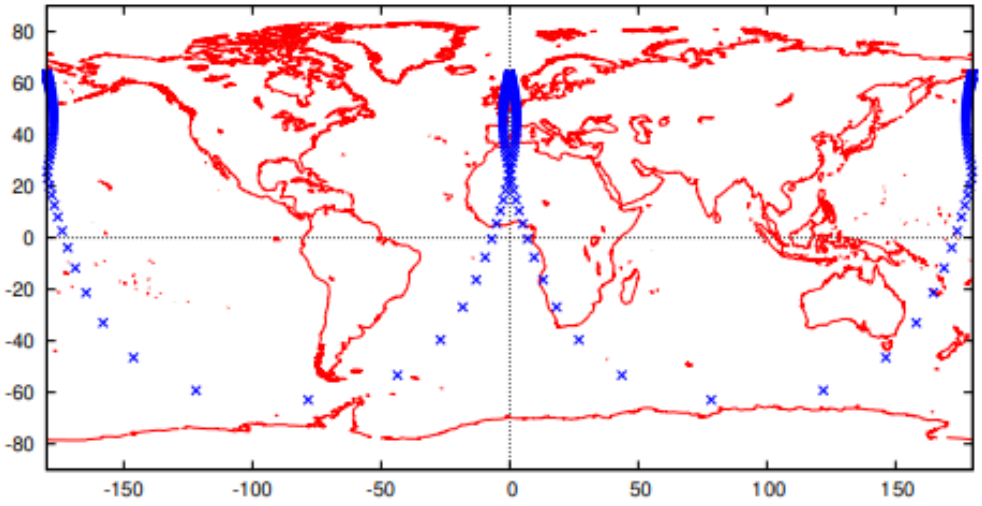
Pràctica 2 de Càlcul Numèric: Representació de "ground tracks" d'òrbites de satèl·lits artificials



Gerard Lahuerta Martín

4 de Juny del 2021

Índex

1 Introducció

1.1 Motivació del treball

Aquest treball ha estat motivat pel **Departament de Control de Missió de l'Agència Espacial Catalana (AEC)** en requerir una eina de software que li permeti determinar la posició d'un satèl·lit artificial en el seu pla orbital.

1.2 Metodologia

1.2.1 Explicació de la metodologia

La metodologia aplicada a l'hora de programar la biblioteca que sol·licitava l'**AEC** ha estat la següent:

1. Llenguatge de programació ràpid i eficient
   Utilitzem el llenguatge C gràcies a la rapidesa en els càlculs i l'alta eficiència.

2. Fàcil d'executar
   La programació ha estat orientada a la fàcil utilització d'aquest per tal de ser intuïtiva, a més, s'afegeix el manual d'ús del software per les consultes necessàries.

3. Fàcil modificació
   El programa està forçament comentat i enfocat per a ser fàcilment intuïtiu per si calgués alguna modificació a posterior i, així, facilitar la tasca.

4. Màxima optimització possible a la vegada que eficàcia i fiabilitat dels càlculs
   Combinem la velocitat del mètode de Newton amb la robustesa del mètode de bisecció en els càlculs orbitals.

1.2.2 Comentaris

El disseny de les rutines ha estat pensat i implementat per a evitar els càlculs i interaccions innecessàries. També, s'ha evitat la creació de rutines i variables redundants per a obtenir la millor optimització possible.

Tal com hem comentat anteriorment, per tal d'evitar confusions i ajudar a l'enteniment de les accions del programa s'han implementat línies de codi comentat, aquestes expliquen els conceptes més ambigus del programa i estan localitzades al costat de cada línia on s'ha cregut necessària una explicació.

També, afegir que els càlculs utilitzats en la utilitat *kpl2nu.c* (explicats a Secció 1.3) han estat els proporcionats per l'**AEC** i han estat aplicats tal com ens els han proporcionat i demanat que s'apliquin.

