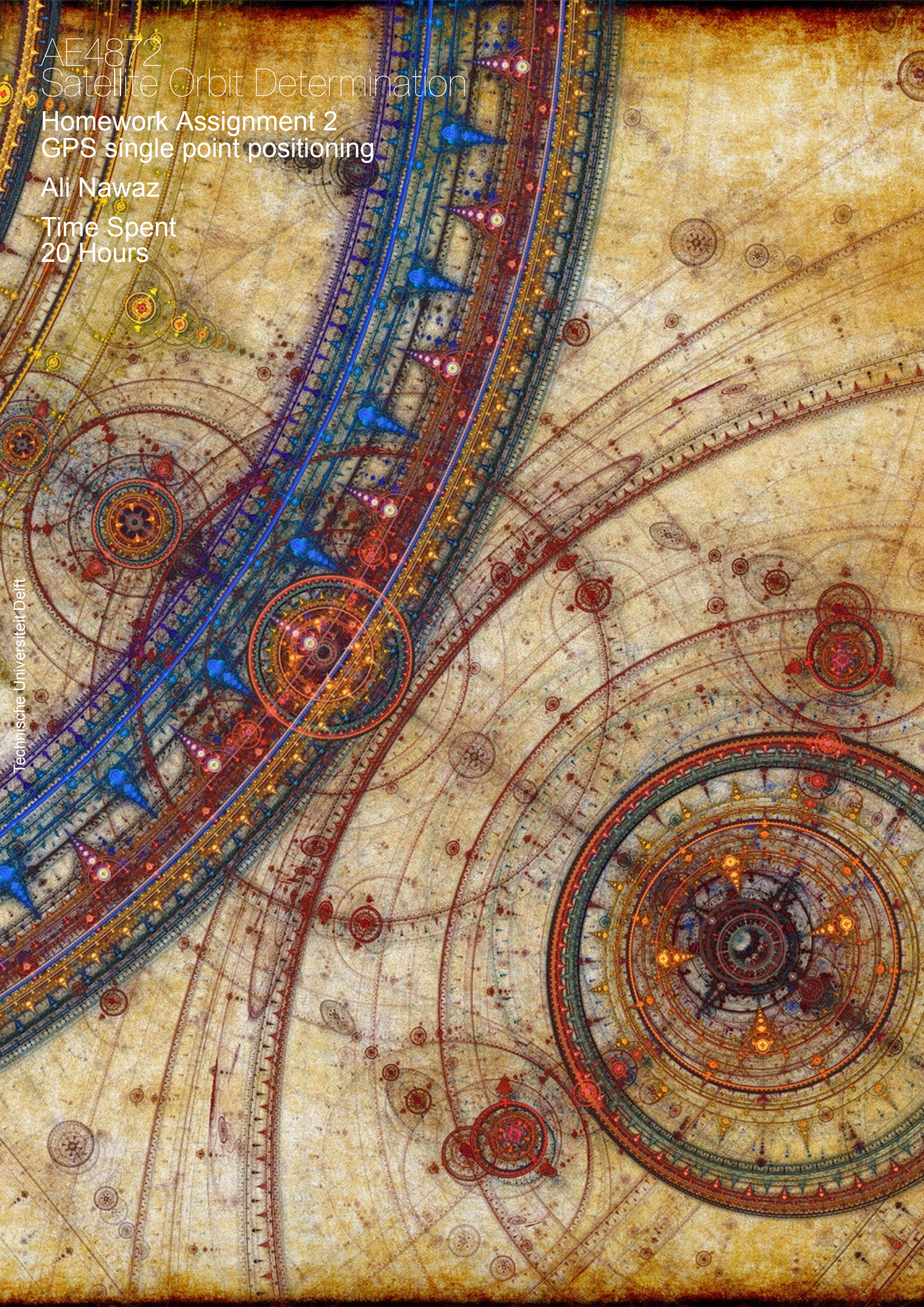
AE4872
Satellite Orbit Determination

Homework Assignment 2
GPS single point positioning

Ali Nawaz

Time Spent
20 Hours

Technische Universiteit Delft



List of symbols

Table 1: List of symbols

Symbol/Acronym	*Description*	*Unit*
A	Information/Data Matrix	-
LSQ	Least Squares	-
WLSQ	Weighted Least Squares	-
MSL	Matlab Script Lines	-
n	Number of data points	-
pinv.	Pseudo Inverse	-
SVD	Singular Value Decomposition	-
t	Time	s
var.	Variance	Hz^2
x	State vector	-
$\bar{y}$	Observation vector	-
$\hat{y}$	Estimated observation vector	-
$\hat{x}$	Estimated state vector	-
ρ	Range measurement	m
$\tilde{\rho}$	Pseudo range	m
t_T	Transmitter time	s
t_R	Receiver time	s
x_{gps}	GPS x location	m
y_{gps}	GPS y location	m
z_{gps}	GPS z location	m
s	State vector	-
s_0	Initial state vector	-
c	speed of light, 299792458	m/s

Cover Image: Fractal Star Navigation http://i.imgur.com/E3UbhwX.jpg.

Contents

1

Data Retrieval

The objective of this assignment is to determine the position of the SWARM A satellite (one out of the three SWARM satellites) with GPS pseudo-range measurements. Iterative LSQ is used to determine the position. The assignment is conducted on the Matlab file `Assignment2_start.m`, provided by course instructor Dr.ir. W. (Wouter) van der Wal [1]. A copy of the file can be found at the github repository [2].

In order to compute the position of the SWARM satellite, position of the GPS satellites must be known. Since the data is post processed, one can make use of the precise GPS orbits and clock corrections provided by the International GNSS Service[3]. GPS co-ordinates and clock corrections are provided by the course instructor in the file `igs19301.sp3` and can be found in the Git repository.

The main Matlab file reads precise GPS co-ordinates and clock corrections with the aid of function readsp3.m which in turn requires a Matlab routine `read_strval.m`. GPS co-ordinate measurements are presented in the variables `x_`gps, `y_gps` and `z_gps` along with the clock correction presented in `clock_correction`.

SWARM GPS pseudo-ranges are presented in the file `Data_SWARM_A.dat`. Since the co-ordinates and clock corrections are not presented for the same time as the GPS observations from the SWARM, they are interpolated. Matlab file `polint.m` uses an 11^{th} order Lagrange polynomial interpolation for the co-ordinates and 3^{rd} order polynomial interpolation for clock correction. Interpolated variables can be accessed in the following variables: `x_gps_interpolated`, `y_gps_interpolated`, `z_gps_interpolated` and `clock_correction_interpolated`. Finally, precise orbit of SWARM A satellite is provided in the file `PreciseOrbit_SWARM_A.dat` against which the results of this assignment are compared.

