Tutorial 7

Differential positioning with code measurements

Professors Dr. Jaume Sanz Subirana, Dr. J. M. Juan Zornoza and Dr. Adrià Rovira Garcia

Research group of Astronomy & Geomatics (gAGE)
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain

gAGE/UPC
Research group of Astronomy & Geomatics
Technical University of Catalonia

Aim of this tutorial

- This tutorial is devoted to analysing and assessing the differential positioning with code pseudorange measurements. Four different permanent receivers and two baselines 280km and 51km are considered.

- The atmospheric effects on differential positioning are the subject of the first session. The differential tropospheric and ionospheric delays are analysed for the two baselines considered and the absolute and differential user solution are computed and assessed.

- The ephemeris error on differential positioning is the subject of the second session. A 2000 metres of Along-Track error is simulated by manipulating the broadcast message, and the impact of this error on range and user domains is assessed for absolute and differential positioning. A theoretical expression to predict the range error on a given baseline is also verified in the last part of this session.

- Detailed **guidelines** for **self-learning students** are provided in this tutorial and in its associated ***notepad*** text file.

- **All software tools** (including ***gLAB***) and associated files for the laboratory session are included in the USB stick associated with this tutorial.

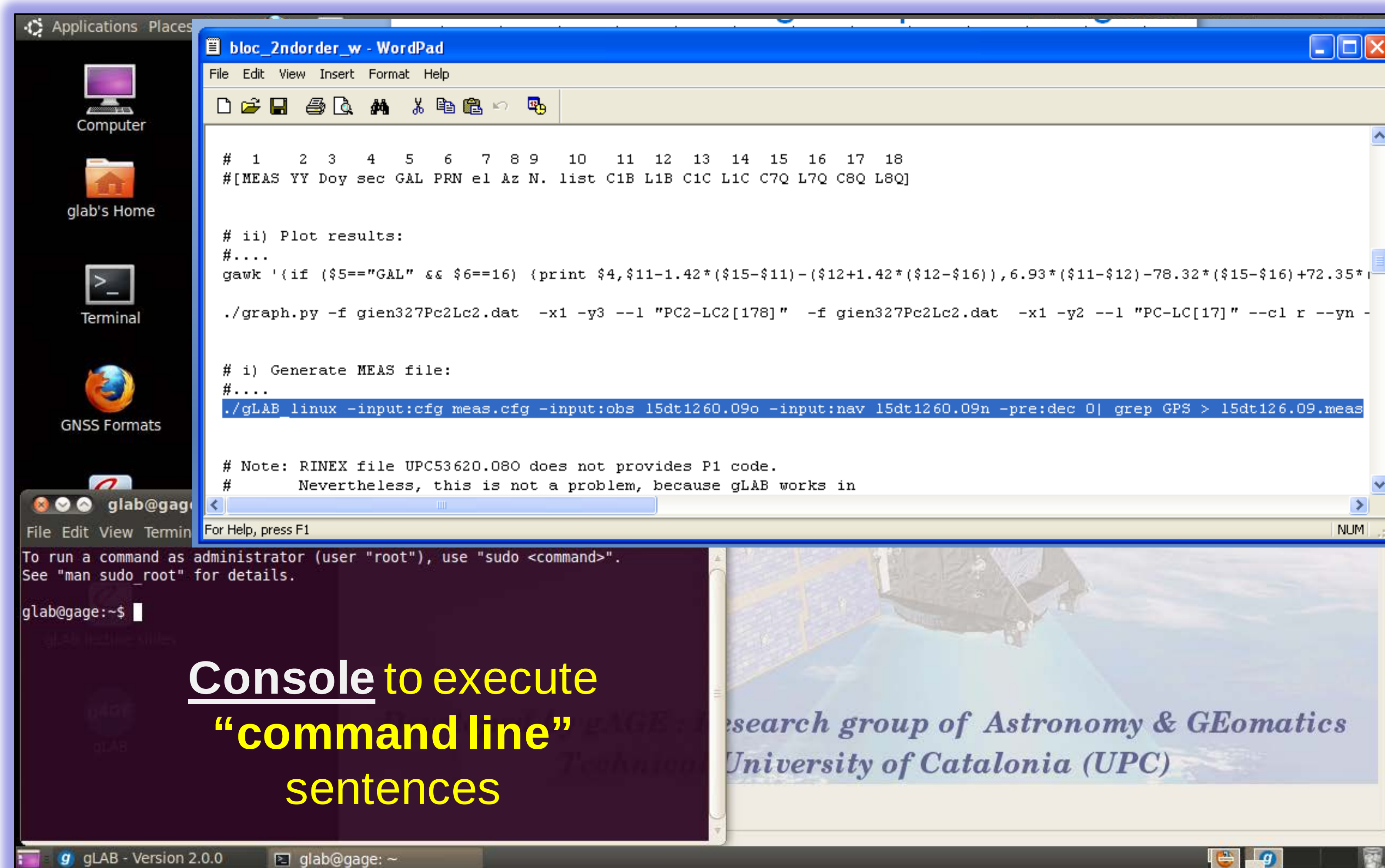
OVERVIEW

- **Introduction: gLAB processing in command line**

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

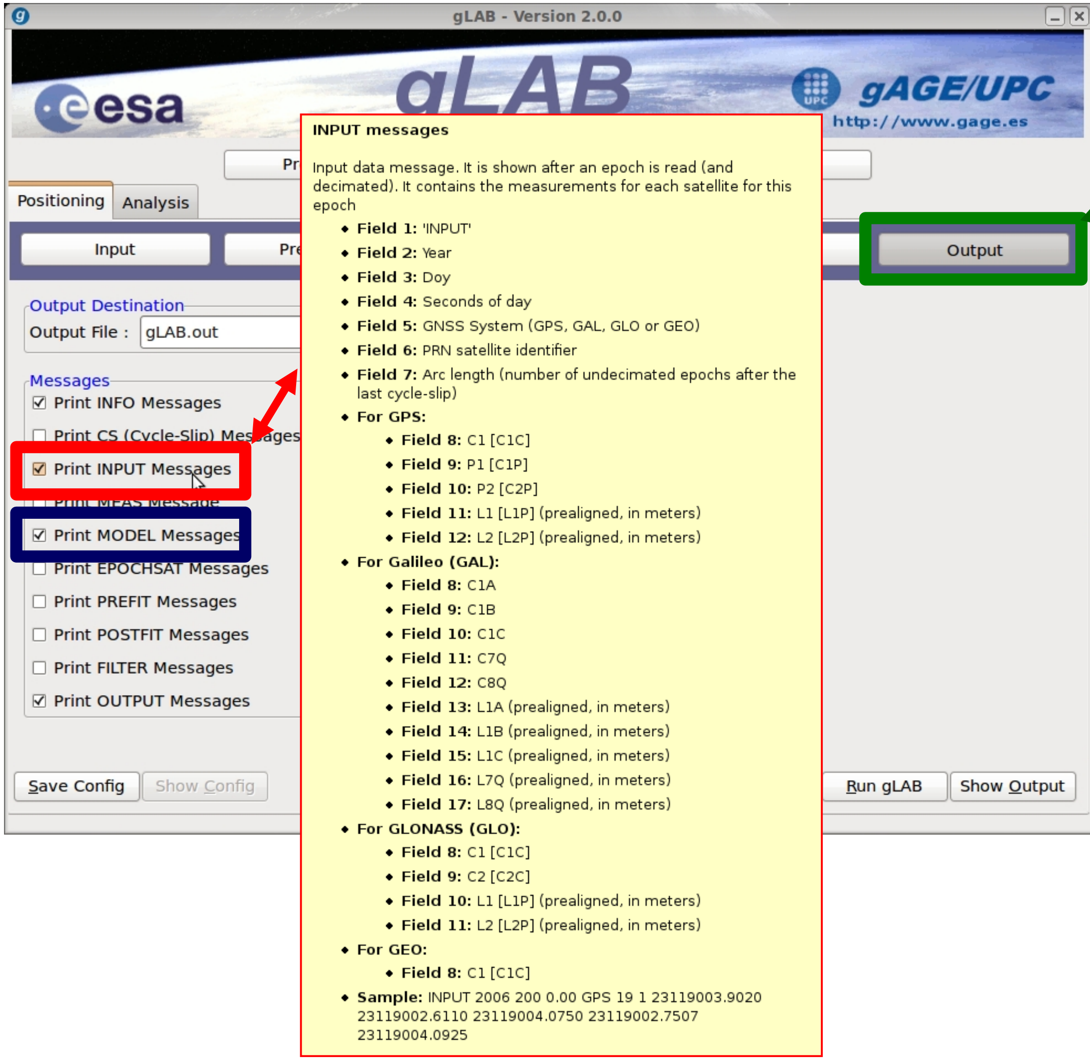
- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

gLAB processing in command line



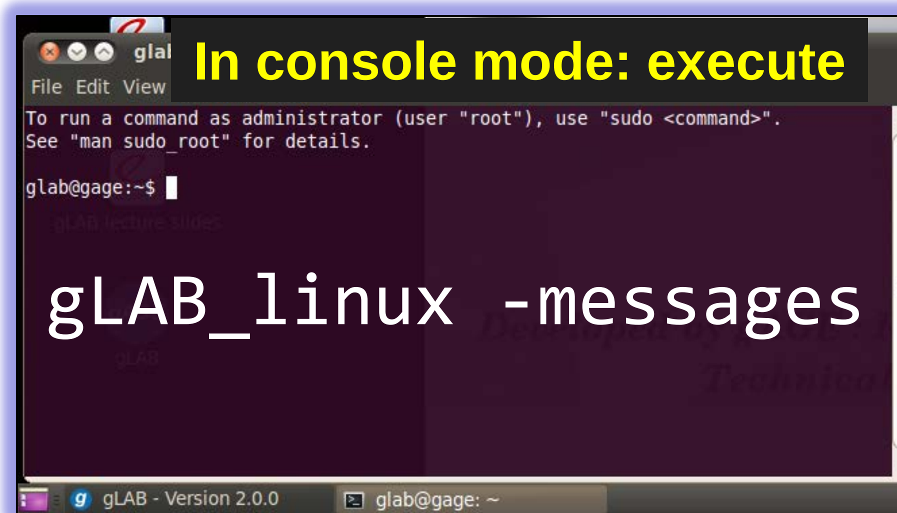
A "**notepad**" with the command line sentence is provided to facilitate the sentence writing: just **"copy" and " paste" from notepad to the working terminal.**

gLAB processing in command line



The different messages provided by gLAB and its content can be found in the [OUTPUT] section.

By placing the mouse on a given message name, a tooltip appears describing the different fields.



OVERVIEW

- **Introduction:** gLAB processing in command line

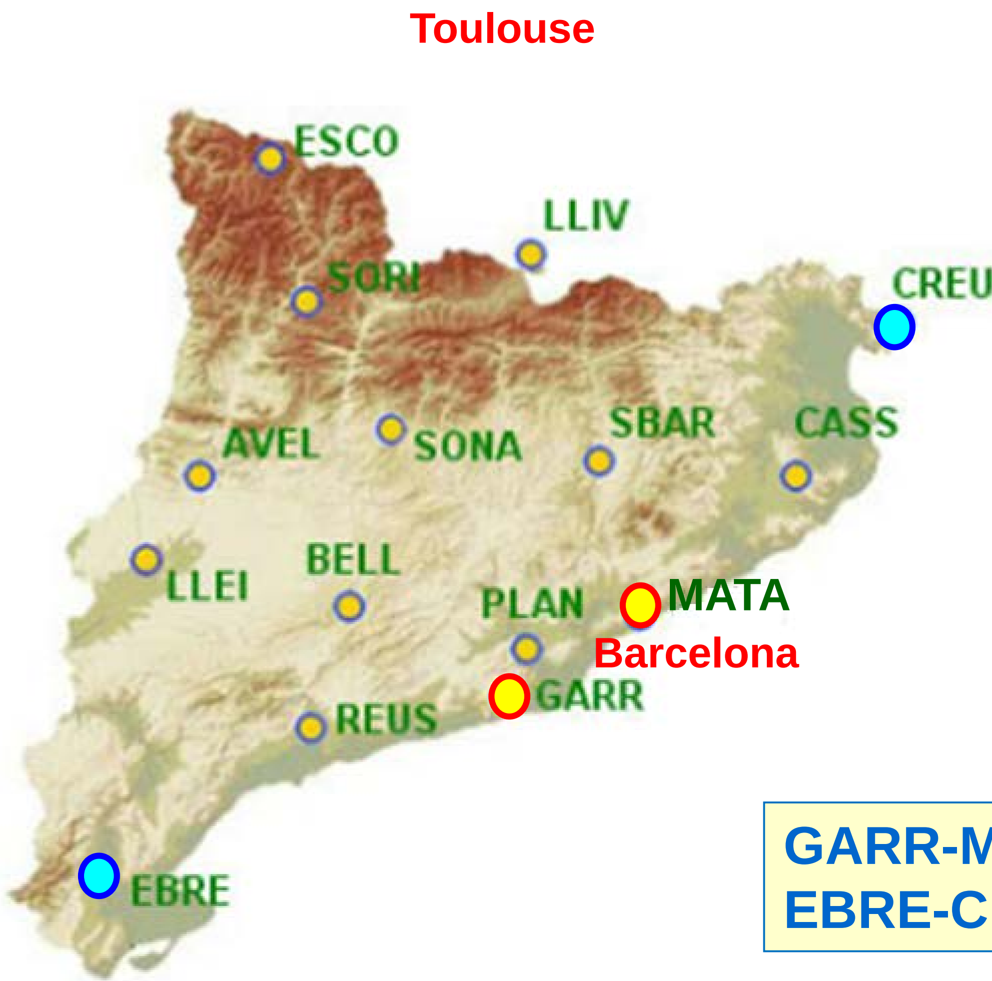
- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

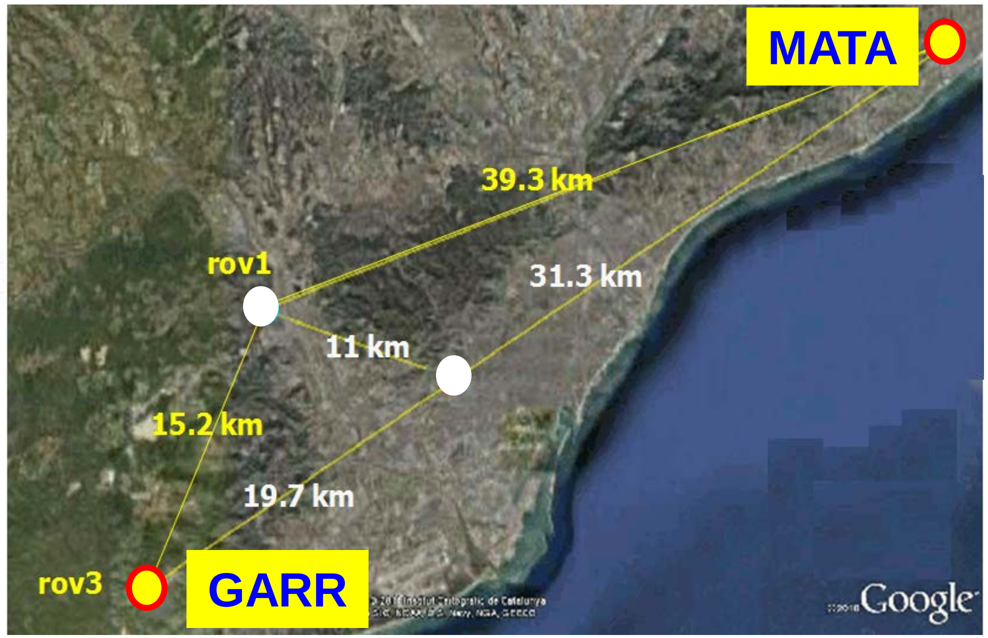
- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

Session A

Atmospheric effects
on differential positioning
with code measurements

Session A: Atmospheric effects





GARR-MATA: 51 km
EBRE-CREU: 288 km

Session A: Atmospheric effects

Data sets:

Measurement files:

CREU0770.10o, EBRE0770.10o, GARR0770.10o, MATA0770.10o

Orbit and clocks files: **brdc0770.10n.**

```
Receiver coordinates:

MATA 4776835.5870 202618.9947 4207574.6304 41.539929530 2.428858625 123.6209
GARR 4796983.5767 160309.1177 4187340.3773 41.292941528 1.914040080 634.6040
EBRE 4833520.1197  41537.2015 4147461.6263 40.820888849 0.492363266 107.8284
CREU 4715420.3054 273177.8809 4271946.7957 42.318843076 3.315603563 133.4780
```

Note: the receivers were not moving (**static receivers**) during the data collection.

OVERVIEW

- **Introduction:** gLAB processing in command line

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Model Components computation

- The script "**`ObsFile1.scr`**" generates a data file "**`STA.obs`**" with the following content

```
  1    2    3    4    5   6  7  8   9    10   11   12    13   14
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion Elev Azim Prefit]
```

- Run this script for EBRE and CREU receivers:

```
ObsFile1.scr EBRE0770.10o brdc0770.10n
ObsFile1.scr CREU0770.10o brdc0770.10n
```

- Generate the navigation equations system for absolute positioning for each receiver and compute the user solution (see next two slides).

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Justify that the next sentence builds the navigation equations system for absolute positioning:

[Prefit]=[Los_k 1]*[dx]

See file content **in previous slide**

```
cat EBRE.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
              $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> EBRE.mod
```

$$\begin{bmatrix} \mathrm{Pref}^{1} \\ \mathrm{Pref}^{2} \\ \cdots \\ \mathrm{Pref}^{n} \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^{1}\right)^{T} & 1 \\ -\left(\hat{\boldsymbol{\rho}}^{2}\right)^{T} & 1 \\ \cdots & \\ -\left(\hat{\boldsymbol{\rho}}^{n}\right)^{T} & 1 \end{bmatrix} \mathbf{dx}$$

time	[Pref]	[	Los_k		1]
	-------	------------	------------	------------	---
30.00	-3.3762	0.3398	-0.1028	0.0714	1
30.00	-7.1131	0.1725	0.5972	0.0691	1
30.00	4.3881	-0.6374	0.0227	0.2725	1

$$\hat{\boldsymbol{\rho}}^{k} \equiv \left[\cos(El_k)\sin(Az_k), \quad \cos(El_k)\cos(Az_k), \quad \sin(El_k)\right]$$

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

The program **kalman.f** implements the kalman filter for code positioning (see source code).

