Midterm Test 1. 2020/21

Solve the problems in one m or mlx-file and use separated cells for the different tasks. Please add comments to your script for the easier understanding. The results should be clearly visible for each subtask. ***Start your script with your name, Neptun ID, and Group ID in comment****. All statements that fail to execute are ignored. Your solutions (with all files necessary for execution, e.g. *.txt, *.dat) must be uploaded into the Moodle system (edu.epito.bme.hu). At the end click the 'Submit assignment' button! Time to finish the whole test is 90 minutes.*

1. We are to determine the plane position of a target $T(x, y)$ with a multi-static radar. Emitter A is located at the point $A = (x_A, y_A) = (10, 10)$ while receivers B and C are at the points $B = (x_B, y_B) = (0, 5)$ and $C = (x_C, y_C) = (13, 0)$. The measured distances of A-T-B and A-T-C are $t_1 = 15$ and $t_2 = 12$, respectively. Determine the two possible locations of the target as the intersection points of two ellipses that share a common focus A. The implicit equations of the ellipses are:

$$\sqrt{(x-x_A)^2+(y-y_A)^2}+\sqrt{(x-x_B)^2+(y-y_B)^2}=t_1$$

$$\sqrt{(x-x_A)^2+(y-y_A)^2}+\sqrt{(x-x_C)^2+(y-y_C)^2}=t_2$$

The following tasks are to be completed (15 points):

a) Write Matlab functions for the two ellipses using the given values of the coordinates of A, B, C and distances t_1 and t_2. (2 points)

b) Plot the two ellipses on the same figure with x-axis range [-5, 15]. Plot a grid and make scaling of both axes equal (3 points)

c) Make a vectorial function for the ellipses using the two scalar functions (2 points)

d) Calculate the two intersection points of the ellipses by using Matlab's built-in numerical method for solving nonlinear equations. Find approximate values of the roots from the figure. (4 points)

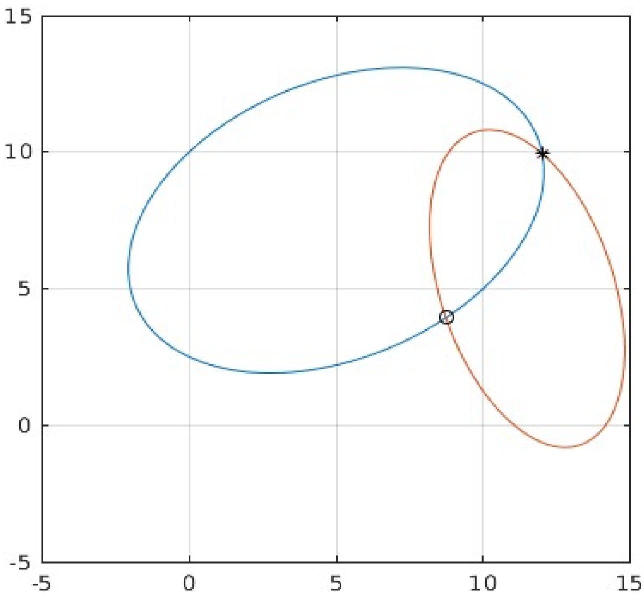
e) Check the solutions numerically by substituting the roots into the equations. Check the solutions graphically by plotting the roots with different symbols (3 points)

f) Calculate the distance between the two possible locations of the target $T_1(x_1, y_1)$ and $T_2(x_2, y_2)$ by the formula

$$t_{12}=\sqrt{((x_2-x_1)^2+(y_2-y_1)^2)}$$ (1 points)

```
% Problem 1.
% multi-static radar system with one emitter and two receivers
clc; clear all; close all
% emitter A
xA=10; yA=10;
% receiver B
```

```
xB=0; yB=5;
% receiver C
xC=13; yC=0;

% measured distance A-T-B
t1 = 15;
% measured distance A-T-C
t2 = 12;

% equations of ellipses
eq1 = @(x,y) sqrt((x-xA).^2+(y-yA).^2)+sqrt((x-xB).^2+(y-yB).^2)-t1
eq2 = @(x,y) sqrt((x-xA).^2+(y-yA).^2)+sqrt((x-xC).^2+(y-yC).^2)-t2

% plot of the ellipses

figure(1)
fimplicit(eq1,[-5,15])
hold on
fimplicit(eq2,[-5,15])
grid on; axis equal

% vectorial form of the equations
f = @(x) [eq1(x(1),x(2)); eq2(x(1),x(2))];
% find solutions
x01 = [9; 4] % initial value for the first inter-section point
x02 = [12; 10] % initial value for the second intersection point

% numerical solution - fsolve
x1 = fsolve(f,x01) % x1 = [8.7348;  3.9305]
x2 = fsolve(f,x02) % x2 = [12.0118;  9.9383]

% check both solutions by substitution
norm(f(x1)) % 5.7394e-11
norm(f(x2)) % 1.1849e-07

% check both solutions by plotting
plot(x1(1),x1(2),'ko')
plot(x2(1),x2(2),'k*')

% distance between the two solutions
t12 = sqrt(sum((x1-x2).^2))

% t12 = 6.8433
```



