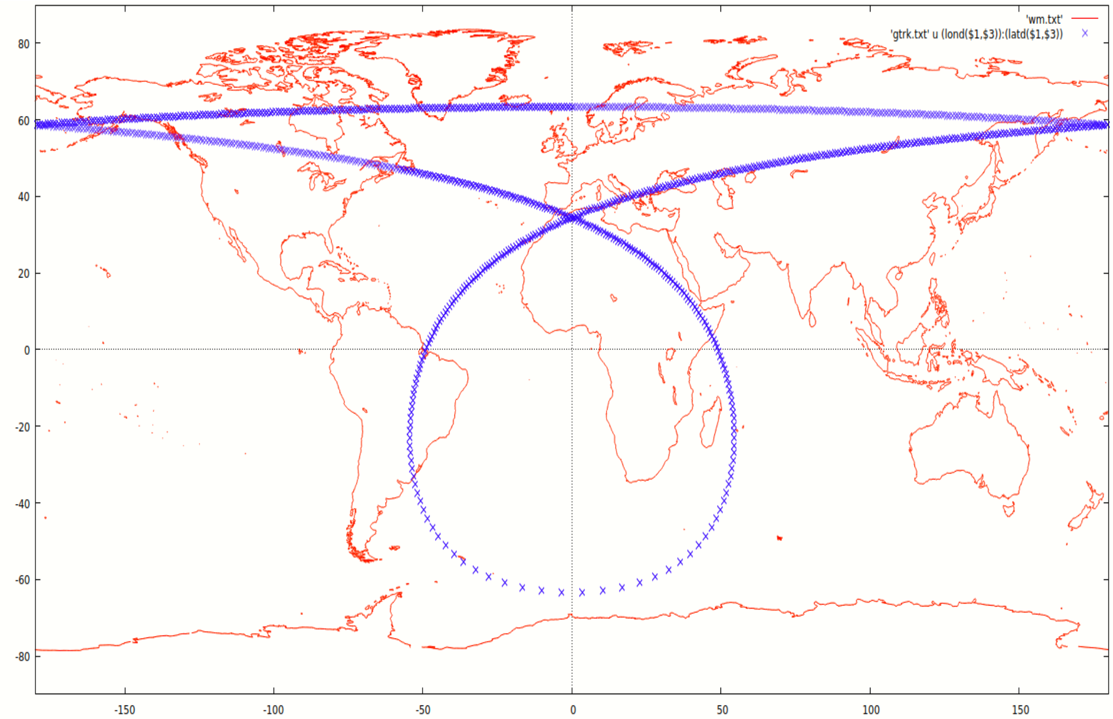
Manel Carrillo Maíllo
1633426
1r de Matcad UAB, Càlcul numèric

REPRESENTACIÓ DE TRACES (GROUND TRACKS) D'ÒRBITES DE SATÈL·LITS ARTIFICIALS



ÍNDEX

1. INTRODUCCIÓ.

Aquesta pràctica tracta sobre la realització d'una eina de software que a partir de la resolució numèrica de l'equació de Kepler, permeti determinar la posició d'un satèl·lit artificial en el seu pla orbital.

Aquesta eina està construïda per dues parts: La primera, que consisteix en la programació d'una rutina per buscar zeros de funcions mitjançant l'ús conjunt del mètode de bisecció i de newton, i la segona, que és una utilitat que calcula les anomalies mitjanes i vertaderes per intervals de temps, utilitzant la rutina anterior.

2. RUTINA BISNWT.

2.1 Objectiu:

L'objectiu d'aquesta rutina és realitzar una rutina per buscar zeros de funcions mitjançant el mètode de bisecció fins trobar un interval petit i fer Newton amb el punt mig de l'últim interval de bisecció per trobar l'arrel de la funció.

L'objectiu principal de fer bisecció abans de Newton, és trobar un punt proper a l'arrel per poder fer newton de manera molt més ràpida i eficient.

Això es deu a que quan som a prop de l'arrel i utilitzem el mètode de Newton, a cada iterat, es duplica nombre de decimals correctes de l'arrel buscada.

Contràriament, quan hi som lluny de l'arrel, el mètode de Newton pot convergir molt lentament (o no convergir) **.**

2.2 Implementació De Software:

La idea en aquesta rutina consisteix en realitzar el mètode de bisecció a partir de dos punts (a i b) mentre l'amplada d'aquest interval (que conté l'arrel) sigui > δ (la qual està prefixada).

Aquest interval s'anirà actualitzant (el valor de a i b) amb el del seu punt mig, de manera que cada cop anirem reduint la distància entre els punts a cada iterat.

En aquesta part, cal comprovar un cas molt poc freqüent però a tindre en compte i és quan $\delta \leq$ tol, que significaria hem trobat l'arrel sense haver fet cap iteració en el mètode de bisecció o Newton.

Un cop la distància entre a i b és més petita que δ, realitzem el mètode de Newton tenint en compte que x0 = c, i iterarem fins que $|xn - xn{-}1| <$ tol o superem els màxim iterats corresponents.

Si Newton ha convergit, tornem l'arrel a un apuntador *arr i via return hi tornem el nombre d'iterats que hem fet.

Si Newton no convergeix, vol dir que el punt c encara està lluny de l'arrel de manera que Newto-n ha passat dels màxims iterats que li corresponen, de manera que dividim δ entre dos i tornem a fer bisecció amb els valors de a i b de l'ultima iteració de bisecció amb l'objectiu de trobar una c propera a l'arrel per tal que el mètode de Newton pugui convergir.

