

# Pràctica 2: Memòria

Biel González Garriga

5 de juny de 2021

# 1 Motivació

El Departament de Control de Missió de l'AEC ens demana una eina que li permeti determinar la posició d'un satèl·lit artificial en el seu pla orbital.

## 1.1 Objectius

- Desenvolupar la rutina bisnwt per resoldre numèricament l'equació de Kepler i el codi client que permeti executar el codi per determinar l'òrbita.
- Desenvolupar la utilitat secreta per representar orbites amb gnuplot.

# 2 Implementació

## 2.1 Bisnwt

La implementació de la rutina bisnwt s'ha dut a terme mitjançant l'aplicació dels algorismes de la bisecció i del mètode de newton. Al combinar els dos algorismes assolim una robustesa i velocitat que serà necessària per a que el client pugui fer un correcte seguiment dels satèl·lits.

## 2.2 Kplt2nu

Kplt2nu.c és el que s'encarrega de rebre els arguments. A partir dels arguments el programa calcularà a cada iteració:

- L'instant de temps en el que es fa la mesura.
- L'anomalia mitjana.
- l'anomalia excèntrica.
- L'anomalia vertadera.

A cada iteració kplt2nu passarà per la línia d'ordres la posició del satèl·lit en aquell instant de temps.

## 3 Manual d'ús

### 3.1 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);
```

- a: cota inferior de l'interval de la bisecció.
- b: cota superior de l'interval de la bisecció.
- \*dlt: amplada mínima del interval de bisecció.
- tol: tolerància per Newton.
- maxit: màxim d'iterats de Newton.
- (\*f)(double,void\*): funció sobre la que farem bisnwt.
- (\*df)(double,void\*): derivada de la funció sobre la que farem bisnwt.
- \*prm: paràmetres de les funcions f i df.
- Valors generats:
  - \*arr: arrel de l'equació trobada numèricament.
- Valors retornats:
  - -1 en el cas que dlt sigui més petita o igual que tol abans d'entrar a fer Newton.
  - El nombre d'iterats en el cas que Newton convergeixi.

## 3.2 Kplt2nu

És necessari estar situat al directori on estigui l'arxiu `kplt2nu.c` `gcc -o kplt2nu -g -Wall kplt2nu.c bisnwt.c -lm`  
`./kplt2nu e T M0 tf nt`

- e: excentricitat de l'òrbita.
- T: període de l'òrbita.
- M0: anomalia mitjana a l'inici de la simulació.
- tf: durada de la simulació en segons.
- nt: nombre de punts de la simulació.

A mesura que es realitzin les iteracions, s'aniran imprimint a la línia de comandes els resultat de la següent manera:

```
InstantDeTemps AnomaliaMitjana AnomaliaVertadera
```

## 3.3 Gnuplot

Les comandes de gnuplot s'han d'executar després de compilar el codi principal, que estarà a la mateixa carpeta que `ctants.gnu`, `dibgte.gnu` i `wm.txt`. Mitjançant comandes s'ha de donar valor a les diferents variables:

- lonesp: longitud del lloc que espia.
- dlapo: correcció per baixar la latitud de l'apogeu.
- xc: excentricitat.
- T: període de l'òrbita.
- i: inclinació de l'òrbita.
- tf: temps de la simulació.
- nt: nombre de punts.

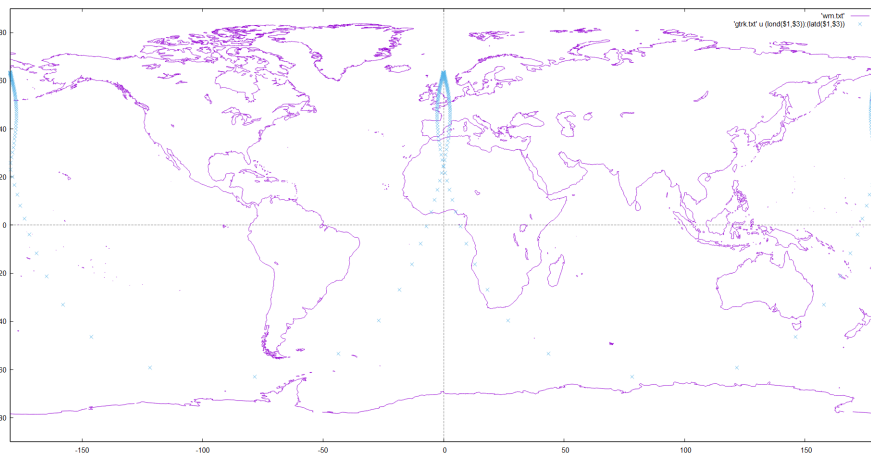
També es pot fer que gnuplot representi un gràfic amb la distància al centre de la terra en funció al temps amb la comanda:

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

### 3.3.1 Exemple d'execució amb gnuplot

```
gcc -o kplt2nu -g -Wall kplt2nu.c bisnwt.c -lm
load 'ctants.gnu'
lonesp=0
dlapo=0
xc=0.74105
T=dias/2
i=63.4*(pi/180)
tf=dias
nt=333
load 'dibgte.gnu'
```

El resultat és:



La gràfica de la distància és:

