Problem 3

1. The position of four satellites are given in the table below in the unit of Mm (megameter).

Satellite No.	Satellite Position *x* (Mm)	Satellite Position *y* (Mm)	Satellite Position *z* (Mm)
1	101	16	207
2	52	21	302
3	17	53	350
4	-15	159	208

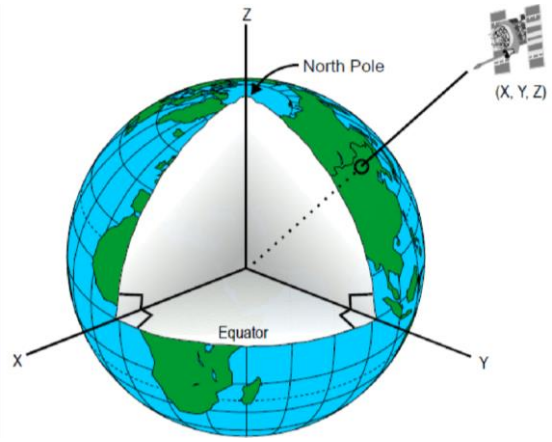
A GPS unit receives signals from the four satellites and the signals incorporate the positions of each satellite and also the following information:

- Sending times of each signal at the satellites, which are stored in file "st.mat"
- Receiving times of each signal at the GPS unit, which are stored in file "rt.mat"

Please calculate the position of the GPS unit in the unit of Mm and also the clock difference between the GPS unit clock and Satellite system clock in the unit of s (second) using numerical optimization function in Matlab. You need to submit both the solution presentation and codes.

Additional Notes:

- Use Matlab command, <load 'Name.mat'>, to load the data. For example, command <load 'st.mat'> will load a variable vector "st" and "st(1), st(2), st(3) and st (4)" will be the sending times of satellite 1, 2, 3 and 4 respectively. Do the same for "rt.mat".
- Speed of GPS radio wave: c = 300 Mm/s
- All the satellites share the same clock – satellite system clock.
- The available numerical optimization functions with examples in Matlab can be found at https://www.mathworks.com/help/optim/referencelist.html?type=function. Please explore and try different functions, such as "fminsearch", "fminumc", "fmincon", etc., to find out the best appropriate function.
- The radius of the earth is 6.378 Mm. The range of the GPS unit's position coordinates is [5, 20] Mm and the range of clock differences is [-10, 10] s. These information can be used to check the correctness of your results or set constraints for the numeric optimization functions.
- Coordinate system of the GPS system is defined as:



MATLAB CODE:

```
clear all
clc


%% ---------------Calculating the location of GPS and Clock time---------------
--------------:
syms x [1 4]
%Combined objective function for all satellites
func = Objective(x)
```

```
func =
```

