COMPUTER LAB INTRODUCTION

In this elementary computer lab session, we recall vector/matrix calculation, 2D and 3D plot for functions with one and two variables, respectively. We also want to plot a function $u = u(x, t)$, which is in terms of a series.

1. Vector/Matrix calculation

We can create vectors and matrices in MATLAB, using different commands.

One simple way to create a vector is, for example, using `colon` (:). You can get help from MATLAB by typing `doc colon` or `help colon`. What is the difference between `doc` and `help`? You can use the one that you prefer the better.

Create three 1×4 vectors:

```
>> a=1:4; b=1:0.5:2.5; c=2*b;
```

What do you get if instead of semi-colon (;) you use comma (,) between the commands? That is, if you run:

```
>> a=1:4, b=1:0.5:2.5, c=2*b
```

Now, perform these commands to see some simple vector/vector operators, and see which one is not a correct MATLAB command:

```
>> a * c
>> a .* c
>> a^2
>> a .^ 2
>> a' .^ 2
>> a' * c
>> a * c'
>> sum(a)
```

Note that `.*` and `.^` are component-wise operators.

What is the difference between `length` and `size`? You can check it by:

```
>> length(a), size(a)
```

Now, we create some matrices. For example, try these commands and see how they work (you can use `doc`, if you need help to understand each command):

```
>> [a c], [a,c]
>> [a;c]
>> diag(a)
>> ones(3), ones(3,2)
>> zeros(3), zeros(3,2)
>> eye(3), eye(4,3)
>> A=[1 2 3; 4 5 6; 7 8 9; 10 11 12]
>> B=[diag(a) zeros(4,1) ; ones(1,5)]
```

We can access specific row(s) or column(s) of a matrix. For example, try:

```
>> A(1,:), A(2,:), A(:,2), A(2:3, :)
>> B(end,:), B(:,end), B(:, 1:3)
```

Now, some vector/matrix manipulation:

```
>> C=repmat(a,3,1), D=repmat(a,3,2), E=repmat(a,1,2)
>> F=reshape(E,2,4), G=reshape(E,4,2)
```

Try also:

```
>> sort(E), sum(E)
>> sum(G), sum(G,1), sum(G,2)
```

Exercise 1. Create this matrix in one line command:

$$A = \begin{bmatrix} 1 & 8 & 0 & 0 & 0 & 0 \\ -1 & 2 & 8 & 0 & 0 & 0 \\ 0 & -1 & 3 & 8 & 0 & 0 \\ 0 & 0 & -1 & 4 & 8 & 0 \\ 0 & 0 & 0 & -1 & 5 & 8 \\ 0 & 0 & 0 & 0 & -1 & 6 \end{bmatrix}$$

2. Plot in 2D and 3D

First we recall how to plot the graph of a one-variable function $y = f(x),\ x \in [a, b]$. To this end, first we need a partition for the domain $[a, b]$, that is, dividing $[a, b]$ into small sub-domains. We can divide the interval $[a, b]$ either by considering a mesh step h, and then use `colon` (:), as:

```
>> x=a:h:b;
```

or, using `linspace` for some positive integer N, as:

```
>> x=linspace(a,b,N);
```

Note that, if we choose $h = \frac{b-a}{N-1}$ then the vectors x would be the same, for both methods.

For example, to plot $y = sin(x),\ x \in [-\pi, \pi]$, we can write:

```
>> x=-pi:0.2:pi; y=sin(x); plot(x,y)
```

or, to plot with more options:

```
>> x=-pi:0.2:pi; y=sin(x); plot(x,y,'bo-')
>> title('2D-plot')
>> xlabel('x'); ylabel('y')
```

To plot more than one functions in one `figure`:

```
>> % plot y1(x)=sin^2(x), y2(x)=sin(x)+x^2, for x in [-5,5]
>> x=-5:0.2:5; y1=sin(x).^2; y2=sin(x)+x.^2;
>> plot(x,y1,'bo-',x,y2,'r*--');
>> title('2D-plot')
>> xlabel('x'); ylabel('y')
>> legend('y1','y2')
```

See `doc plot` for more examples and options of `plot`. You can also find more MATLAB commands which are related to `plot`, at the bottom of the page, as 'See Also'.

Now, we plot a surface defined by a two-variables function $u = f(x, y)$, $x \in [a, b]$, $y \in [c, d]$. To this end, first we should compute the function values. We can divide the intervals $[a, b]$ and $[c, d]$ either by considering mesh steps h and k, and then use `colon` (:), as:

```
>> x=a:h:b; y=c:k:d;
```

or, using `linspace` for some positive integers N and M, as:

```
>> x=linspace(a,b,N); y=linspace(c,d,M);
```

Note that, if we choose $h = \frac{b-a}{N-1}$ and $k = \frac{d-c}{M-1}$, then the vectors x and y would be the same, for both methods.

Now, we can simply use a vector/vector multiplication, to calculate the function values $u(x, y)$ for all x and y. Or, we can also use `for`-loops to compute the function values.

For example, we compute $u(x, y) = sin(x) * sin(y)$ for $x \in [0, 5]$, $y \in [0, 10]$. We can simply write:

```
>> x=0:0.2:5; y=0:0.1:10;    % We have chosen h=0.2 and k=0.1
>> u=sin(x)'*sin(y);
```

(note that we multiply the transpose $sin(x)'$ and $sin(y)$). We can also compute it using `for`-loops, as

```
>> x=0:0.2:5; y=0:0.1:10;    % We have chosen h=0.2 and k=0.1
>> for i=1 : length(x)
>>     for j=1 : length(y)
>>         u(i,j)=sin(x(i))*sin(y(j));
>>     end
>> end
```

Then, we can plot the surface as:

```
>> surf(y,x,u)
>> xlabel('y'); ylabel('x')
```

Note that `xlabel` is 'y' and `ylabel` is 'x', while we write `surf(y,x,u)`. So the rule of x and y changes for `surf`. You can also use `mesh`, to plot the surface. Compare it with `surf`.

Exercise 2. Let k, L, T be some positive constants. Consider the one variable function

$$f(x) = \begin{cases} \frac{2k}{L} x & 0 \le x \le \frac{L}{2}, \\ \frac{2k}{L}(L - x) & \frac{L}{2} \le x \le L. \end{cases} \tag{1}$$

And the two variables function

$$u(x, t) = \frac{8k}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \sin\big(\frac{n\pi}{2}\big) \cos\big(\frac{n\pi}{L} t\big) \sin\big(\frac{n\pi}{L} x\big), \tag{2}$$

for $x \in [0, L]$ and $t \in [0, T]$.

Set $k = 1, L = 6$ and $T = 10$.

(a) Plot the function (1), and note that its first derivative does not exist at $x = \frac{L}{2}$ (the curve is sharp at that point).

(b) Plot the 3D graph of the function (2) for $(x,t) \in [0,L] \times [0,T]$. Note that to evaluate the series in (2), we should truncate the sum to, say $N = 100$ terms, since we cannot compute the series for ∞ terms. You can use the Matlab command "surf" for 3D plot. Do not forget to put the "xlabel" and "ylabel", to make sure that you use the variables x and t in the right order.

(c) Now, we want to see the behaviour (oscillation) of $u(x,t)$ in 2D.
Let $0 = t_1 < t_2 < \cdots < t_M = T$ be the partition of the time interval $[0,T]$, that you have used for plotting in part (b). In a `for` loop plot $u(x,t_j)$ for $j = 1, 2, \cdots, M$, using `hold on` and `pause(0.05)`, to see the oscillation.