

Memòria de pràctiques: Representació de traces (ground tracks) d'òrbites de satèl·lits artificials

Marc Roig Oliva

Índex

1	Motivació	2
2	Implementació del treball	2
3	Manual del software	3
3.1	bisnwt	3
3.2	kplt2nu	3
4	Validació del software	4
4.1	Trobar arrels amb "bisnwt"	4
4.2	Càlcul de les posicions del satèl·lit amb "kplt2nu"	4
5	Representació de traces amb Gnuplot	5
6	Top secret	6

1 Motivació

La AEC demana una eina que permeti determinar la posició d'un satèl·lit al seu pla orbital. Es realitzara el càlcul mitjançant la resolució numèrica de l'equació de Kepler¹, que combinarà la robustesa del mètode de bisecció i la velocitat del mètode de Newton.

2 Implementació del treball

La AEC requereix un programa principal: **kplt2nu** i una llibreria **bisnwt** amb la rutina per trobar zeros de funcions per separat

- **bisnwt**: Aquesta rutina combina els mètodes de la bisecció i de Newton per trobar zeros de funcions de la següent manera:
 1. El programa fa bisecció mentre que la longitud de l'interval sigui major a una certa δ seleccionada per l'usuari.
 2. Si δ es menor o igual que la tolerància desitjada la rutina guarda el punt mig de l'interval com a resultat, sino s'aplica el mètode de Newton prenent el punt mig de l'interval com a x_0 .
 3. Si Newton convergeix guardem el resultat, i si no convergeix tornem a començar amb bisecció pero prenem l'últim interval de la última bisecció com a interval inicial.
- **kplt2nu**: Aquesta utilitat calcula a partir dels arguments rebuts de la anterior rutina les anomalies mitjana, excèntrica i vertadera així com l'instant de temps en el que es fa la mesura. La utilitat imprimeix per consola la posició del satèl·lit en cada instant de temps

Tant el programa principal com la llibreria **bisnwt** utilitzen les llibreries pre-programades de C:

- `stdio.h`
- `stdbool.h`
- `math.h`

¹ $M = E - e \sin e$

3 Manual del software

3.1 bisnwt

Consta de la rutina **bisnwt**

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

Paràmetres:

- a (double): Límit inferior de l'interval de bisecció.
- b (double): Límit superior de l'interval de bisecció.
- *arr (double): Resultat numèric de l'arrel.
- *dlt (double): Longitud mínima de l'interval de bisecció.
- tol (double): Tolerància de convergència pel mètode de Newton.
- maxit (int): Nombre màxim d'iteracions per al mètode de Newton.
- *f (double, void*): Funció sobre la que s'aplica la rutina bisnwt.
- *df (double, void*): Derivada de la funció sobre la que s'aplica bisnwt.
- *prm: Vector amb els paràmetres de f i df.

3.2 kplt2nu

És una utilitat que rep 5 arguments:

- e (double): Excèntricitat de l'òrbita
- T (double): Període de l'òrbita
- M0 (double): Anomalia mitjana a l'inici de la simulació
- tf (double): Durada de la simulació en segons
- nt (int): Nombre de punts de la simulació

Aquesta utilitat imprimeix per pantalla l'instant de temps, l'anomalia mitjana i l'anomalia vertadera.

4 Validació del software

4.1 Trobar arrels amb "binswt"

La validació de la rutina de trobar arrels consta de trobar el zero de la funció $f(x) = e^x - 2$ amb diferents valors de δ

a	b	dlt	tol	maxit	x	iters
-9	1	10	10^{-12}	10	0.6931471805599453	6
-9	1	2.5	10^{-12}	10	0.6931471805599453	7
-9	1	0.01	10^{-12}	10	0.6931471805599453	4

Podem observar que la millor δ és 0.01 ja que es la que convergeix a la solució en el menor número de iteracions, els altres dos valors de δ requereixen un nombre d'iteracions molt semblant encara ser valors molt diferents.

Una altre cosa a notar es que a aquest test la llargada de l'interval mai es major que δ per tant no s'aplica el mètode de bisecció. Fent una ampliació de la llargada de l'interval recollim les següents dades:

a	b	dlt	tol	maxit	x	iters
-9	2	10	10^{-12}	10	0.6931471805599453	8
-9	2	2.5	10^{-12}	10	0.6931471805599453	5
-9	2	0.01	10^{-12}	10	0.6931471805599453	3

4.2 Càlcul de les posicions del satèl·lit amb "kplt2nu"

Per validar la rutina "kplt2nu" es fa una crida al programa de la forma:

```
./kplt2nu 0.74105=e 43081.95859068188=T 3.141592653589793=M0 \  
86163.91718136377=tf 333=nt > orb.txt
```

i es fa la comparació amb els resultats esperats.

5 Representació de traces amb Gnuplot

Per representar les "ground tracks" es fa ús del script **dibgte.cnu** que executa el programa ja compilat i redirigeix la sortida de les dades a un fitxer "gtrk.txt", prèviament però cal definir certes constants i paràmetres. Per fer això s'utilitza el script **ctants.gnu** que defineix el radi de la terra, el paràmetre gravitatori de la Terra, la freqüència angular en rad/s, el dia sideri en segons i l'ipert.

La crida dins de Gnuplot seria de la forma:

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

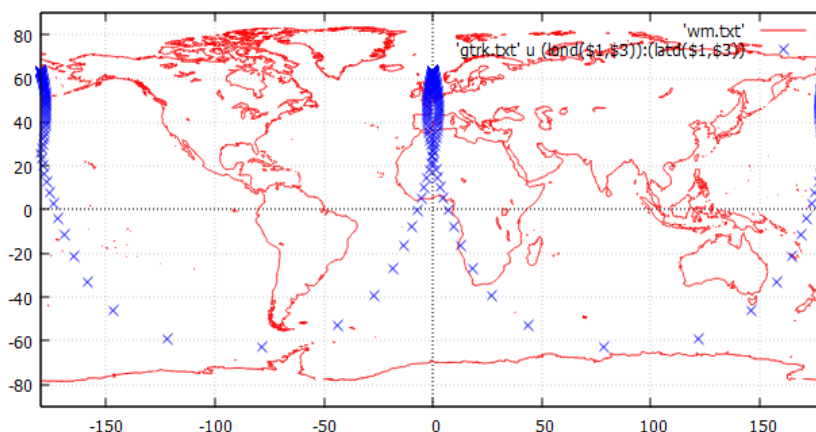


Figure 1: Representació de "Ground Tracks"

Per comprobar que el satèl·lit no arriba a trespasar la atmòsfera terrestre és necessari mostrar la gràfica de la elevació del satèl·lit amb la ajuda de Gnuplot, de la forma:

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