[1] Course Instructor: https://www.tudelft.nl/staff/w.vanderwal/
[2] Supporting files : https://github.com/Alixir/Satellite-Orbit-Determination
[3] IGNSS : http://www.igs.org/

2

Epoch-wise LSQ

2.1. Position estimation and residual behaviour

The aim of this section is to estimate the position of the SWARM satellite for each epoch. A receiver clock correction is not estimated. Observed range measurement by SWARM A satellite is non-linear as expressed in Equation 2.1. In order to make use of linear LSQ, the problem must be linearized such that it can be expressed in the from of Equation 2.2.

$$\rho = \sqrt{(x - x_{GPS})^2 + (y - y_{GPS})^2 + (z - z_{GPS})^2} \tag{2.1}$$

$$\bar{y} = A \cdot \bar{x} \tag{2.2}$$

The problem expressed by Equation 2.1 can be linearized around the state vector $s_0 = [x\ y\ z]^T$ as expressed in Equation 2.3. The linearised equation presented in Equation 2.3 can be extended to multiple satellites per epoch to facilitate epoch-wise or batch-wise LSQ.

$$\rho(s) - \rho(s_0) = \begin{bmatrix} \frac{\delta\rho(s)}{\delta x} & \frac{\delta\rho(s)}{\delta y} & \frac{\delta\rho(s)}{\delta z} \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{2.3}$$

Terms forming the fundamental basis of Equation 2.3 can be expressed as shown in Equation 2.4

$$\begin{aligned} \frac{\delta\rho}{\delta x} &= 2 \cdot \frac{x - x_{GPS}}{2} \cdot ((x - x_{GPS})^2 + (y - y_{GPS})^2 + (z - z_{GPS}^2))^{-\frac{1}{2}} \\ \frac{\delta\rho}{\delta x} &= \frac{x - x_{GPS}}{\rho} \end{aligned}$$

Simlarly, for the last two terms in design matrix: (2.4)

$$\begin{aligned} \frac{\delta\rho}{\delta y} &= \frac{y - y_{GPS}}{\rho} \\ \frac{\delta\rho}{\delta z} &= \frac{z - z_{GPS}}{\rho} \end{aligned}$$

The observation vector $\bar{y}$ in Equation 2.2 is comprised of the terms stated in Equation 2.5. Where $\tilde{\rho}$ is the observed range, known as the pseudorange. While, $c \cdot t_T$ indicates the effect of GPS clock correction; and ρ is the measured range, which is updated every iteration based on the current state s. The information matrix in Equation 2.2 now simplifies to the one stated in Equation 2.5. Where x, y and z are updated every iteration.

$$\begin{aligned} \bar{y} &= \tilde{\rho} + c \cdot t_T - \rho \\ A &= \begin{bmatrix} \frac{x - x_{GPS}}{\rho} & \frac{y - y_{GPS}}{\rho} & \frac{z - z_{GPS}}{\rho} \end{bmatrix} \end{aligned} \tag{2.5}$$

State estimation for the LSQ problem can be obtained via pseudo inverse of the information matrix or with the aid of Singular Value Decomposition. Equation 2.6 illustrates LSQ with pseudo-range.

$$\begin{aligned} \hat{x} &= (A^T \cdot A)^{-1} \cdot A^T \cdot \bar{y} \\ \hat{y} &= A \cdot \hat{x} \end{aligned} \tag{2.6}$$

While Equations 2.8 to 2.11 illustrate how the state estimation in Equation 2.7 can be extended to a Singular Value Decomposition problem with the aid of pseudo inverse . Rank of the information matrix provides information regarding information content of the matrix. A might undergo rank loss during the pseudo-inverse process, this might result in singularities in the solution. These singularities often lead to in-accurate results. Since loss of rank represents valuable loss of information. Even if rank loss was not observed during the pseudo inverse process, Matlab presented warnings regarding the presence of singularities in the numerical solution. However, with the implementation of SVD algorithm, neither the loss in rank nor the presence of singularities in Matlab pertains. Thus LSQ is conducted by first applying SVD on the information matrix. The residual of the measurement and observation can be obtained as shown in Equation 2.12.

$$\hat{x} = (A^t \cdot A)^{-1} \cdot A^t \cdot \bar{y} \tag{2.7}$$

$$A^t A = V\Lambda^2 U^t U\Lambda V^t = V\Lambda^2 V^t \tag{2.8}$$

$$A^t \bar{y} = V\Lambda U^t \bar{y} \tag{2.9}$$

$$V\Lambda^2 V^t \bar{x} = V\Lambda U^t \bar{y} \quad \rightarrow \quad V^t \bar{x} = U^t \bar{y} \tag{2.10}$$

$$\bar{x} = V\Lambda^{-1} U^t \bar{y} \tag{2.11}$$

$$\text{residual} = \bar{y} - \hat{y} \tag{2.12}$$

Now that the fundamentals of LSQ is described, the next step is to conduct an epoch-wise LSQ. To aid the reader in the visualisation process, flowchart presented in Figure 2.1 simplifies the outline of the algorithm used to facilitate epoch-wise position estimation of SWARM A satellite.

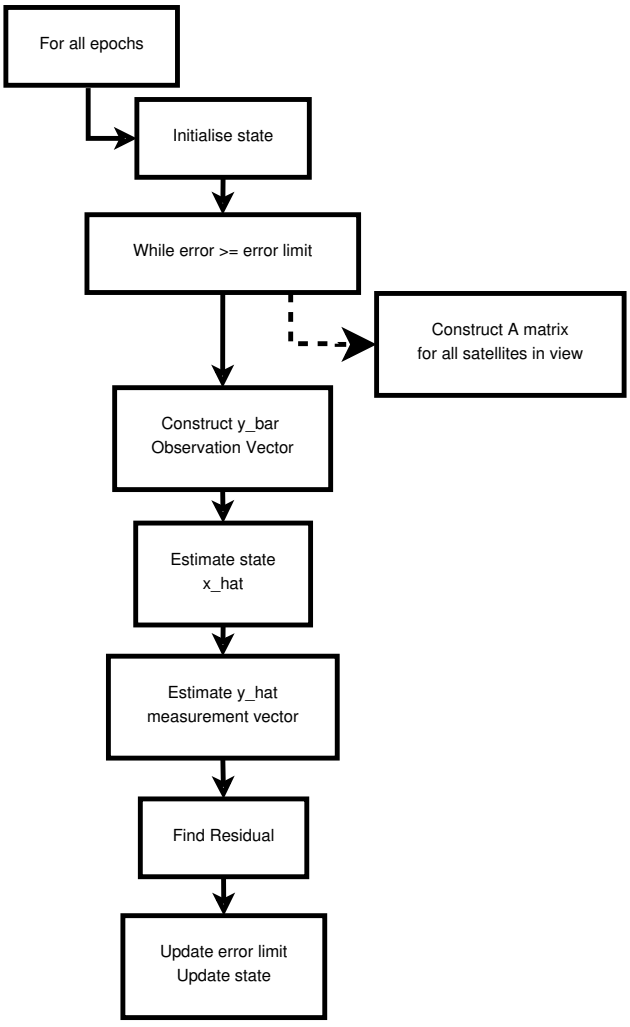


Figure 2.1: Flowchart illustrating steps undertaken to conduct epochwise LSQ.

Supplementary script for the algorithm presented in Figure 2.1 is presented in Appendix . The script is used to estimate the position of the SWARM satellite per epoch. Epoch-wise norm of the residuals for all satellites in view/epoch is presented in Figure 2.2.