1.2.3 Informació d'interès

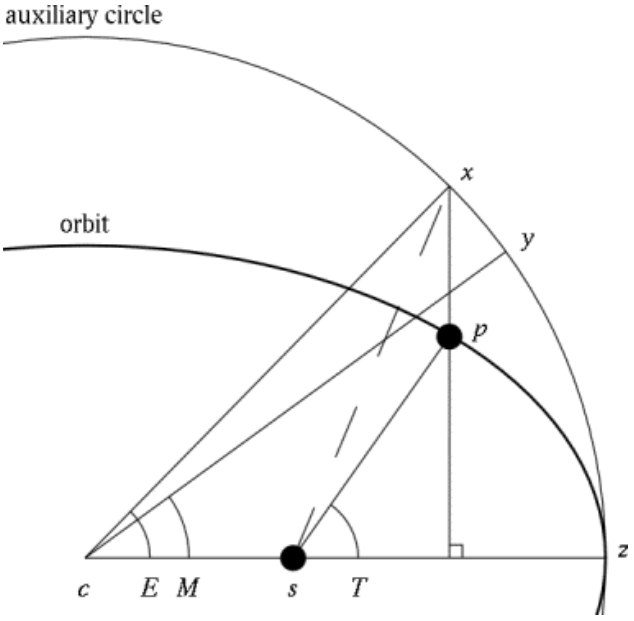
La biblioteca ha estat programada amb l'editor d'ús públic Notepad++, encara que pot ser modificat en altres editors com podrien ser l'editor de notes o d'altres més complexos com VisualStudio.

Durant la programació de les rutines s'ha ideat un programa per testejar el correcte funcionament de la biblioteca i així en cas de possibles errors poder comprovar el correcte funcionament d'aquesta.

L'ús d'aquest test que està anomenat com *bisnwt_test.c* és fàcilment executable i és explicat més a fons a la Secció 2.3.

1.3 Càluls utilitzats en la utilitat *kpl2nu.c*

Per a fer els càlculs orbitals se'ns és necessari les següents dades:



- **Anomalia Mitjana:** M

$$M = 2\pi \frac{t - t_p}{T}$$

- **Anomalia Excentrica:** E

$$M = E - e \sin E$$

- **Anomalia Vertadera:** ν

$$\nu = \arccos \frac{e - \cos E}{e \cos E - 1}$$

Figura 1: Relació entre anomalies

On t, tp, T i e són respectivament el temps actual, el temps que tarda en passar pel perigeu (un dels màxims de l'òrbita el·líptica), el període de l'òrbita i l'excentricitat de l'el·lipse. Recalcar que: $t \in [0, +\infty)$, $tp \in \Re$, $T \in \Re$ i $e \in [0, 1)$ són coneguts i proporcionats per l'**EAC**. També afegir que: $M \in \Re$, $E \in [0, 2k\pi]$ i $\nu \in [0, 2k\pi]$, essent $k \in \mathbb{N}$

2 Explicació de la biblioteca *bisnwt.c*

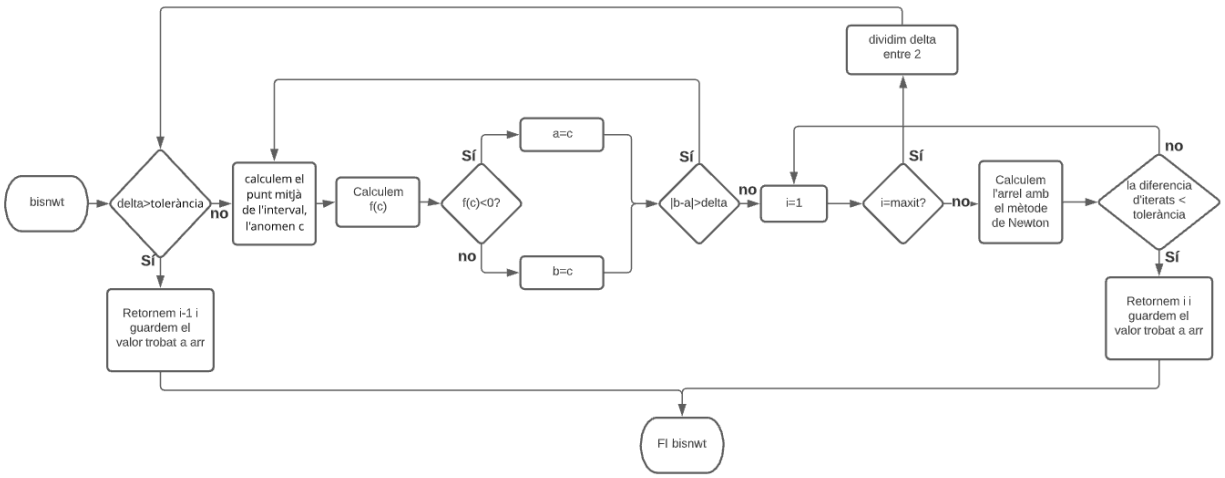
En aquest capítol parlarem a fons sobre l'estructura, els materials que implementa, el funcionament, les estratègies utilitzades i el correcte ús i execució del programa.

