PRÀCTICA 2: MEMÒRIA

Ona Sánchez Núñez: 1601181

5 de Juny del 2021

1 Problema plantejat

Fer un programa pel Departament de Control de Missió de l'Agència Espacial Catalana (AEC), que requereix d'una eina de software que permeti determinar la posició d'un satèl·lit artificial en el seu pla orbital.

Sabem que el càlcul està basat en la resolució numèrica de l'equació de Kepler i, a part, l'A-EC necessita una rutina per buscar zeros de funcions, que ha de combinar el mètode de Newton amb el mètode de bisecció.

La motivació real d'aquest problema és informació confidencial de l'AEC, i es descriurà a la darrera part de la memòria.

2 Requeriments tècnics

2.1 Metodologia

Per a trobar la posició d'un satèl·lit artificial dins el seu pla orbital s'ha de trobar la seva anomalia vertadera com a funció del temps. Per a això, s'ha de resoldre l'equació de Kepler,

$$M = E \text{ - } e\sin E$$

que permet tromar l'anomalia excèntrica, E, a partir de l'anomalia mitjana M.

La relació entre el temps i l'anomalia mitjana ve donada per:

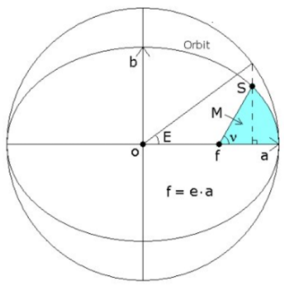
$$M = 2\pi\ \frac{t \text{ - } tp}{T}$$

La relació entre l'anomalia excèntrica, E, i l'anomalia vertadera, v, ve donada per:

$$\cos v = \frac{e-cosE}{ecosE-1}$$

Per triar la determinació de l'arc cosinus, cal tenir en compte que $v \in$ [k2π, k2π + π] si $E \in$ [k2π, k2π + π], i $v \in$ [k2π + π, k2π + 2π] si $E \in$ [k2π + π, k2π + 2π].

La relació entre v, M i E es pot veure representada a continuació:



2.2 Algorisme: Mètode de bisecció

$a_1 := a, b_1 := b, I_1 := [a_1, b_1]$
$\forall i$ = 1,2,3,...
  $p_i := (a_i+b_i)\frac{1}{2}$
  si $f(a_i)f(p_i) \leqslant 0$
    $a_{i+1} := a_i, b_{i+1} := p_i$
  si no
    $a_{i+1} := p_i, b_{i+1} := b_i$
  $I_{i+1} := [a_{i+1}, b_{i+1}]$
Fins un criteri de parada.

2.3 Algorisme: Mètode de Newton

$\forall i$ = 0,1,...,n_{max} -1

$x_{i+1} = x_i$ - $\frac{f(x_i}{f'(x_i)}$

si $|x_{i+1} - x_i| \leqslant \epsilon$, tornem x_{i+1}, STOP

error (no convergència)

Per més informació sobre aquests mètodes consultar `https://mat.uab.cat/~jmm/cnmc/material/apunts-v001.pdf`

3 Estratègia per programar

- Evitar repetir codi.
- Tabular el codi.
- Noms de variables que permetin una lectura entenedora.
- Comentar el codi.
- Validar cada funció un cop realitzada.

4 Manual del software

4.1 Per a les biblioteques

4.1.1 bisnwt()

Prototipus:

```
int bisnwt (double a, double b, double *arr, double *dlt, double tol, int maxit,
double (*f)(double,void*), double (*df)(double,void*), void *prm);
```

Descripció: Rutina que busca el zero d'una funció de classe C^1 mitjançant el mètode de bisecció i el mètode de Newton.

Arguments que rep:

- a : double, primer extrem de l'interval.
- b : double, segon extrem de l'interval.
- *arr : punter de tipus double, guarda el valor de l'arrel.
- *dlt : punter de tipus double, delta.
- tol : double, tolerància.
- maxit : int, màxim nombre d'iterats.
- *prm : punter de tipus void utilitzat com una llista que guarda a la posició 0 l'excentricitat de l'òrbita i a la posició 1 l'anomalia mitjana.
- (*f)(double, void*) : punter a funció, avalua f en el punt que passem (el double).
- (*df)(double, void*) : punter a funció, avalua la derivada de f en el punt que passem (el double).