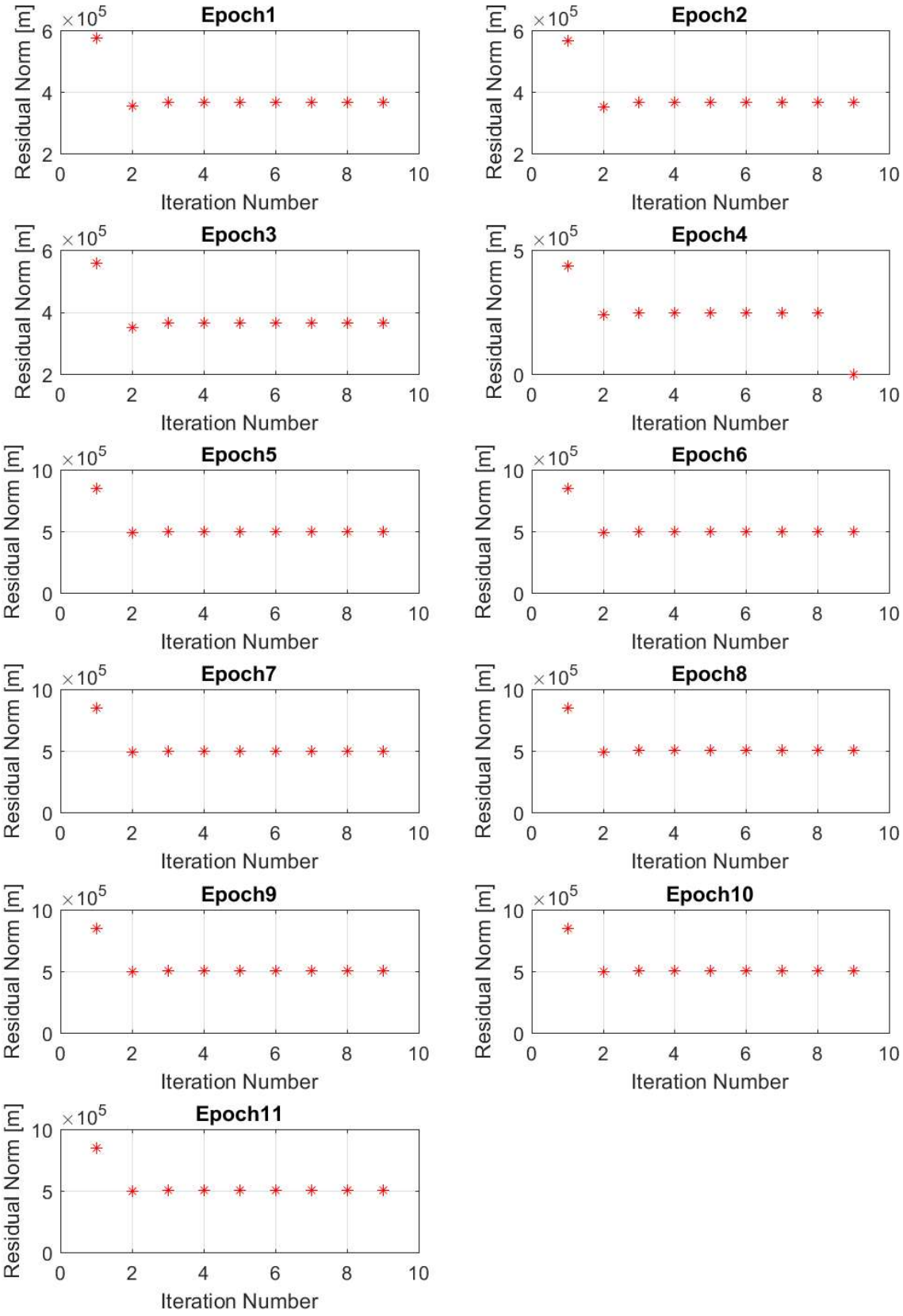


Figure 2.2: Epoch wise norm of the residuals for all satellites in view/epoch.

2.2. Difference between measured and precise orbit

This section outlines the difference between the measured altitude of the satellite and the precise orbit direction available. Figure 2.3 illustrates the measured orbital altitude and the precise orbital altitude observed over 11 epochs. While, Figure 2.4 illustrates the difference between the measured and the precise orbital altitude observed over 11 epochs. An important observation is the jump in Figure 2.4 between epochs 4 and 5. Furthermore, the curve changes direction between epochs 3 and 4 as well. This can be explained by the loss of lock/cycle slip of GPS link for the 4th epoch, where some link cycles are lost.

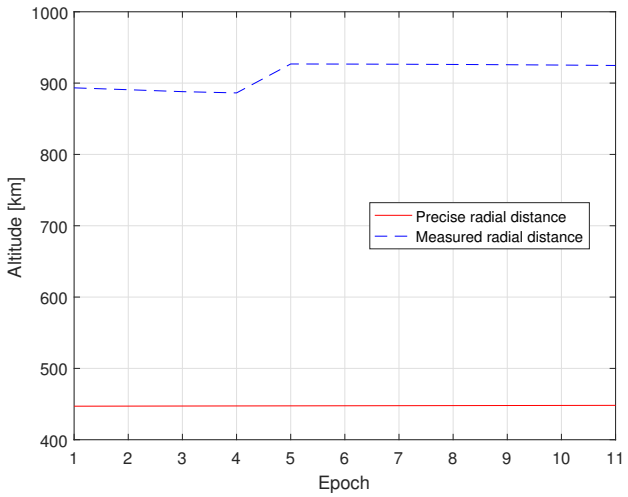


Figure 2.3: Measured and precise orbit direction for SWARM A satellite.

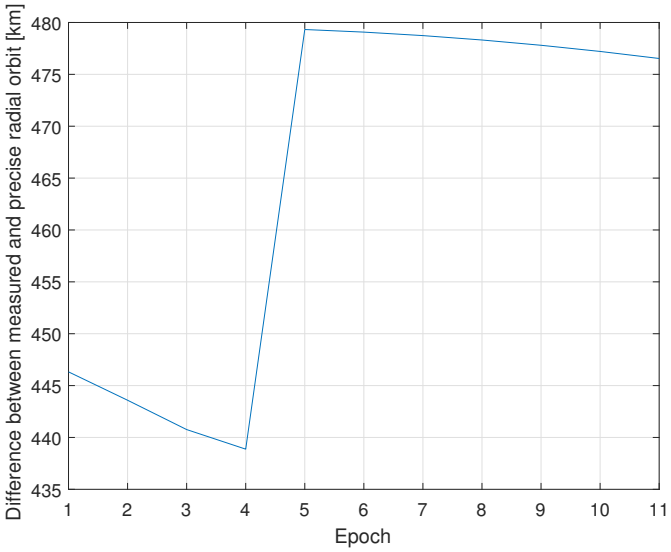


Figure 2.4: Difference between the measured and precise orbit direction for SWARM A satellite.

3

Batch wise LSQ

This chapter focuses on estimating one receiver clock correction for all epochs. First receiver clock correction is estimated by batch processing observations from all epochs. An intermezzo is provided to observe the behaviour of residuals in batch processing. The clock correction is later included in the iteration loop to observe the effect of including light-time effect. The residuals and orbital positions are re-analysed after the inclusion of light time effect. This is followed by a brief discussion outlining the possible sources of errors and their importance is the observed results. To conclude this chapter implications of the choice of weights in case of a weighted LSQ solution is analysed.Flowchart illustrated in Figure 3.1 outlines the fundamental basis of the batch wise LSQ algorithm used to estimate the receiver clock error for all epochs.