Per a fer aquesta rutina he realitzat un while que engloba tota la funció i que para quan indiquem que hem trobat l'arrel. D'aquesta manera podem realitzar de nou bisecció en cas que Newton no convergeixi.

Per altra banda, per a la implementació del mètode de bisecció he realitzat un while que acaba quan la distancia entre a i b és més petita que el valor de delta indicat.

Per a poder determinar l'interval a cada iterat, s'ha avaluat la multiplicació de f(c)*f(b). De manera que si el resultat és negatiu, vol dir que el valor de la funció en el punt de c és negatiu (com el de a) però més a prop de l'arrel, de manera que el nou valor de a serà c.

En cas contrari, si la funció en el punt c té una imatge positiva més propera a l'arrel, el valor de c es guarda a b.

En aquesta part de bisecció, tal com s'ha mencionat abans, s'ha contemplat el cas en que $\delta \leq$ tol, de manera que si es compleix aquesta condició, guardem c dins *arr i tornem -1 via return, ja que significaria que hem trobat l'arrel sense haver fet cap iteració en el mètode de bisecció.

Un cop es trenca el while de la bisecció entrem en el while del mètode de Newton que iterarà metre la variable que controla les iteracions sigui més petita que el màxim d'iterats i mentre que la distancia en valor absolut de xn (x1) i xn+1(c) sigui ,més gran o igual a al tolerància.

En el mètode de Newton s'ha contemplat el cas en que s'entri una c que ja sigui 0 de manera que voldria dir que ja hem trobat l'arrel. També el cas en que la derivada en el punt c doni 0, en aquest cas parem i fem un return -1 ja que estaríem dividint entre 0.

Si superem el màxim d'iteracions a fer i Newton no ha convergit, es divideix entre dos la delta i tornem a fer bisecció amb els valors de a i b guardats en la ultima iteració del primer mètode per després fer de nou Newton i veure si amb la nova c i la δ reajustada, Newton aconsegueix convergir .

2.3 Manual Del Software Encarregat

Els arguments entrats són *:* ***'double a'***: extrem esquerra de l'interval on es troba l'arrel o solució, *'**double b'*** :extrem dret de l'interval on es troba la solució, ***'double *arr':*** apuntador a una variable externa on guardarem l'arrel o solució en cas de trobar-la, ***'double *dlt'***: delta que anirem actualitzant en cas que el mètode de newton no convergeixi*, '**double tol'***: tolerància que implementarem al mètode de newton per saber si ha convergit o no, ***'int maxit'***: iteracions màximes que podrem realitzar al fer el mètode de newton, ***'double (*f)(double, void)'*** :serveix per calcular el resultat de la funció en un punt, ***'double (*df)(double,void),'void *prm '***: serveix per calcular el resultat de la derivada d'una funció en un punt.

Per poder validar la rutina biswnt s'ha comprovat amb un programa en c anomenat prova.c on cridem la funció bisnwt amb la a, b i delta introduïda per nosaltres. Avaluant la funció f(x) = e^x – 2.

a=-9, b=1, delta=10:

Amb aquesta delta tan gran, no entra a bisecció sinó que va directament a fer newton amb un valor molt llunyà a l'arrel de manera que Newton no convergeix i delta s'ha d'anar reajustant cada cop més i més fins que finalment pot convergir.

```
ITERACIÓ: 8
VALOR DE C: 0.7571955585655736

ITERACIÓ: 9
VALOR DE C: 0.695155180382807

ITERACIÓ: 10
VALOR DE C: 0.6931491952428697

BISECCIÓ:
 -----------------------------
a -1.500000 b 1.000000 c: -0.250000

NEWTON
-----------------------------
ITERACIÓ: 1
VALOR DE C: 1.318050833375483

ITERACIÓ: 2
VALOR DE C: 0.8533638353104991

ITERACIÓ: 3
VALOR DE C: 0.7053230232747401

ITERACIÓ: 4
VALOR DE C: 0.6932210061992887

ITERACIÓ: 5
VALOR DE C: 0.6931471832849908

ITERACIÓ: 6
VALOR DE C: 0.6931471805599453

ITERACIÓ: 7
VALOR DE C: 0.6931471805599453
```

a=-9, b=1, delta=2.5:

Amb delta 2.5, surt per pantalla dos cops el mètode de Newton i bisecció, de manera que deduïm que el punt c agafat en el mètode de bisecció no era prou bo per fer newton i no ha convergit de manera que la delta s'ha actualitzat i newton ha convergit amb 6 iteracions.

```
ITERACIÓ: 7
VALOR DE C: 1.073738216825968

ITERACIÓ: 8
VALOR DE C: 0.7571955585655736

ITERACIÓ: 9
VALOR DE C: 0.695155180382807

ITERACIÓ: 10
VALOR DE C: 0.6931491952428697

BISECCIÓ:
 ----------------------------
a -1.500000 b 1.000000 c: -0.250000

NEWTON
----------------------------
ITERACIÓ: 1
VALOR DE C: 1.318050833375483

ITERACIÓ: 2
VALOR DE C: 0.8533638353104991

ITERACIÓ: 3
VALOR DE C: 0.7053230232747401

ITERACIÓ: 4
VALOR DE C: 0.6932210061992887

ITERACIÓ: 5
VALOR DE C: 0.6931471832849908

ITERACIÓ: 6
VALOR DE C: 0.6931471805599453

ITERACIÓ: 7
VALOR DE C: 0.6931471805599453
```

a=-9, b=1, delta=0.01:

Surt per pantalla molts iterats de bisecció ja que la delta és molt petita de manera que com la c es troba molt a prop de l'arrel només fa 3 iterats de Newton.

```
BISECCIÓ:
 -----------------------------
a -9.000000 b 1.000000 c: -4.000000
a -4.000000 b 1.000000 c: -1.500000
a -1.500000 b 1.000000 c: -0.250000
a -0.250000 b 1.000000 c: 0.375000
a 0.375000 b 1.000000 c: 0.687500
a 0.687500 b 1.000000 c: 0.843750
a 0.687500 b 0.843750 c: 0.765625
a 0.687500 b 0.765625 c: 0.726562
a 0.687500 b 0.726562 c: 0.707031
a 0.687500 b 0.707031 c: 0.697266

NEWTON
-----------------------------
ITERACIÓ: 1
VALOR DE C: 0.6931556497216683

ITERACIÓ: 2
VALOR DE C: 0.6931471805958085

ITERACIÓ: 3
VALOR DE C: 0.6931471805599454

ITERACIÓ: 4
VALOR DE C: 0.6931471805599454
```

3 UTILITAT kplt2nu

3.1 Objectiu:

Aquest programa principal es pot definir com un solucionador de l'equació de Kepler.

El que ha de fer és imprimir per pantalla el valor corresponent a l'anomalia mitjana i el valor corresponent a l'anomalia vertadera per cadascun dels intervals de temps de la simulació que també serà mostrat per pantalla.

Per trobar els intervals de temps on analitzar cada anomalia, els trobem amb la formula ti = i(tf/nt). Per trobar la M, utilitzem la formula: $M = 2\pi^* (t - tp)/ T$ i per trobar la V utilitzarem $\cos \nu = e - \cos E/(e \cos (E) - 1)$ i aïllant amb l'arccosinus, v=acos((e-cos(E))/(e*cos(E)-1)).

En aquest cas, la M i la E (anomalia excèntrica) estan centrades gracies a l'equació de Kepler. En canvi, la v i la E no, ja que la v va de 0 a π i la E de 0 a 2 π de manera que les he, de centrar fent que estiguin a la mateixa mitja volta. En resum, que $\nu \in [k2\pi, k2\pi+\pi]$ si $E \in [k2\pi, k2\pi + \pi]$, i $\nu \in [k2\pi + \pi, k2\pi + 2\pi]$ si $E \in [k2\pi + \pi, k2\pi + 2\pi]$.

3.2 Implementació De Software:

Per a fer-ho, creem un bucle que comenci en i igual a 0 i acabi quan i assoleixi el valor de nt, dintre d'aquest bucle anirem imprimint tant el temps com la M i la v.

Per tindre l'output dels valors del temps i de la M anem imprimint ti = i(tf/nt) i per a la M, M =((2*PI)*(ti)/T)+M0;

Aquesta formula a diferència de la que he posat abans, no depen de la tf i això es deu a que si és cert que la formula de la M (anomalia mitjana) depèn de tf que és el temps en el que el satèl·lit passa pel perigeu que coincideix quan t=0, en cas contrari, el valor de l'anomalia calculada seria igual a 0 fet que no és possible.

Per últim cal calcular el valor de v, però abans hem de guardar la solució de l'equació de Kepler cridant a la funció bisnwt a la qual li passem a i b sent $a= M - \pi$ i $b= b = M + \pi$,

que és un interval que conté l'arrel i calculem v fent l'arccosinus de la formula explicada abans amb l'arrel de Kepler trobada (E).

Per poder cridar a la funció bisnwt, abans hem de crear una estructura de Kepler la qual conté la e i la M i després dues funcions que criden a la estructura anterior, la primera que avaluï el valor de E amb la funció de Kepler i un altre amb la de la derivada per poder fer Newton.

A diferencia de la M , calculem les voltes dividint l'arrel (resultat d'haver cridat a bisnwt) / 2 π i a partir de la funció .

Per poder saber la volta en la que ens trobem utilitzem la funció de c anomenada "floor" que separa la part entera de la decimal perdent aquesta, de manera que per obtenir la part decimal, fem la resta entre el nombre de voltes i la part entera.

Per poder passar la v a una volta o a una altre, comparem el valor de la part decimal amb 0.5 per saber si està a menys mitja volta o a més de mitja volta.

Si la part decimal és més gran que 0.5 vol dir que estem a la 2ª mitja volta de manera que per centrar-la li restem al valor de v el mateix valor de v més la part entera *2 +2 i tot per π, v=-v+(part_entera*2+2)* π en cas contrari, sumem a v la part entera *2 i tot per π , v=v+(2*part_entera)* π, i imprimim els resultats com hem fet amb els temps i amb la M.

3.3 Manual Del Software Encarregat:

Els arguments entrats són: ***'double e'*** excentricitat de l'òrbita, ***'double T'*** el període de l'òrbita en segons, ***'doublé M0'*** valor de l'anomalia mitjana a l'inici de la simulació, ***'double tf'*** és la durada de la simulació en segons, i ***'int nt'*** ´es el nombre de punts de la simulació.

Aquesta utilitat ha de treure per standard output tres columnes de valors, de les quals la primera correspon al temps que ha passat, la segona correspon a la anomalia mitjana en relació al temps que ha transcorregut i per últim la anomalia vertadera corresponent al temps que ha passat.

La primera columna extreu uns outputs que van del valor al valor 0 al valor 86163.91718136377

La segona columna extreu uns valors que van del valor 3.141592653589793 al valor 15.70796326794896

La tercera columna extreu uns valors que van del valor 3.149952256959617 al valor 15.70796326794897.