$$\left((x_2-16)^2+(x_1-101)^2+(x_3-207)^2-\left(x_4-\frac{42684205511175}{137438953472}\right)^2\right.\quad (x_2-21)^2+(x_1-52)^2+(x_3-302)^2-\left(x_4-\frac{26629467094725}{68719476736}\right)^2\quad (x_1-17)^2+(x_2-$$

```
%Solving the above function to find out the location of the GPS and clock time
of GPS.
%Considering first value of the result array because of the distance and time
range..
%mentioned in the question.
res=solve(func,x);

x=vpa(res.x1(1))
```

```
x =
5.199999988557892922355936903893 5
```

```
y=vpa(res.x2(1))
```

```
y =
7.599999982665580858374138853357 3
```

```
z=vpa(res.x3(1))
```

```
z =
8.499999964281982527223186329334 5
```

```
%calculating gps clock time:
c=300;
ts=vpa(res.x4(1))/c
```

```
ts =
0.29999999987246866615747342118609
```

```
%% ----------------------------Optimization-----------------------------------:
joint_obj = @(x) OPTM_A(x).^2 + OPTM_B(x).^2+OPTM_C(x).^2 + OPTM_D(x).^2;
%adding constraints through equations
joint_constraint = @(x) merge_constraints(x);


%% Initial guesses x=5,y=5,z=5, t=0.2
x0=[5 5 5 0.2];
[x1,fval1,exitflag1,output1] = fmincon(@(x)joint_obj(x),
x0,[],[],[],[],[],[],joint_constraint,[])
```

```
Feasible point with lower objective function value found.


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>
x1 = 1×4
    5.1649    7.5490    8.3978   89.8908
fval1 = 0.1010
exitflag1 = 2
output1 = struct with fields:
         iterations: 56
          funcCount: 324
    constrviolation: 0
           stepsize: 1.1062e-09
          algorithm: 'interior-point'
      firstorderopt: 2.5390
       cgiterations: 24
            message: '↵Local minimum possible. Constraints satisfied.↵↵fmincon stopped
because the size of the current step is less than↵the value of the step size tolerance and
constraints are ↵satisfied to within the value of the constraint tolerance.↵↵<stopping
criteria details>↵↵Optimization stopped because the relative changes in all elements of x
are↵less than options.StepTolerance = 1.000000e-10, and the relative maximum
```

```
constraint↲violation, 0.000000e+00, is less than options.ConstraintTolerance = 1.000000e-
06.↲↲'
       bestfeasible: [1×1 struct]
```

```
[x2,fval2,exitflag2,output2] = fminimax(@(x)joint_obj(x),
x0,[],[],[],[],[],[],joint_constraint,[])
```

```
Local minimum possible. Constraints satisfied.

fminimax stopped because the size of the current search direction is less than
twice the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>
x2 = 1×4
     4.7777     6.9779     7.2547    88.6694
fval2 = 18.4614
exitflag2 = 18.4614
output2 = 4
```

```
[x3,fval3,exitflag3,output3] = fminsearch(@(x)joint_obj(x), x0)
```

```
Exiting: Maximum number of function evaluations has been exceeded
         - increase MaxFunEvals option.
         Current function value: 29.522387
x3 = 1×4
     5.0470     7.3636     7.9943    89.4697
fval3 = 29.5224
exitflag3 = 0
output3 = struct with fields:
    iterations: 475
     funcCount: 800
     algorithm: 'Nelder-Mead simplex direct search'
       message: 'Exiting: Maximum number of function evaluations has been exceeded↲
- increase MaxFunEvals option.↲         Current function value: 29.522387 ↲'
```

```
[x4,fval4,exitflag4,output4] = fminunc(@(x)joint_obj(x), x0)
```

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
x4 = 1×4
   -7.5610   -11.4323   -30.8550    47.7209
fval4 = 7.0943e+03
exitflag4 = 1
output4 = struct with fields:
       iterations: 16
        funcCount: 95
         stepsize: 4.1044e-04
```

```
      lssteplength: 1
     firstorderopt: 152.9360
         algorithm: 'quasi-newton'
           message: '↵Local minimum found.↵↵Optimization completed because the size of the
gradient is less than↵the value of the optimality tolerance.↵↵<stopping criteria
details>↵↵Optimization completed: The first-order optimality measure, 4.553806e-07, is less
↵than options.OptimalityTolerance = 1.000000e-06.↵↵'
```

```
%based on the function values for each of the optimization method,
the %error/function value(fval) is least for fmincon function and
other factors like number of iterations taken to arrive to these
values is also least for fmincon which tells about the performance for
same set of inputs

function [merged_c, merged_ceq] = merge_constraints(x)
    c1 =  OPTM_A(x) ;
    c2 =  OPTM_B(x);
    c3 =  OPTM_C(x);
    c4 =  OPTM_D(x);

    ceq1=0;ceq2=0;ceq3=0;ceq4=0;

    merged_c = [c1(:); c2(:);c3(:);c4(:)];
    merged_ceq = [ceq1(:); ceq2(:);ceq3(:);ceq4(:)];
end

Answer:
%%%based on the function values for each of the optimization method,
the %error/function value(fval) is least for fmincon function and
other factors like number of iterations taken to arrive to these
values is also least for fmincon which tells about the performance for
same set of inputs.

x1 = 1×4
    5.1649    7.5490    8.3978   89.8908
fval1 = 0.1010
```