2.1 Implementació

La biblioteca està pensada per a ser implementada en els entorns d'execució de l'**EAC**. L'únic mètode que s'inclou a la biblioteca (del qual parlem a continuació) pot ser utilitzat independentment del programa que l'implementi i no precisa cap altra eina externa més enllà de la llibreria *math.h* del conjunt de llibreries del llenguatge C.

2.2 Funcionament

El funcionament del mètode *bisnwt* és senzill. Aplica inicialment el mètode de bisecció per a trobar un interval menor a un valor δ introduït per a després aplicar amb el mètode de Newton per a trobar l'arrel (així ens assegurem una millor viabilitat i eficiencia en els càlculs). Finalment, si el mètode de newton després de *maxit* iterats (introduït també com a paràmetre) no ha convergit, el redueix el valor de δ en la mitad i es repeteix el procés fins que el mètode Newton convergeixi o l'interval sigui menor que un tolerancia (*tol*) prèviament també introduïda.



2.3 Testatge de la biblioteca: *test* i *bisnwt_test.c*

Expliquem a continuació el programa *bisnwt_test.c* i la correcta implementació de la biblioteca en els codis de programació de l'**EAC** així com en el mateix codi.

2.4 Compilació de la utilitat de testatge: *bisnwt_test.c*

Per a implementar a biblioteca *bisnwt.h* en qualsevol codi en C cal a l'inici del codi escriure la comanda següent:

```
#include "bisnwt.h"
```

Una vegada escrita la comanda el programa implementarà l'extensió al codi.
Per a compilar *test_bisnwt.c* i obtenir la utilitat *test* es pot utilitzar l'arxiu *MakeFile* que s'inclou.
Aquest es pot utilitzar de les maneres següents:

- **Si es tracta del sistema Operatiu Unix**, en la consola/terminal de l'ordinador executar la comanda *make -f Makefile.txt* per crear l'executable *test*.
  Anàlogament, executar la comanda *clean -f Makefile.txt* per netejar *d'arxius objecte* innecessaris el directori de treball o *realclean -f Makefile.txt* per esborrar tot el treball fet per l'arxiu Makefile.

- **Si es tracta del sistema Operatiu Windows**, en la consola/terminal de *mingw-w64* executar la comanda *mingw32-make -f Makefile_Windows.txt* per crear l'executable *test.exe*.
  Anàlogament, executar la comanda *minw32-make clean -f Makefile_Windows.txt* per netejar d'arxius objecte innecessaris el directori de treball o *minw32-make realclean -f Makefile_Unix.txt* per esborrar tot el treball fet per l'arxiu Makefile.

Si no s'utilitza l'arxiu Makefile, es pot compilar els arxius que continguin aquesta biblioteca amb la comanda següent al terminal/consola:

```
gcc -o nom_del_executable -Wall nom_del_programa.c bisnwt.c -lm
```

Per obtenir l'executable que ens interesa en aquest cas utilitzem la comanda abans explicada,

```
gcc -o test -Wall test_bisnwt.c bisnw.c
```

O també podem utilitzar l'arxiu Makefile amb:

```
make -f test Makefile.txt
```

Analogament en cas de Windows, aplicar la comanda explicada pel sistema operatiu amb l'afegit de la comanda *test* abans del nom de l'arxiu.

2.5 Output i executable *test*

Per tal d'assegurar el funcionament correcte de la biblioteca hem posat a disposició de l'**EAC** el codi *test_bisnwt* que té com a objectiu assegurar que el mètode que actualment conté la biblioteca funcioni de manera correcta.
El programa *test_bisnwt.c* és molt senzill, únicament crida la funció que conté *bisnwt.h* per fer uns càlculs de prova i eventualment mostrar-los per pantalla.
Per a executar la utilitat cal introduir pel terminal les següents comandes i comparar-les amb els respectius outputs:

- **test -9 1 10 1e-12 10**

  ```
  valor: 0.6931471805599453 estat del valor: 7
  ```

- **test -9 1 2.5 1e-12 10**

  ```
  valor: 0.6931471805599453 estat del valor: 7
  ```

- **test -9 1 0.1 1e-12 10**

  ```
  valor: 0.6931471805599453 estat del valor: 5
  ```

3 Explicació de la utilitat

3.1 Funcionament de *kpl2nu.c*

En aquesta secció explicarem el funcionament del programa *kpl2nu.c*, el seu funcionament intern i les seves dependències.

