ENG 4350 6.0 FW16/17- Space Hardware

Lab P4 – High Level Function Implementation and Debugging

Submitted by:

Keith Menezes & James Brook

Submitted to:

Prof. Hugh Chesser

Contents

Contents 2
Purpose 3
Fixes – Software/Code 3
Advancements 4
Interface 4
Editing the Station File Interface 4
Editing the TLE File Interface 6
Software/Code 8
STK Testing 9
Future work 10
Conclusion 10
Appendix A – Main Interface – Code 10
Appendix A.1. – Altering the Station File options 18
Appendix A.2. – Altering the TLE File options 20
Appendix A.3. – AOS/LOS Table Code 23

Purpose

The purpose of the document is to outline the progress that we've made since P3 where tests had been developed and we confirmed functions that worked and identified ones that didn't. We will explain the recent advances made with the code including the fixes as well as the revised user interface and what next steps are involved.

Fixes – Software/Code

This section outlines what problems we identified with our code and the changes we made to fix them.

- Mean_anomaly_motion()

The problem occurred when inputting the time as Julian date and not multiplying the time interval by 86400 (the number of seconds in a day).

The second problem we observed was that we forgot that when converting to n_dot_mean_motion_rad_p_s we needed to multiply the input (n_dot_mean_motion) by (2*PI/pow(86400,2)) and subsequently when converting to n_2dots_mean_motion_rad_p_s we needed to multiply the input (n_2dots_mean_motion) by (2*PI/pow(86400,3)).

The last thing we realized was the way we were using the function. The satellite epoch has to be in Julian as we specified inside the Propogate.c file and seen below is the call we make in the main.

```
double mA, mM; double t = curTime; double satEpoch = jdatep(sats[j].refepoch);
double mA0 = sats[j].meanan;
double nMM = sats[j].meanmo;
double ndMM = sats[j].ndot;
double n2dMM = sats[j].nddot6;
mean_anomaly_motion(&mA, &mM, t, satEpoch, mA0, nMM, ndMM, n2dMM);
```

- Sat_ECI()

The realization is that the true anomaly may come out negative with the computations. Therefore, we added an if(true<0){ add 2*PI}.

Another realization was that we need to have the angles input in radians, here is the call/conversion made in the main:

```
double RA=sats[j].raan*(PI/180);
double Arg_Per=sats[j].argper*(PI/180);
double i=sats[j].incl*(PI/180);
sat_ECI(eciP, eciV, sats[j].eccn, eccAnom, sMA, RA, Arg_Per, i, mM);
```

As well as correcting the semi-major-axis input mentioned below.

- Semi-major-axis

This is a simple calculation made within the main for determining the AOS/LOS.

The realization is that the satellite structure holds the TLE data that needs to be converted to revolutions per second (by dividing by 2*PI) before input is made into the semi-major-axis calculation.

```
double meanMorev = sat.meanmo/(2*PI);

double SemiMajorAxis = CUBE_ROOT(398600.4418/(4*PI*PI*meanMorev* meanMorev));
```

Advancements

This section outlines the advancements made with the software and interface, as well as, with the Systems Tool Kit (STK) simulation.

Interface

The user interface and experience has been improved by allowing the user to manipulate the data entries and results within the program. In the most recent version of the code, the user is now able to select an option to edit parameters of individual values for the station data or the imported TLE data. The user can select a parameter for the station file, or select a satellite and then a parameter for the TLE data. This is done by adding an additional UI option. Viewing the changes to the station or satellite data can be viewed in the same manner as in P3.

Here is the flow of the interface (See Appendix A for full main code):

1. Print Banner
2. Import station file
3. User input
    a. View station data
    b. Change station data
        i. Select parameter
            1. Enter new parameter
    c. Continue
4. Import TLE file
5. User input
    a. View satellite data
        i. Select satellite
    b. Change satellite data
        i. Select satellite
            1. Select parameter
                a. Enter new parameter
    c. Continue
6. Import schedule dates and time step
7. Print schedule dates and time step
8. Generate AOS/LOS table

Editing the Station File Interface

```
--------------------
ENG4350 Team Info:
Keith Menezes
James Brook
Team Pointer - Memory Allocation - Masters
2017-03-26
Version 4
Welcome
```

```
 --------------------

Importing station data...

Complete

Enter the number next to the corresponding option:
1   view station file data
2   edit station file data
3   continue

Entry: 1

Station File Contents:
1  Name:  ARO
2  Station Latitude: 45.955503
3  Station Longitude: 281.926960
4  Station Altitude: 260.420000
5  UTC Offset: -4.000000
6  Azimuth Elevation nlim: 1
7  Azimuth Elevation Limit Azimuth: 0.000000
8  Azimuth Elevation Limit Elevation Min: 9.000000
9  Azimuth Elevation Limit Elevation Max: 89.000000
10 Station Azimuth Speed Max: 3.000000
11 Station Elevation Speed Max: 3.000000

Enter the number next to the corresponding option:
1   view station file data
2   edit station file data
3   continue

Entry: 2

1    name
2    stnlat
3    stnlong
4    stnalt
5    utc_offset
6    az_el_nlim
7    az_el_nlim.az
8    az_el_nlim.elmin
9    az_el_nlim.elmax
10   st_az_speed_max
11   st_el_speed_max

What parameter would you like to edit: 2

stnlat entry: 123.456

Enter the number next to the corresponding option:
1   view station file data
2   edit station file data
3   continue
Entry: 1
```

```
Station File Contents:
1  Name:  ARO
2  Station Latitude: 123.456000
3  Station Longitude: 281.926960
4  Station Altitude: 260.420000
5  UTC Offset: -4.000000
6  Azimuth Elevation nlim: 1
7  Azimuth Elevation Limit Azimuth: 0.000000
8  Azimuth Elevation Limit Elevation Min: 9.000000
9  Azimuth Elevation Limit Elevation Max: 89.000000
10 Station Azimuth Speed Max: 3.000000
11 Station Elevation Speed Max: 3.000000

Enter the number next to the corresponding option:
1   view station file data
2   edit station file data
3   continue

Entry:
```

