 2, 3];
>> B = [3, 4, 5];
>> C = [ A, B]
```

method 5: iterations

```
A(1) = 0;
for i = 2:6
        A(i) = i
 end
```

% this creates a new vector [0, 2, 3, 4, 5, 6], and it does so by appending new pieces onto an existing vector, one piece at a time.

method 6: A = rand(5) % this creats a vector with all element values between 0 and 1.

method 7: A = linspace(0,2*pi, 10);

7. **Vectors in a Matrix**

1) row vector: A(5) means A(1, 5)

2) column vector A(5,1)

3) Transpose operator: A(5)' is now a column vector

8. **Address vector elements**

1) meaning of A(1,5) and A(5)

2) meaning of A(5,1)

3) A(1:3) is a new vector

4) A(0.2) is invalid; A(0) is also invalid

9. **Vector operations**

1) Vector added by a scalar

```
>> A = [1, 2, 3];
>> B = A + 3.0
 B =
   4   5   6
```

3) Vector multiplied by a scalar

```
>> A = [1, 2, 3];
>> B = A* 3.0;
 B =
    3   6   9
```

2) Adding two vectors

```
>> A = [1, 2, 3]
```

```
>> B = [3, 4, 5 ]
>> C = A+B
 C =
   [4, 6, 8]
```

4) Vector summation (sum of all vector elements)

```
>> C = [4, 6, 8];
>> sum(C)
 ans = 20
```

5) Element-by-element operation (Attach . to the first vector) (extra credit)

```
>> C = A. * B
 C =
    [3, 8, 15]

>> C = [1 1 1]. / A;
 C =
     1   0.5  0.33333
>> C= [1,2,3].^ 2
   C =
      1  4   9
```

6) Transpose operation

```
>> A = rand(5)   % this creates a 5x5 random matrix
>> B = A'        % the transposed matrix is assigned to B now
```

10. Vector as a string

```
>> A = 'ABC abc'
>> A(3)
    ans =
          C
>> A(5)
    ans =
          a
```

11. Create Matrices

method 1: A(4,5) = 0.0;

method 2: A = [1, 2, 3; 3, 4, 5];

method 3:
```
>> B = [1:4];
>> C = [4:7];
>> A = [B; C];
```

method 4:
```
rand(10,8);  % This creates a 10x8 matrix filled with random numbers [0,1]
zeros(10,8);  % Create a 10x8 matrix filled will 0's.
```

```
ones(10,8);  % Create a  10x8 matrix filled with 1's.
```

Example: How to create a 5x5 matrix with random numbers between -1 and 1?

```
A = rand(5) * 2 – 1;
```

12. **Address matrix elements**

```
A = rand(5,5)
```

(1) A(5, 4)
(2) A(1:2, [2,3])
(3) A([2,4,5], [1,3,4])
(4) A(:, 2:3)
(5) A(: , :)
(6) A(2, :)

13. **Matrix Operations**

(1) matrix added by a scalar

```
>>  A=[1, 2, 3; 3, 4, 5];
>> B = A + 2.0;
 B =
    3     4     5
    5     6     7
```

(2) matrix multiplied by a scalar
(3) matrix addition
(4) element-by-element operation (dot operator)

```
>> A = [1, 2, 3; 4, 5, 6];
>> A.*A
 ans =
       1     4     9
      16    25    36
```

(5) Sum of matrix

```
>> sum(A)
 ans =
       5     7     9
>> sum((sum(A))
  ans =
        21
```

(6) Mean and standard deviation of A

```
>> mean(A)
  ans =
>> std (A)
  and =
```

14. How to load data into Matlab

Save the file into the working directory of matlab, e.g, data.dat (alternatively, you can name it as data.mat).
Now you can load the data using
load data.dat

or (if it has the extension .mat).
load data

15. How to check the size of a matrix?

A = rand(4,5);

To check the size, you have two commands to use:

>> whos A;

>> [M,N] = size(A) ; % In this case, M is the number of rows; and N is the number of columns

16. Iterations (for-loops)

(1) syntax:

```
for i=1:10
    do something

end
```

Example 1:

```
A = [1, 3, 7]
for myVar = A
    A = A + myVar;
end
```

Example 2:

Considering the following iteration code, what is A(12)?