2. We are to determine a sphere that passes through 4 given points $P_i(X_i, Y_i, Z_i)$, $i = 1, \ldots, 4$. The coordinates of the vertex points are given in the file `sphere.txt` where the three columns correspond to X, Y, Z coordinates of the vertices. For each vertex $P_i(X_i, Y_i, Z_i)$ the following linear equation can be written in terms of the vector of unknowns $\mathbf{x} = [x_0, y_0, z_0, w]^T$ as

$$2X_i x_0 + 2Y_i y_0 + 2Z_i z_0 - w = c_i \quad i = 1, \ldots, 4$$

where $c_i = X_i^2 + Y_i^2 + Z_i^2$, $C(x_0, y_0, z_0)$ is the center of the sphere, $w = x_0^2 + y_0^2 + z_0^2 - R^2$, and R denotes the radius of the sphere. These 4 equations define a linear system $\mathbf{Ax} = \mathbf{b}$.

The following tasks are to be completed (15 points):

a) Load vertices from file `sphere.txt` and separate X, Y, Z coordinates of the vertices into vectors. (2 points)

b) Set up the linear system $\mathbf{Ax} = \mathbf{b}$ with 4 equations and 4 unknowns to be solved. Check whether there is a solution and whether it is unique. (3 points)

c) Solve the linear system with LU factorization and check the norm of the residuals. (3 points)

d) Solve the linear system with Matlab's built-in inverse function and check the norm of the residuals. Determine which solution is more accurate. (2 points)

e) Calculate the center $C(x_0, y_0, z_0)$ and radius R of the sphere from $\mathbf{x} = [x_0, y_0, z_0, w]^T$. (2 points)

f) Write a Matlab function to calculate the distance from any point $P(x_P, y_P, z_P)$ from the center of the sphere $C(x_0, y_0, z_0)$ using the formula $d = \sqrt{((x_0 - x_P)^2 + (y_0 - y_P)^2 + (z_0 - z_P)^2)}$ (2 points)

g) Calculate distances between the center of the sphere and the 4 vertices each and check whether these distances are equal with the radius R. (1 points)

```
% Determination of a sphere that passes through 4 points
clc; clear all; close all
% load coordinates of the points
XYZ = load("sphere.txt")
% separate coordinates
X = XYZ(:,1)
Y = XYZ(:,2)
Z = XYZ(:,3)

% set up the linear system
c = X.^2 + Y.^2 + Z.^2;
b = c;
A = [2*X, 2*Y, 2*Z, -ones(4,1)];

% Check whether we have a unique solution
size(A,2)  % there are 4 columns
```

```
rank(A), rank([A b]) % 4=4,  solution exists and unique: n=4

% Check whether the determinant is non-zero:
det(A) % =-542.8480 not zero, we have a unique solution

% solve A*x = b with LU factorization
[L U P] = lu(A);
% 1) solve L*y = d for y
d = P*b
opt1.LT=true
y = linsolve(L,d,opt1);
%
% 2) solve U*x = y for x
opt2.UT=true
x = linsolve(U,y,opt2)

% check solution
residual1 = norm(A*x - b)

% solve with inverse
x2 = inv(A)*b

% check solution
residual2 = norm(A*x2 - b)

% which is better in terms of accuracy?
if residual1 > residual2
   disp('The solution with inverse is more accurate')
else
   disp('The solution with linsolve is more accurate')
end
% The solution with linsolve is more accurate

% calculate center and radius
xc = x(1:3)
R = sqrt(x(1)^2+x(2)^2+x(3)^2-x(4))
% R = 4.2361

% check distances of points from the center of sphere
d = @(xc,P) sqrt(sum((xc-P).^2))

d1 = d(xc,XYZ(1,:)') % 4.2361
d2 = d(xc,XYZ(2,:)') % 4.2361
d3 = d(xc,XYZ(3,:)') % 4.2361
d4 = d(xc,XYZ(4,:)') % 4.2361
```

3. Four stations receive a signal from a source which was emitted at an unknown time t. The positions $P_i(x_i, y_i)$ of the stations and signal reception times t_i at each station are known. Determine the location $P(x, y)$ of the source and the emission time t of the signal that propagates with speed c.