Per validar la utilitat es compila el programa fent gcc -o kplt2nu -g -Wall kplt2nu.c bisnwt.c –lm i per executar-ho :./kpl 0.74105=e 43081.95859068188=T

3.141592653589793=M0 86163.91718136377=tf 333=nt gcc -o kplt2nu -g -Wall kplt2nu.c bisnwt.c –lm

289	74520.144589	14.00980507681935	15.26649515919258
290	74778.895091	14.04754192551112	15.27989748135837
291	75037.645593	14.08527877420289	15.2929689695172
292	75296.396095	14.12301562289466	15.30573072027851
293	75555.146597	14.16075247158643	15.31820214721686
294	75813.897100	14.1984893202782	15.33040115689056
295	76072.647602	14.23622616896996	15.34234430307603
296	76331.398104	14.27396301766174	15.35404692234537
297	76590.148606	14.3116998663535	15.36552325360944
298	76848.899108	14.34943671504527	15.37678654383333
299	77107.649610	14.38717356373704	15.38784914178937
300	77366.400112	14.42491041242881	15.39872258142921
301	77625.150614	14.46264726112058	15.40941765622117
302	77883.901116	14.50038410981235	15.41994448560241
303	78142.651618	14.53812095850412	15.43031257453086
304	78401.402120	14.57585780719589	15.44053086698366
305	78660.152622	14.61359465588766	15.45060779413173
306	78918.903124	14.65133150457943	15.46055131782196
307	79177.653626	14.6890683532712	15.47036896991407
308	79436.404128	14.72680520196296	15.48006788794825
309	79695.154630	14.76454205065473	15.48965484755867
310	79953.905132	14.8022788993465	15.49913629199571
311	80212.655634	14.84001574803827	15.50851835907505
312	80471.406136	14.87775259673004	15.51780690583327
313	80730.156638	14.91548944542181	15.52700753113637
314	80988.907140	14.95322629411358	15.53612559645885
315	81247.657642	14.99096314280535	15.54516624502611
316	81506.408145	15.02869999149712	15.55413441949149
317	81765.158647	15.06643684018889	15.56303487830024
318	82023.909149	15.10417368888066	15.5718722108763
319	82282.659651	15.14191053757243	15.58065085175582
320	82541.410153	15.17964738626419	15.58937509377015
321	82800.160655	15.21738423495597	15.598049100388
322	83058.911157	15.25512108364773	15.60667691729457
323	83317.661659	15.2928579323395	15.61526248329487
324	83576.412161	15.33059478103127	15.62380964061269
325	83835.162663	15.36833162972304	15.63232214465232
326	84093.913165	15.40606847841481	15.64080367328471
327	84352.663667	15.44380532710658	15.64925783571474
328	84611.414169	15.48154217579835	15.65768818098215
329	84870.164671	15.51927902449012	15.66609820614511
330	85128.915173	15.55701587318189	15.67449136419235
331	85387.665675	15.59475272187366	15.6828710717271
332	85646.416177	15.63248957056543	15.69124071646411
333	85905.166679	15.6702264192572	15.69960366457914
334	86163.917181	15.70796326794896	15.70796326794897

4.TOP SECRET (PART EXPERIMENTAL)

Aquesta part tracta de dissenyar orbites de satèl·lits espia i estudiar com estudiar a diferents països o zones concretes amb orbites Molinya que són òrbites excèntriques,amb una inclinació de 63.4 graus per evitar que, l'orbita roti dins el seu pla orbital.

D'aquesta manera, el perigeu es pot mantenir al punt de latitud més petita de la traça de l'òrbita i l'apogeu al punt de latitud més gran.

El període es pren de mig dia sideri , i el satèl·lit se situa en aquesta òrbita de manera que les passades per l'apogeu alternin entre estar per sobre del lloc espiat i sobre el lloc que vol espiar.

Per a fer aquesta part disposem dos scripts gnuplot per representar traces d'òrbites d'aquests tipus: el primer és el fitxer ctants.gnu, que defineix les constants necessàries.

En segon lloc, el fitxer dibgte.gnu, que mitjançant una crida al kplnt2nu i diversos canvis de coordenades, representa la traça de l'orbita.

Per poder representar la primera òrbita hem d'escriure aquestes comandes a gnuplot.

```
load 'ctants.gnu'

lonesp=0 # longitud del lloc que espia

xc=0.74105 # excentricitat

T=dias/2 # període de l'òrbita (mig dia sideri)

i=63.4*(pi/180) # inclinació de l'òrbita (en radians)

tf=dias # simulem durant un dia

nt=333 # a la traça se representen 333 punts

load 'dibgte.gnu'
```