Retorna:
- Si s'aconsegueix un interval més petit a la tolerància amb el mètode de bisecció, es retorna un -1 i es guarda el punt mig de l'interval (c) a *arr.

- En cas contrari, es fa Newton i, si convergeix, retorna el nombre d'iteracions i es guarda la arrel a *arr, si no convergeix es torna a fer bisecció.

4.1.2 kplt2nu

Prototipus:

```
int bisnwt (double a, double b, double *arr, double *dlt, double tol, int maxit,
double (*f)(double,void*), double (*df)(double,void*), void *prm);
```

Descripció: Conté el main. Ha de mostrar per pantalla l'anomalia mitjana i l'anomalia vertadera per cada temps ti.

Arguments que rep:

- e : double, exentricitat de l'òrbita.
- T : double, període de l'òrbita.
- M0 : double, anomalia mitjana.
- tf : double, durada de la simulació en segons.
- nt : int, nombre de punts de la simulació.

Altres variables importants dins la rutina són:

- ti : double, fa referència a cada instant de temps calculat com i*(tf/nt).
- Mi : double, anomalia mitjana per ti.
- tp : double, constant, s'utilitza per calcular Mi.
- vi : double, anomalia vertadera per ti.
- a,b : doubles, es passen com a paràmetres a bisnwt(), corresponen als extrems del intervals que contenen l'arrel a trobar.

Procés:

- Calcular l'anomalia mitjana M_i.
- Utilitzar la rutina bisnwt() amb la fòrmula de Kepler per calcular l'anomalia excèntrica E.
- Calcular la corresponent anomalia vertadera v_i.
- Mostrar-ho per pantalla.

4.2 Per a les utilitats

4.2.1 Com executar kplt2nu per Windows

1. Tenir completes les funcions dels fitxers *bisnwt.c*, *bisnwt.h* i *kplt2nu.c.*
2. Obrir el terminal i situar-se al directori on es tenen guardats els fitxers anteriors.
3. Compilar el programa amb la comanda : gcc -o kplt2nu -g -Wall kplt2nu.c bisnwt.c -lm
4. Executar el programa amb la comanda : kplt2nu 0.74105 43081.95859068188 3.141592653589793 86163.91718136377 333

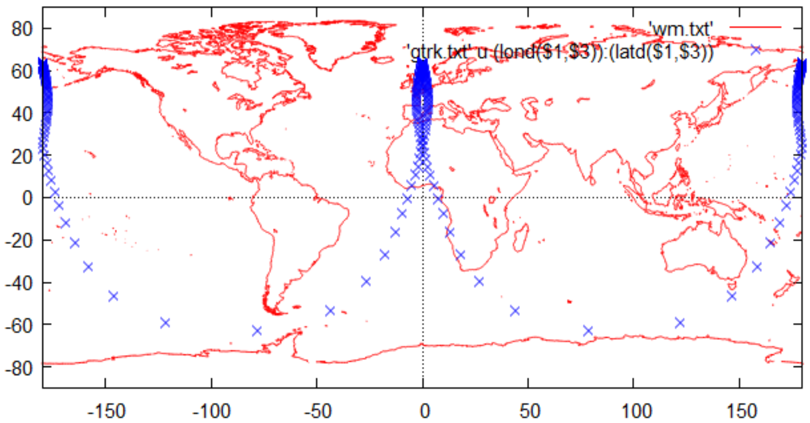
4.2.2 Com dibuixar òrbites

1. Obrir gnuplot i situar-se al directori on es tenen guardats els fitxers *ctants.gnu* i *dibgte.gnu*.
2. Executar la comanda : load 'ctants.gnu'
3. Donar valors a lonesp, xc, T, i, tf i nt.
4. Executar la comanda : load 'dibgte.gnu'

Apunts tècnics: lonesp = longitud del lloc que espiem, xc = excentricitat, T = període de l'òrbita, i = inclinació de l'òrbita, tf = dies, nt = punts que es representen a la traça. La inclinació de l'òrbita i ha de ser sempre 63.4*(pi/180), la resta de variables poden variar.