It can be seen the station latitude had been changed to `123.456` and the output on the console was confirmed (See Appendix A.1 for code).

Editing the TLE File Interface

```
Importing TLE file sats...

Complete

Enter the number next to the corresponding option:
1   view TLE data
2   edit TLE data
3   continue

Entry: 1

Enter the satellite number you would like to view: 20

Information for sat number 20

    name is GPS BIIF-2  (PRN 01)
    refepoch is 17077.719618
    incl is 55.395600
    raan is 104.382400
    eccn is 0.006232
    argper is 28.737400
    meanan is 157.288500
    meanmo is 2.005617
    ndot is -0.000000
    nddot6 is 0.000000
    bstar is 0.000000
    orbitnum is 0.000000


Enter the number next to the corresponding option:
1   view TLE data
```

```
2   edit TLE data
3   continue

Entry: 2

Enter the satellite number you would like to edit: 20

1    name
2    refepoch
3    incl
4    raan
5    eccn
6    argper
7    meanan
8    meanmo
9    ndot
10   nddot6
11   bstar
12   orbitnum

What parameter would you like to edit: 2

refepoch entry: 123.456

Enter the number next to the corresponding option:
1   view TLE data
2   edit TLE data
3   continue

Entry: 1

Enter the satellite number you would like to view: 20

Information for sat number 20

    name is GPS BIIF-2  (PRN 01)
    refepoch is 123.456000
    incl is 55.395600
    raan is 104.382400
    eccn is 0.006232
    argper is 28.737400
    meanan is 157.288500
    meanmo is 2.005617
    ndot is -0.000000
    nddot6 is 0.000000
    bstar is 0.000000
    orbitnum is 0.000000


Enter the number next to the corresponding option:
1   view TLE data
2   edit TLE data
3   continue

Entry:
```

The TLE reference epoch has changed from its original value to what the user has set (123.456) and has been confirmed at the output. (See Appendix A.2. for code).

Software/Code

We now have calculations being made for the AOS/LOS table. The user must input a file called trackingschedule.txt or .dat into the file directory with the following format:

```
Tracking start date/time: 2017-03-19-00:00:00
Tracking stop date/time: 2017-03-19-01:00:00
Output time step (sec): 300.00
```

Above we can change the time to when Prof. Chesser allocates us the tracking times at ARO. The output computed is below as well as stored to a file named AOSLOS.txt (See Appendix A.3 for code):

```
Sat No.            Name              AOS                 LOS
5       GPS BIIR-7  (PRN 18)   2017-03-19 00:00:00   2017-03-19 00:59:59
6       GPS BIIR-8  (PRN 16)   2017-03-19 00:00:00   2017-03-19 00:59:59
9       GPS BIIR-11 (PRN 19)   2017-03-19 00:29:59   2017-03-19 00:59:59
10      GPS BIIR-12 (PRN 23)   2017-03-19 00:00:00   2017-03-19 00:59:59
11      GPS BIIR-13 (PRN 02)   2017-03-19 00:29:59   2017-03-19 00:59:59
12      GPS BIIRM-1 (PRN 17)   2017-03-19 00:00:00   2017-03-19 00:59:59
17      GPS BIIRM-6 (PRN 07)   2017-03-19 00:00:00   2017-03-19 00:59:59
25      GPS BIIF-7  (PRN 09)   2017-03-19 00:00:00   2017-03-19 00:59:59
26      GPS BIIF-8  (PRN 03)   2017-03-19 00:00:00   2017-03-19 00:59:59
27      GPS BIIF-9  (PRN 26)   2017-03-19 00:59:59   2017-03-19 00:59:59
29      GPS BIIF-11  (PRN 10)  2017-03-19 00:00:00   2017-03-19 00:59:59
30      GPS BIIF-12  (PRN 32)  2017-03-19 00:00:00   2017-03-19 00:59:59
31      GPS BIIF-12  (PRN 32)  2017-03-19 00:00:00   2017-03-19 00:59:59
```

To start confirming our visibility calculations we developed a test routine for our satellite coordinate transformation functions. The summary below displays the resulting error compared to the STK values.