The INPUT file is the file **EBRE.mod** generated in the previous slide.
The OUTPUT is the **user solution** file:

```
     1    2   3   4    5
 [time dE dN dU, dt]
```

The filter configuration is done by the namelist **kalman.nml**.

- The namelist **kalman_wn.nml** configures the filter to process the coordinates and clock as white nose (i.e. **Kinematic** solution).
- The namelist **kalman_ct.nml** configures the filter to process the coordinates as constants and clock as white nose (i.e. **static** solution).

The program is executed as:

```
cp kalman.nml_wn kalman.nml

cat EBRE.mod | kalman > EBRE.pos
```

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Using the files **`EBRE.obs`** and **`CREU.obs`**, follow the next steps:

1. Build the navigation equations system for each receiver, for absolute positioning.
2. Compute the user solutions in kinematic mode.
3. Plot the results and compare the positioning errors.

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

1.- Building the navigation equations system for absolute positioning:

```
cat EBRE.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
           $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> EBRE.mod
```

2.- Computing the user solution for absolute positioning:

```
cp kalman.nml_wn kalman.nml

cat EBRE.mod | kalman > EBRE.pos
```

3.- Plotting results:

```
graph.py  -f EBRE.pos -x1 -y2 -s.- -l "North error"
          -f EBRE.pos -x1 -y3 -s.- -l "East error"
          -f EBRE.pos -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
          -t "EBRE: Standard Point Positioning"
```

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

1.- Building the navigation equations system for absolute positioning:

```
cat CREU.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
   printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
           $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> CREU.mod
```

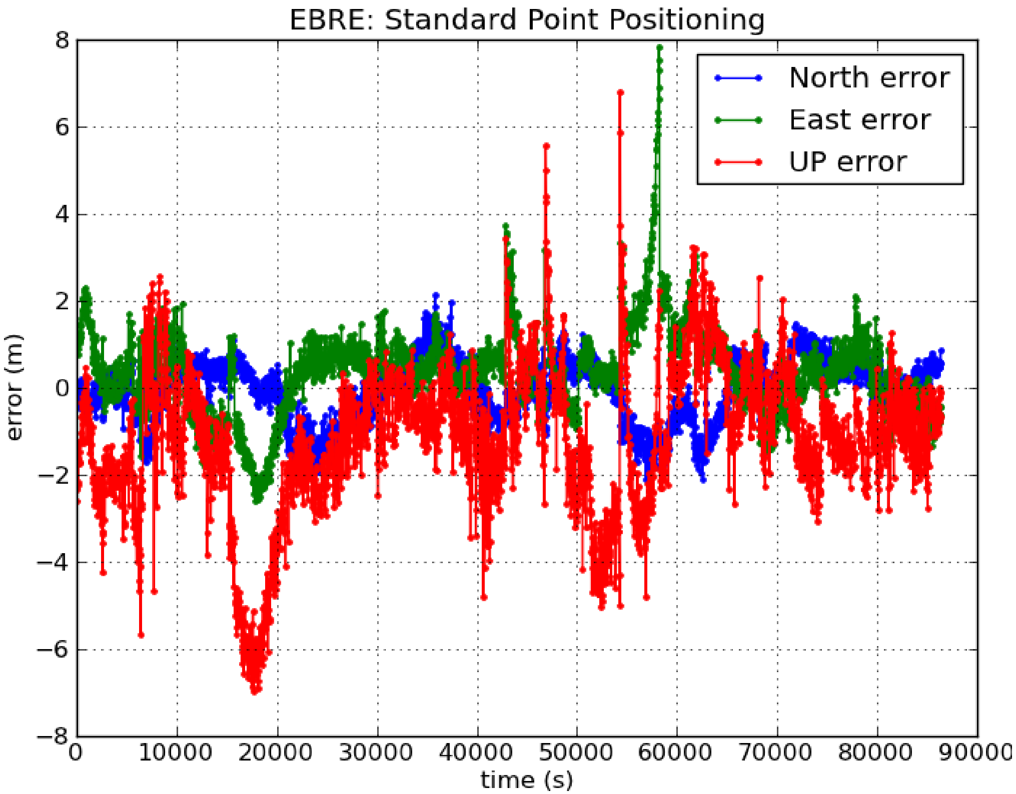
2.- Computing the user solution for absolute positioning:

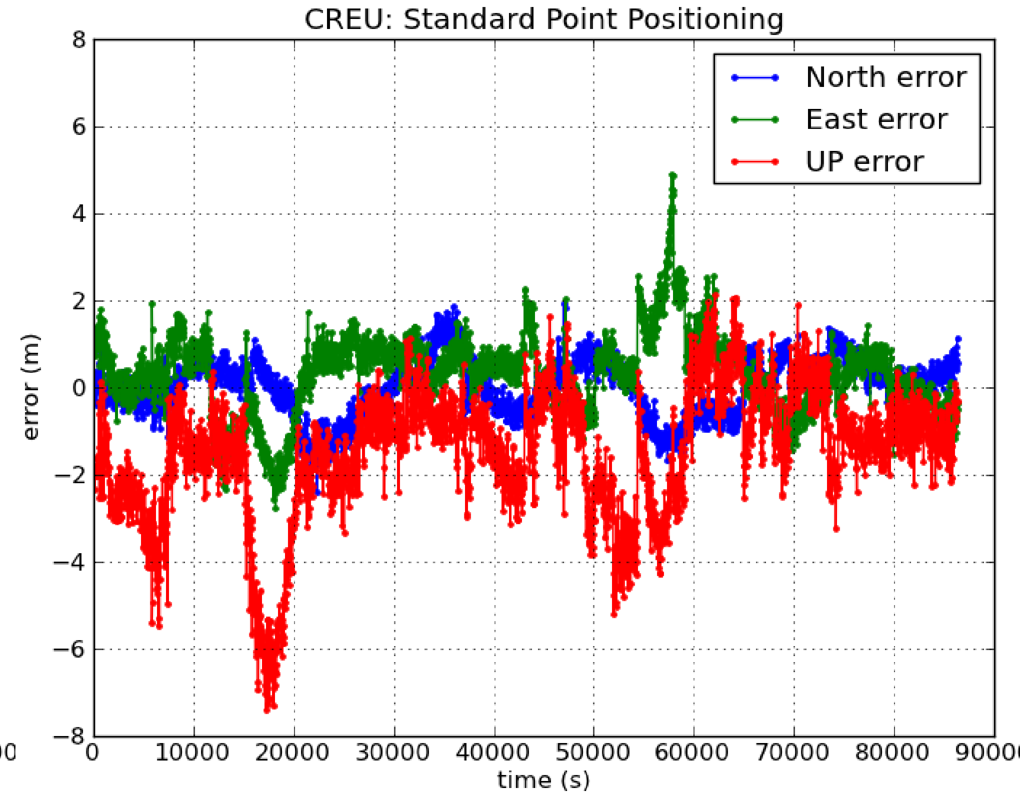
```
cp kalman.nml_wn kalman.nml

cat CREU.mod | kalman > CREU.pos
```

3.- Plotting results:

```
graph.py  -f CREU.pos -x1 -y2 -s.- -l "North error"
          -f CREU.pos -x1 -y3 -s.- -l "East error"
          -f CREU.pos -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
          -t "CREU: Standard Point Positioning"
```

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)





Questions:

•What is the expected accuracy of the computed coordinates (in absolute positioning)?

•Why are the patterns seen for the receivers EBRE and CREU are so similar?

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

The script **Dobs.scr** computes the single difference of measurements files **sta1.obs** (reference station) and **sta2.obs** (user).

It can be executed by the sentence:

```
Dobs.scr sta1.obs sta2.obs
```

The output file (**D_sta1_sta2.obs**) has the following content:

```
  1     2    3    4    5   6     7    8    9    10    11   12  13    14
[sta2 sat DoY sec DP1 DL1 DP2 DL2 DRho DTrop DIon El2 Az2 DPrefit]
```

Where: $D(o) = (o)_{sta2} -- (o)_{sta1}$

EL2 and AZ2 are the satellite azimuth and elevation from sta2.

In our case, executing:

```
Dobs.scr EBRE.obs CREU.obs
```

the OUTPUT file **D_EBRE_CREU.obs** is generated.

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Justify that the next sentence builds the navigation equations system for differential positioning of CREU (user) relative to EBRE (reference station):

[DPrefit]=[Los_k 1]*[dx]

Question: Is this combination equivalent to use **$14**? Why?

```
cat D_EBRE_CREU.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
$4,$5-$9-$10-$11 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> D_EBRE_CREU.mod
```

$$
\begin{bmatrix} DPref^1 \\ DPref^2 \\ \cdots \\ DPref^n \end{bmatrix} = \begin{bmatrix} -\left(\hat{\boldsymbol{\rho}}^1_{\mathrm{creu}}\right)^T & 1 \\ -\left(\hat{\boldsymbol{\rho}}^2_{\mathrm{creu}}\right)^T & 1 \\ \cdots & \\ -\left(\hat{\boldsymbol{\rho}}^n_{\mathrm{creu}}\right)^T & 1 \end{bmatrix} \mathbf{dx}
$$

time	[DPref]	[	Los_k		1]
	-------	-----	-----	-----	-----
30.00	-3.3762	0.3398	-0.1028	0.0714	1
30.00	-7.1131	0.1725	0.5972	0.0691	1
30.00	4.3881	-0.6374	0.0227	0.2725	1

$$
\hat{\boldsymbol{\rho}}^k \equiv \left[\cos(El_k)\sin(Az_k), \quad \cos(El_k)\cos(Az_k), \quad \sin(El_k)\right]
$$

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Using the files **`EBRE.obs`** and **`CREU.obs`**, follow the next steps:

1. Compute the single differences of measurements and model components.
2. Build the navigation equations system for the differential positioning of CREU receiver (user) relative to EBRE (reference station).
3. Compute the user solution in kinematic mode and in static mode.
4. Plot the results and discuss the positioning error found.

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

1. Computing the single differences of measurements and model components.

```
Dobs.scr EBRE.obs CREU.obs
```

2.- Building the navigation equations system for differential positioning:

```
cat D_EBRE_CREU.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
 $4,$5-$9-$10-$11,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'> D_EBRE_CREU.mod
```

3.- Computing the user solution for differential positioning:

Kinematic:

```
cp kalman.nml_wn kalman.nml
cat D_EBRE_CREU.mod | kalman > EBRE_CREU.posK
```

Static:

```
cp kalman.nml_ct kalman.nml
cat D_EBRE_CREU.mod | kalman > EBRE_CREU.posS
```

A.1.3.1. Differential positioning. Full model

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

4.- Plotting results:

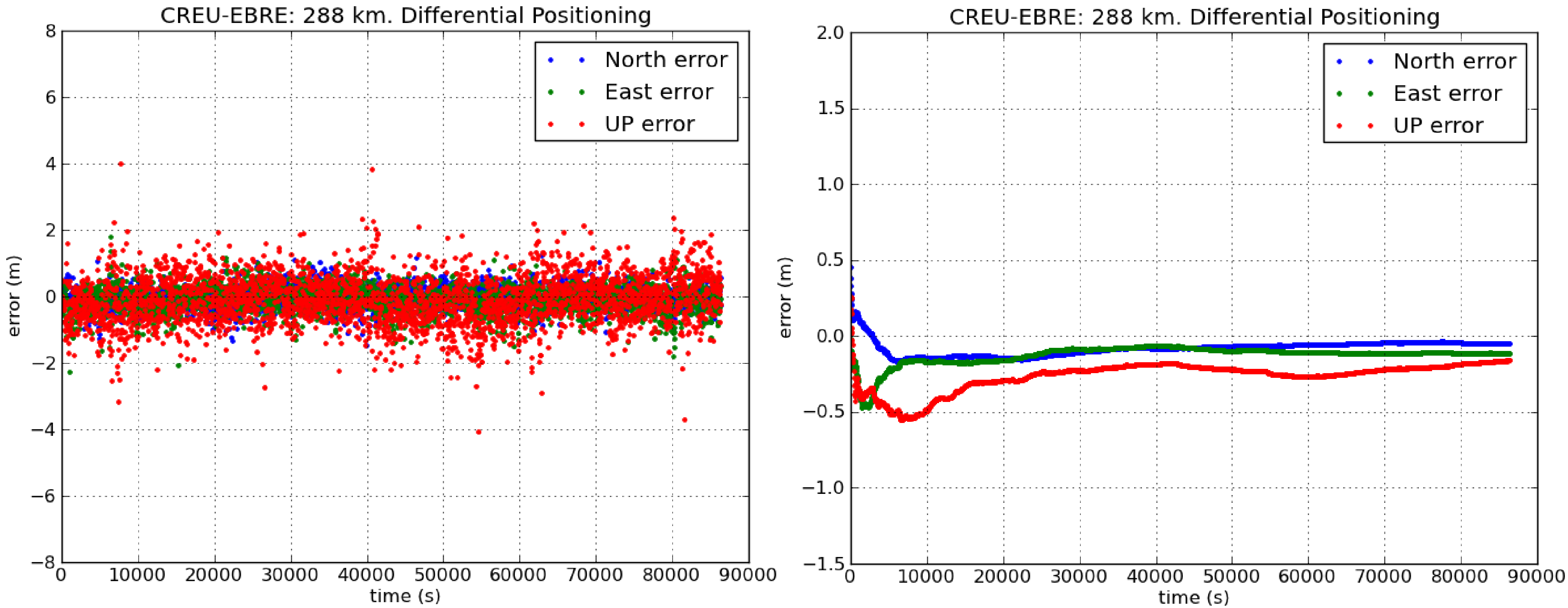
- kinematic mode:

```
graph.py -f EBRE_CREU.posK -x1 -y2 -s.  -l "North error"
         -f EBRE_CREU.posK -x1 -y3 -s.  -l "East error"
         -f EBRE_CREU.posK -x1 -y4 -s.  -l "UP error"
         --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
         -t "CREU-EBRE: 288 km: Differential Positioning"
```

- Static mode:

```
graph.py -f EBRE_CREU.posS -x1 -y2 -s.  -l "North error"
         -f EBRE_CREU.posS -x1 -y3 -s.  -l "East error"
         -f EBRE_CREU.posS -x1 -y4 -s.  -l "UP error"
         --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
         -t "CREU-EBRE: 288 km: Differential Positioning"
```

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)



Questions:

•Looking at these results, justify intuitively why the errors are reduced in differential positioning.

•Why is the error lower in static than in kinematic mode?

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Analyze the effect of the differential ionosphere on the differential positioning of CREU relative to EBRE station:

Follow the next steps:

1. Using file **`D_EBRE_CREU.obs`** plot the differential ionospheric correction (from Klobuchar model) between EBRE and CREU.

2. **Repeat** the previous process, **but without applying the ionospheric correction**. Compare the results

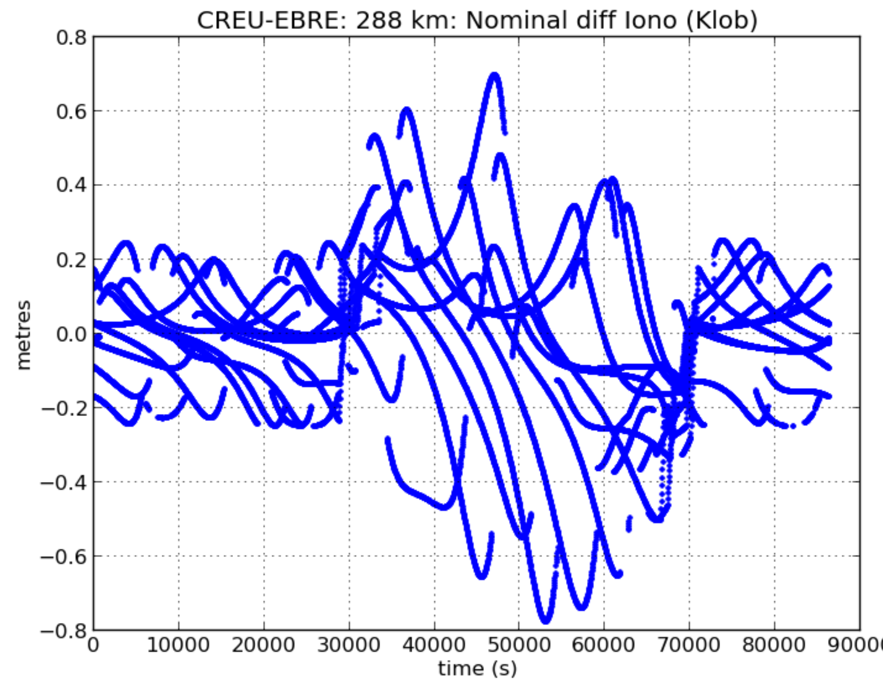
A.1.3.2. Differential positioning.
No ionospheric corrections

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Plotting the Klobuchar differential ionospheric correction:

```
graph.py -f D_EBRE_CREU.obs -x4 -y'11' --yn -0.8 --yx 0.8
         -t "CREU-EBRE: 288 km: Nominal diff Iono (Klob)"
```

Question:
Justify the pattern seen in the nominal (Klobuchar) ionospheric corrections.