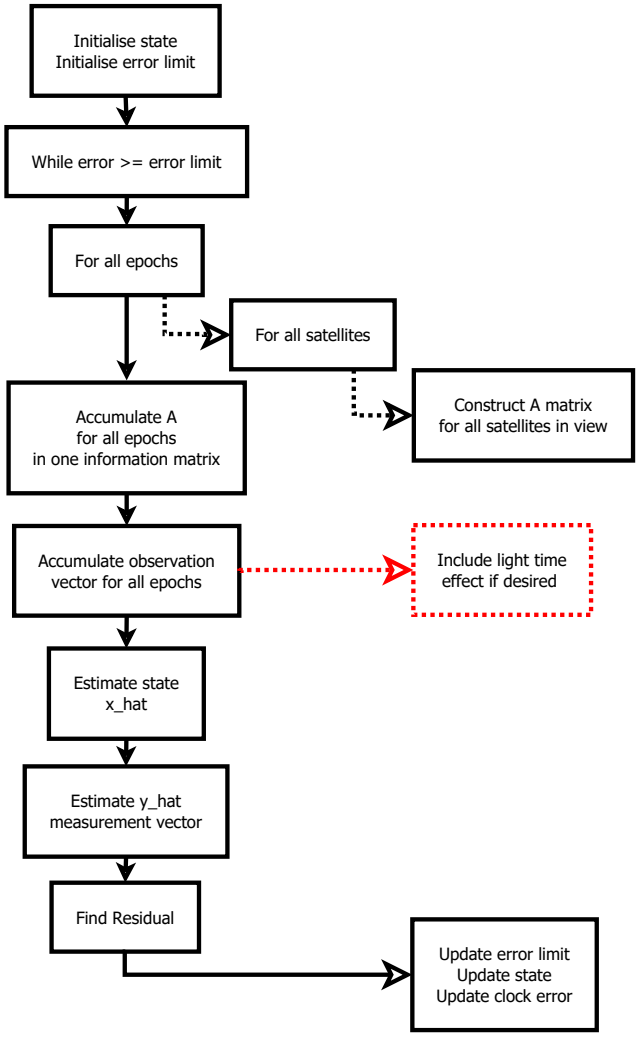


Figure 3.1: Flowchart illustrating steps undertaken to conduct batch wise LSQ.

3.1. Receiver clock correction for all epochs

Supplementary script for the receiver clock correction is presented in the Appendix. One receiver clock is estimated for all epochs. Figure 3.2 illustrates the evolution of the receiver clock error over increasing iterations. The observation vector is not corrected for the light time effect yet. It can be observed that the receiver clock error linearly becomes more and more negative over iterations.

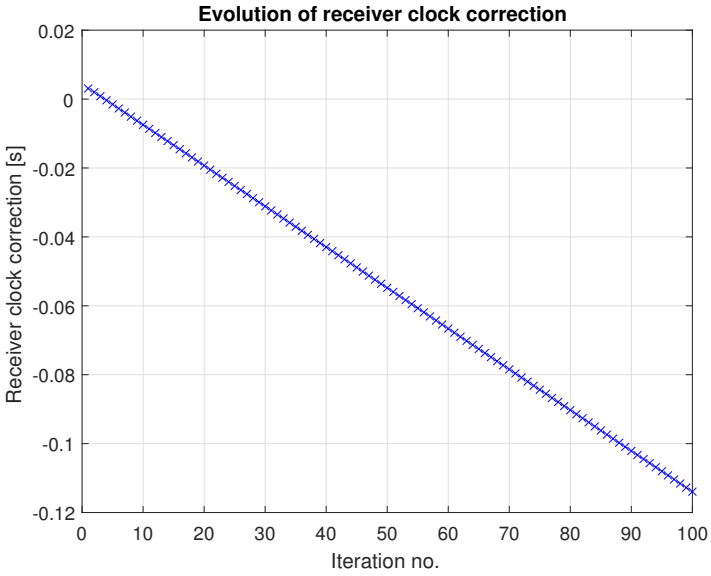


Figure 3.2: Evolution of receiver clock correction over 100 iteration. Light time effect is not included yet.

3.2. Residual Behaviour

Figure 3.5 illustrates the behaviour of the residual norm per epoch for first 100 iterations. While Figures 3.3 and 3.4 illustrate the orbital altitude measured via batch LSQ against the precise orbital altitude, and the difference in between them. Comparing Figures 3.3 and 3.4 against 2.3 and 2.4, the accuracy has improved significantly. Furthermore, comparing the residual norms/epoch in Figures 3.5 and 2.2, it can be observed that residual norms have reduced in significant orders.

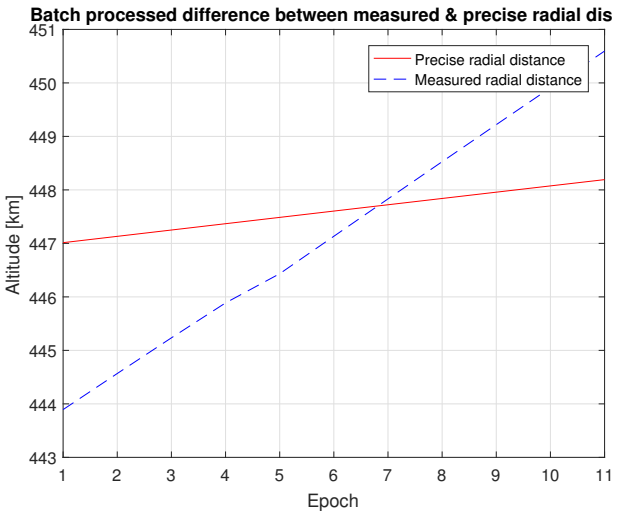


Figure 3.3: Measured and precise orbit direction for SWARM A satellite via batch LSQ.

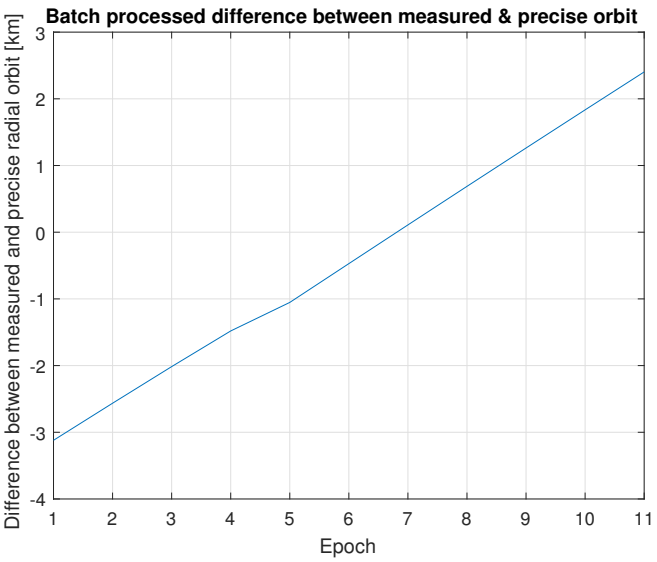


Figure 3.4: Difference between the measured and precise orbit direction for SWARM A satellite via batch LSQ.