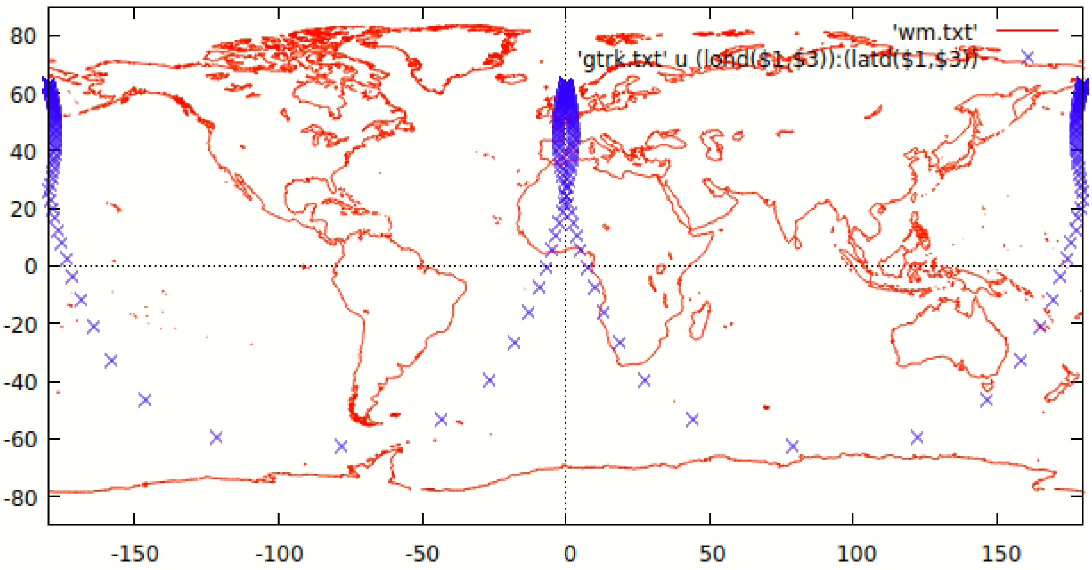
4.1 PRIMERA ÒRBITA OBSERVADA

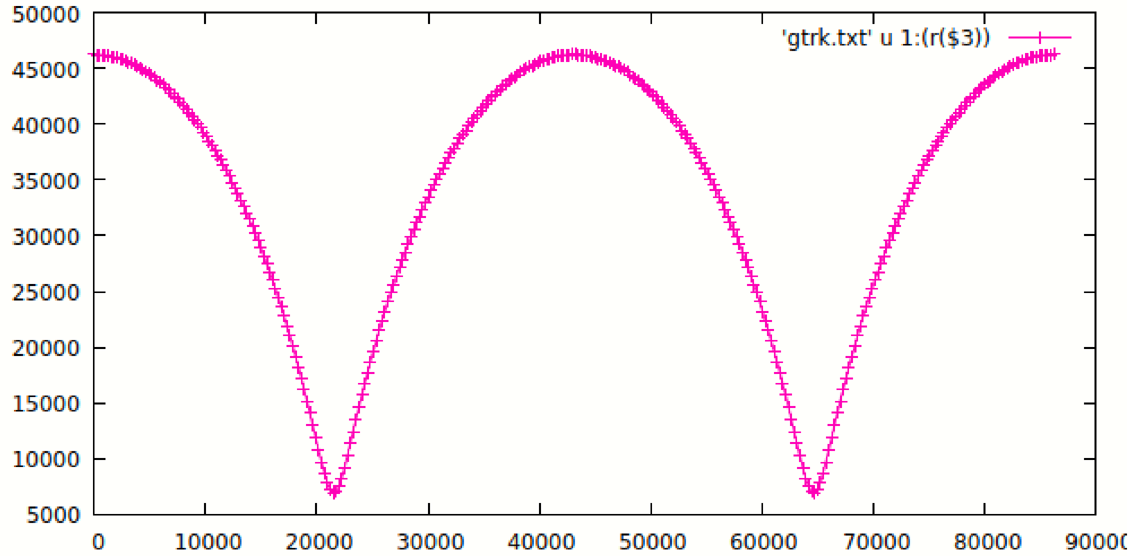
load'ctants.gnu';lonesp=0;xc=0.74105;T=dias/2;i=-63.4*(pi/180);tf=dias;nt=333;load 'dibgte.gnu'

En aquesta òrbita podem observar que les zones espiades són Espanya, Irlanda, Regne Unit, França, Itàlia i la Península de Chukotka.

Tal i com està la situació avui dia, crear una orbita per espiar, aquestes zones del món no podríem extreure molta informació.

En canvi si aconseguim espiar les zones que estan amb conflictes bèl·lics importants com és el cas de Rússia i Ucraïna, podríem extreure encara més dades rellevants.