A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Computing the differential solution, but without using ionospheric corrections:

1.- Building the navigation equations system for differential positioning, but without using the ionospheric corrections:

```
cat D_EBRE_CREU.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
   $4,$5-$9-$10,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'> D_EBRE_CREU.mod
```

2.- Computing the user solution for differential positioning:

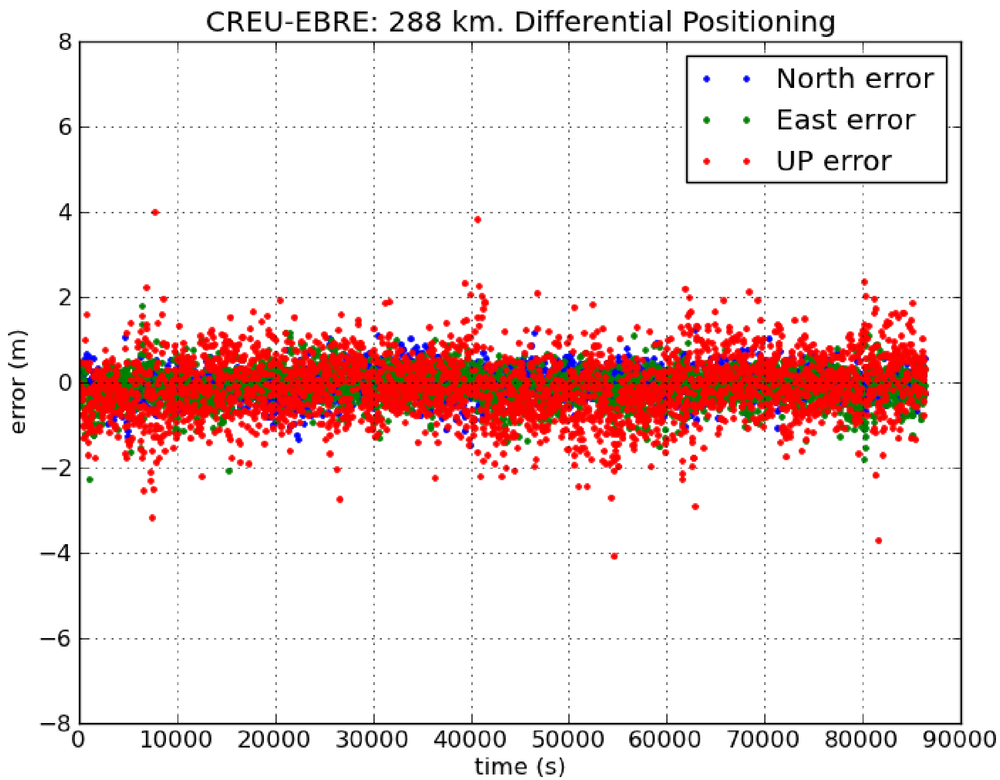
Kinematic:

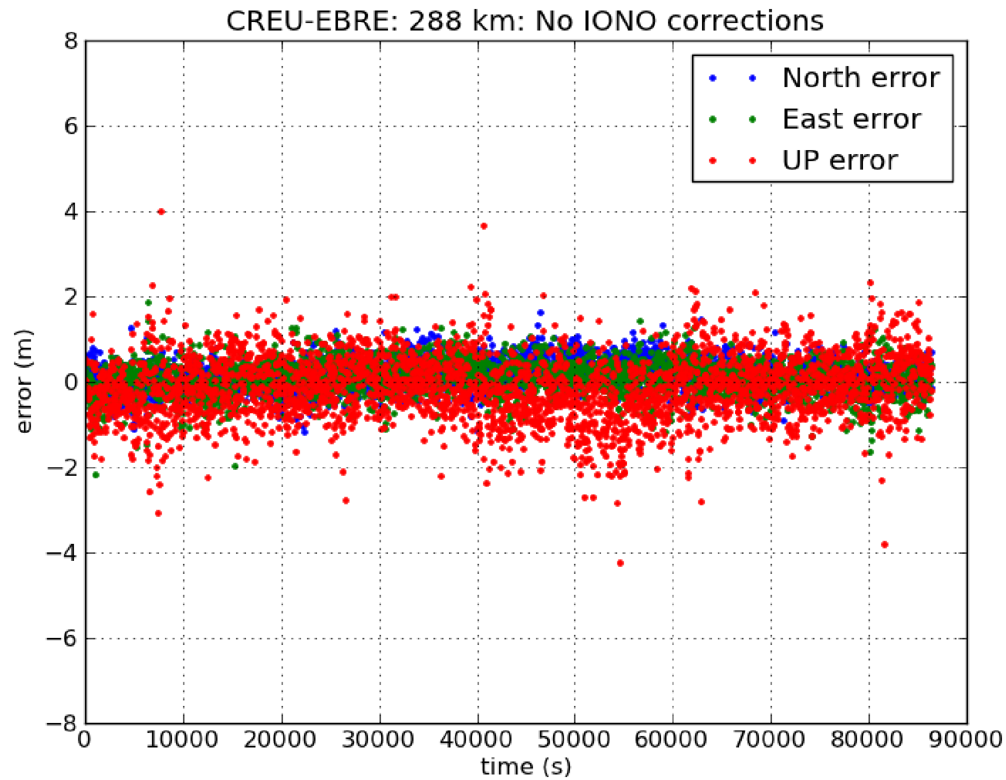
```
cp kalman.nml_wn kalman.nml
cat D_EBRE_CREU.mod | kalman > EBRE_CREU.posK
```

Static:

```
cp kalman.nml_ct kalman.nml
cat D_EBRE_CREU.mod | kalman > EBRE_CREU.posS
```

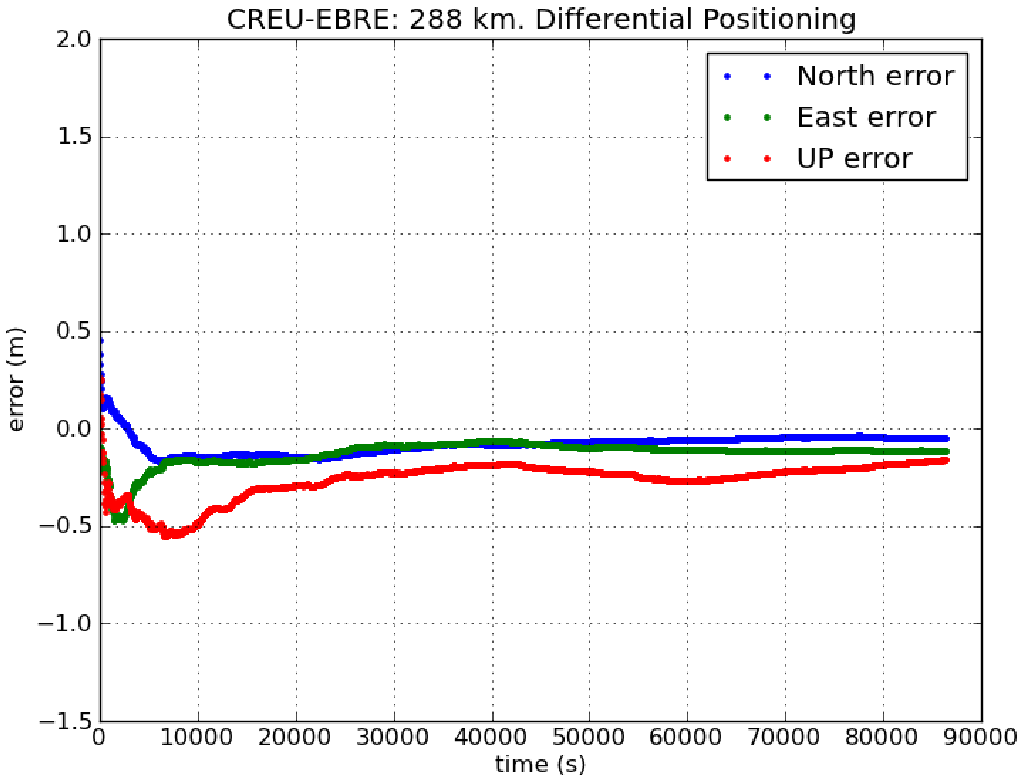
A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

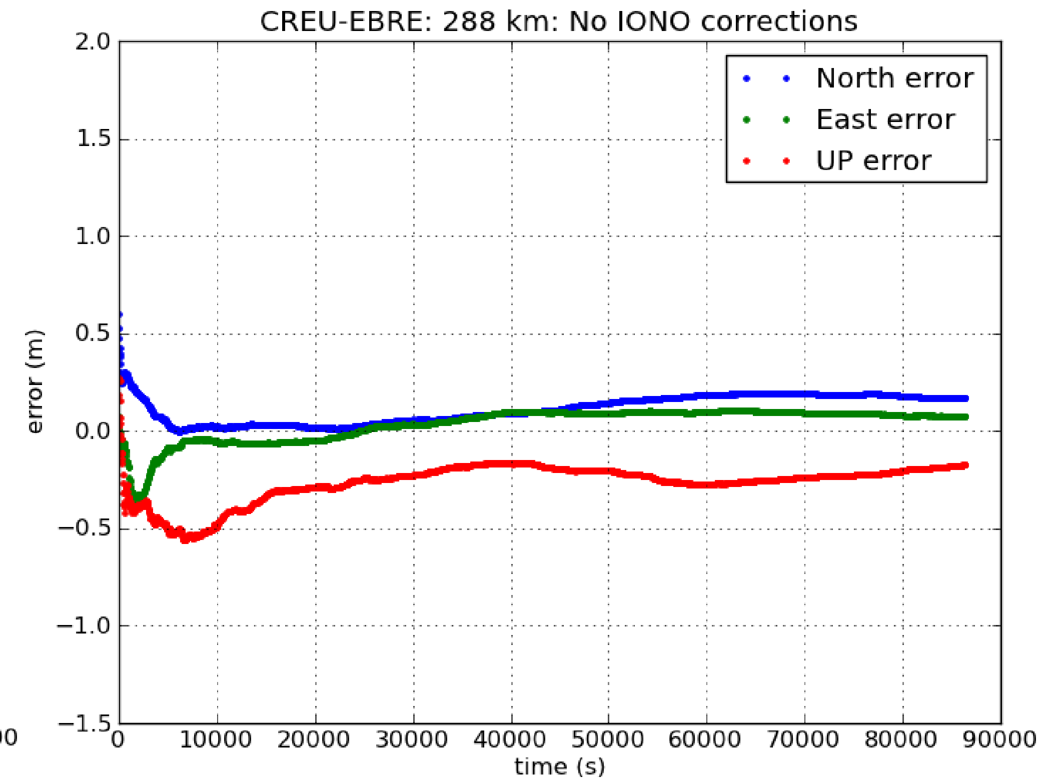




Question:
Taking into account the previous plot of the differential ionospheric corrections (slide #26), discuss the effect seen on the position domain.

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)





Question:
Discuss the pattern seen in the figure. Why is the effect of the differential ionospheric error is higher in these static positioning results?

A.1.3.2. Differential positioning. No ionospheric corrections

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Analyze the effect of the differential Troposphere on the differential positioning of CREU relative to EBRE station:

Follow the next steps:

1. Using file `D_EBRE_CREU.obs` plot the differential trpospheric correction (from nominal model) between EBRE and CREU.

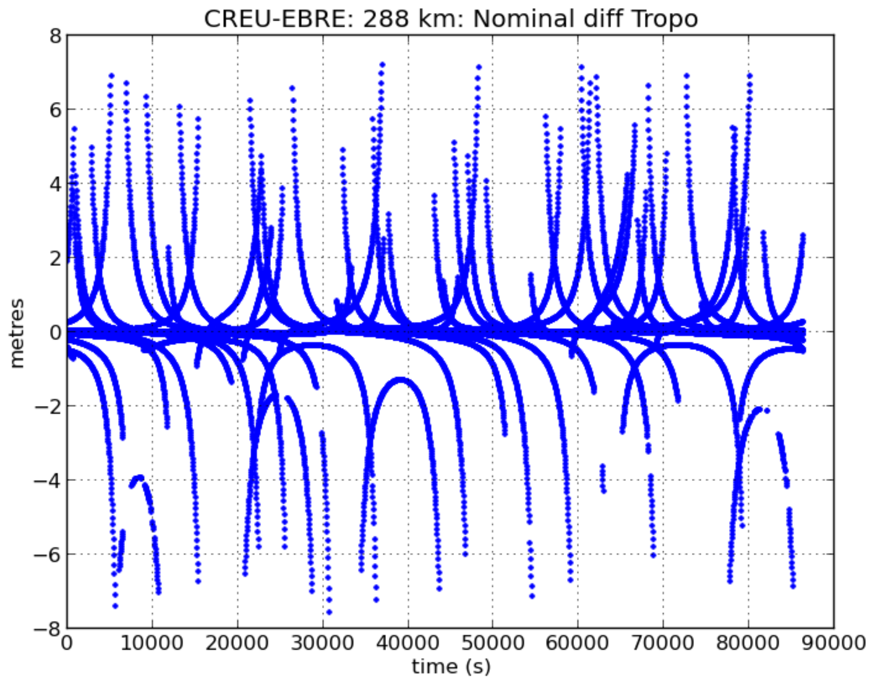
2. Repeat the previous process, **but without applying the tropospheric correction**. Compare the results

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Plotting the nominal differential tropospheric correction:

```
graph.py -f D_EBRE_CREU.obs -x4 -y'10' --yn -8 --yx 8
         -t "CREU-EBRE: 288 km: Nominal diff Tropo"
```

Question:
Justify the pattern seen in the nominal tropospheric corrections.



A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Computing the differential solution, but without using tropospheric corrections:

1.- Building the navigation equations system for differential positioning, but without using the tropospheric corrections:

```
cat D_EBRE_CREU.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
   $4,$5-$9-$11,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'> D_EBRE_CREU.mod
```

2.- Computing the user solution for differential positioning:

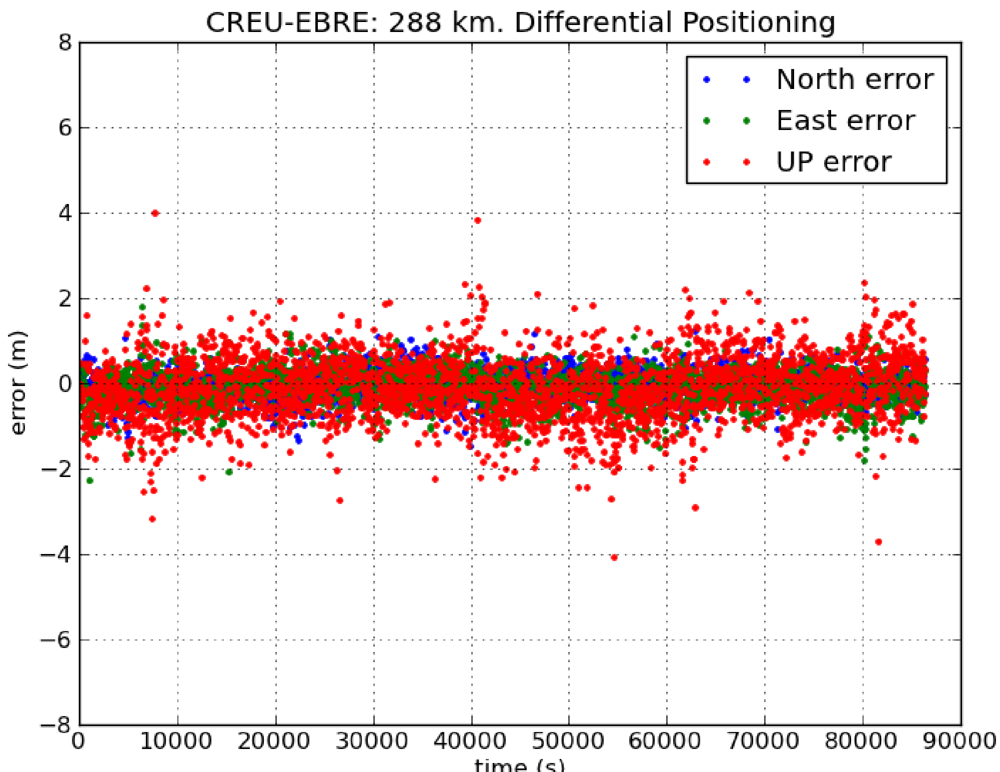
Kinematic:

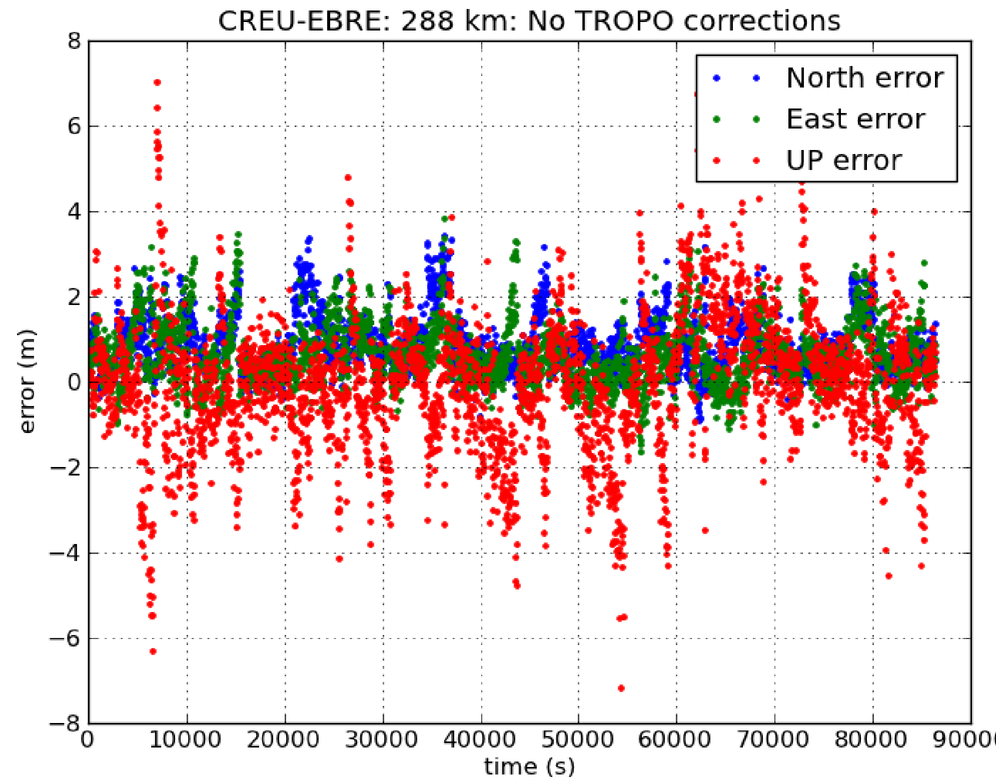
```
cp kalman.nml_wn kalman.nml
cat D_EBRE_CREU.mod | kalman > EBRE_CREU.posK
```