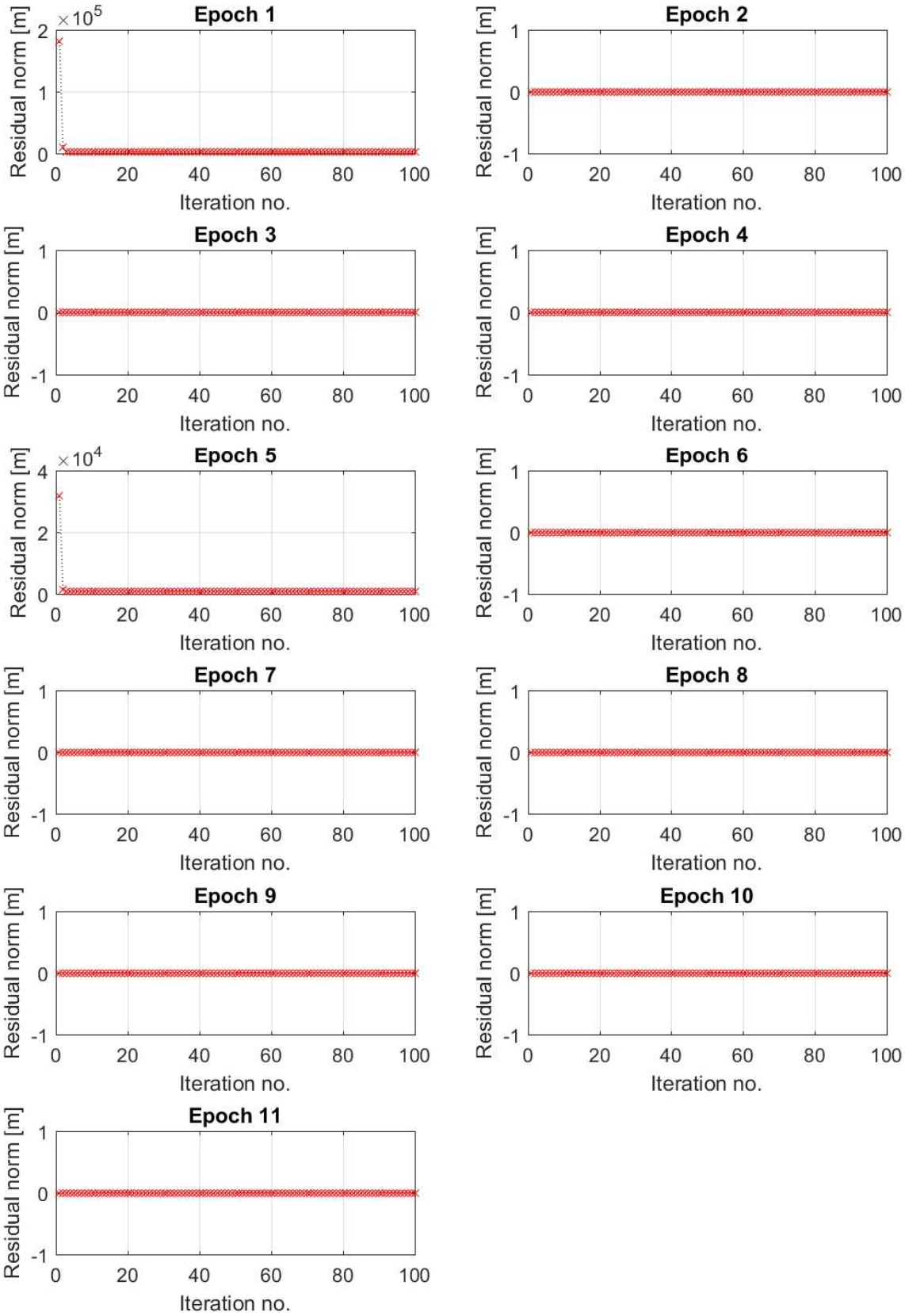


Figure 3.5: Evolution of batch-wise norm of the residuals for all satellites in view/epoch for 100 iterations.

However, one interesting observation is observed for epochs 1 and 5. The residual norm for all epochs reach 0m, apart from these two epochs. Where, for the first epoch the residual norm reduces to 2586m and that of the fifth epoch settles down to 872.2m. The 5th epoch is the epoch where the number of satellites in view rise from 7 to 8. Number of satellites in view before the first epoch is not known. It could be that Epoch 1 undergoes an increase in the view of GPS sat compared to previous epoch. Overall, it can be concluded that

the residual norms decrease and the accuracy improves for batch wise LSQ when compared to epoch wise LSQ.

3.3. Effect of light-time effect inclusion

Now the updated receiver clock correction is included in the observation vector every iteration. The updated observation Equation is presented in Equation 3.1. This is nothing more than inclusion of light-time effect in Equation 2.5.

$$\bar{y} = \tilde{\rho} - (c \cdot t_R - c \cdot t_T) - \rho \tag{3.1}$$

Figure 3.6 illustrates the behaviour of the receiver clock correction over increasing iterations. While Figures 3.7 and 3.8 illustrate the orbital altitude measured via batch LSQ against the precise orbital altitude, and the difference in between them. Now, the light time effect is updated and included in the observation vector every epoch.

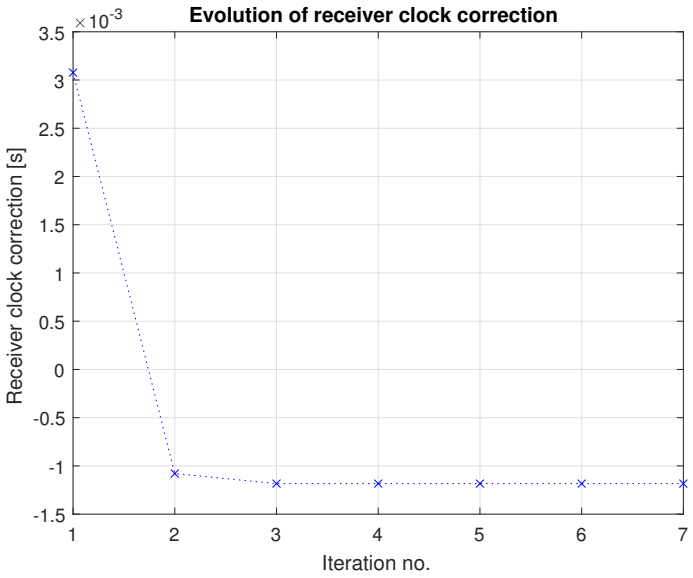


Figure 3.6: Evolution of receiver clock correction over increasing iterations. Light time effect is iterated and included in the observation vector.

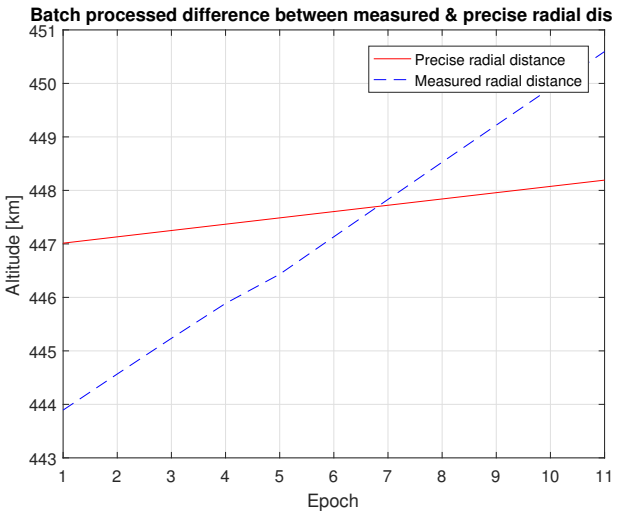


Figure 3.7: Measured and precise orbit direction for SWARM A satellite via batch LSQ. Light time effect is iterated and included in the observation vector.