Table 1: Testing Satellite functions

Parameter [km] or [km/s]	**Systems Tool Kit**	**ARO-Tracking-Software**	**Error**
Time: 19 Mar 2017 01:00:43.9	**ECI**	**ECI**	
X	-15749.3	-15796.9	-0.30%
Y	4039.78	3976.651	1.56%
Z	20444.58	20428.77	0.08%
VX	-0.73611	-0.70835	-3.77%
VY	-3.86446	-3.80366	-1.57%
VZ	0.186824	0.183263	1.91%
Time: 19 Mar 2017 01:00:43.9	**ECF**	**ECF**	
X	14631.27	-8350.55	157.07%
Y	-7166.67	13950.97	-294.66%
Z	20418.3	20444.58	0.13%
VX	0.984513	-2.22455	325.95%
VY	2.566927	-1.61956	163.09%
VZ	0.18576	0.186824	0.57%

It is clear that the changes made to sat_ECI() have converged to the STK solution. Next the ECF will be fixed so that those results converge to the STK solution as well.

STK Testing

Advancements with the STK simulation include adding a receiver to the Ground Station Facility file as well as a GPS specific transmitter on the target spacecraft. This enables us to calculate the link budget information and confirm our computations. Below is the demonstration with the key data sections we are interested in highlighted namely, Receiver Gain and Received Isotropic Power.

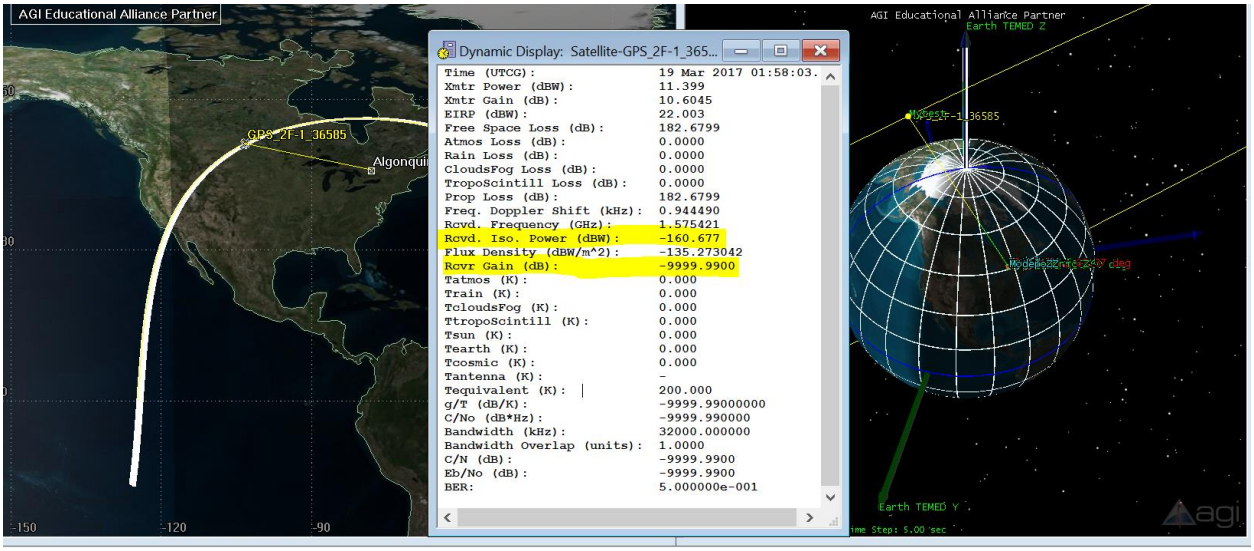

Figure 1: Visualization of STK simulation link budget calculations

The next thing we learned is to create custom angles for debugging the true anomaly and rate of change of that angle and which helped us confirm our sat_ECI() and mean_anomaly_motion() calculations. Below is the confirmation:

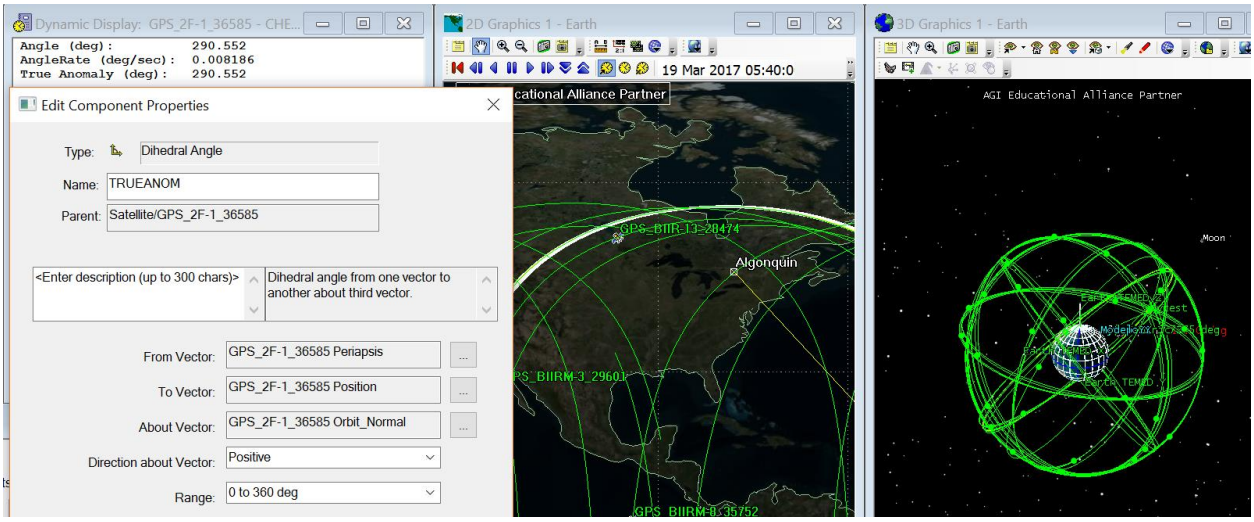

Figure 2: Making a new angle called TRUEANOM, and the strip chart above demonstrating it represents the classical definition of true anomaly therefore the rate of change can be determined.