Static:

```
cp kalman.nml_ct kalman.nml
cat D_EBRE_CREU.mod | kalman > EBRE_CREU.posS
```

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

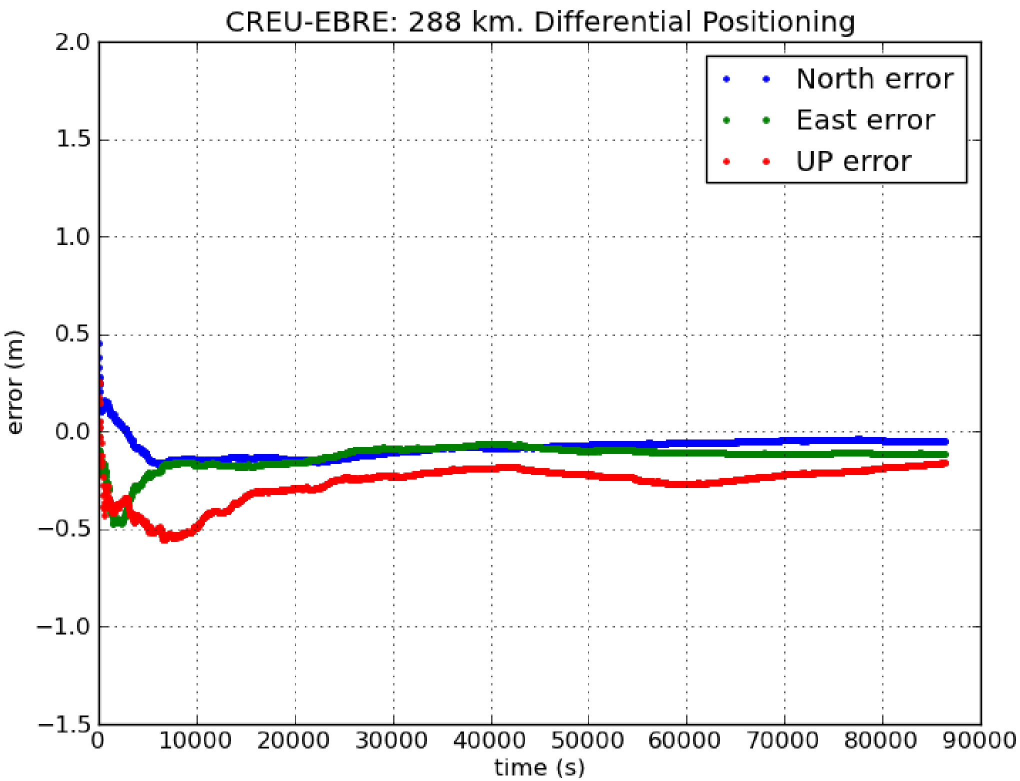


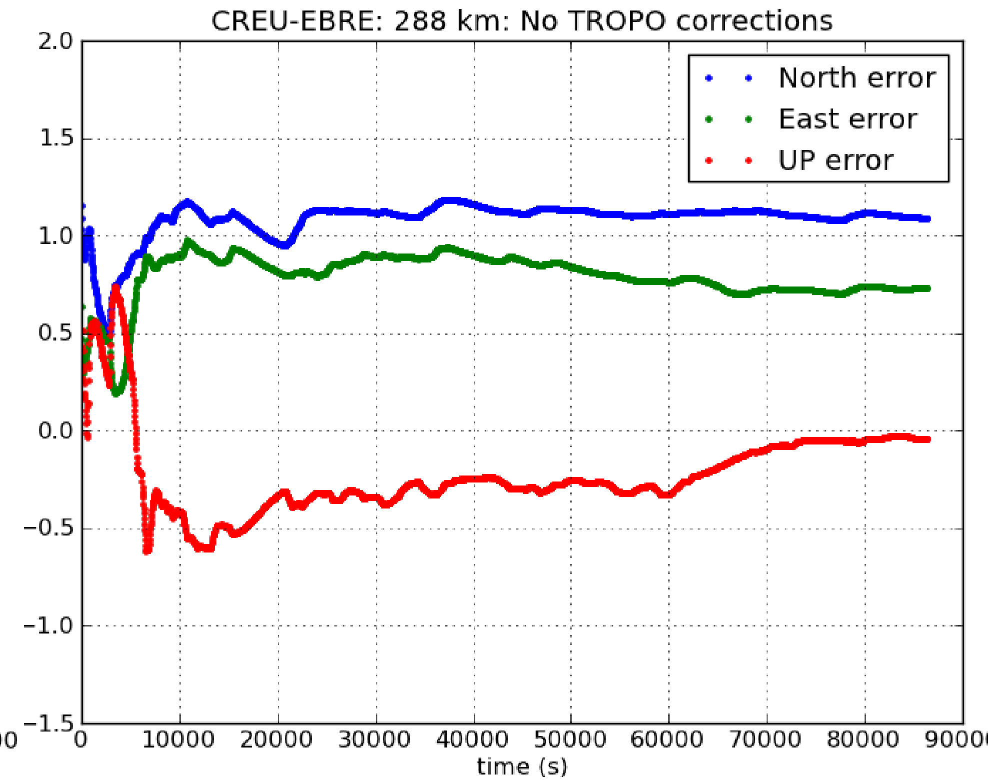


Questions:

- ***Which error source has the higher impact on the position domain higher the ionosphere or the troposphere?***
- ***Which is easier to model?***

A.1. Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)





Question:

- ***Compare the results with those of the previous case, when not considering the ionospheric error.***

A.1.3.3. Differential positioning. No tropospheric corrections

OVERVIEW

- **Introduction:** gLAB processing in command line

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Repeat the previous exercise, but using the permanent receivers **GARR** and **MATA**, with only **51 km of baseline**.

Model Components computation

- The script "**ObsFile1.scr**" generates a data file "**STA.obs**" with the following content

```
  1   2   3   4   5  6  7  8  9   10   11  12   13   14
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion Elev Azim Prefit]
```

- Run this script for GARR and MATA receivers:

```
ObsFile1.scr GARR0770.10o brdc0770.10n
ObsFile1.scr MATA0770.10o brdc0770.10n
```

- Generate the navigation equations system for absolute positioning for each receiver and compute the user solution (see next two slides).

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Using the files **`GARR.obs`** and **`MATA.obs`**, follow the next steps:

1. Build the navigation equations system for each receiver, for absolute positioning.
2. Compute the user solutions in kinematic mode.
3. Plot the results and compare the positioning errors.

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

1.- Building the navigation equations system for absolute positioning:

```
cat GARR.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
            $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> GARR.mod
```

2.- Computing the user solution for absolute positioning:

```
cp kalman.nml_wn kalman.nml

cat GARR.mod | kalman > GARR.pos
```

3.- Plotting the results:

```
graph.py  -f GARR.pos -x1 -y2 -s.- -l "North error"
          -f GARR.pos -x1 -y3 -s.- -l "East error"
          -f GARR.pos -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
          -t "GARR: Standard Point Positioning"
```

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

1.- Building the navigation equations system for absolute positioning:

```
cat MATA.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
            $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> MATA.mod
```

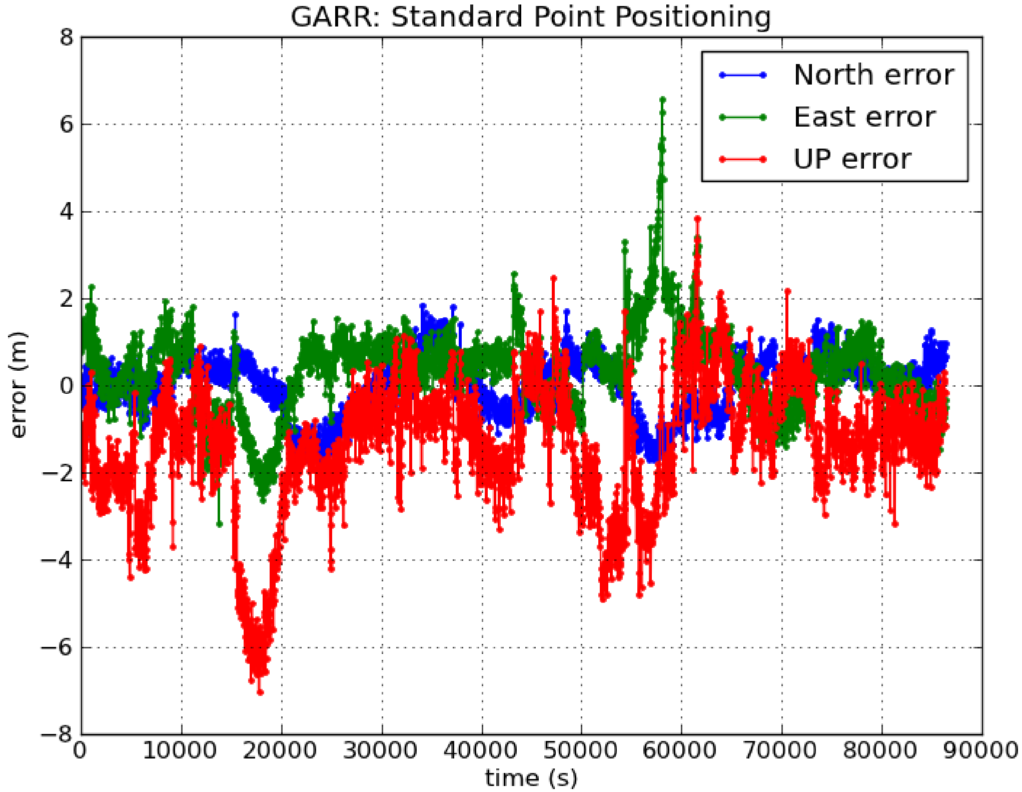
2.- Computing the user solution for absolute positioning:

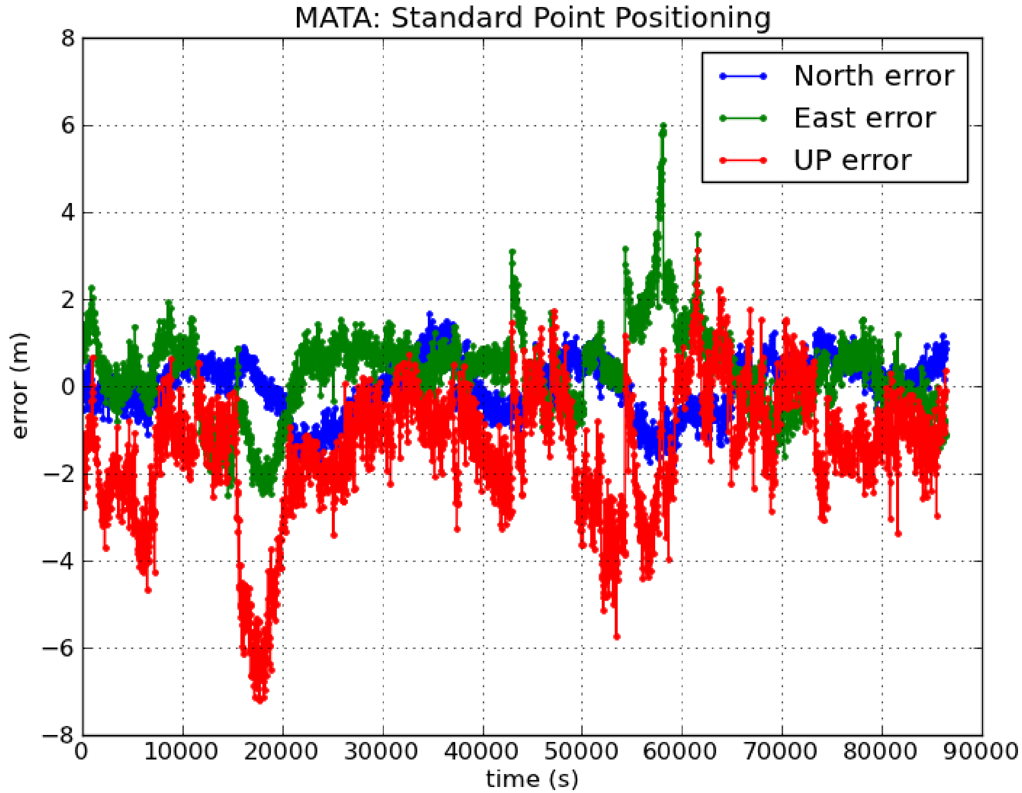
```
cp kalman.nml_wn kalman.nml

cat MATA.mod | kalman > MATA.pos
```

3.- Plotting the results:

```
graph.py  -f MATA.pos -x1 -y2 -s.- -l "North error"
          -f MATA.pos -x1 -y3 -s.- -l "East error"
          -f MATA.pos -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
          -t "MATA: Standard Point Positioning"
```

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)





Questions:

- ***Compare the results with the previous case with a baseline of 280 km.***
- ***Are the patterns even more similar now? Why?***

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Using the files **`GARR.obs`** and **`MATA.obs`**, follow the next steps:

1. Compute the single differences of measurements and model components.
2. Build the navigation equations system for the differential positioning of MATA receiver (user) relative to GARR (reference station).
3. Compute the user solution in kinematic mode and in static mode.
4. Plot the results and discuss the positioning error found.

A.2.3.1 Differential positioning. Full model

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

1. Computing the single differences of measurements and model components.

```
Dobs.scr GARR.obs MATA.obs
```

2.- Building the navigation equations system for differential positioning:

```
cat D_GARR_MATA.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
 $4,$5-$9-$10-$11,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'> D_GARR_MATA.mod
```

3.- Computing the user solution for differential positioning:

Kinematic:

```
cp kalman.nml_wn kalman.nml
cat D_GARR_MATAA.mod | kalman > GARR_MATA.posK
```

Static:

```
cp kalman.nml_ct kalman.nml
cat D_GARR_MATA.mod | kalman > GARR_MATA.posS
```

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

4.- Plotting the results:

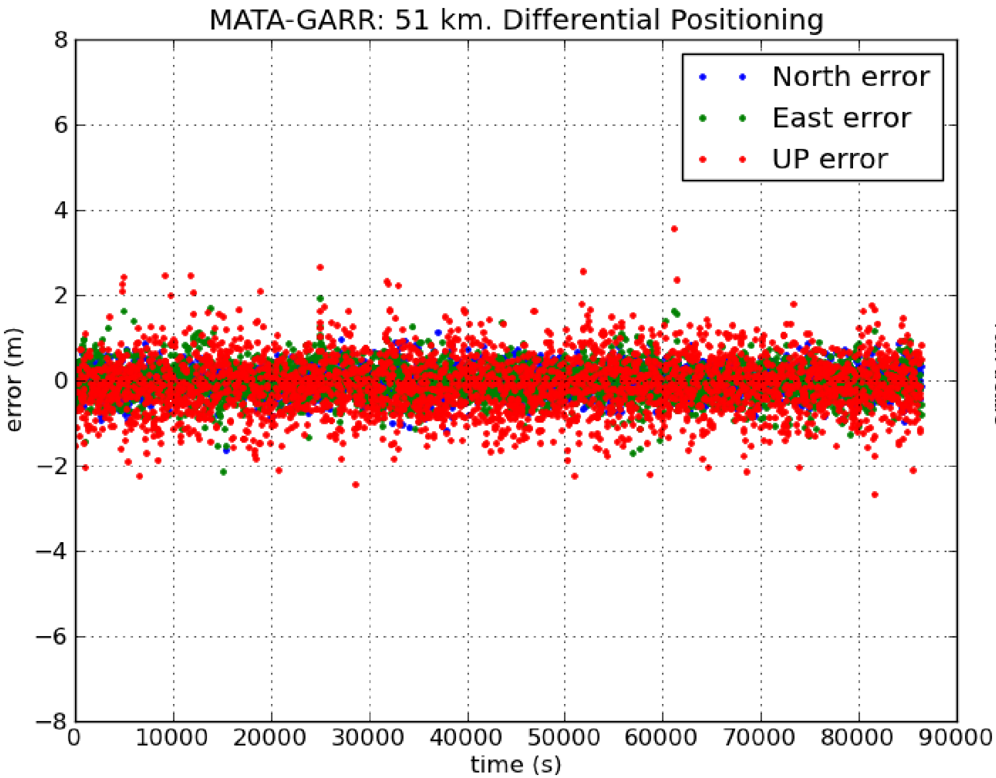
- Kinematic mode:

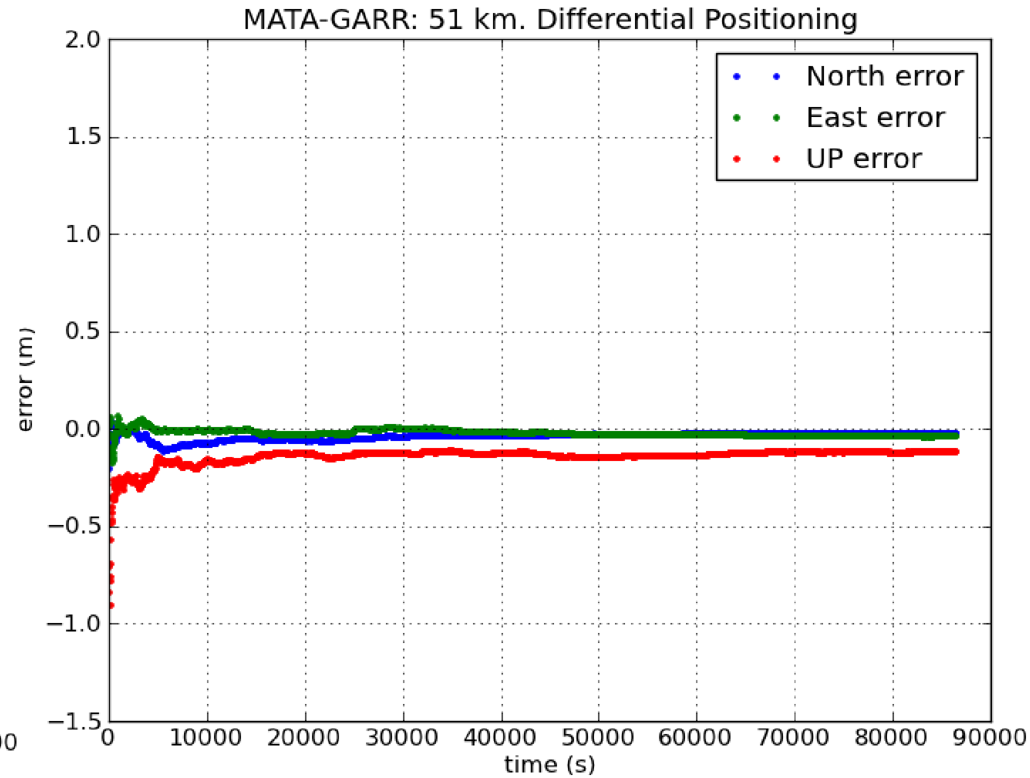
```
graph.py -f GARR_MATA.posK -x1 -y2 -s.  -l "North error"
         -f GARR_MATA.posK -x1 -y3 -s.  -l "East error"
         -f GARR_MATA.posK -x1 -y4 -s.  -l "UP error"
         --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
         -t "GARR_MATA: 51 km: Differential Positioning"
```

- Static mode:

```
graph.py -f GARR_MATA.posS -x1 -y2 -s.  -l "North error"
         -f GARR_MATA.posS -x1 -y3 -s.  -l "East error"
         -f GARR_MATA.posS -x1 -y4 -s.  -l "UP error"
         --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
         -t "GARR_MATA: 51 km: Differential Positioning"
```

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)





Questions:
- ***Compare the results with the previous case with a baseline of 280 km.***
- ***Are we reaching a similar level of accuracy? Why?***

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Analyze the effect of the differential ionosphere on the differential positioning of MATA relative to GARR station:

Follow the next steps:

1. Using file **`D_GARR_MATA.obs`** plot the differential ionospheric correction (from Klobuchar model) between EBRE and CREU.