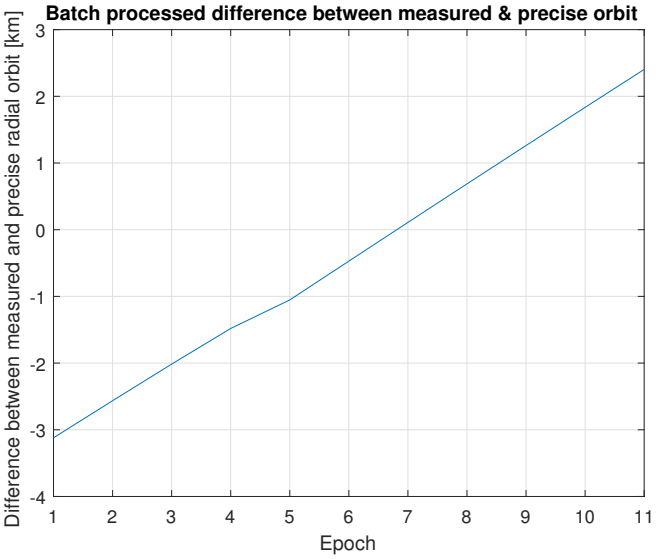


Figure 3.8: Difference between the measured and precise orbit direction for SWARM A satellite via batch LSQ. Light time effect is iterated and included in the observation vector.

Finally for the sake of completion evolution of the residual norm/epoch over increasing iterations, with inclusion of iterative light time effect is presented in Figure 3.9. Similar trend is observed compared to the scenario

where light time effect was not included.

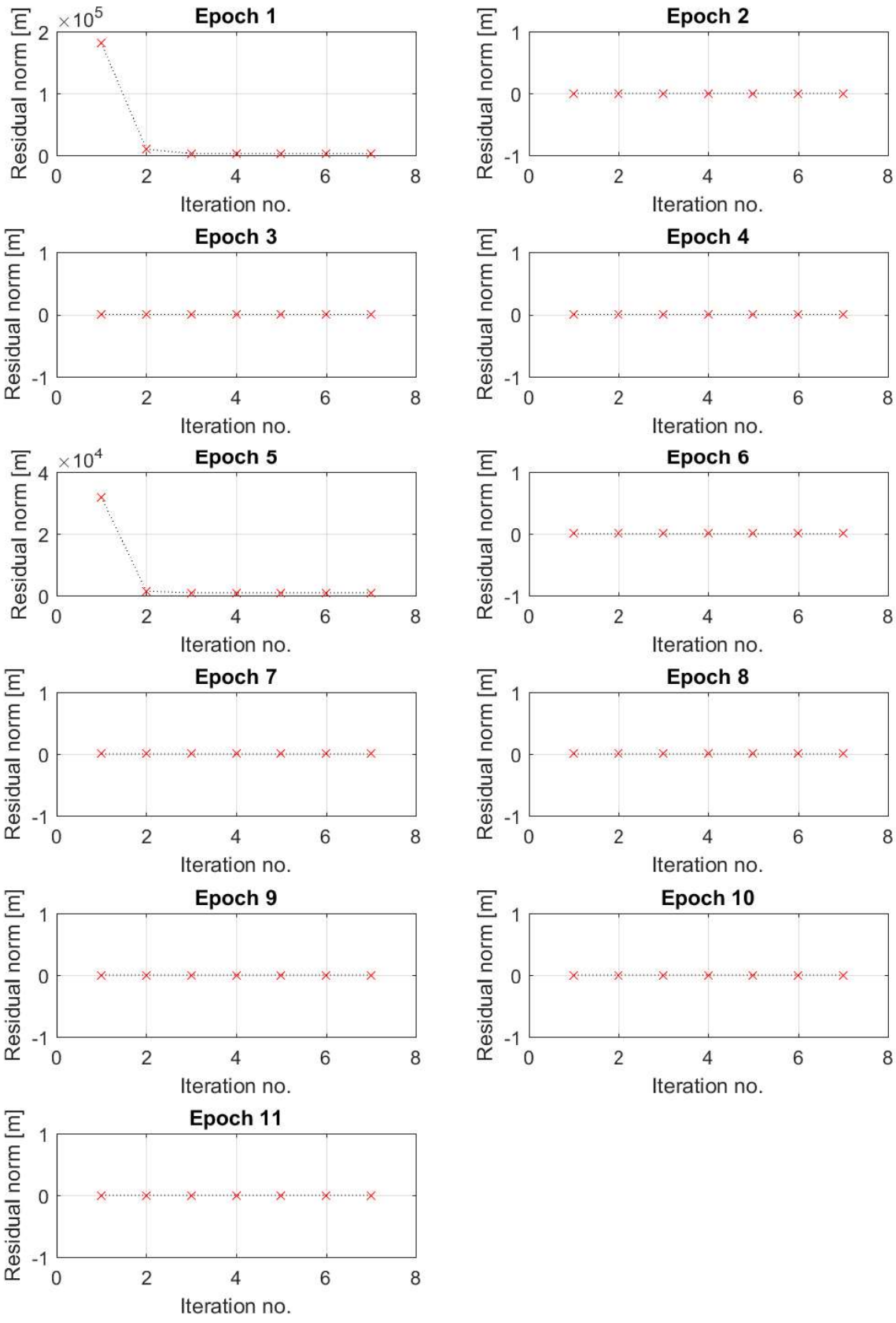


Figure 3.9: Evolution of batch-wise norm of the residuals for all satellites in view/epoch for increasing iterations. Light time effect is included.

3.4. Sources of errors

Possible sources of errors that cause the difference between the measured solution and the precise orbit are presented as follows in order of importance:

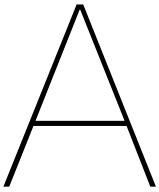
- Time Synchronisation of receiver and transmitter clock. Since no single or double difference is carried out, it might as well be that the clocks on GPS and the receiver are not synchronised resulting in divergence.
- Loss of signal : Loss of signal cycles in between the GPS and SWARM A satellites. Including quantization errors and packet loss. Especially seen for Epoch 4.
- Instrumentation errors and noise present in the data. The data is not filtered and structured before processing. Co-variance information of the available pseudo-range data is not used in the LSQ process. In an ideal scenario, the LSQ process would make use of filters and co-variance information of the observed dataset and possible error sources.
- Ionospheric effects: Since the satellites are in ionosphere, the signals are dispersed by the ionosphere.
- Ephemeris errors: Errors present in the provided precise orbit.
- Interpolation errors: The GPS locations and transmitter clock correction are provided after interpolation. Accuracy of the interpolated result is not verified as part of this assignment.
- Relativistic effects: Not all of them are taken into account.
- Tropospheric effects for GPS satellites close to the horizon, such that the signal is dispersed as it passed through the atmospheric layers.

3.5. Weight determination for weighted LSQ

Methods of least square to determine the estimated state parameters and hence the measured vectors, did not involve inclusion of weights on observation vector. Hence problems of such nature are known as unweighted least square problem. Inclusion of co-variance information of the observation is extremely crucial to remove the errors and obtain a better fit of the model to the real observed dataset. This is where Weighted Least Square (WLSQ) comes in. The weights in WLSQ are often illustrated in form of a diagonal matrix. Where the diagonal terms are inverse of the standard deviation for a given observation [1] [2]. This means higher the variance or deviation of an observation, lower the importance is paid to it while re-constructing the observation vector via LSQ. In this specific case of SWARM A satellite, it was observed that at Epoch 4 there is a loss in the number of GPS satellites in view, thereby it is a perfect choice for lower weight. However while batch processing the dataset, it can be observed from Figure 3.9 and 3.6 that significant effect of re-gaining connection with new satellite is seen on epoch 5. While for both epoch 1 and 5 the residual norms do not tend to zero. These observations indicate it being reliable to denoted less weights to epochs 1 and 5 as well. Thus taking all observations in view, application of less weights to epochs 1,4 and 5 might result in better fit with WLSQ method. Furthermore, in case of Batch WLSQ, less weight should also be applied to the terms associated with receiver clock correction as well. Since, nothing is known about it primarily.