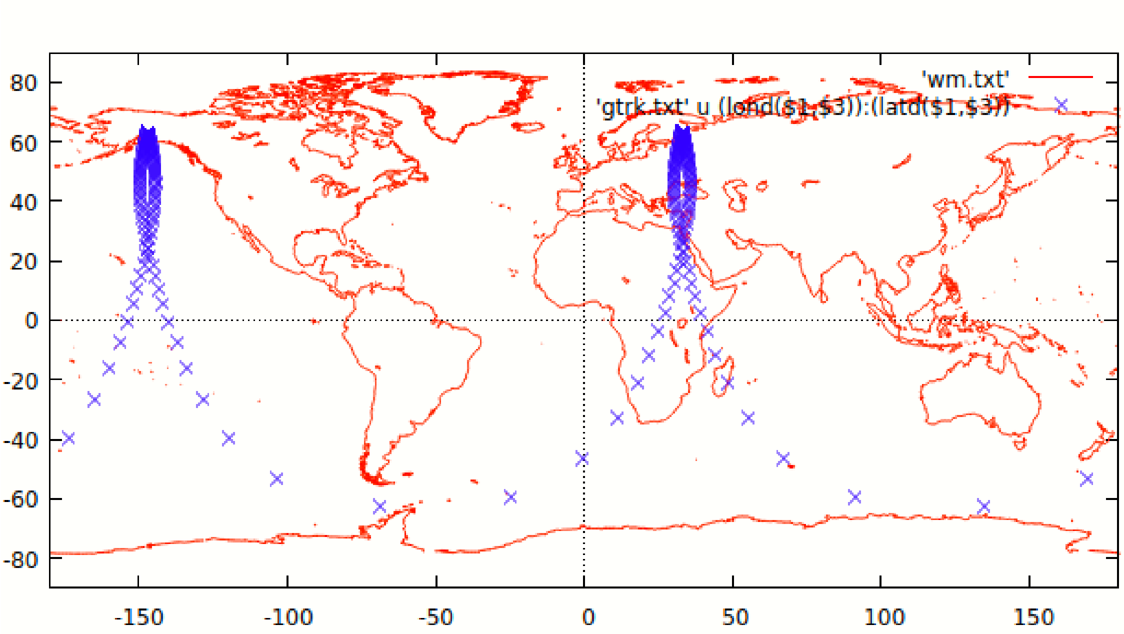


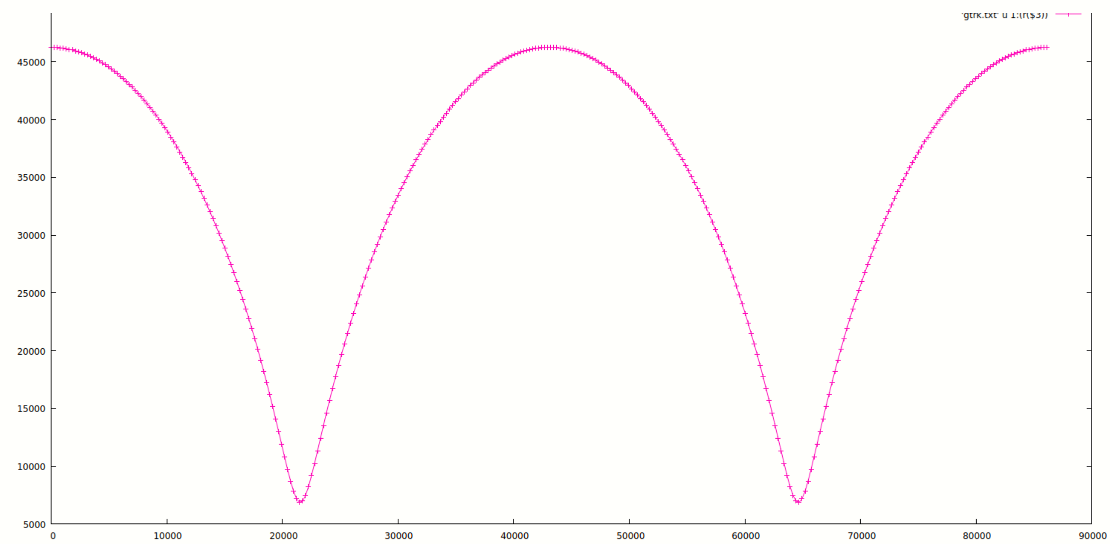


4.2 SEGONA ÒRBITA

load’ctants.gnu’;lonesp=10;xc=0.74105;T=dias/;i=63.4*(pi/180;tf=días;nt=333;load ’dibgte.gnu’

El primer paràmetre que he volgut moure en referència a l’anterior és la lonesp =10,que, d’aquesta manera podem veure que l’orbita s’ha desplaçat cap a la dreta sent les zones espiades (Fairbanks, les golfes d’Alaska, a la franja esquerra i a la dreta la zona d’Arcàngel o el mar blanc, passant per Moscou o Geòrgia) amb aquesta orbita podem espiar zones molt conflictives avui dia com és el cas de la capital de Rússia, i Ucraïna però per la part de l’esquerra només podríem espiar la part del nord d’Amèrica la majoria de la qual correspon al mar. Si el nostre objectiu és espiar zones bèl·liques importants, aquesta òrbita és bastant bona.