Future work

For the next deliverable, we would like to:

A. Continue to test our sat_ecf() and fix it.
B. Fully test our range (range_ECF2topo and range_topo2look_angles) functions over a variety of cases, if problems arise we will need to troubleshoot them.
C. Test and include the Doppler equation into the AOS/LOS table
D. Tracking Data table
E. Antenna Pointing File for STK debugging

Conclusion

Significant progress is being made every Monday and Wednesday where we work during the lab period to tackle problems and debug our software. Before we progress further to the tracking data table we would like to ensure all our functions work nominally, through continual testing and checking with STK.

Appendix A – Main Interface – Code

```
#include <stdio.h>
#include <math.h>
#include "Propagate.h"
#include "Basic.h"
#include "FileIO.h"
#include "Vector.h"
#include "STKout.h"
#include "DateAndTimeCalculations.h"
#include "Matrix.h"
#include "Vector.h"
#include <time.h>
#include <stdlib.h>
#include <string.h>
#define CUBE_ROOT(X) (exp(log(X)/ 3.))
#define PI
3.14159265358979323846264338327950288419716939937510582097494459230781640628
6

int main(void){

	Banner();
	printf("\nImporting station data...\n\n");
	Station *stn = (Station*) malloc(sizeof(Station));
	ReadStationFile(stn, '0');
	printf("Complete\n\n");
	int sf;
	for (sf = 0; sf < 1;){
		printf("\nEnter the number next to the corresponding option:\n");
		printf("1   view station file data\n");
		printf("2   edit station file data\n");
		printf("3   continue\n\n");
		int input1;
		printf("Entry: ");
		fflush(stdout);
		scanf("%d", &input1);
```

```
            if (input1 == 1){
                  printf("\nStation File Contents:\n");
                  printf("1  Name: %s\n", stn->name);
                  printf("2  Station Latitude: %f\n", stn->stnlat);
                  printf("3  Station Longitude: %f\n", stn->stnlong);
                  printf("4  Station Altitude: %f\n", stn->stnalt);
                  printf("5  UTC Offset: %f\n", stn->utc_offset);
                  printf("6  Azimuth Elevation nlim: %d\n", stn->az_el_nlim);
                  printf("7  Azimuth Elevation Limit Azimuth: %f\n", stn-
>az_el_lim.az);
                  printf("8  Azimuth Elevation Limit Elevation Min: %f\n", stn-
>az_el_lim.elmin);
                  printf("9  Azimuth Elevation Limit Elevation Max: %f\n", stn-
>az_el_lim.elmax);
                  printf("10 Station Azimuth Speed Max: %f\n", stn-
>st_az_speed_max);
                  printf("11 Station Elevation Speed Max: %f\n", stn-
>st_el_speed_max);
            }
            if(input1 == 2){

                  printf("\n1    name\n");
                  printf("2    stnlat\n");
                  printf("3    stnlong\n");
                  printf("4    stnalt\n");
                  printf("5    utc_offset\n");
                  printf("6    az_el_nlim\n");
                  printf("7    az_el_nlim.az\n");
                  printf("8    az_el_nlim.elmin\n");
                  printf("9    az_el_nlim.elmax\n");
                  printf("10   st_az_speed_max\n");
                  printf("11   st_el_speed_max\n");

                  printf("\nWhat parameter would you like to edit: ");
                  int num2;
                  fflush(stdout);
                  scanf("%d", &num2);
                  printf("\n");
                  if(num2 == 1){
                        printf("name entry: ");
                        char *n;
                        fflush(stdout);
                        scanf("%c", n);
                        //stn->name = n;
                  }
                  if(num2 == 2){
                        printf("stnlat entry: ");
                        double n;
                        fflush(stdout);
                        scanf("%lf", &n);
                        stn->stnlat = n;
                  }
                  if(num2 == 3){
                        printf("stnlon entry: ");
                        double n;
```

```
        fflush(stdout);
        scanf("%lf", &n);
        stn->stnlong = n;
}
if(num2 == 4){
        printf("stnalt: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->stnalt = n;
}
if(num2 == 5){
        printf("utc_offset entry: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->utc_offset = n;
}
if(num2 == 6){
        printf("az_el_nlim entry: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->az_el_nlim = n;
}

if(num2 == 7){
        printf("az_el_lim.az entry: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->az_el_lim.az = n;
}
if(num2 == 8){
        printf("az_el_lim.elmin entry: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->az_el_lim.elmin = n;
}

if(num2 == 9){
        printf("az_el_lim.elmax entry: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->az_el_lim.elmax= n;
}

if(num2 == 10){
        printf("st_az_speed_max entry: ");
        double n;
        fflush(stdout);
        scanf("%lf", &n);
        stn->st_az_speed_max = n;
```

```
                }

                if(num2 == 11){
                        printf("st_el_speed_max entry: ");
                        double n;
                        fflush(stdout);
                        scanf("%lf", &n);
                        stn->st_el_speed_max = n;
                }


        }
        if (input1 == 3){sf++;}

}

printf("\nImporting TLE file sats...\n\n");
char *file = "TLE.txt";
Satellite sats[32];
ReadNoradTLE(sats, file);
printf("Complete\n");
int x;
for(x = 0;x < 1;){
        printf("\n\nEnter the number next to the corresponding option:\n");
        printf("1   view TLE data\n");
        printf("2   edit TLE data\n");
        printf("3   continue\n\n");
        int input2;
        printf("Entry: ");
        fflush(stdout);
        scanf("%d", &input2);

        if(input2 == 1){
                printf("\nEnter the satellite number you would like to view: ");
                int num;
                fflush(stdout);
                scanf("%d", &num);

                printf("\nInformation for sat number %d\n", num);
                printf("\n    name is %s", sats[num].name);
                printf("    refepoch is %f\n", sats[num].refepoch);
                printf("    incl is %f\n", sats[num].incl);
                printf("    raan is %f\n", sats[num].raan);
                printf("    eccn is %f\n", sats[num].eccn);
                printf("    argper is %f\n", sats[num].argper);
                printf("    meanan is %f\n", sats[num].meanan);
                printf("    meanmo is %f\n", sats[num].meanmo);
                printf("    ndot is %f\n", sats[num].ndot);
                printf("    nddot6 is %f\n", sats[num].nddot6);
                printf("    bstar is %f\n", sats[num].bstar);
                printf("    orbitnum is %f\n", sats[num].orbitnum);
        }
        if(input2 == 2){
                printf("\nEnter the satellite number you would like to edit: ");
                int num;
```

```
fflush(stdout);
scanf("%d", &num);

printf("\n1    name\n");
printf("2    refepoch\n");
printf("3    incl\n");
printf("4    raan\n");
printf("5    eccn\n");
printf("6    argper\n");
printf("7    meanan\n");
printf("8    meanmo\n");
printf("9    ndot\n");
printf("10   nddot6\n");