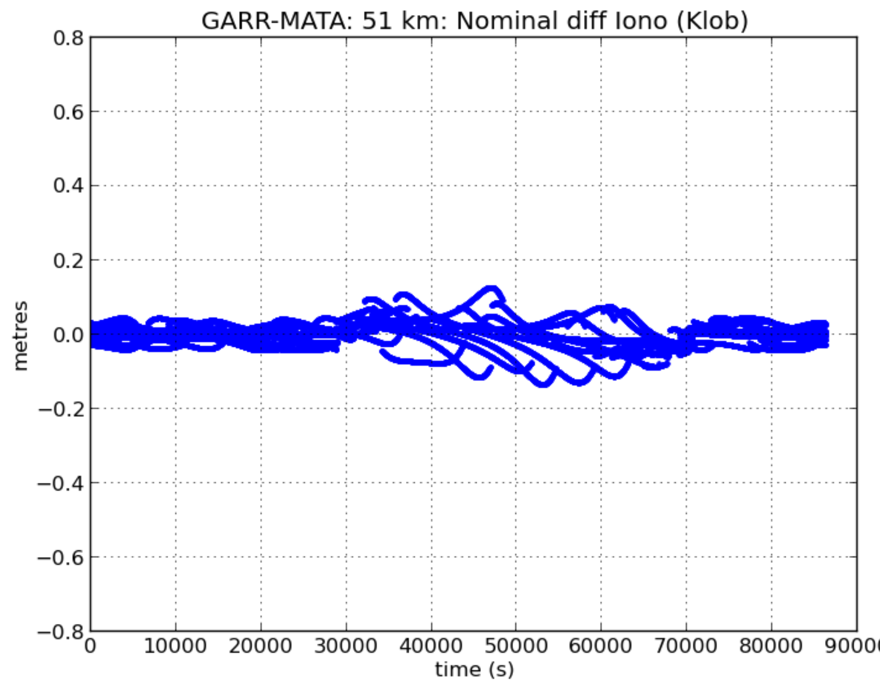
2. **Repeat** the previous process, **but without applying the ionospheric correction**. Compare the results

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Plotting the Klobuchar differential ionospheric correction:

```
graph.py -f D_GARR_MATA.obs -x4 -y'11' --yn -0.8 --yx 0.8
         -t "GARR-MATA: 51km: Nominal diff Iono (Klob)"
```

Question:
Justify the pattern seen in the nominal (Klobuchar) ionospheric corrections.
Compare the plot with that of the 280km baseline (slide #26). Do the results perform as expected?



A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Computing the differential solution, but without using ionospheric corrections:

1.- Building the navigation equations system for differential positioning, but without using the ionospheric corrections:

```
cat D_GARR_MATA.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
   $4,$5-$9-$10,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'> D_GARR_MATA.mod
```

2.- Computing the user solution for differential positioning:

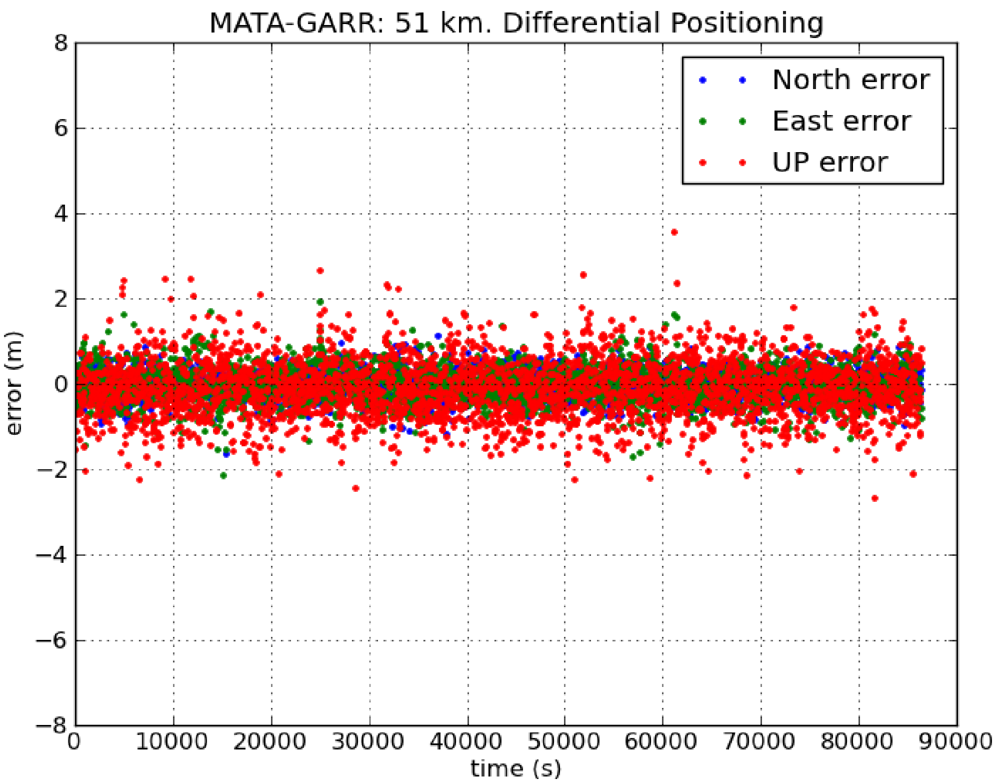
Kinematic:

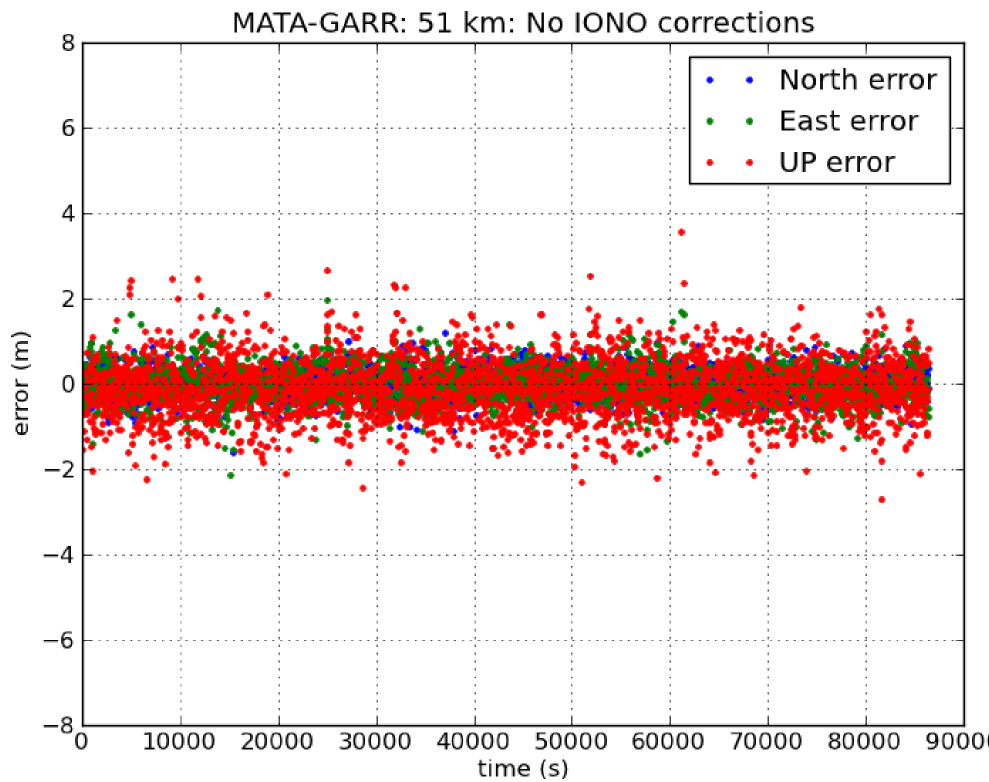
```
cp kalman.nml_wn kalman.nml
cat D_GARR_MATA.mod | kalman > GARR_MATA.posK
```

Static:

```
cp kalman.nml_ct kalman.nml
cat D_GARR_MATA.mod | kalman > GARR_MATA.posS
```

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)



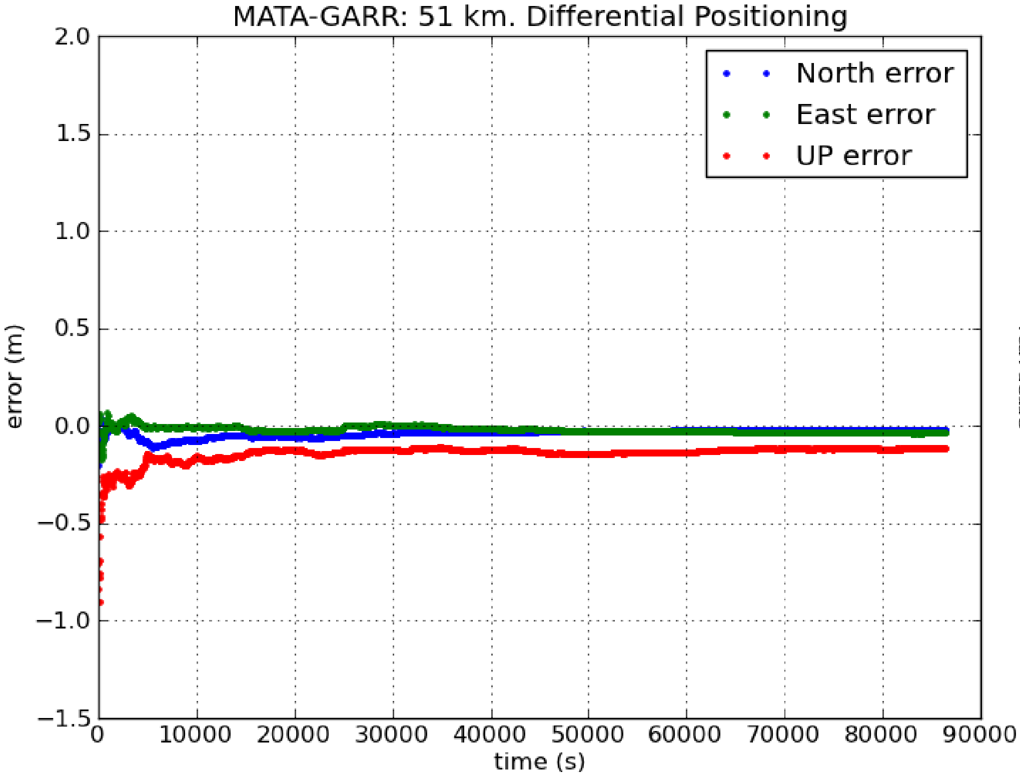


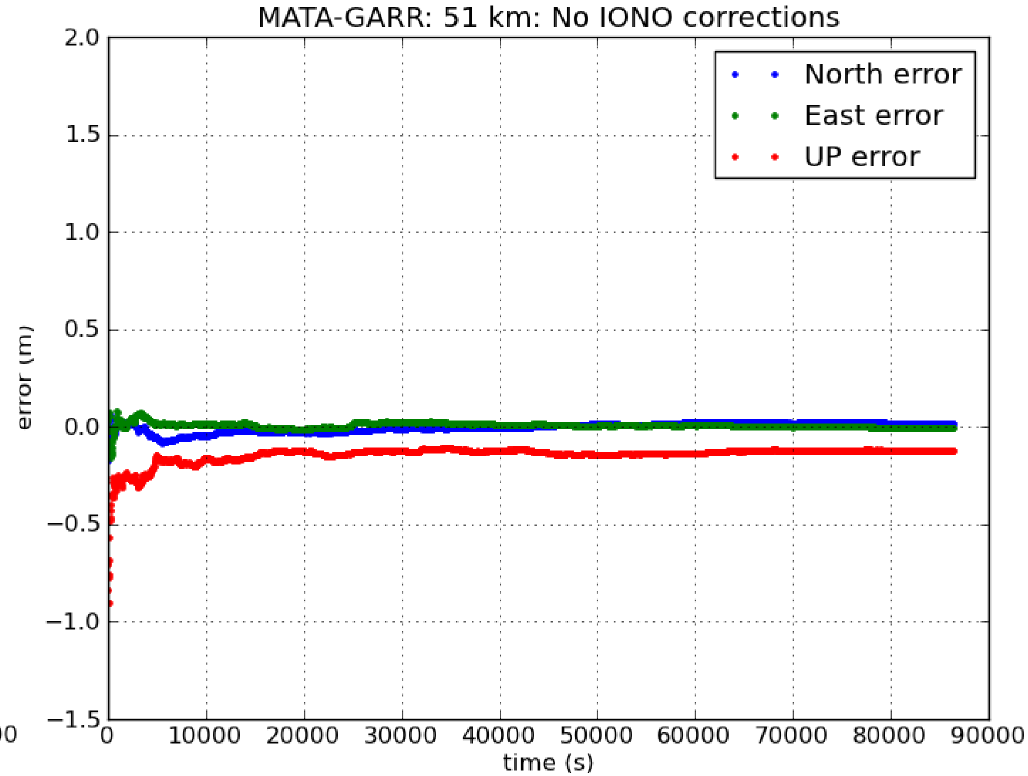
Question:

- ***Do the results confirm the expected effect of neglecting the ionospheric corrections for this baseline (using code measurements) at the user level?***

A.2.3.2. Differential positioning. No ionospheric corrections

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)





Question:
- ***Do the results confirm the expected effect of neglecting the ionospheric corrections for this baseline (using code measurements) at the user level?***

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Analyze the effect of the differential Troposphere on the differential positioning of MATA relative to GARR station:

Follow the next steps:

1. Using file **`D_GARR_MATA.obs`** plot the differential trpospheric correction (from nominal model) between EBRE and CREU.

2. Repeat the previous process, **but without applying the tropospheric correction**. Compare the results.

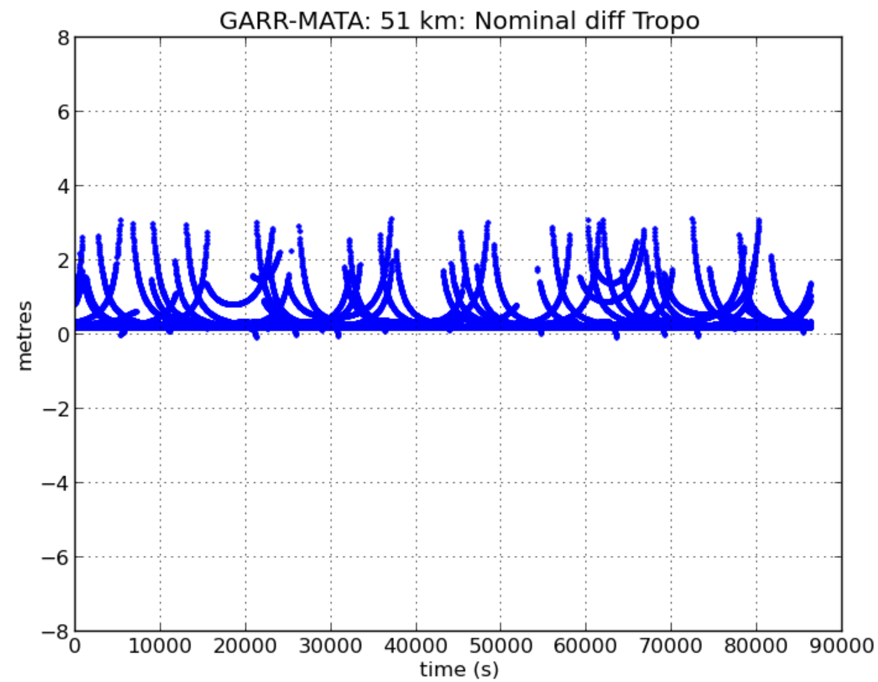
A.2.3.2. Differential positioning.
No tropospheric corrections

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Plotting the nominal differential tropospheric correction:

```
graph.py -f D_GARR_MATA.obs -x4 -y'10' --yn -8 --yx 8
         -t "GARR_MATA: 51km: Nominal diff Tropo"
```

Question:
Justify the pattern seen in the nominal tropospheric corrections. Compare the plot with that of the 280km baseline (slide #31). Do the results perform as expected?