```
clear all; clc;
A(9)=13;
for i=[10:12] ;
   A(i)=A(i-1)+37;
end
```

Example 3:

using a for loop to calcualte $2^2+2^3+2^4+2^5+...+2^{20}$

Matlab code:

```
clear all; clc;
sum = 0;
for i=2:20
  sum= sum + 2^i;
end
sum
```

Example 4

using for loops to create the following vector:

[1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199]

Matlab code:

```
clear all; clc;
A(1) = 1; A(2) = 3;
for i=3:11
    A(i) = A(i-2) + A(i-1);
end
```

17. Double nested for-loops

```
for i=1:100
    for j =i:100
     do something
    end
end
```

Example (1): Using nested for-loops to create/modify matrix elements

11 12 13 14 15
13 14 15 16 17
15 16 17 18 19

Matlab code:

```
for i=1:3
    for j = 1:5
        A(i,j) = 8+i *2+j
    end
end
```

Example (2): What does the following Matlab code generate?

```
clear all; clc;
for i=1:4
    sum = 0;
    for j = i:4
        sum = sum + j;
       A(i,j) = sum;
 end
end
A

 ans =
```

18. Colors

additive color model

RGB

[1,1,0]; [1,0,1]; [0 1, 1,], [0, 0, 0]

subtractive color model

CYMK

RGB color model used in matlab:

```
Red=[1,0,0],  % assign this color to variable Red
a= [0.5, 0.5, 1];  % assign this color to variable a
```

19. Matrices as images:

Indexed images

How to create a image

(1) Create a colormap matrix: (the dimension is n x 3, n colors with each color represented by r, g, b components in 0-1).

(2) Create a matrix: M=rand(50); Designate/alter matrix elements

(3) imagesc(M) and/or imagesc

(4) coloarbar

Example: How to create a 6x6 black and white checkerboard?

Matlab code:

```
clear all; clc;
```

```
% create a color map with two colors
  map = [0,0,0; 1, 1, 1];

% create an image matrix
for i=1:6
   for j= 1:6
       M(i,j) = mod (i+j, 2);
 end
end
% display the image
image (M)
colormap (map)
```

20. **Plot**

Steps to make a plot:
step 1: Guess the plot range, a, b. (-10, 10)
step 2: create x and y vectors. Use x as an independent variable, and use y as a dependent vector: x =-10:0.01:10
y = x.^3 -1
(note: dot operators)
step 3: make a plot: plot(x, y)
Example: Is the following Matlab statement correct or not?

```
plot([2,3,4, 5], [2,3,4,5], 'ok');
```

You can also plot multiple curves using plot.
More advanced command of plot:

```
plot (x,y, '--gs', 'LineWidth', 2, 'MarkerSize',10, 'MarkerEdgeColor', 'b', 'MarkerFaceColor', [1.0,0.5,0.3]):
```

21. **Write text in the graph**

syntax:

```
text(x, y, 'text is here', 'FontSize', 14, 'Color', [0,10]);
```

22. **Plot 2-D graphs:**

understand the meanings of:

```
figure       % generate a new figure;
hold on     % the plot will be held in place
hold off     % the plot will be erased;
shg          % bring the figure to front;
axis square   % make a square figure
```

```
axis equal     % the units used in the x axis and y axis are equal
axis off       % the axes will not be displayed
axis ([xmin, xmax, ymin, ymax)
print -dpng 'filename.png'
drawnow    % the plot will be drawn as soon as this command is issued.
pause(n)     % pause temporairly stops matlab execution for n seconds before continuing.
```

Example:

```
x = linspace(0, 2*pi, 100);
y = sin(x):
plot(x,y, '-o'):
axis([0, 2*pi ,-1.5,1.5])
```

Example 2: How to draw a circle?

```
clear all; clc;
theta = linspace(0, 2*pi, 1000);
r = 1.0;
x = r* cos(theta);
y = r* sin(theta);
plot (x,y, '--b', 'MarkerSize', 16);
axis off
axis square
print -dpng 'circle.png'
```

23. **Fill**

Use 'fill' to generate polygons filled with colors.

```
fill (x, y, 'k')
```

Example:

```
x = [0, 0, 1, 1 ];
y = [1, 2, 2, 1 ];
c  =[1, 0, 0 ];
fill (x, y, c);
%note: the coordinates of  x and y must match. The ploting sequence of  the points is
important.
```