printf("11   bstar\n");
printf("12   orbitnum\n");

printf("\nWhat parameter would you like to edit: ");
int num2;
fflush(stdout);
scanf("%d", &num2);
printf("\n");
if(num2 == 1){
	printf("name entry: ");
	char *n;
	fflush(stdout);
	scanf("%c", n);
	//sats[num].name = n;
}
if(num2 == 2){
	printf("refepoch entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].refepoch = n;
}
if(num2 == 3){
	printf("incl entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].incl = n;
}
if(num2 == 4){
	printf("raan entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].raan = n;
}
if(num2 == 5){
	printf("eccn entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].eccn = n;
```

```
}
if(num2 == 6){
	printf("argper entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].argper = n;
}

if(num2 == 7){
	printf("meanan entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].meanan = n;
}

if(num2 == 8){
	printf("meanmo entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].meanmo = n;
}
if(num2 == 9){
	printf("ndot entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].ndot = n;
}

if(num2 == 10){
	printf("nddot6 entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].nddot6 = n;
}

if(num2 == 11){
	printf("bstar entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].bstar = n;
}

if(num2 == 12){
	printf("orbitnum entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].orbitnum = n;
}
```

```
		}
		if(input2 == 3){x++;}
	}

	//------------------------------AOS/LOS-------------------------------------
------


	printf("\nOpening tracking file...\n\n");
	FILE *fp = fopen("tracking_sched.txt","r+");

	char line1[50]; //The first line in tracking_sched.txt -- Specifies the start
date/time
	char line2[50]; //The second line in tracking_sched.txt -- Specifies the stop
date/time
	char line3[31]; //The third line in tracking_sched.txt -- Specifies the time
step
	fgets(line1, 50, fp);
	fgets(line2, 50, fp);
	fgets(line3, 31, fp);

	fclose(fp);

	//Pull out the dates from each line
	char date_start[20], date_stop[20], time_step[6];

	strncpy(date_start, line1+26, 19);
	strncpy(date_stop, line2+25, 19);
	strncpy(time_step, line3+24, 5);
	printf("Printing out the start date %s\n",date_start);
	printf("Printing out the stop date %s\n",date_stop);
	printf("Printing out the time step %s\n",time_step);

	printf("\nCalculating AOS and LOS...\n");
	//Convert each time_step to a double
	double step;
	step = atof(time_step);

	double JulianDateStart, JulianDateStop;

	JulianDateStart = dat2jd(date_start);
	JulianDateStop = dat2jd(date_stop);


	char *NAME[31];
	double AOS[31], LOS[31];
	int NUM[31];
	int num = 0;


	for(int j=1; j<32; j++){//Run through each satellite

		double currentTime;
		currentTime = JulianDateStart;
```

```
            int acquired=0; //acquired=0 if the AOS has not been obtained and =1 if
it has been obtained
            int lost = 0; //lost = 0 if the sat has not been lost yet and =1 if the
sat has been lost and is out of view

            for( ;currentTime<JulianDateStop; currentTime =
currentTime+frcofd(0,0,step)){//Run through each time step until we reach the end of
the interval
                  double mA, mM;
                  double satEpoch = sats[j].refepoch;
                  double mA0 = sats[j].meanan;
                  double nMM = sats[j].meanmo;
                  double ndMM = sats[j].ndot;
                  double n2dMM = sats[j].nddot6;
                  mean_anomaly_motion(&mA, &mM, currentTime, satEpoch, mA0, nMM,
ndMM, n2dMM);
                  double mMrev=mM/(2*PI);

                  double eccAnom = KeplerEqn(mA, sats[j].eccn);

                  Vector *eciPos, *eciVel;
                  eciPos = (Vector*)malloc(sizeof(Vector));
                  eciVel = (Vector*)malloc(sizeof(Vector));

                  double sMA = CUBE_ROOT(398600.4418/(4*PI*PI*mMrev*mMrev));
                  sat_ECI(eciPos, eciVel, sats[j].eccn, eccAnom, sMA,
sats[j].raan*PI/180,sats[j].argper*PI/180, sats[j].incl*PI/180, mM);
                  Vector *ecfPos, *ecfVel;
                  ecfPos = (Vector*)malloc(sizeof(Vector));
                  ecfVel = (Vector*)malloc(sizeof(Vector));

                  double thetat = THETAN(sats[j].refepoch);
                  sat_ECF(ecfPos, ecfVel, thetat, eciPos, eciVel);
                  Vector *stnPos, *rtPos, *rtVel;
                  stnPos = (Vector*)malloc(sizeof(Vector));
                  station_ECF(stnPos, stn->stnlong, stn->stnlat, stn->stnalt);
                  rtPos = (Vector*)malloc(sizeof(Vector));
                  rtVel = (Vector*)malloc(sizeof(Vector));
                  range_ECF2topo(rtPos, rtVel, *stnPos, ecfPos, ecfVel, stn-
>stnlong, stn->stnlat);

                  double az;
                  double el;
                  double azV;
                  double elV;
                  LookAngles *LA =(LookAngles*) malloc(sizeof(LookAngles));
                  range_topo2look_angles(LA, az, el, azV, elV, rtPos, rtVel);
                  double ss[31];
                  //ss[num] = linkstrength(rtPos->mag);
                  if (LA->elevation <= stn->az_el_lim.elmax && LA->elevation >=
stn->az_el_lim.elmin && acquired == 0){//Go in to this loop if the satellite is
acquired.
                        //If the satellite has already been acquired then don't
add it again
                        NUM[num] = j;
```

```
                        NAME[num] = sats[j].name;
                        NAME[num][strlen(NAME[num])-1] = '\0'; //Just some
formatting of the Name of the sat for printing.
                        NAME[num][strlen(NAME[num])-2] = '\0';//Without these two
lines there are two \n operators in NAME[num]
                        AOS[num] = currentTime;
                        acquired = 1;
                    }
                    if(LA->elevation >= stn->az_el_lim.elmax && LA->elevation <= stn-
>az_el_lim.elmin && acquired==1){// Go in to this loop if the satellite is lost.
                        // You can only lose the satellite after it has been
acquired
                        // hence acquired==1
                        LOS[num] = currentTime;
                        lost = 1; //the sat is now out of view
                        break;}}if(acquired==1){if(lost==0){//if the sat had been
acquired but not lost then there is no LOS time.
                                LOS[num] = JulianDateStop;}num++;}}
      printf("\nComplete\n\n");
      //Print the AOS/LOS table to the console and write it to a file
      FILE *filepoint;
      filepoint = fopen("AOSLOS.txt", "w+");
      printf("Sat No.          Name              AOS                LOS\n");
      fprintf(filepoint, "Sat No.                Name
      AOS                          LOS                          Min.
ExpectedLevel (dBm)\n");
      for(int i=0;i<num;i++){
            printf("%d    %s %s       ", NUM[i], NAME[i], jd2dat(AOS[i]));
            printf("%s\n", jd2dat(LOS[i]));
            fprintf(filepoint, "%d          %s     %s     ", NUM[i], NAME[i],
jd2dat(AOS[i]));
            fprintf(filepoint, "%s", jd2dat(LOS[i]));
      //    fprintf(filepoint,"%f\n",linkstrength(rtPos->mag));
      }
      fclose(filepoint);
      return 0;
}
```