[1] Weighted Least Squares Method: https://ww2.amstat.org/sections/srms/Proceedings/y2013/files/308377_80748.pdf

[2] M. Verhaegen, Filtering and Sytem Identification: http://tinyurl.com/filtering-and-sys-ident



Matlab Script

Following script can be also be found at this Github Repository:
https://github.com/Alixir/Satellite-Orbit-Determination

```
% Single point positioning
% Wouter van der Wal
% Delft University of Technology

clear all
close all
clc

% Fundamental GPS constants
gpsconst;

dt                  = 10;       % time step
number_of_epochs    = 11;
order               = 11;       % order of interpolation of precise orbits GPS satellites
order_clk           = 3;        % order of interpolation of clock correction
t_orbit             = 60*[-78:15:73];  % time vector for precise orbits for GPS satellite provided by IGS every 15 minutes
time                = 0:dt:(number_of_epochs-1)*dt; % time vector for SWARM satellite
c                   = 299792458; % velocity of light m/sec
earth_radius        = 6371; % in km
%% Read the IGS coordinates and clock corrections for all the GPSsatellites
[x_gps,y_gps,z_gps,clock_correction] = readsp3('igs19301.sp3');


%% Read the Swarm GPS data
fid_rinex = fopen('Data_SWARM_A.dat','r');
for ii = 1:15
    line = fgets(fid_rinex);
end
% Read the observations
for epoch = 1:number_of_epochs
    line = fgets(fid_rinex);
    ft = sscanf(line,'%c %f %f %f %f %f %f %f %f');
    no_sat(epoch) = ft(9);
    si = 0;
    for no = 1:no_sat(epoch)
        si = si + 1;
        line = fgets(fid_rinex);
        ft = sscanf(line,'%c %f %f %f %f %f %f %f %f %f');
        prns(epoch,no) = ft(2);        % PRN number of satellite
        n = prns(epoch,no);
        C1(epoch,si) = ft(5);
       % Interpolate the coordinates of the GPS satellites to the current epoch
        [x_gps_interpolated(epoch,si),x_error(epoch,si)] = polint(t_orbit,x_gps(:,n),time(epoch),order);
        [y_gps_interpolated(epoch,si),y_error(epoch,si)] = polint(t_orbit,y_gps(:,n),time(epoch),order);
        [z_gps_interpolated(epoch,si),z_error(epoch,si)] = polint(t_orbit,z_gps(:,n),time(epoch),order);
        [clock_correction_interpolated(epoch,si),t_error(epoch,si)] = polint(t_orbit,clock_correction(:,n),time(epoch),order_clk);
    end
```

```
end
fclose(fid_rinex);
ccGPS = clock_correction_interpolated*c;
x_gps_interpolated = x_gps_interpolated*1e3;                    % Convert to meters
y_gps_interpolated = y_gps_interpolated*1e3;
z_gps_interpolated = z_gps_interpolated*1e3;
r_gps_interpolated = sqrt(x_gps_interpolated(:,1).^2+y_gps_interpolated(:,1).^2+z_gps_interpolated
    (:,1).^2);
% Plot for checking
% figure; plot((r_gps_interpolated-6371e3)/1e3); title('altitude PRN 7 [km]')


%% Read the precise Swarm orbit
fid_obs = fopen('PreciseOrbit_SWARM_A.dat','r');
for ii = 1:22
    line = fgets(fid_obs);
end
for epoch = 1:11
        line = fgets(fid_obs);
        line = fgets(fid_obs);
        x_precise(epoch) = read_strval(line(5:18)); % These are in km from the center of the earth
    - Ali
        y_precise(epoch) = read_strval(line(19:32));
        z_precise(epoch) = read_strval(line(33:46));
        line = fgets(fid_obs);
end
fclose(fid_obs);
% Convert to spherical coordinates for checking
r_precise = sqrt( x_precise.^2+y_precise.^2+z_precise.^2 );
% Plot for checking
% figure; plot(r_precise-6371); title('altitude Swarm [km]')
%% ========== START OF STUDENT SCRIPT ==============
%% Author: Ali Nawaz
% Course: AE4872 Satellite Orbit Determination
% Homework Assignment 2: GPS single point positioning
%%================================================
%% Data retrieval
xg = x_gps;
yg = y_gps;
zg = z_gps;
cc = clock_correction;

xgi = x_gps_interpolated;
ygi = y_gps_interpolated;
zgi = z_gps_interpolated;
rgi = r_gps_interpolated;
cci = clock_correction_interpolated;
ccf = ccGPS; % Final clock correction = cci*c

xp = x_precise;
yp = y_precise;
zp = z_precise;
rp = r_precise;

%% Part 1a: Epoch wise position estimation of SWARM satellite
rank_e = []; % initialising rank check per epoch for the
%Scan through each epoch
for e = 1:epoch
    clear A dx dy dz drho;
    s0 = [ 10^3; 10^3; 10^3 ]; % initial guess x0, y0, z0 in meters
    diff = norm(s0); % Initialise error limit
    iter =0; % initialising iteration number
    while diff>=10^(-15)
        iter = iter +1; % current iteration number
        for s = 1:no_sat(e)
            dx(s) = s0(1) - xgi(e,s); % x - x_gps for sth satellite in view
            dy(s) = s0(2) - ygi(e,s); % y - y_gps for sth satellite in view
            dz(s) = s0(3) - zgi(e,s); % z - z_gps for sth satellite in view

            drho(s) = sqrt(dx(s)^2 + dy(s)^2 + dz(s)^2); % measured range for nth satellite in view
            A(s,:) = [ dx(s)/drho(s) dy(s)/drho(s) dz(s)/drho(s)]; % Assembling epochwise
    information matrix. Final matrix for each epoch= no. of satellites in view/ epoch by 3
        end
        % Determining observation vector y_bar
        % y_bar = rho(s) - rho(s0) + c*t_T
```

```
        % y_bar = observed pseudorange - measured range for given initial
        % guess + effect of GPS clock correction
        y_bar = transpose(C1(e,1:no_sat(e)) - drho + ccf(e,1:no_sat(e)));
        rank_e(e) = rank(A); % Rank check before application of LSQ with pinv
        % Unwieghted LSQ to obtain the state vector x_hat
        % LSQ with pseudo-inverse
%         x_hat = (A'*A)\A'*y_bar;
        % LSQ with SVD
        [U,S,V] = svd(A,'econ'); % Singular value decomposition of A, to avoid singularity errors or solution manifold due to rank deficit
%         condition_number = S(1,1)/S(end,end); % Indicates the number of potential digit that can be lost in any numerical calculations when A is computed directly.
        S_inv_diag = [];
        for nn=1:length(S)
            S_inv_diag = [S_inv_diag,inv(S(nn,nn))];
        end
        S_inv = diag(S_inv_diag);
        x_hat = V*S_inv*U'*y_bar; % Parameter estimation via SVD aided LSQ

        % Estimated vector
        y_hat = A*x_hat;

        res = y_bar - y_hat; % residual in meters
        res_norm(e,iter) = norm(res);% Epoch wise norm of the residual vector of all satellites