A.2.3.2. Differential positioning.
No tropospheric corrections

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)

Computing the differential solution, but without using tropospheric corrections:

1.- Building the navigation equations system for differential positioning, but without using the tropospheric corrections:

```
cat D_GARR_MATA.obs | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
   $4,$5-$9-$11,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'> D_GARR_MATA.mod
```

2.- Computing the user solution for differential positioning:

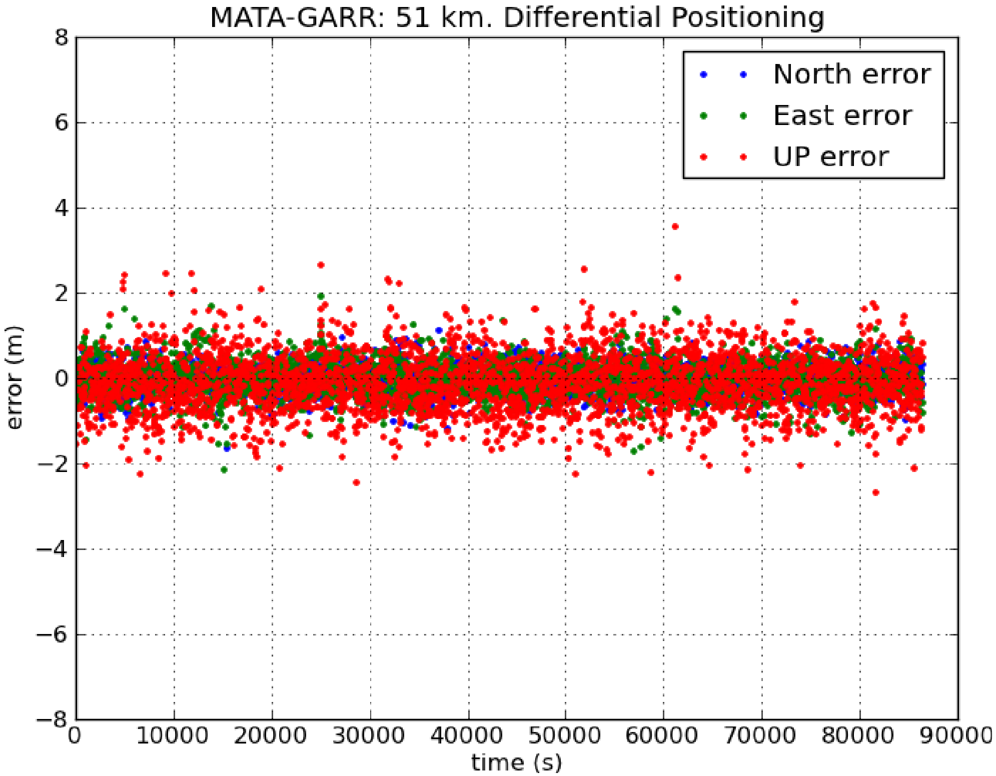
Kinematic:

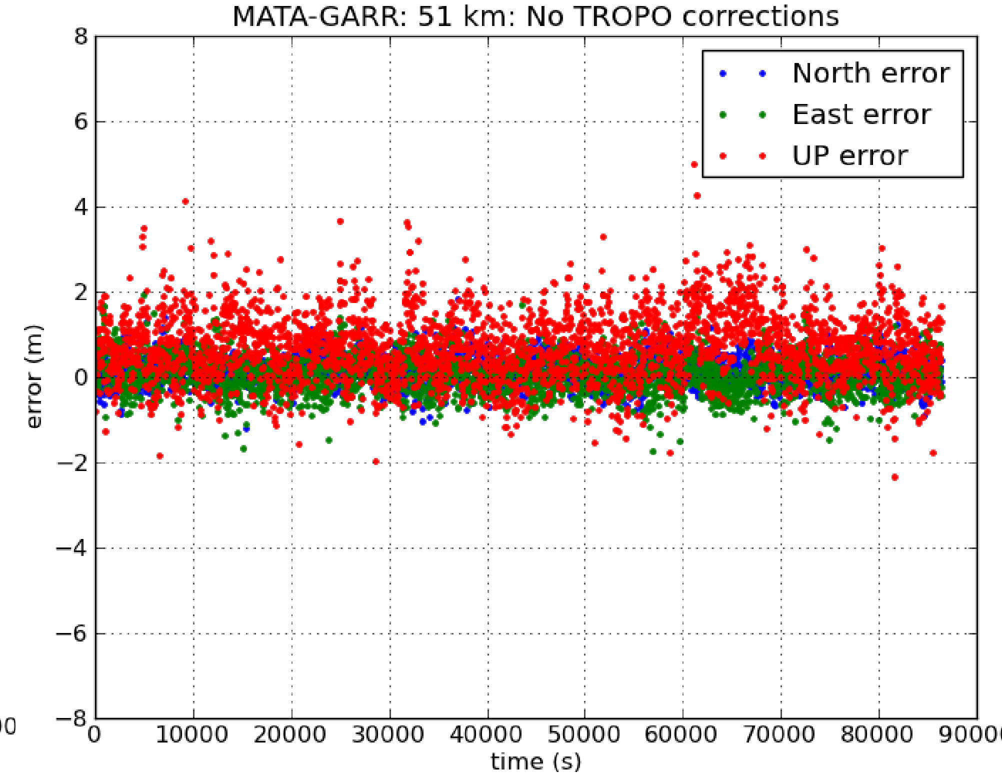
```
cp kalman.nml_wn kalman.nml
cat D_GARR_MATA.mod | kalman > GARR_MATA.posK
```

Static:

```
cp kalman.nml_ct kalman.nml
cat D_GARR_MATA.mod | kalman > GARR_MATA.posS
```

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)



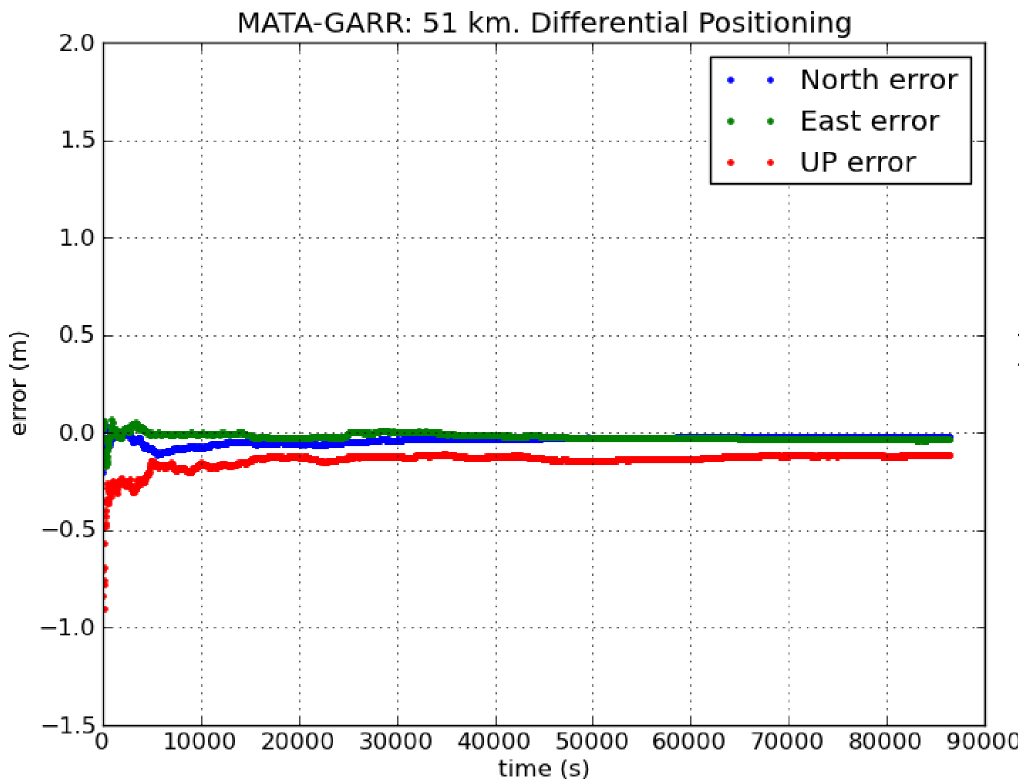


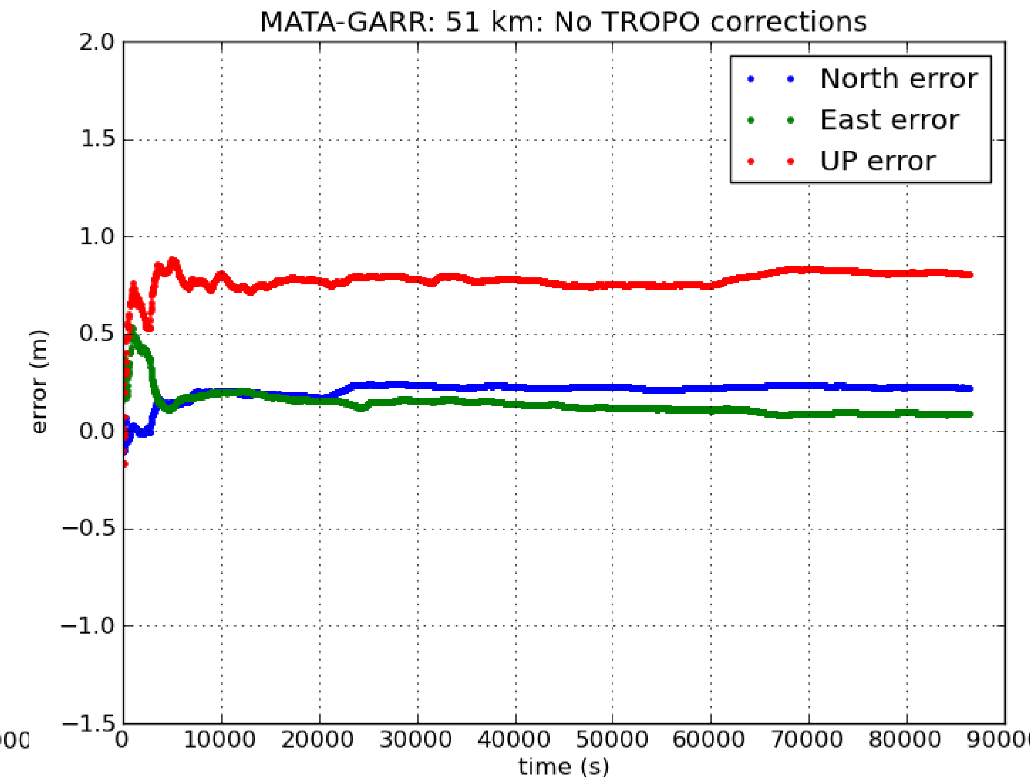
Question:

- ***Do the results confirm the expected effect of neglecting the tropospheric corrections for this baseline (using code measurements) at the user level?***

A.2.3.2. Differential positioning.
No tropospheric corrections

A.2. Differential positioning of GARR_MATA receivers (Short baseline: 51 km)





Question:

- ***Discuss the error found at the user level for the 50 Km of baseline.***
- ***Is the error level similar to the error seen with the 280km of baseline (slide 34#)?***
- ***Should the tropospheric corrections still be applied for these baselines?***

A.2.3.2. Differential positioning. No tropospheric corrections

OVERVIEW

- **Introduction:** gLAB processing in command line

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

Session B

Differential positioning
Orbit errors

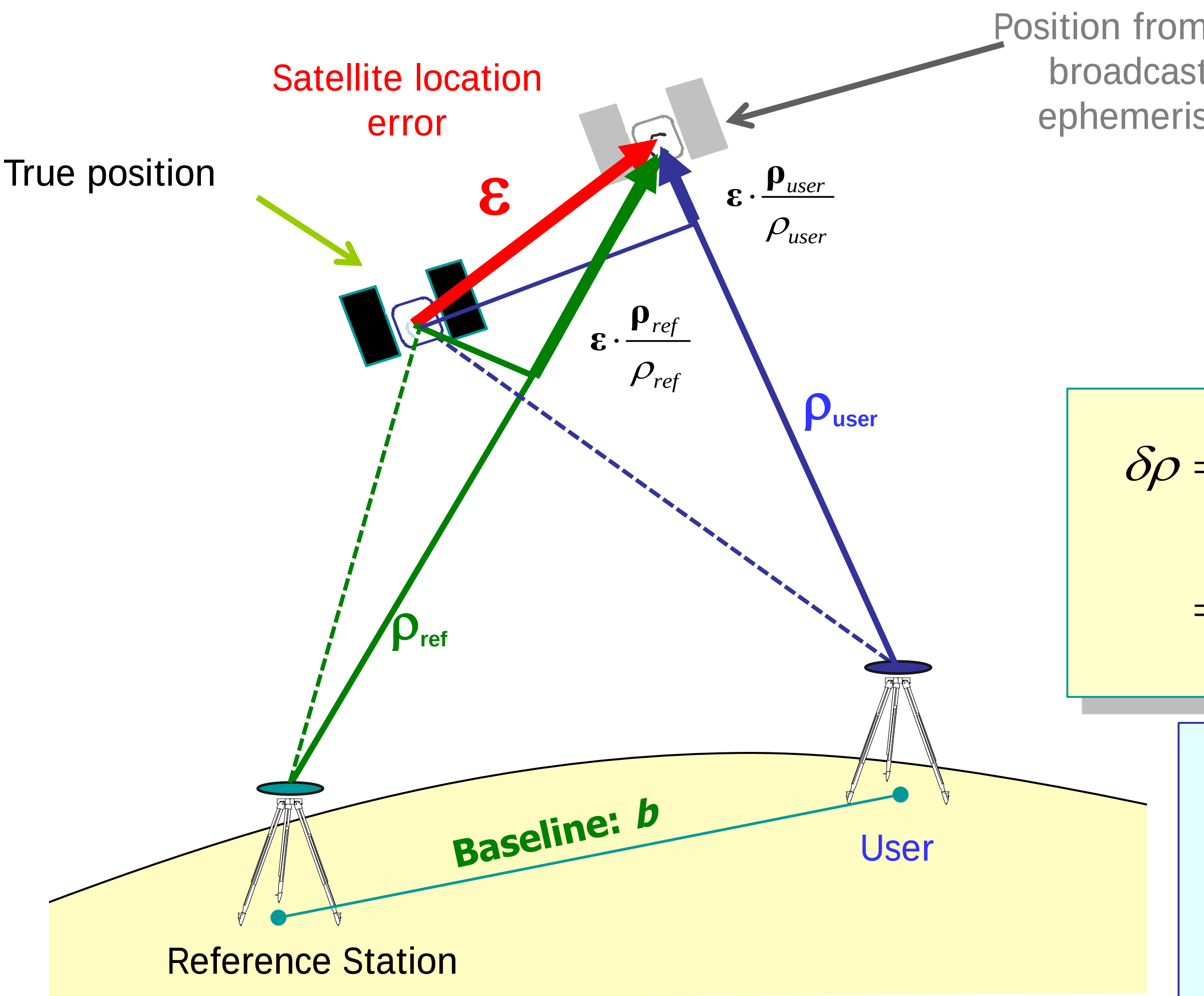
B. Orbit error effect on Differential positioning.

The target of this exercise is to asses the satellite orbit error on absolute and differential positioning.

This will be done by modifying the broadcast orbit parameters to generate an Along-track orbit error of about 2000m and, positioning two receivers with these corrupted orbits.

The user positioning error using the original and the corrupted orbits will be compared for absolute and differential positioning, with the receivers and baselines used in the previous session.

Ephemeris Errors and Geographic decorrelation



Differential range error
due to satellite obit error

$$\delta\rho = \boldsymbol{\varepsilon}\cdot\frac{\boldsymbol{\rho}_{user}}{\rho_{user}} - \boldsymbol{\varepsilon}\cdot\frac{\boldsymbol{\rho}_{ref}}{\rho_{ref}}$$

$$\delta\rho = -\frac{b\sin\phi}{\rho}\boldsymbol{\varepsilon}^T\cdot\hat{\mathbf{u}} = -\vec{\boldsymbol{\varepsilon}}^T\cdot(\sin\phi\,\hat{\mathbf{u}})\frac{b}{\rho}$$

$$= -\boldsymbol{\varepsilon}^T\left(\mathbf{I}-\hat{\boldsymbol{\rho}}\cdot\hat{\boldsymbol{\rho}}^T\right)\frac{\mathbf{b}}{\rho}$$

Where: $\mathbf{b} = b\,\hat{\mathbf{b}}$
is the baseline vector

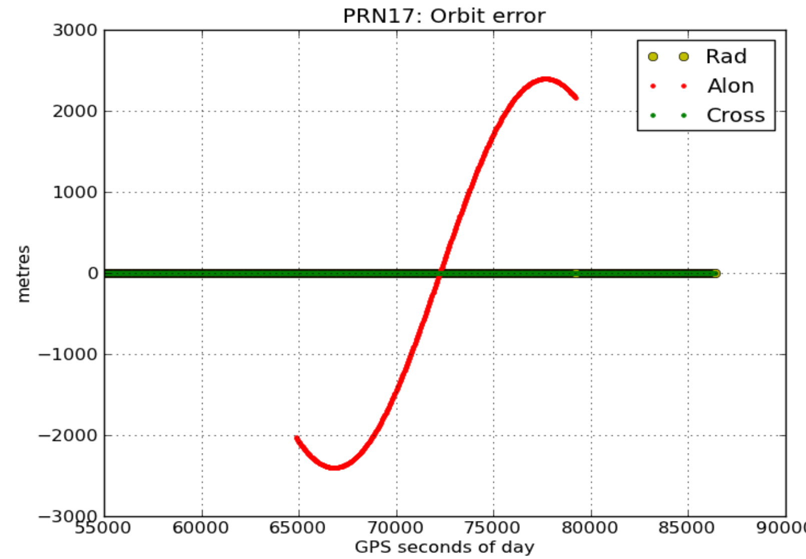
A conservative bound:

$$\boxed{\delta\rho < \frac{b}{\rho}\varepsilon}$$

with a baseline $b = 20km$

$$\delta\rho < \frac{20}{20000}\varepsilon = \frac{1}{1000}\varepsilon$$

B. Orbit error effect.



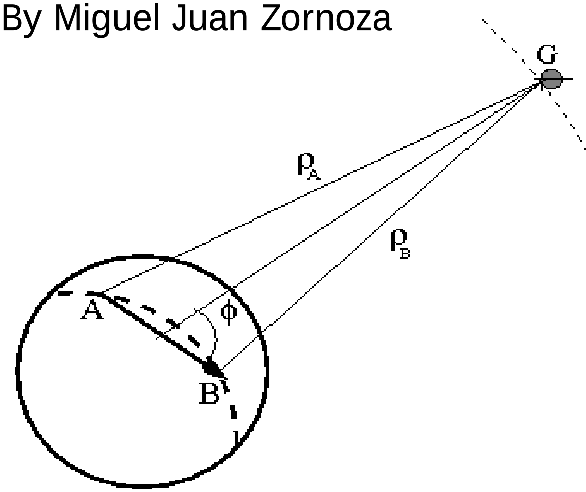
Injecting an error to broadcast orbits:

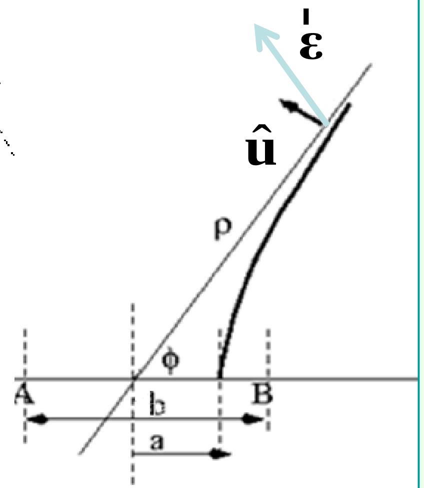
Along-track Error (PRN17)

The following ephemeris block of PRN17 is modified
In order to simulate an Along-track error of about 2000m.
The corrupted file is renamed as: **brdc0770.10nERR**

```
17 10  3 18 20  0  0.0 1.379540190101E-04 2.842170943040E-12 0.000000000000E+00
    7.800000000000E+01-5.059375000000E+01 4.506973447820E-09-2.983492318682E+00
   -9.257976353169E-05 5.277505260892E-03 8.186325430870E-06 5.153578153610E+03
    4.176000000000E+05-5.401670932770E-08-4.040348681654E-01-7.636845111847E-08
    9.603630515702E-01 2.215312500000E+02-2.547856603060E+00-7.964974630307E-09
   -3.771585673111E-10 1.000000000000E+00 1.575000000000E+03 0.000000000000E+00
    2.000000000000E+00 0.000000000000E+00-1.024454832077E-08 7.800000000000E+01
    4.104180000000E+05 4.000000000000E+00


diff brdc0770.10n brdc0770.10nERR- --------------------------------------------------
<     -2.579763531685E-06 5.277505260892E-03 8.186325430870E-06 5.153578153610E+03
>     -9.257976353169E-05 5.277505260892E-03 8.186325430870E-06 5.153578153610E+03
---------------------------------------------------------------------------------
```

By Miguel Juan Zornoza





- Errors over the hyperboloid (i.e. $\rho_B - \rho_A = ctt$) will not produce differential range errors.
- The highest error is given by the vector $\hat{\mathbf{u}}$, orthogonal to the hyperboloid and over the plain containing the baseline vector $\hat{\mathbf{b}}$ and the LoS vector $\hat{\boldsymbol{\rho}}$.

  Note:
  Being the baseline b much smaller than the distance to the satellite, we can assume that the LoS vectors from A and B receives are essentially identical to ρ.
  That is, $\rho_B \cong \rho_A \cong \rho$

$$\mathbf{u} = \hat{\boldsymbol{\rho}} \times \left(\hat{\mathbf{b}} \times \hat{\boldsymbol{\rho}}\right) = \hat{\mathbf{b}}\left(\hat{\boldsymbol{\rho}}^T \cdot \hat{\boldsymbol{\rho}}\right) - \hat{\boldsymbol{\rho}}\left(\hat{\boldsymbol{\rho}}^T \cdot \hat{\mathbf{b}}\right)$$

$$= \mathbf{I}\,\hat{\mathbf{b}} - \left(\hat{\boldsymbol{\rho}} \cdot \hat{\boldsymbol{\rho}}^T\right)\hat{\mathbf{b}} = \left(\mathbf{I} - \hat{\boldsymbol{\rho}} \cdot \hat{\boldsymbol{\rho}}^T\right)\hat{\mathbf{b}}$$

Note: $\mathbf{u} = \sin\phi\, \hat{\mathbf{u}}$

$$a = (\rho_B - \rho_A)/2 \ : \ \text{hyperboloid semiaxis}$$

$$b/2 \ : \ \text{focal length}$$

$$where \quad a = \frac{1}{2} b \cos\phi$$

Note: in this 3D problem ϕ is NOT the elevation of ray.

Differential range error $\delta\rho$ produced by an orbit error ε_P parallel to vector $\hat{\mathbf{u}}$

Let $\delta\varepsilon \equiv \varepsilon_P$

$$\delta\rho \equiv \delta(\rho_B - \rho_A) = 2\delta a =$$

$$= 2\frac{\partial a}{\partial \varepsilon}\delta\varepsilon = 2\frac{\partial a}{\partial \phi}\frac{\partial \phi}{\partial \varepsilon}\delta\varepsilon = -b\sin\phi\frac{\partial \phi}{\partial \varepsilon}\delta\varepsilon$$

$$\approx -b\sin\phi\frac{1}{\rho}\delta\varepsilon$$

Note: $\boldsymbol{\varepsilon}_P \perp \boldsymbol{\rho} \Rightarrow \delta\varepsilon \ ; \ \rho\,\delta\phi$

Note: being $\hat{\mathbf{u}}$ a vector orthogonal to the LoS $\hat{\boldsymbol{\rho}}$, thence, $\varepsilon_P = \boldsymbol{\varepsilon}^T \hat{\mathbf{u}}$

Thence:

$$\delta\rho = -\frac{b\sin\phi}{\rho}\boldsymbol{\varepsilon}^T \cdot \hat{\mathbf{u}} = -\boldsymbol{\varepsilon}^T \cdot \left(\sin\phi\,\hat{\mathbf{u}}\right)\frac{b}{\rho}$$

$$= -\boldsymbol{\varepsilon}^T\left(\mathbf{I} - \hat{\boldsymbol{\rho}} \cdot \hat{\boldsymbol{\rho}}^T\right)\frac{\mathbf{b}}{\rho}$$

Where: $\mathbf{b} = b\,\hat{\mathbf{b}}$
is the baseline vector

OVERVIEW

- **Introduction:** gLAB processing in command line

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

B1. Orbit error: Differential positioning
of EBRE-CREU receivers (Long baseline: 288 km)

Repeat the computation of the absolute positioning of receivers **EBRE** and **CREU**, but using the corrupted ephemeris file **brdc0770.10nERR**.

Model Components computation

- The script "**ObsFile1.scr**" generates a data file "**STA.obs**" with the following content

```
  1    2   3   4   5  6  7  8  9   10   11  12   13   14
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion Elev Azim Prefit]
```

- Run this script for EBRE and CREU receivers. Rename as **STA.obsERR** the output files

```
ObsFile1.scr EBRE0770.10o brdc0770.10nERR
mv EBRE.obs EBRE.obsERR
ObsFile1.scr CREU0770.10o brdc0770.10nERR
mv EBRE.obs EBRE.obsERR
```

- Generate the navigation equations system for absolute positioning for each receiver and compute the user solution (see the next two slides).

B.1.1. Absolute positioning. Orbit error

B1. Orbit error: Differential positioning
of EBRE-CREU receivers (Long baseline: 288 km)

1.- Building the navigation equations system for absolute positioning:

```
cat EBRE.obsERR | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
        $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> EBRE.modERR
```

2.- Computing the user solution for absolute positioning:

```
cp kalman.nml_wn kalman.nml

cat EBRE.modERR | kalman > EBRE.posERR
```

3.- Plotting the results:

```
graph.py  -f EBRE.posERR -x1 -y2 -s.- -l "North error"
          -f EBRE.posERR -x1 -y3 -s.- -l "East error"
          -f EBRE.posERR -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -300 --yx 300
          -t "EBRE: SPP: 2000m Along-Track orbit error"
```

B1. Orbit error: Differential positioning
of EBRE-CREU receivers (Long baseline: 288 km)

1.- Building the navigation equations system for absolute positioning:

```
cat CREU.obsERR | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
        $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> CREU.modERR
```

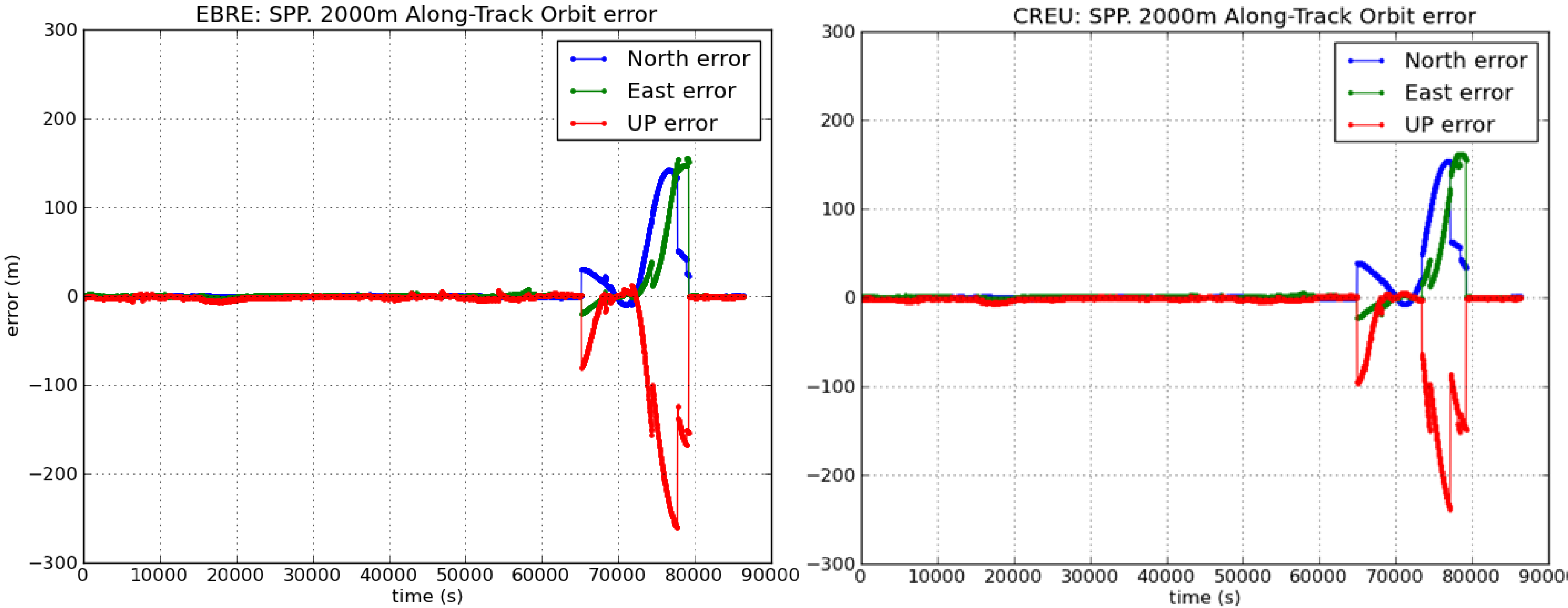
2.- Computing the user solution for absolute positioning:

```
cp kalman.nml_wn kalman.nml

cat CREU.modERR | kalman > CREU.posERR
```

3.- Plotting the results:

```
graph.py  -f CREU.posERR -x1 -y2 -s.- -l "North error"
          -f CREU.posERR -x1 -y3 -s.- -l "East error"
          -f CREU.posERR -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -300 --yx 300
          -t "CREU: SPP: 2000m Along-Track orbit error"
```

B1. Orbit error: Differential positioning
of EBRE-CREU receivers (Long baseline: 288 km)



Question:
- ***Is the impact on user positioning error similar for both receivers?***
- ***From these plots, what is the expected level of error in differential positioning?***

B.1.1. Absolute positioning. Orbit error

B1. Orbit error: Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

Analyse the effect of the 2000m orbit error on the differential positioning of CREU receiver relative to EBRE, with 280km of baseline.

Using the files **`EBRE.obsERR`** and **`CREU.obsERR`**, follow next steps:

1. Compute the single differences of measurements and model components.
2. Build the navigation equations system for the differential positioning of CREU receiver (user) relative to EBRE (reference station).
3. Compute the user solution in kinematic mode.
4. Plot the results and discuss the positioning error found.

B1. Orbit error: Differential positioning
of EBRE-CREU receivers (Long baseline: 288 km)

1.- Computing the single differences of measurements and model components.

```
Dobs.scr EBRE.obsERR CREU.obsERR
mv D_EBRE_CREU.obs D_EBRE_CREU.obsERR
```

2.- Building the navigation equations system for differential positioning:

```
cat D_EBRE_CREU.obsERR | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
$4,$5-$9-$10-$11,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'>D_EBRE_CREU.modERR
```

3.- Computing the user solution for differential positioning:

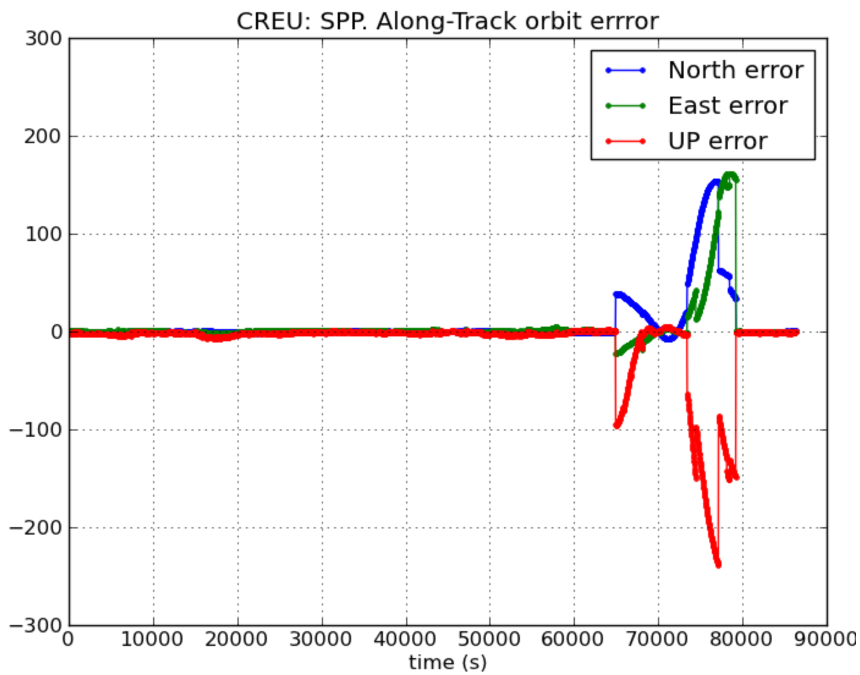
Kinematic:

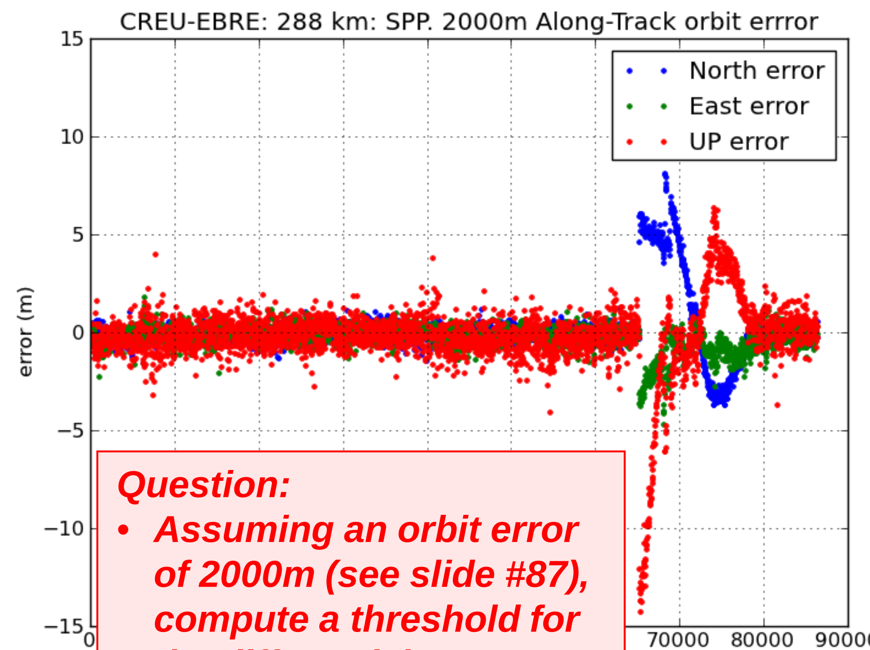
```
cp kalman.nml_wn kalman.nml
cat D_EBRE_CREU.modERR| kalman > EBRE_CREU.posKERR
```

B1. Orbit error: Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)

4.- Plotting results:

```
graph.py -f D_EBRE_CREU.modERR -x1 -y2 -s.  -l "North error"
         -f D_EBRE_CREU.modERR -x1 -y3 -s.  -l "East error"
         -f D_EBRE_CREU.modERR -x1 -y4 -s.  -l "UP error"
         --xl "time (s)" --yl "error (m)" --yn -15 --yx 15
         -t "EBRE-CREU 280km: SPP: 2000m Along-Track orbit error"
```





Question:
- ***Assuming an orbit error of 2000m (see slide #87), compute a threshold for the differential error.***

OVERVIEW

- **Introduction:** gLAB processing in command line

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

B2. Orbit error: Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

Repeat the computation of the absolute positioning of receivers **GARR** and **MATA**, but using the corrupted ephemeris file **brdc0770.10nERR**.

Model Components computation

- The script "**ObsFile1.scr**" generates a data file "**STA.obs**" with the following content

```
  1    2   3   4   5  6  7  8  9   10   11  12    13   14
[sta sat DoY sec P1 L1 P2 L2 Rho Trop Ion Elev Azim Prefit]
```

- Run this script for GARR and MATA receivers. Rename as **STA.obsERR** the output files

```
ObsFile1.scr GARR0770.10o brdc0770.10nERR
mv GARR.obs GARR.obsERR
ObsFile1.scr MATA0770.10o brdc0770.10nERR
mv MATA.obs MATA.obsERR
```

- Generate the navigation equations system for absolute positioning for each receiver and compute the user solution (see next two slides).