A continuació es pot veure un exemple que correspon a una traça d'òrbita de tipus Molniya.

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



5 Descripció de les proves

5.1 Validació

Per validar el funcionament de la rutina *bisnwt.c* es va crear una rutina *bisnwt test.c*. Aquesta rutina busca l'arrel de la funció e^x-2. El resultat teòric és ln(2) $\sim$ 0.6931471805...

La rutina retorna 0.693147 dins *arr.

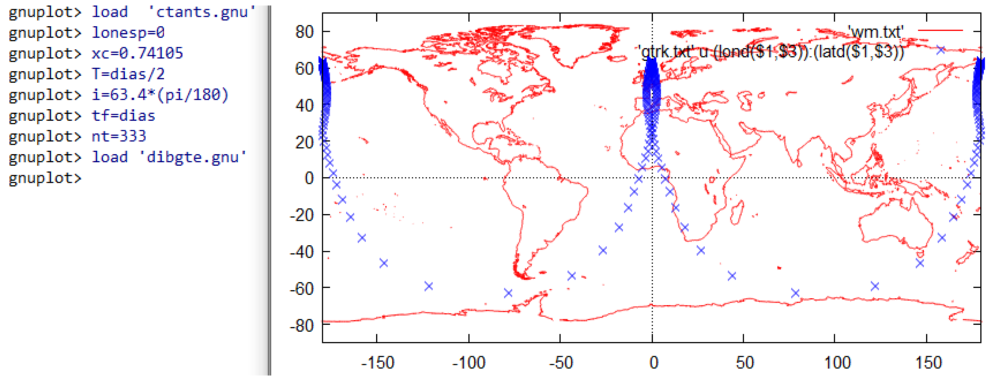
El funcionament de *kplt2nu.c* s'ha validat amb la comparació de la sortida per pantalla de la rutina (els valors ti, Mi i vi) amb el fitxer *orb.txt* del campus virtual.
En la validació de la rutina s'han utilitzat els valors dels paràmetres fets servits per crear *orb.txt*.

6 CONFIDENCIAL

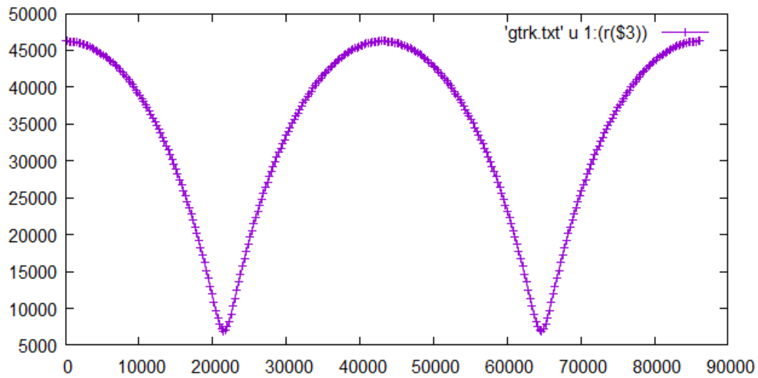
A continuació es troba un estudi extraoficial que només poden conèixer col·laboradors de confiança de l'AEC.

Per evitar que la informació que hi ha a continuació arribi a tercers, es demana la immediata destrucció d'aquest document després de la seva lectura.

En realitat, la finalitat del treball realitzat fins ara consisteix en determinar possibles llocs a espiar amb òrbites semblants a l'òrbita de tipus Molniya:



Per tenir compte que les òrbites que es consideren no poden entrar dins l'atmosfera, es pot representar la distància al centre de la terra com a funció del temps:

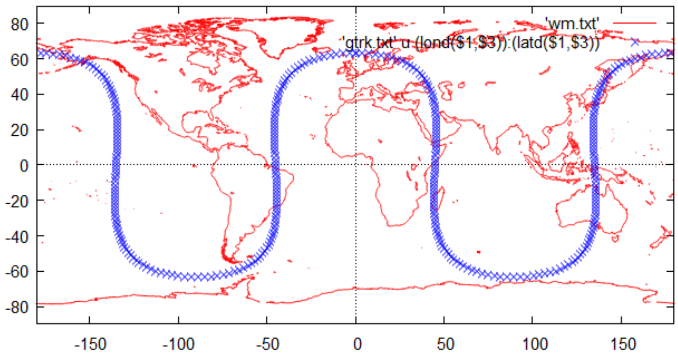


S'ha utilitzat la comanda:

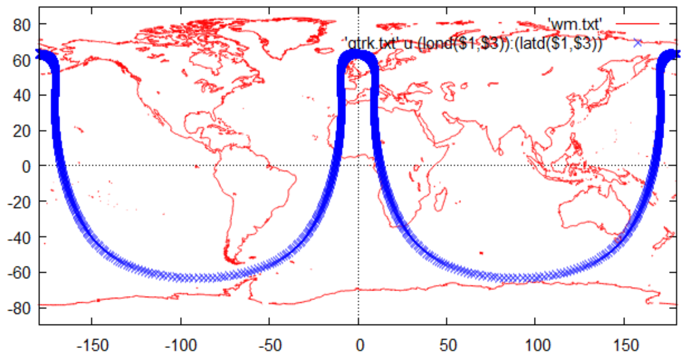
```
set autoscale ; plot 'gtrk.txt' u 1:(r($3)) w lp
```

6.0.1 Proves

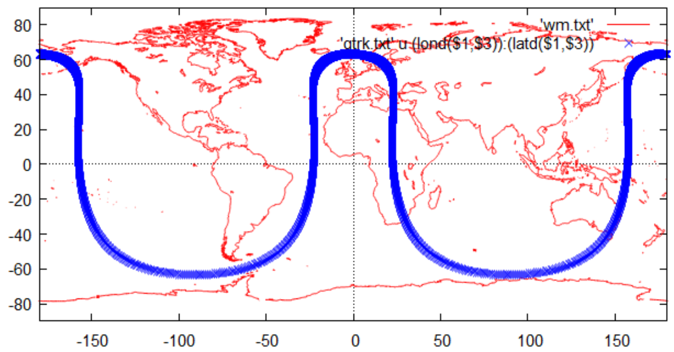
Si canviem l'exentricitat, **xc**, donant-li valor 0, veiem que la forma de l'òrbita es torna més estreta, cosa que permetria l'espionatge de zones del centre d'Austràlia, Japó, Aràbia Saudí i algunes zones de Rússia.



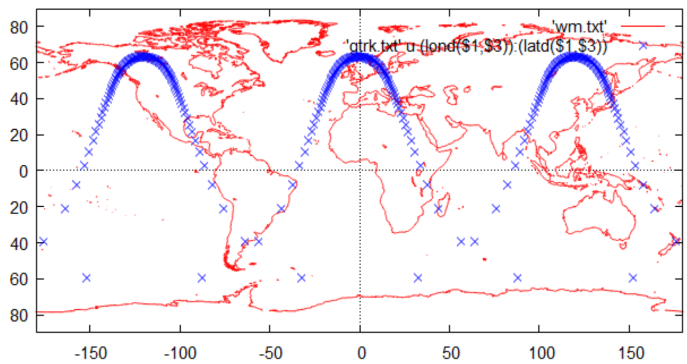
Amb xc = 0.6

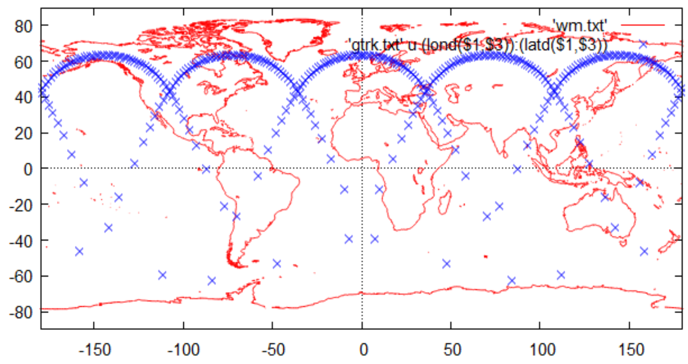


Amb xc = 0.4



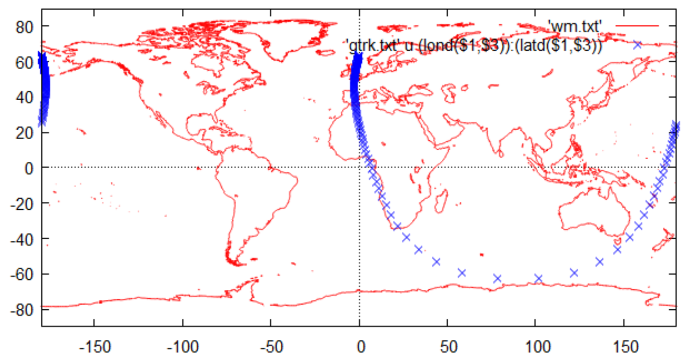
Si es canvia el període de l'òrbita, **T**, posant T=dias/3 o T=dias/5 veiem que l'òrbita canvia completament i ja no s'assembla a l'òrbita Molniya.





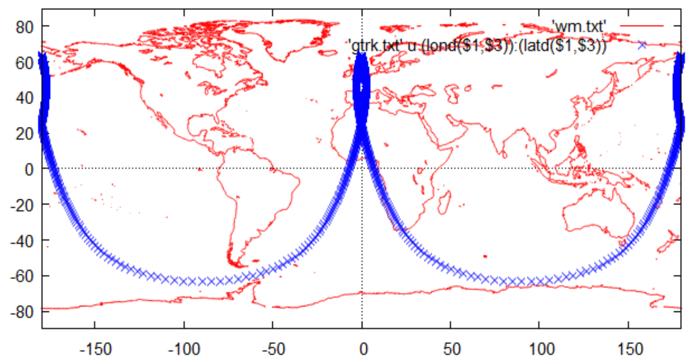
Degut a que les òrbites aconseguides s'allunyen de l'objectiu, es descarta l'opció de canviar el període de l'òrbita.

Canviant el temps que dura la simulació, **tf**, per tf=dias/dos, la representació mostra només la meitat de l'òrbita, de manera que canviar aquest paràmetre serveix per saber quin recorregut ha fet el satèl·lit en el temps tf.



La modificació d'aquest paràmetre no permet l'espionatge de noves zones, de manera que es descarta l'opció de canviar el temps de la simulació.

Al canviar la traça que es representa, **nt**, aconseguim més precisió a l'hora de veure per on passa el satèl·lit i determinar quines zones es poden espiar. A continuació es mostra l'òrbita de Molniya amb nt = 3000.



6.0.2 Conclusions

Per tal d'assolir l'objectiu del treball i determinar zones que es poden espiar amb les òrbites desitjades, s'ha de modificar el paràmetres xc i augmentar nt per guanyar precisió.

Es treballarà amb nt = 3000.

Les zones que es poden espiar són:

- Amb xc = 0.74104: Mali, Argèlia i una petita part d'Europa central.

- Amb xc = 0.0: Algunes zones de Rússia, Aràbia Saudí, el Japó i zones del centre d'Austràlia.

- Amb xc = 0.6: Mali, Argèlia, Nigèria, Alemanya i algunes zones d'Austràlia

- Amb xc = 0.4: Una petita part d'Alaska i d'Austràlia, Polonia, Rumania, zones d'Europa del nord i zones d'Àfrica com Libia, la República Democràtica del Congo i Zàmbia.

7 Bibliografia

Apunts de Càlcul Numèric, Josep Maria Mondelo
pr02.pf : CÀLCUL NUMÈRIC GMATCAD, Pràctica 2