3.2 Dependències

Primerament cal esmentar i explicar les necessitats i dependències del programa *kpl2nu.c*. Aquest requereix la biblioteca solicitada i explicada anteriorment (*bisnwt.h*). Per altra banda, per a dur a terme els càlculs se l'introdueix com arguments en l'execució els valors de e, T, $M0$, tf i nt on:

- **Variable *e*:** double que guardarà l'excentricitat de l'el·lipse que traça el satel·lit.
- **Variable *T*:** double que guardarà el valor del període de l'òrbita del satel·lit.
- **Variable *M0*:** double que guardarà el valor de l'anomalia mitjana inicial (en $t_0 = 0$).
- **Variable *tf*:** double que guardarà el temps final en la trajectora del programa.
- **Variable *nt*:** integuer que guardarà el nombre d'intervals a càlcular de l'òrbita.

Amb totes aquestes dades, el programa ya pot treballar en calcular la traça de la trayectoria del satèl·lit.

3.2.1 Explicació de *kpltnu2.c*

El funcionament del programa, en concepte és senzill. A partir de les dades inicials calcula el nombre d'intervals en els quals calcular on es troba el satel·lit en la seva òrbita per així, una vegada calculats tots els intervals, mostrar les dades per pantalla.
Per poder calcular l'anomalia Excèntrica E utilitzem la biblioteca *bisnwt.h* cridant el mètode *bisnwt*.

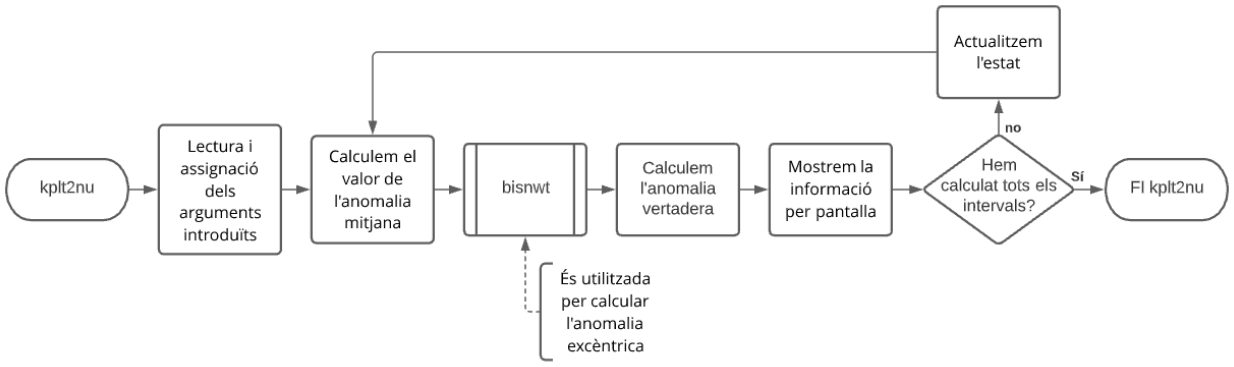


Figura 2: Diagrama del codi kplt2nu

Cal recalcar que els valors que fan referència al màxim d'iteracions $maxit$, la δ així com la tolerancia tol són ja paràmetres definits en el codi; concretament 100, 0.01 i 10^{-12} respectivament.
També afegir que a l'hora de programar *kpl2nu.c* es va programar per necessitat del càlculs dues funcions ($Kepler\_fxE$ i $Kepler\_dfxE$) que són les introduïdes com apuntadors a funcions en el mètode *bisnwt* que, per a poder obtenir totes les dades necessàries, se'l passà un l'adreça de memoria de l'estructura *dades_kpl* que conté l'excentricitat de l'el·lipse e i l'anomalia mitjana actualizada per la posició que es vol calcular M.

4 Execució *kplt2nu*

4.0.1 Compilació de *kpl2nu.c*

Per a compilar *kpl2nu.c* i obtenir l'executable *kplt2nu* es pot utilitzar la comanda següent:

```
gcc -o kplt2nu.c -Wall kplt2nu.c bisnw.c
```

O també podem utilitzar l'archiu Makefile amb:

```
make -f kplt2nu Makefile.txt
```

Anàlogament en cas de Windows, aplicar la comanda explicada pel sistema operatiu amb l'afegit de la comanda

4.0.2 Output del programa

L'output de la utilitat són tres columnes de dades que són impreses per pantalla ordenada de la següent manera:

$$\begin{matrix} t_0 & M_0 & \nu_0 \\ t_1 & M_1 & \nu_1 \\ \vdots & \vdots & \vdots \\ t_{nt} & M_{nt} & \nu_{nt} \end{matrix}$$

Essent t el temps, M l'anomalia mitjana, ν l'anomalia excèntrica i nt el valor d'intervals en el que dividir l'òrbita del satèl·lit.

Pel fet que l'output mostrat tendeix a ser molt gran, es recomana afegir en el moment de l'execució una redirecció d'entrada a un fitxer que guardi tota la informació.

5 Anàlisis de les dades obtingudes

Per analitzar millor les dades obtingudes s'hi ha optat en utilitzar el programa informàtic *gnuplot*.

Per a poder representar l'òrbita sobre el mapa terrestre del satèl·lit cal executar les següents comandes dins del terminal de *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 es representen 333 punts
> load 'dibgte.gnu'
```

En cas que es vulgui representar un diagrama amb l'altura que adquireix el satèl·lit cal executar després de les comandes anteriors la línia següent:

```
> set autoscale ; plot 'gtrk.txt' u 1:(r($3)) w lp
```

Amb aquest exemple d'aquí s'obté les següents imatges:

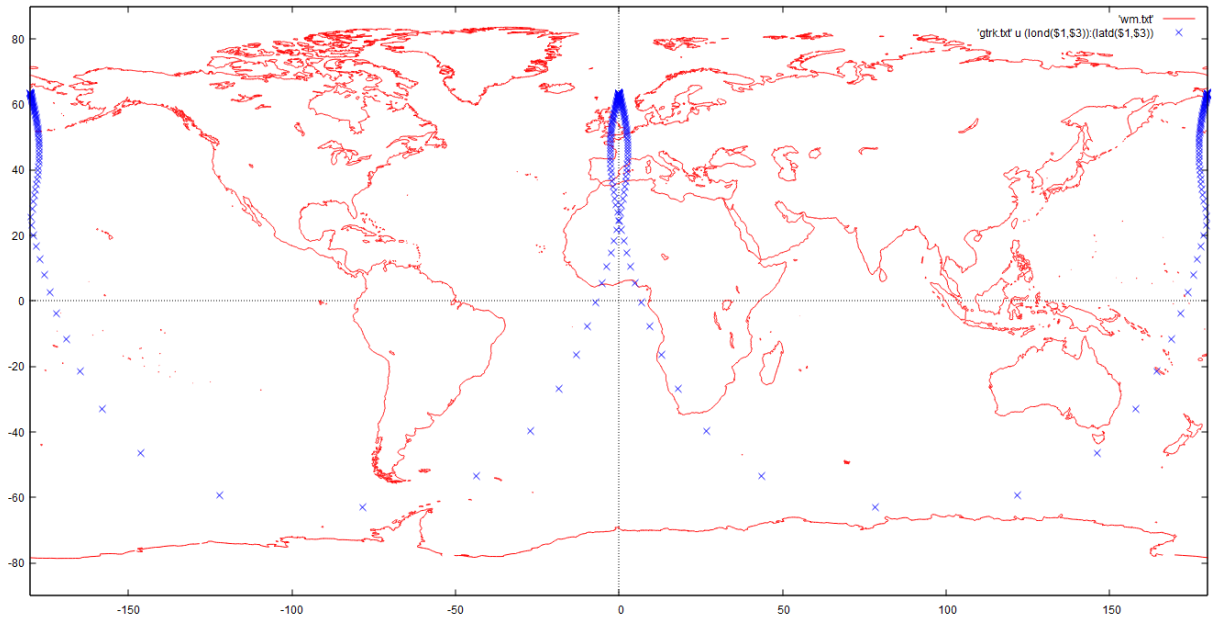


Figura 3: Traça de la configuració d'exemple

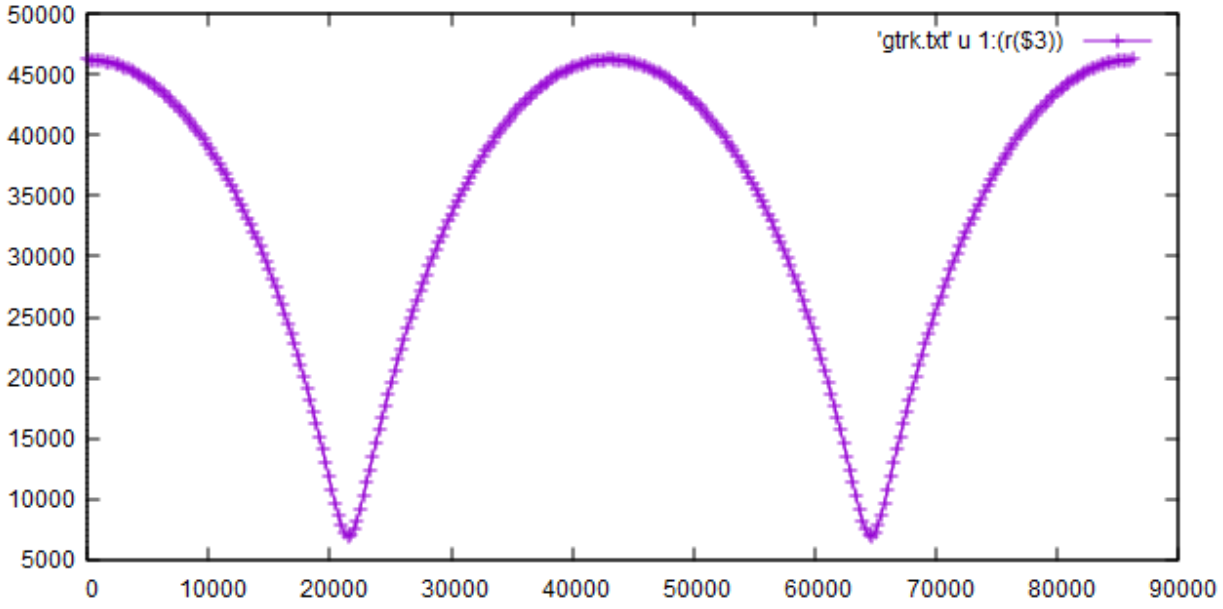


Figura 4: Gràfic de l'altura de la configuració d'exemple

Propostes sobre d'investigació i espionatge per l'**EAC**

Com a part de la nostra tasca en la programació de la utilitat *kplt2nu*, hem fet una recopilació de països als que l'**EAC** podria estar interesada en investigar/espiar les seves activitats aprofitant les qualitats de les òrbites Molniya.

Una de les possibilitats de llocs a estudiar serien Nord Corea i els Estats Units d'Amèrica, que poden ser espiats mitjançant una òrbita amb una excentricitat $e = 0.5$, tingui un període de $T = \frac{\text{dia sireri}}{3}$ i una longitud del meridià $lonesp = -4$. Tot així no recomenem del tot aquesta configuració orbital a causa del gran risc de caure en l'atmosfera a causa del fregament amb ella (ja que hi passa molt prop d'ella).

Per altra banda, també es proposa l'espionatge al Regne Unit mitjançant un satèl·lit que orbiti amb una excentricitat $e = 0.543210987654$ una longitud de meridia $lonesp = 0$ i un període d'òrbita $T =$ dia sireri. Aquesta órbita s'allunya suficientment de l'atmosfera i pot espiar el Regne Unit a la vegada que envia la informació que es rep a la seu de control de l'**EAC**.

Finalment, l'òrbita més factible és la que té una excentricitat de $e = 0.498765678923$, un període $T = \frac{\text{dia sireri}}{2}$ i una longitud del meridia $lonesp = 0$. Aquesta està feta per a observar els moviments en l'estret de Bering entre Rusia i Estats Units.

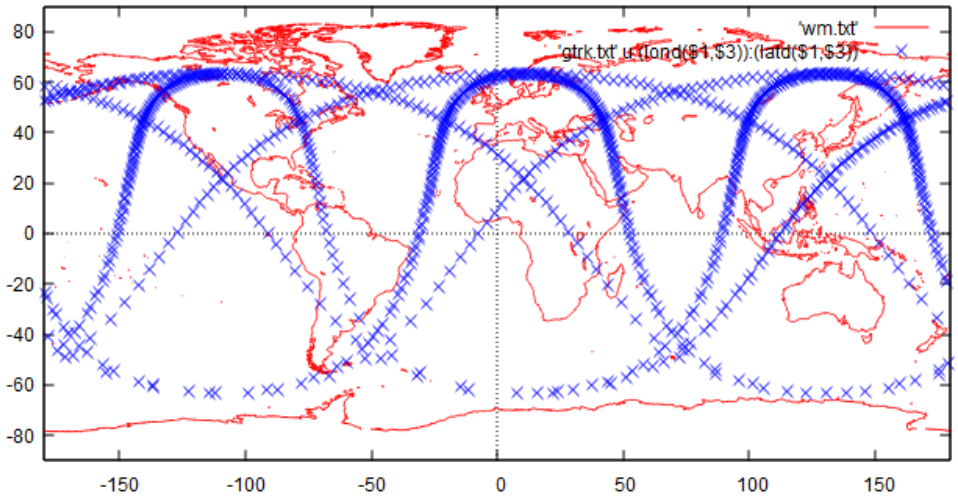


Figura 5: Traça de l'òrbita per a espiar a Nord Corea i USA amb 1000 punts

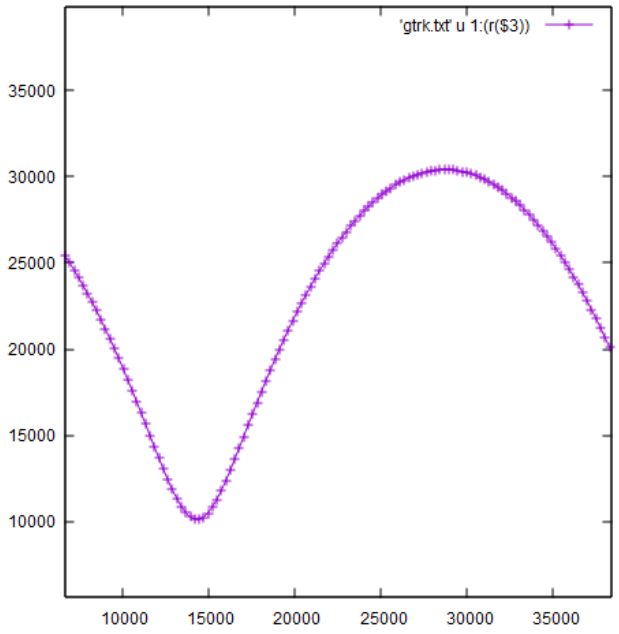


Figura 6: Altituds de l'òrbita per a espiar a Nord Corea i USA amb 1000 punts

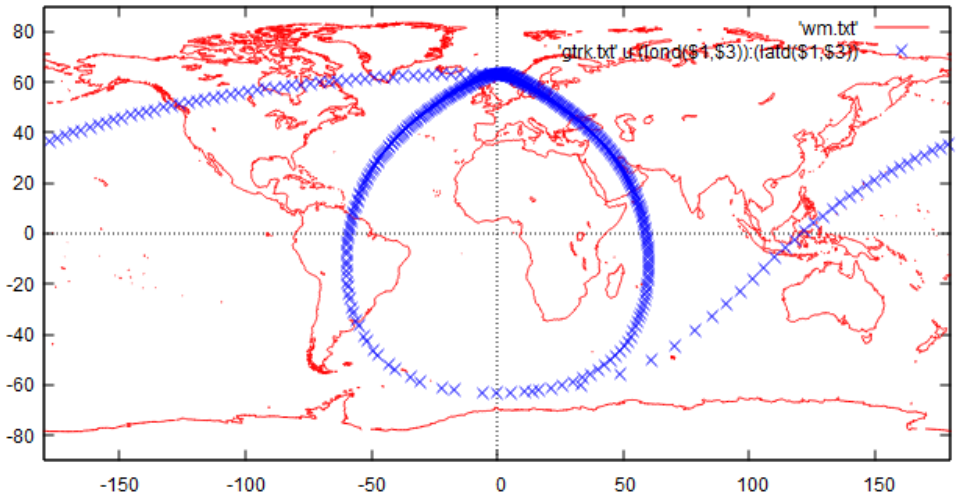


Figura 7: Traça de l'òrbita per a espiar el Regne Unit amb 1000 punts

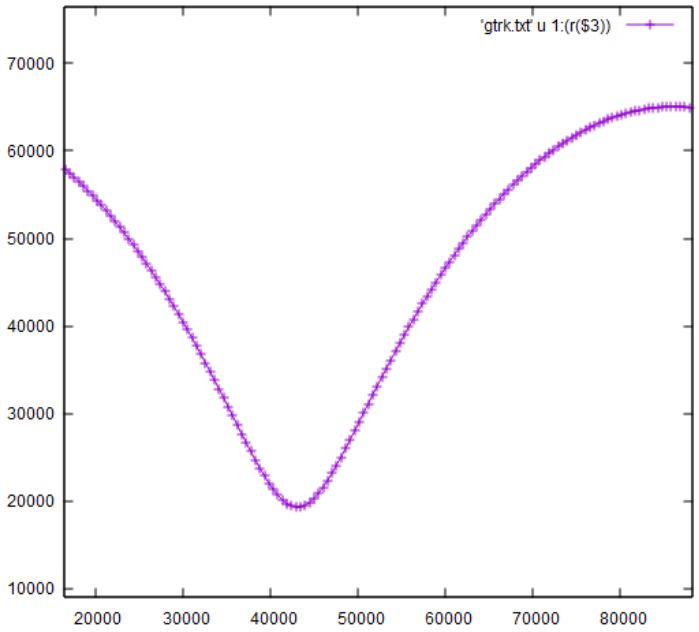


Figura 8: Altituds de l'òrbita per a espiar el Regne Unit amb 1000 punts

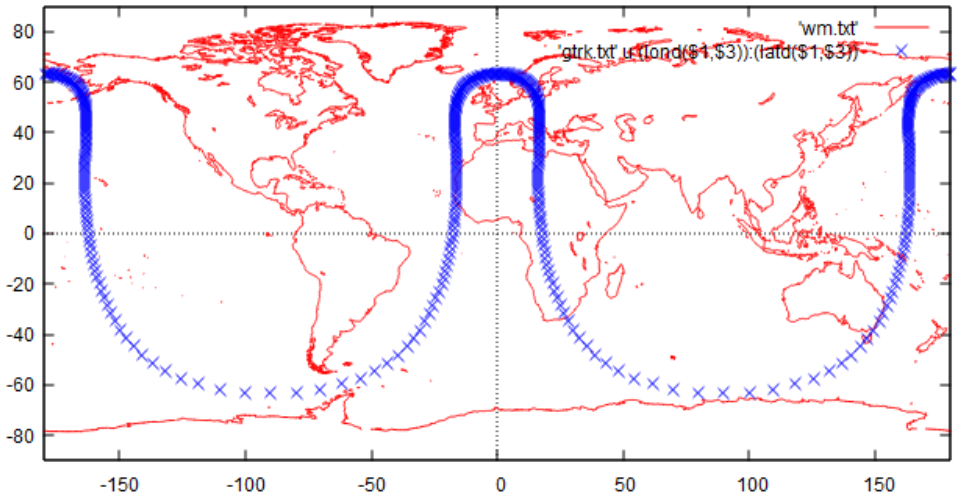


Figura 9: Traça de l'òrbita per a espiar l'estret de Bering amb 1000 punts

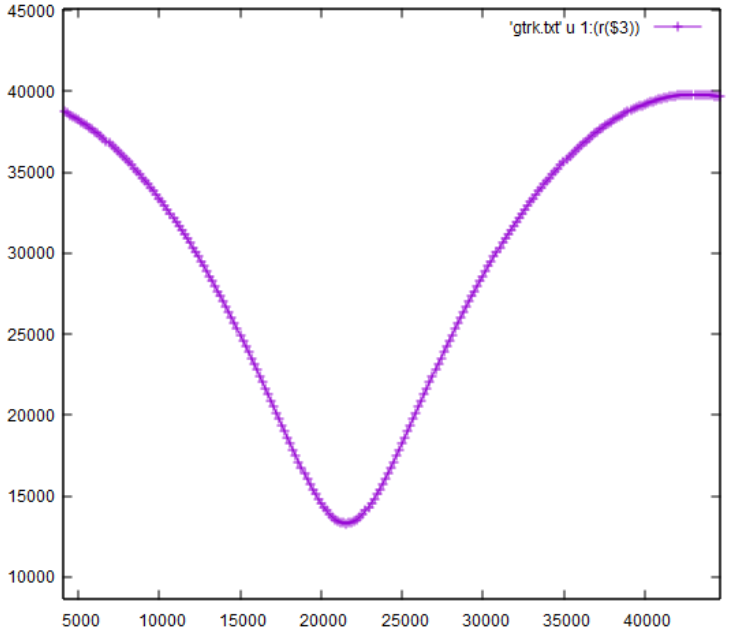


Figura 10: Altitud de l'òrbita per a espiar l'estret de Bering 1000 punts