Appendix A.1. – Altering the Station File options

```
if(input1 == 2){

                                                    printf("\n1    name\n");
                                                    printf("2    stnlat\n");
                                                    printf("3    stnlong\n");
                                                    printf("4    stnalt\n");
                                                    printf("5
utc_offset\n");
                                                    printf("6
az_el_nlim\n");
                                                    printf("7
az_el_nlim.az\n");
                                                    printf("8
az_el_nlim.elmin\n");
```

```
                                                printf("9
az_el_nlim.elmax\n");
                                                printf("10
st_az_speed_max\n");
                                                printf("11
st_el_speed_max\n");

                                                printf("\nWhat parameter
would you like to edit: ");
                        int num2;
                        fflush(stdout);
                        scanf("%d", &num2);
                        printf("\n");
                        if(num2 == 1){
                                printf("name entry: ");
                                char *n;
                                fflush(stdout);
                                scanf("%c", n);
                                //stn->name = n;
                        }
                        if(num2 == 2){
                                printf("stnlat entry: ");
                                double n;
                                fflush(stdout);
                                scanf("%lf", &n);
                                stn->stnlat = n;
                        }
                        if(num2 == 3){
                                printf("stnlon entry: ");
                                double n;
                                fflush(stdout);
                                scanf("%lf", &n);
                                stn->stnlong = n;
                        }
                        if(num2 == 4){
                                printf("stnalt: ");
                                double n;
                                fflush(stdout);
                                scanf("%lf", &n);
                                stn->stnalt = n;
                        }
                        if(num2 == 5){
                                printf("utc_offset entry: ");
                                double n;
                                fflush(stdout);
                                scanf("%lf", &n);
                                stn->utc_offset = n;
                        }
                        if(num2 == 6){
                                printf("az_el_nlim entry: ");
                                double n;
                                fflush(stdout);
                                scanf("%lf", &n);
                                stn->az_el_nlim = n;
                        }
```

```
if(num2 == 7){
	printf("az_el_lim.az entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	stn->az_el_lim.az = n;
}
if(num2 == 8){
	printf("az_el_lim.elmin entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	stn->az_el_lim.elmin = n;
}

if(num2 == 9){
	printf("az_el_lim.elmax entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	stn->az_el_lim.elmax= n;
}

if(num2 == 10){
	printf("st_az_speed_max entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	stn->st_az_speed_max = n;
}

if(num2 == 11){
	printf("st_el_speed_max entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	stn->st_el_speed_max = n;
}
```