        % Updating error limit and state
        diff = abs( ( norm(x_hat + s0) - norm(s0) )/ ( norm(s0)));
        s0 = s0 + x_hat;
        % Setting a limit for iteration
        if iter>=1000
            disp(['Iteration exceeds for epoch:' num2str(e)]);
        end
        s_final(e,:) = transpose(s0); % Final value of s = [x;y;z] per epoch
        r_measured(e,:) = (norm(s_final(e,:)) * 10^(-3)); % Measured distance in km.
        dist_measured(e,:) = r_measured(e,:) - earth_radius; % Measured distance in radial direction in km.
    end
    % Plotting epoch wise residual norms
            figure(11)
            subplot(6,2,e);
            plot(1:length(res_norm(e,:)),res_norm(e,:),'r*');
            xlabel('Iteration Number');
            ylabel('Residual Norm [m]');
            grid on
            title(strcat('Epoch',num2str(e)));
end

%% Part 1b: Difference between measured solution and precise orbit direction
% Plotting measured and observed radial distances in km
dist_precise = rp - earth_radius;
figure(21)
plot(1:epoch,dist_precise,'r-',1:epoch,dist_measured,'b--');
legend('Precise radial distance','Measured radial distance','Location','Best');
xlabel('Epoch');
ylabel('Altitude [km]');
grid on;
% Plotting the difference between measured and observed orbit
figure(22)
plot(1:epoch,dist_measured - dist_precise');
xlabel('Epoch');
ylabel('Difference between measured and precise radial orbit [km]');
grid on;
```

```
%% Author: Ali Nawaz
% Course: AE4872 Satellite Orbit Determination
% Homework Assignment 2: GPS single point positioning
%%=====================================================
%% Part 1C,D
% Initialize state vector and error limit
s0_rcc = (10^3)*ones(3*epoch +1,1);
diff_rcc = norm(s0_rcc);
iter_rcc = 0;
del_t = 0;
while diff_rcc>=10^-15
    A_rcc =[];
```

```
    y_bar_rcc = [];
    iter_rcc = iter_rcc +1;
    n= 0;
    for e=1:epoch
        clear Ac dx_rcc dy_rcc dz_rcc drho_rcc;
        for s = 1:no_sat(e)
            dx_rcc(s) = s0_rcc(n+1) - xgi(e,s); % x - x_gps for sth satellite in view
            dy_rcc(s) = s0_rcc(n+2) - ygi(e,s); % y - y_gps for sth satellite in view
            dz_rcc(s) = s0_rcc(n+3) - zgi(e,s); % z - z_gps for sth satellite in view

            drho_rcc(s) = sqrt(dx_rcc(s)^2 + dy_rcc(s)^2 + dz_rcc(s)^2); % measured range for nth satellite in view
            Ac(s,:) = [ dx_rcc(s)/drho_rcc(s) dy_rcc(s)/drho_rcc(s) dz_rcc(s)/drho_rcc(s)]; % Assembling epochwise information matrix. Final matrix for each epoch= no. of satellites in view / epoch by 3
        end
        % Accumalating primary information matrix from all the epochs into
        % one matrix
        A_rcc = blkdiag(A_rcc,Ac);
        % Accumulating observation for all epochs
%         y_bar_rcc = [y_bar_rcc;transpose(C1(e,1:no_sat(e)) + ccGPS(e,1:no_sat(e)) - drho_rcc) ] ;
        % Inclusion of light time effect
        y_bar_rcc = [y_bar_rcc;transpose(C1(e,1:no_sat(e)) + (ccGPS(e,1:no_sat(e))- del_t(end)*c) - drho_rcc) ] ;
        n= n+3;
    end
    % Appending clock correction part to the all-epoch information matrix.
    mat_height = size(A_rcc);
    A_rcc = [A_rcc,ones(mat_height(1),1)];

    % rank check to check for information loss in the information matrix
    rank_rcc(iter_rcc) = rank(A_rcc);

    % LSQ with the aid of SVD
    [Urcc,Srcc,Vrcc] = svd(A_rcc,'econ'); % Singular value decomposition of A, to avoid singularity errors or solution manifold due to rank deficit
    S_inv_diag_rcc = [];
    for nn=1:length(Srcc)
        S_inv_diag_rcc = [S_inv_diag_rcc,inv(Srcc(nn,nn))];
    end
    S_inv_rcc = diag(S_inv_diag_rcc);

    x_hat_rcc = Vrcc*S_inv_rcc*Urcc'*y_bar_rcc; % Parameter estimation via SVD aided LSQ
    y_hat_rcc = A_rcc*x_hat_rcc; % Measured vector

    nn = 0;
    for ee = 1:epoch
        res_rcc = y_bar_rcc(nn+1:no_sat(ee)) - y_hat_rcc(nn+1:no_sat(ee)); % residual in meters
        res_norm_rcc(ee,iter_rcc) = norm(res_rcc);% Epoch wise norm of the residual vector of all satellites
        nn = no_sat(ee);
    end

    % Update error and state
    diff_rcc = abs( ( norm(x_hat_rcc + s0_rcc) - norm(s0_rcc) )/ ( norm(s0_rcc + x_hat_rcc)));
    s0_rcc = s0_rcc + x_hat_rcc;
    % Clock correction per iteration
    del_t(iter_rcc) = s0_rcc(end)/c;
    if iter_rcc>=100
        disp('Iteration limit of 100 exceeds');
        break
    end
end
%% Plotting the behaviour of epochwise residual, inclusion of clock correction before batch LSQ for all epochs
for ee = 1:epoch
        figure(31)
        subplot(6,2,ee)
        plot(1:length(res_norm_rcc(ee,:)),res_norm_rcc(ee,:),'rx');
        hold on
        plot(1:length(res_norm_rcc(ee,:)),res_norm_rcc(ee,:),'k:');
        hold off
        title(['Epoch ' num2str(ee)]);
        ylabel('Residual norm [m]');
        xlabel('Iteration no.');
```

```
        grid on
end
%% Plotting the behaviour of clock correction per iteration
figure(33)
plot(1:length(del_t),del_t,'b:x');
ylabel('Receiver clock correction [s]');
xlabel('Iteration no.');
grid on
title('Evolution of receiver clock correction');

%% Evaluation the difference between measured and precise radial orbit
s_final_rcc = [];
n = 0;
for e = 1:epoch
    s_final_rcc(e,:) = s0_rcc(n+1:n+3,:); % final value of s = [x;y;z] per epoch
    n = n+3;
    r_measured_rcc(e,:) = (norm(s_final_rcc(e,:)) * 10^(-3)); % Measured distance in km.
    dist_measured_rcc(e,:) = r_measured_rcc(e,:) - earth_radius; % Measured distance in radial
    direction in km.
end

% Plotting measured and observed radial distances in km
dist_precise_rcc = rp - earth_radius;
figure(34)
plot(1:epoch,dist_precise_rcc,'r-',1:epoch,dist_measured_rcc,'b--');
legend('Precise radial distance','Measured radial distance');
xlabel('Epoch');
ylabel('Altitude [km]');
title('Batch processed difference between measured & precise radial dist.');
grid on;
% Plotting the difference between measured and observed orbit
figure(35)
plot(1:epoch,dist_measured_rcc - dist_precise_rcc');
xlabel('Epoch');
ylabel('Difference between measured and precise radial orbit [km]');
title('Batch processed difference between measured & precise orbit');
grid on;
```