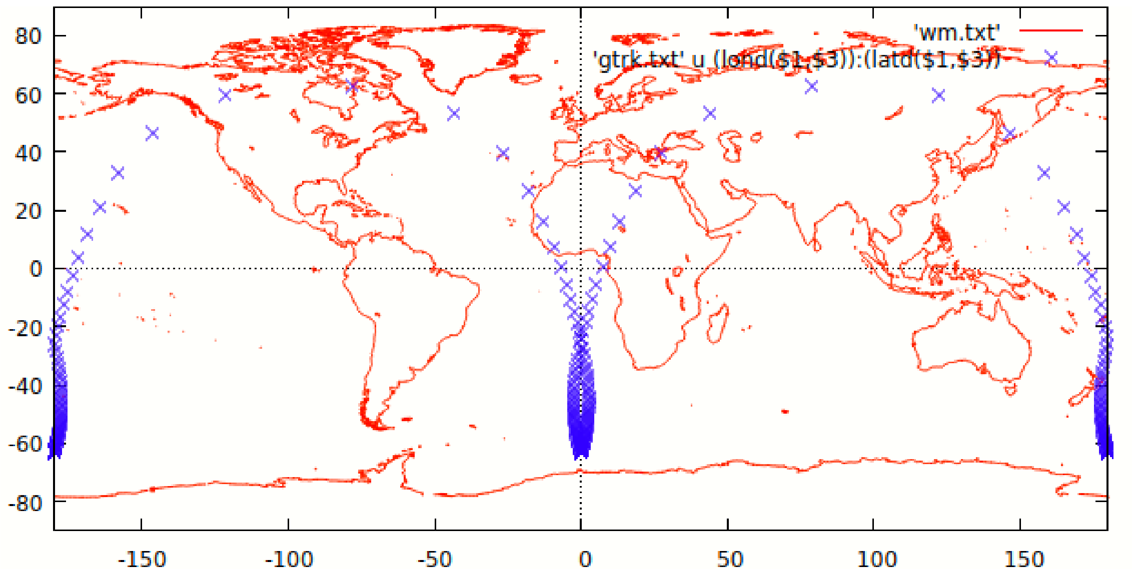


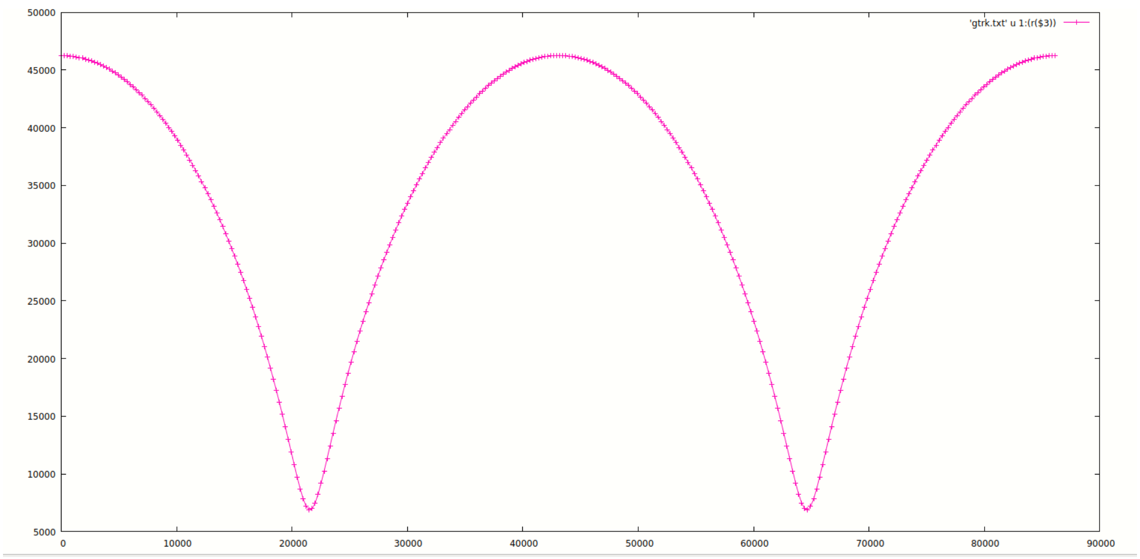


4.3 TERCERA ÒRBITA

load'ctants.gnu';lonesp=0;xc=0.74105.;T=dias;i=-63.4*(pi/180);tf=dias;nt=333;load 'dibgte.gnu'

En aquesta part d'aquí he experimentat canviant de signe la inclinació de la òrbita i surt cap per baix.

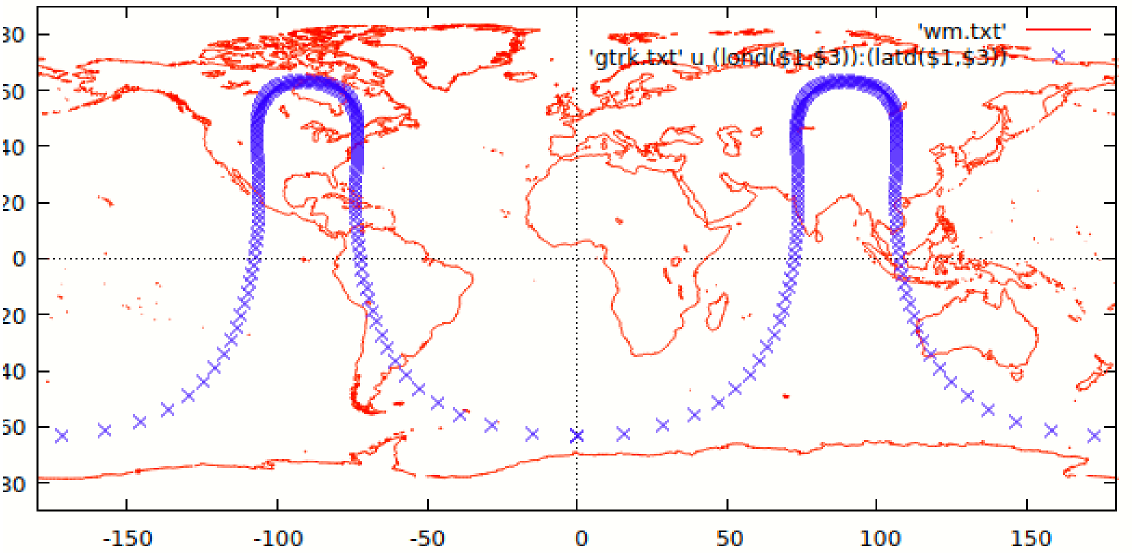


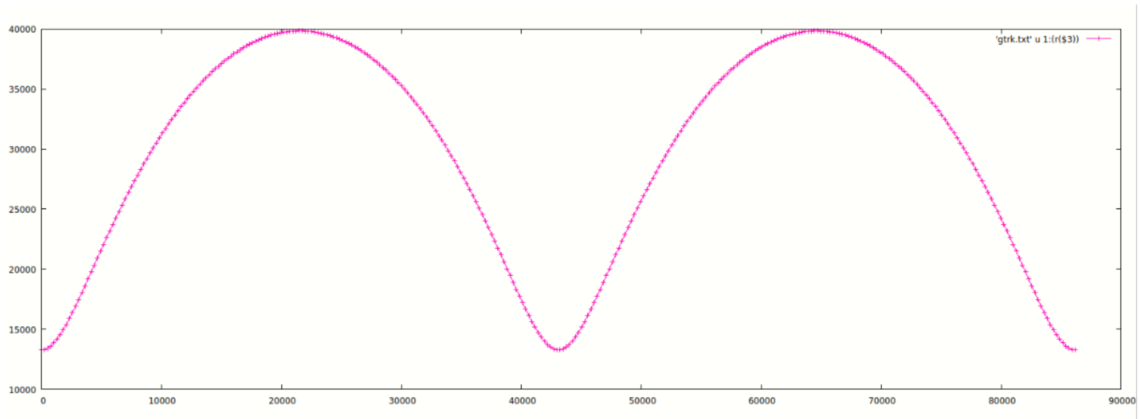


4.4 QUARTA ÒRBITA

Load'ctants.gnu';lonesp=0;xc=-0.5;T=dias/2;i=-63.4*(pi/180);tf=dias;nt=333;load 'dibgte.gnu'

En aquest experiment he calgut saber que passava si posava l'excentricitat en signe negatiu i canviant-la a 0.5 i em sortia la figura de sota girada 180 graus, és a dir la zona espiada enfocant al sud del mapa tocant només la zona del mar, de manera que per poder espiar part de la superfície terrestre he canviat de signe també la inclinació de la orbita.