Appendix A.2. – Altering the TLE File options

```
if(input2 == 2){
	printf("\nEnter the satellite number you would like
to edit: ");
	int num;
	fflush(stdout);
	scanf("%d", &num);

printf("\n1
name\n");
printf("2
refepoch\n");
```

```
                                                                printf("3
incl\n");
                                                                printf("4
raan\n");
                                                                printf("5
eccn\n");
                                                                printf("6
argper\n");
                                                                printf("7
meanan\n");
                                                                printf("8
meanmo\n");
                                                                printf("9
ndot\n");
                                                                printf("10
nddot6\n");
                                                                printf("11
bstar\n");
                                                                printf("12
orbitnum\n");

                                                                printf("\nWhat
parameter would you like to edit: ");
                                int num2;
                                fflush(stdout);
                                scanf("%d", &num2);
                                printf("\n");
                                if(num2 == 1){
                                        printf("name entry: ");
                                        char *n;
                                        fflush(stdout);
                                        scanf("%c", n);
                                        //sats[num].name = n;
                                }
                                if(num2 == 2){
                                        printf("refepoch entry: ");
                                        double n;
                                        fflush(stdout);
                                        scanf("%lf", &n);
                                        sats[num].refepoch = n;
                                }
                                if(num2 == 3){
                                        printf("incl entry: ");
                                        double n;
                                        fflush(stdout);
                                        scanf("%lf", &n);
                                        sats[num].incl = n;
                                }
                                if(num2 == 4){
                                        printf("raan entry: ");
                                        double n;
                                        fflush(stdout);
                                        scanf("%lf", &n);
                                        sats[num].raan = n;
                                }
```

```
if(num2 == 5){
	printf("eccn entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].eccn = n;
}
if(num2 == 6){
	printf("argper entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].argper = n;
}

if(num2 == 7){
	printf("meanan entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].meanan = n;
}

if(num2 == 8){
	printf("meanmo entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].meanmo = n;
}
if(num2 == 9){
	printf("ndot entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].ndot = n;
}

if(num2 == 10){
	printf("nddot6 entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].nddot6 = n;
}

if(num2 == 11){
	printf("bstar entry: ");
	double n;
	fflush(stdout);
	scanf("%lf", &n);
	sats[num].bstar = n;
}

if(num2 == 12){
```

```
            printf("orbitnum entry: ");
            double n;
            fflush(stdout);
            scanf("%lf", &n);
            sats[num].orbitnum = n;
        }
```