   The following linear system that consists of 3 equations in the form of **A*x** = **b** with the unknowns **x** = $[x, y, ct]$ $(ct = c \cdot t)$ can be set up for the problem:

   $$(x_i - x_1)x + (y_i - y_1)y - c(t_i - t_1)ct = k_{i1} \qquad (i = 2, 3, 4)$$

   where $k_{i1} = 0.5\left[(r_i^2 - r_1^2) - c^2(t_i^2 - t_1^2)\right]$ and $r_i^2 = x_i^2 + y_i^2$ $(i = 1, 2, 3, 4)$

   Known station postions are: $P_1 = (5, 8)$, $P_2 = (-10, -10)$, $P_3 = (10, -10)$, $P_4 = (-8, 6)$. Known reception times are: $t_1 = 15.0642$, $t_2 = 15.0780$, $t_3 = 15.1104$, $t_4 = 15.0196$, signal propagation speed is $c = 2$.

The following tasks are to be completed (15 points):

a) Specify the coefficient matrix **A** and vector **b** in Matlab using the given data. (4 points)

b) Check existence and uniqueness of the solution. (2 points)

c) Solve the linear system with LU factorization and check the norm of the residuals. (3 points)

d) Determine signal emission time t from the solution. (1 point)

e) Point $P(x, y)$ is located at the intersection of 3 hyperbolas defined by the following equations:

$$\sqrt{(x-x_2)^2+(y-y_2)^2} - \sqrt{(x-x_1)^2+(y-y_1)^2} = c(t_2 - t_1)$$

$$\sqrt{(x-x_3)^2+(y-y_3)^2} - \sqrt{(x-x_1)^2+(y-y_1)^2} = c(t_3 - t_1)$$

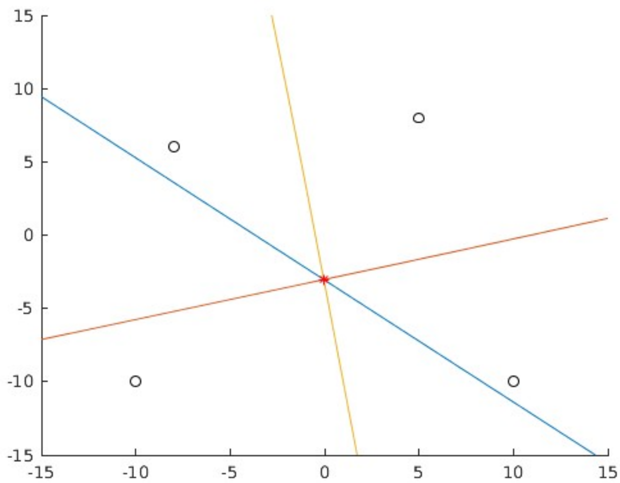
$$\sqrt{(x-x_4)^2+(y-y_4)^2} - \sqrt{(x-x_1)^2+(y-y_1)^2} = c(t_4 - t_1)$$

Define implicit functions of the hyperbolas in Matlab and plot them on the same figure using the x-range [-15, 15]. (4 points)

f) Plot in the same figure with different symbols the locations of stations as well as of the source. (1 point)

```
%  positioning with time differences
clc; clear all; close all
% Station coordinates in a matrix
Pi = [5, 8; -10, -10; 10, -10; -8, 6]
% xi, yi coordinates (i=2,3,4)
xi = Pi(2:4,1); yi = Pi(2:4,2)
% x1, y1 coordinates
x1 = Pi(1,1); y1 = Pi(1,2);
```

```
% reception times
ti = [15.0780, 15.1104, 15.0196]';
t1 = 15.0642;
c = 2;
% coefficient matrix
A = [xi-x1, yi-y1, -c*(ti-t1)]
% vector on the right side
ri2 = xi.^2 + yi.^2;
r12 = x1^2 + y1^2;
b = 0.5*((ri2-r12) - c^2*(ti.^2-t1^2))

% existence and uniqueness of the solution
rank(A) == rank([A,b])  % logical 1, solution exists
det(A)  % 19.9968 ,  unique

% solution with LU factorization
[L U P] = lu(A);
d = P*b;
opt1.LT=true;
y = linsolve(L,d,opt1);
opt2.UT=true;
x = linsolve(U,y,opt2)
% x =
%    -0.0394
%    -3.0319
%    18.0160

% check solution (norm of residuals)
norm(A*x - b)  % 1.7764e-15
```




```
% emission time
t = x(3)/c  % 9.0080

% hyperbolas for station pairs
h1 = @(x,y) sqrt((x-xi(1)).^2+(y-yi(1)).^2)-sqrt((x-x1).^2+(y-y1).^2)-c*(ti(1)-t1)
h2 = @(x,y) sqrt((x-xi(2)).^2+(y-yi(2)).^2)-sqrt((x-x1).^2+(y-y1).^2)-c*(ti(2)-t1)
h3 = @(x,y) sqrt((x-xi(3)).^2+(y-yi(3)).^2)-sqrt((x-x1).^2+(y-y1).^2)-c*(ti(3)-t1)

% plot
figure(1); hold on
fimplicit(h1,[-15,15])
fimplicit(h2,[-15,15])
fimplicit(h3,[-15,15])

% plot stations
plot(Pi(:,1),Pi(:,2),'ko')
% plot source
plot(P(1),P(2),'r*')
```