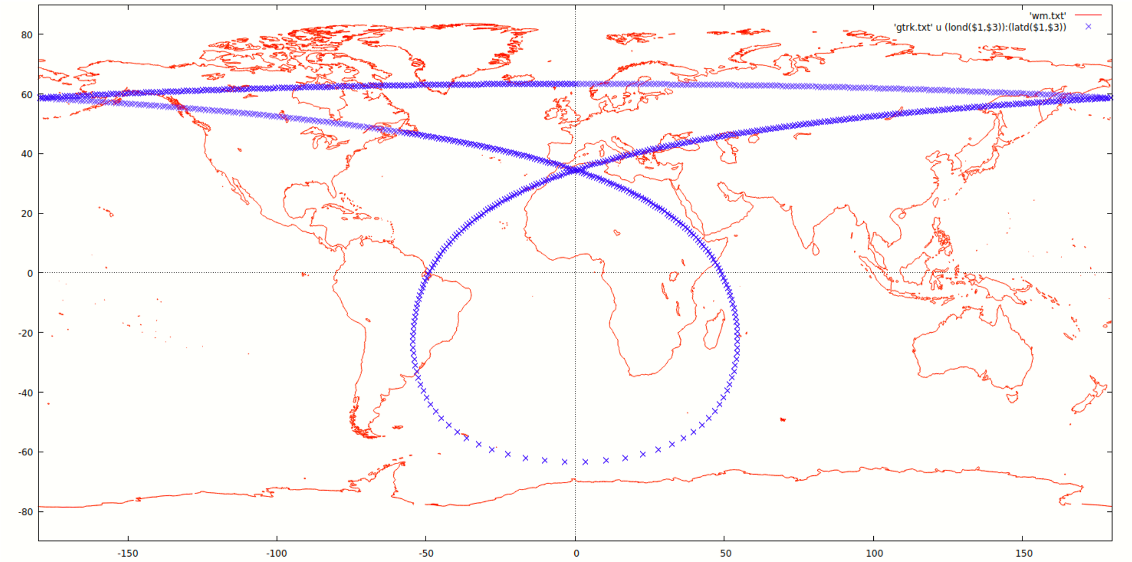


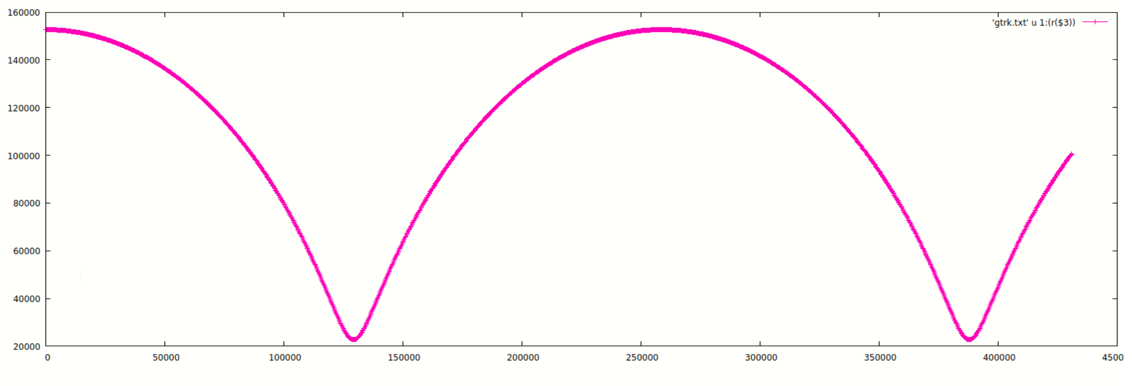
4.5 CINQUNA ÒRBITA

load'ctants.gnu';lonesp=0;xc=0.74105;T=3*dias;i=63.4*(pi/180);tf=5*dias;nt=5*333;load 'dibgte.gnu'.

Aquesta òrbita s'ha generat canviant el nombre de dies, el tf i el nombre de punts representats. Considero aquesta una bona òrbita per espiar ja que pràcticament espia tota la superfície terrestre llevat d'alguns punts d'Amèrica i Austràlia.

A més es pot veure que la posició respecte la superfície és molt més elevada que la primera òrbita calculada.





4.6 SISENA ÒRBITA (ORBITA SEMBLANT A L'ÒRBITA TUNDRA)

load'ctants.gnu';lonesp=0;xc=0.2;T=dias;i=63.4*(pi/180);tf=5*dias;nt=5*333;load'dibgte.gnu' .

Buscant informació sobre les orbites Molinya he trobat una altre orbita que m'ha semblat interessant que és l'anomenada òrbita Tundra la qual és utilitzada per certs satèl·lits de telecomunicacions per cobrir àrees mal servides per l'òrbita geostacionària. Aquestes òrbites van ser desenvolupades pels soviètics per cobrir els territoris del nord i Per poder representar-la he buscat informació sobre aquesta on deia que a diferència de l'orbita Molinya, aquesta el període és un dia i no mig i a més la excentricitat rondava entorns a 0.2