Appendix A.3. – AOS/LOS Table Code

```
    printf("\nOpening tracking file...\n\n");
    FILE *fp = fopen("tracking_sched.txt","r+");

    char line1[50]; //The first line in tracking_sched.txt -- Specifies the start
date/time
    char line2[50]; //The second line in tracking_sched.txt -- Specifies the stop
date/time
    char line3[31]; //The third line in tracking_sched.txt -- Specifies the time
step
    fgets(line1, 50, fp);
    fgets(line2, 50, fp);
    fgets(line3, 31, fp);

    fclose(fp);

    //Pull out the dates from each line
    char date_start[20], date_stop[20], time_step[6];

    strncpy(date_start, line1+26, 19);
    strncpy(date_stop, line2+25, 19);
    strncpy(time_step, line3+24, 5);
    printf("Printing out the start date %s\n",date_start);
    printf("Printing out the stop date %s\n",date_stop);
    printf("Printing out the time step %s\n",time_step);

    printf("\nCalculating AOS and LOS...\n");
    //Convert each time_step to a double
    double step;
    step = atof(time_step);

    double JulianDateStart, JulianDateStop;

    JulianDateStart = dat2jd(date_start);
    JulianDateStop = dat2jd(date_stop);

    char *NAME[31];
    double AOS[31], LOS[31];
    int NUM[31];
    int num = 0;


    for(int j=1; j<32; j++){//Run through each satellite

        double currentTime;
        currentTime = JulianDateStart;
```

```
            int acquired=0; //acquired=0 if the AOS has not been obtained and =1 if
it has been obtained
            int lost = 0; //lost = 0 if the sat has not been lost yet and =1 if the
sat has been lost and is out of view

            for( ;currentTime<JulianDateStop; currentTime =
currentTime+frcofd(0,0,step)){//Run through each time step until we reach the end of
the interval
                  double mA, mM;
                  double satEpoch = sats[j].refepoch;
                  double mA0 = sats[j].meanan;
                  double nMM = sats[j].meanmo;
                  double ndMM = sats[j].ndot;
                  double n2dMM = sats[j].nddot6;
                  mean_anomaly_motion(&mA, &mM, currentTime, satEpoch, mA0, nMM,
ndMM, n2dMM);
                  double mMrev=mM/(2*PI);

                  double eccAnom = KeplerEqn(mA, sats[j].eccn);

                  Vector *eciPos, *eciVel;
                  eciPos = (Vector*)malloc(sizeof(Vector));
                  eciVel = (Vector*)malloc(sizeof(Vector));

                  double sMA = CUBE_ROOT(398600.4418/(4*PI*PI*mMrev*mMrev));
                  sat_ECI(eciPos, eciVel, sats[j].eccn, eccAnom, sMA,
sats[j].raan*PI/180,sats[j].argper*PI/180, sats[j].incl*PI/180, mM);
                  Vector *ecfPos, *ecfVel;
                  ecfPos = (Vector*)malloc(sizeof(Vector));
                  ecfVel = (Vector*)malloc(sizeof(Vector));

                  double thetat = THETAN(sats[j].refepoch);
                  sat_ECF(ecfPos, ecfVel, thetat, eciPos, eciVel);
                  Vector *stnPos, *rtPos, *rtVel;
                  stnPos = (Vector*)malloc(sizeof(Vector));
                  station_ECF(stnPos, stn->stnlong, stn->stnlat, stn->stnalt);
                  rtPos = (Vector*)malloc(sizeof(Vector));
                  rtVel = (Vector*)malloc(sizeof(Vector));
                  range_ECF2topo(rtPos, rtVel, *stnPos, ecfPos, ecfVel, stn-
>stnlong, stn->stnlat);

                  double az;
                  double el;
                  double azV;
                  double elV;
                  LookAngles *LA =(LookAngles*) malloc(sizeof(LookAngles));
                  range_topo2look_angles(LA, az, el, azV, elV, rtPos, rtVel);
                  double ss[31];
                  //ss[num] = linkstrength(rtPos->mag);
                  if (LA->elevation <= stn->az_el_lim.elmax && LA->elevation >=
stn->az_el_lim.elmin && acquired == 0){//Go in to this loop if the satellite is
acquired.
                        //If the satellite has already been acquired then don't
add it again
                        NUM[num] = j;
```

```
                        NAME[num] = sats[j].name;
                        NAME[num][strlen(NAME[num])-1] = '\0'; //Just some
formatting of the Name of the sat for printing.
                        NAME[num][strlen(NAME[num])-2] = '\0';//Without these two
lines there are two \n operators in NAME[num]
                        AOS[num] = currentTime;
                        acquired = 1;
                    }
                    if(LA->elevation >= stn->az_el_lim.elmax && LA->elevation <= stn-
>az_el_lim.elmin && acquired==1){// Go in to this loop if the satellite is lost.
                        // You can only lose the satellite after it has been
acquired
                        // hence acquired==1
                        LOS[num] = currentTime;
                        lost = 1; //the sat is now out of view
                        break;}}if(acquired==1){if(lost==0){//if the sat had been
acquired but not lost then there is no LOS time.
                                LOS[num] = JulianDateStop;}num++;}}
      printf("\nComplete\n\n");
      //Print the AOS/LOS table to the console and write it to a file
      FILE *filepoint;
      filepoint = fopen("AOSLOS.txt", "w+");
      printf("Sat No.          Name              AOS                LOS\n");
      fprintf(filepoint, "Sat No.                Name
      AOS                           LOS                           Min.
ExpectedLevel (dBm)\n");
      for(int i=0;i<num;i++){
            printf("%d    %s %s       ", NUM[i], NAME[i], jd2dat(AOS[i]));
            printf("%s\n", jd2dat(LOS[i]));
            fprintf(filepoint, "%d          %s     %s     ", NUM[i], NAME[i],
jd2dat(AOS[i]));
            fprintf(filepoint, "%s", jd2dat(LOS[i]));
      //    fprintf(filepoint,"%f\n",linkstrength(rtPos->mag));
      }
      fclose(filepoint);
```