4. A receiver with unknown position measures the frequency of a radio signal emitted by a source moving along a known path. The problem is to determine the receiver position $P(x, y)$ by Doppler measurements to the moving source using the following specifications.

The positions of the source are known at three different epochs: $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_3)$ and $P_3 = (x_1, y_3)$ with the coordinates $P_1 = (-12, 3)$, $P_2 = (2, 10)$, $P_3 = (20, 14)$. The receiver was able to measure the integrated Doppler phase cycles $N_{12} = 4950$ and $N_{23} = 35542$ based on its continuous frequency measurements. The source-receiver range differences between epochs 1,2 és 2,3 can then be determined from the following formulas:

$$r_2 - r_1 = 0.00075 \cdot (N_{12} - 20000) \quad \text{and} \quad r_3 - r_2 = 0.00075 \cdot (N_{23} - 20000) \quad .$$

Range differences can also be calculated from the coordinates, hence the receiver position must lie at the intersection points of the following two hyperbolas:

$$\sqrt{((x-x_2)^2 + (y-y_2)^2)} - \sqrt{((x-x_1)^2 + (y-y_1)^2)} = r_2 - r_1 \quad \text{and}$$

$$\sqrt{((x-x_3)^2 + (y-y_3)^2)} - \sqrt{((x-x_2)^2 + (y-y_2)^2)} = r_3 - r_3 \quad .$$

Determine both possible locations $P(x, y)$ of the receiver!

The following tasks are to be completed (15 points):

a) Calculate source-receiver range differences between epochs 1,2 and 2,3. (1 point)

b) Define Matlab implicit functions for both hyperbolas in the form $h_1(x, y) = 0$ and $h_2(x, y) = 0$ (3 points)

c) Plot both hyperbolas on the same figure. Set plot x-range as [-15, 25]. (2 points)

d) Make a vectorial function for the hyperbolas using the two scalar functions (2 points)

e) Calculate the two intersection points of the hyperbolas by using Matlab's built-in numerical method for solving nonlinear equations. Find approximate values of the roots from the figure. (3 points)

f) Check the solutions numerically by substituting the roots into the equations. Check the solutions graphically by plotting the roots with different symbols. Plot the 3 known source positions into the same figure. (3 points)

g) Determine both distances from the two possible receiver postions to the source at the second epoch. (1 point)

```
% Doppler postioning

clc; clear all; close all

% calculate range differences
N12 = 4950; N23 = 35542;
```

```
dr12 = 0.00075*(N12-20000)
dr23 = 0.00075*(N23-20000)

% known source positions
P1 = [-12, 3];
P2 = [2, 10];
P3 = [20, 14];

% implicit equations of the hyperbolas
h1 = @(x,y) sqrt((x-P2(1)).^2+(y-P2(2)).^2) - sqrt((x-P1(1)).^2+(y-P1(2)).^2) - dr12
h2 = @(x,y) sqrt((x-P3(1)).^2+(y-P3(2)).^2) - sqrt((x-P2(1)).^2+(y-P2(2)).^2) - dr23

% plot
figure(1); hold on
fimplicit(h1, [-15,25])
fimplicit(h2, [-15,25])

% vectorial notation
h = @(x) [h1(x(1),x(2));
h2(x(1),x(2))]

% initial values from the figure
x01 = [5; 5];
x02 = [0; 18];

% solution with built-in Matlab func-
tion
x1 = fsolve(h, x01);
x2 = fsolve(h, x02);

% plot with different symbols
plot(x1(1),x1(2),'ko')
plot(x2(1),x2(2),'k*')

% positions of the moving source at three epochs
plot(P1(1),P1(2),'mx')
plot(P2(1),P2(2),'mx')
plot(P3(1),P3(2),'mx')

% receiver-source distances for the second epoch
t2 = @(x,y) sqrt((x-P2(1)).^2+(y-P2(2)).^2)

t21 = t2(x1(1),x1(2)) % 5.8334
t22 = t2(x2(1),x2(2)) % 8.3574
```