B2. Orbit error: Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

1.- Building the navigation equations system for absolute positioning:

```
cat GARR.obsERR | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
    printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
         $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> GARR.modERR
```

2.- Computing the user solution for absolute positioning:

```
cp kalman.nml_wn kalman.nml

cat GARR.modERR | kalman > GARR.posERR
```

3.- Plotting the results:

```
graph.py  -f GARR.posERR -x1 -y2 -s.- -l "North error"
          -f GARR.posERR -x1 -y3 -s.- -l "East error"
          -f GARR.posERR -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -300 --yx 300
          -t "GARR: SPP: 2000m Along-Track orbit error"
```

B.2.1. Absolute positioning. Orbit error

B2. Orbit error: Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

1.- Building the navigation equations system for absolute positioning:

```
cat MATA.obsERR | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
   printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
       $4, $14 ,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1 }'> MATA.modERR
```

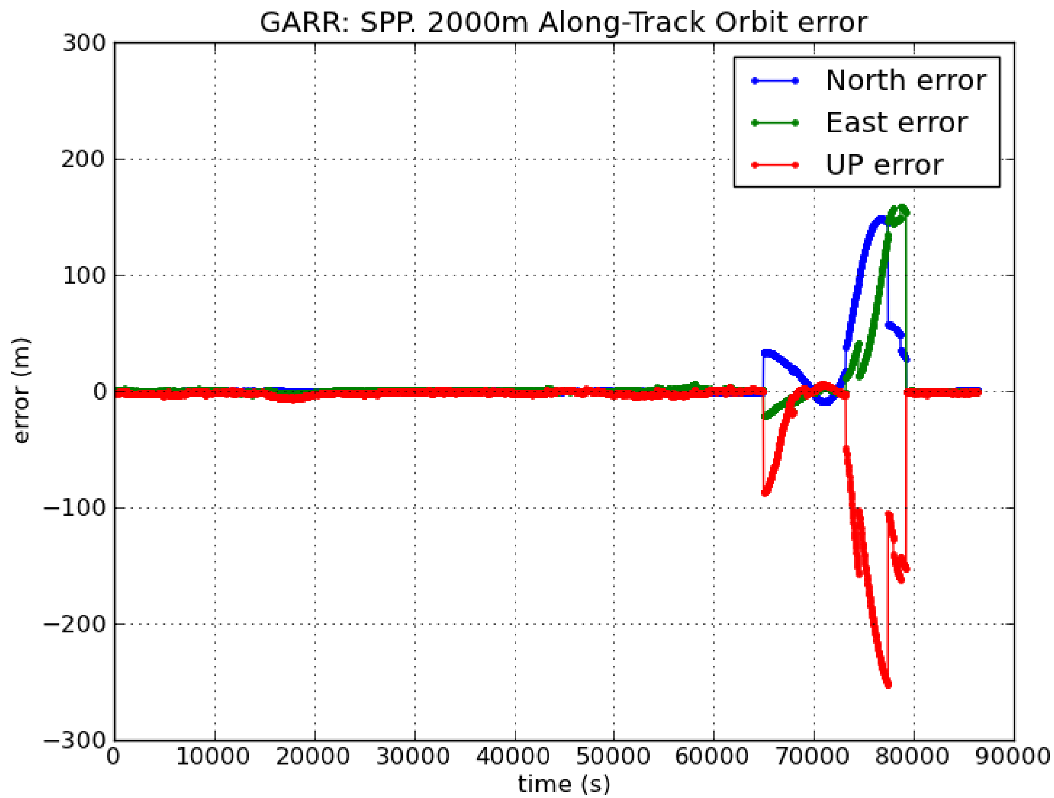
2.- Computing the user solution for absolute positioning:

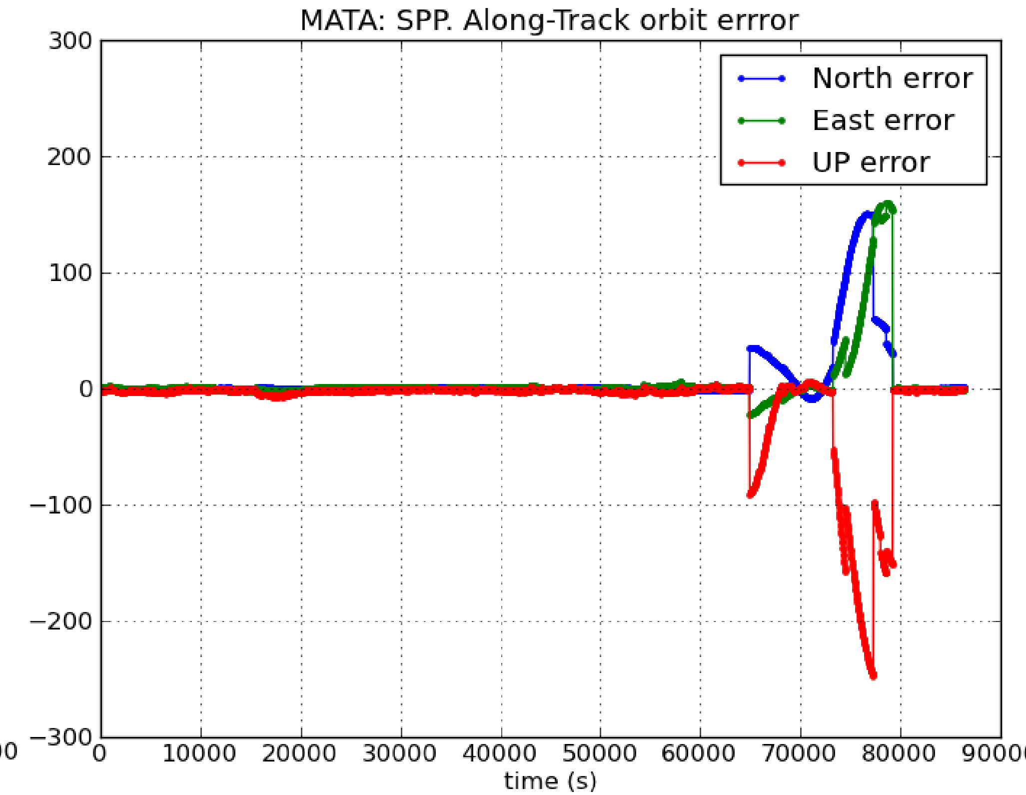
```
cp kalman.nml_wn kalman.nml

cat MATA.modERR | kalman > MATA.posERR
```

3.- Plotting the results:

```
graph.py  -f MATA.posERR -x1 -y2 -s.- -l "North error"
          -f MATA.posERR -x1 -y3 -s.- -l "East error"
          -f MATA.posERR -x1 -y4 -s.- -l "UP error"
          --xl "time (s)" --yl "error (m)" --yn -300 --yx 300
          -t "MATA: SPP: 2000m Along-Track orbit error"
```

B2. Orbit error: Differential positioning
of GARR-MATA receivers (Short baseline: 51 km)





Question:

- ***Is the impact on user positioning error similar for both receivers?***
- ***From these plots, what is the expected level of error in differential positioning?***

B.2.1. Absolute positioning. Orbit error

B2. Orbit error: Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

Analyse the effect of the 2000m orbit error on the differential positioning of MATA receiver relative to GARR, with 51km of baseline.

Using the files **`GARR.obsERR`** and **`MATA.obsERR`**, follow the next steps:

1. Compute the single differences of measurements and model components.
2. Build the navigation equations system for the differential positioning of MATA receiver (user) relative to GARR (reference station).
3. Compute the user solution in kinematic mode.
4. Plot the results and discuss the positioning error found.

B2. Orbit error: Differential positioning
of GARR-MATA receivers (Short baseline: 51 km)

1. Computing the single differences of measurements and model components.

```
Dobs.scr GARR.obsERR MATA.obsERR
```

2.- Building the navigation equations system for differential positioning:

```
cat D_GARR_MATA.obsERR | gawk 'BEGIN{g2r=atan2(1,1)/45}{e=$12*g2r;a=$13*g2r;
     printf "%8.2f %8.4f %8.4f %8.4f %8.4f %1i \n",
$4,$5-$9-$10-$11,-cos(e)*sin(a),-cos(e)*cos(a),-sin(e),1}'>D_GARR_MATA.modERR
```

3.- Computing the user solution for differential positioning:

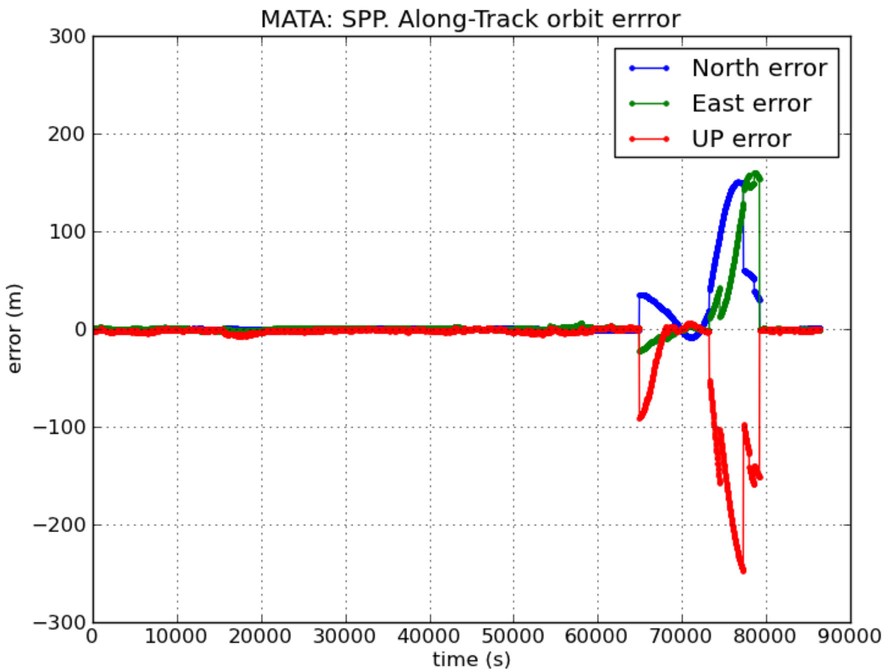
Kinematic:

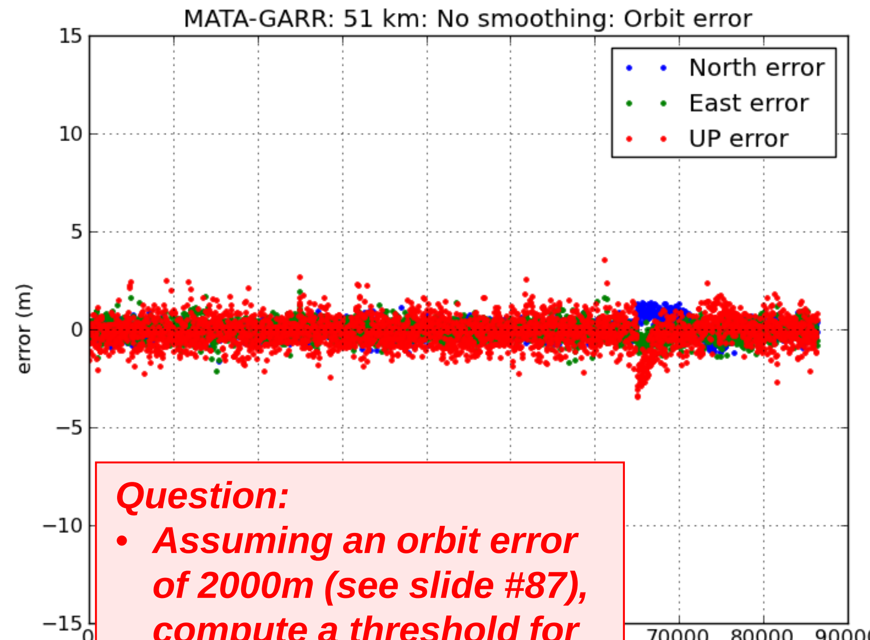
```
cp kalman.nml_wn kalman.nml
cat D_GARR_MATA.modERR| kalman > GARR_MATA.posKERR
```

B2. Orbit error: Differential positioning
of GARR-MATA receivers (Short baseline: 51 km)

4.- Plotting results:

```
graph.py -f D_GARR_MATA.modERR -x1 -y2 -s.  -l "North error"
         -f D_GARR_MATA.modERR -x1 -y3 -s.  -l "East error"
         -f D_GARR_MATA.modERR -x1 -y4 -s.  -l "UP error"
         --xl "time (s)" --yl "error (m)" --yn -8 --yx 8
         -t "GARR_MATA 51km: SPP: 2000m Along-Track orbit error"
```





Question:
- ***Assuming an orbit error of 2000m (see slide #87), compute a threshold for the differential error.***

OVERVIEW

- **Introduction:** gLAB processing in command line

- **Session A: Atmospheric effects**
  - A1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - A2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)

- **Session B: Orbit error effects**
  - B1 Differential positioning of EBRE-CREU receivers (Long baseline: 288 km)
  - B2 Differential positioning of GARR-MATA receivers (Short baseline: 51 km)
  - B3 Range domain orbit error.

B3. Orbit error: Differential positioning
Range Domain Orbit Error

Analyze the orbit error of **PRN17** in the Range Domain.

The following procedure is proposed:

Using the files **brdc0770.10n** and **brdc0770.10nERR**, follow the next steps:

1. Compute the absolute orbit for satellite PNR17 error by comparing the corrupted orbits with the original ones.
2. Compute the range error for the receivers EBRE and CREU
3. Compute the differential range error between EBRE and CREU
4. Compute the predicted range error between EBRE and CREU and compare with the present one.

B3. Orbit error: Differential positioning
Range Domain Orbit Error

B.3.1.- Absolute error computation for satellite PRN17:

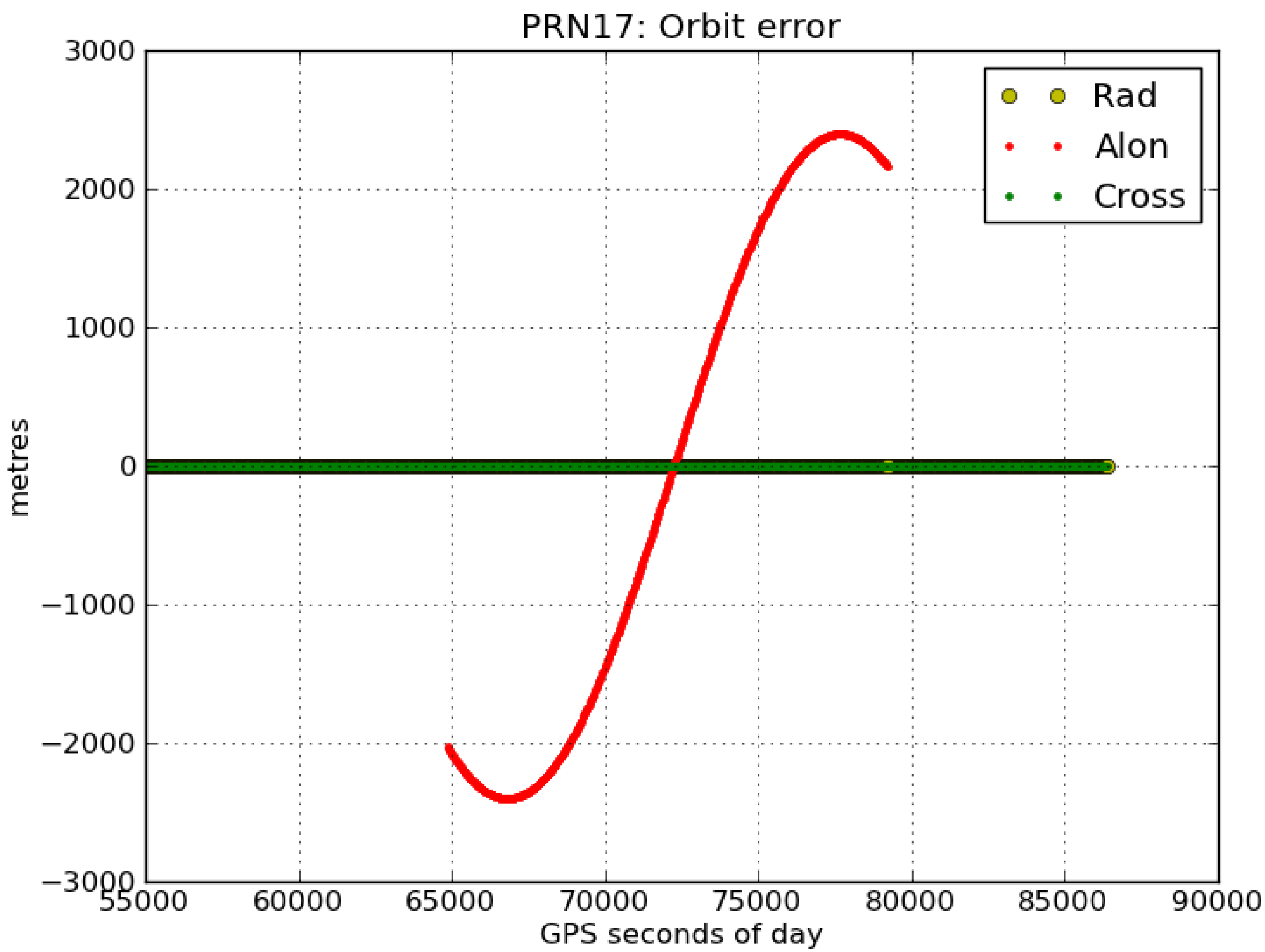
Compute the discrepance between the satellite coordinates predicted from files **brdc0770.10n** and **brdc0770.10nERR**

```
gLAB_linux -input:nav brdc0770.10n -input:nav brdc0770.10nERR -pre:dec 30 |
                                                         grep SATDIFF > dif.sel
```

Plot the results for satellite PRN17:

```
graph.py -f dif.sel -x4 -y11 -so -c '($6==17)' --cl y -l "Rad"
         -f dif.sel -x4 -y12 -s. -c '($6==17)' --cl r -l "Alon"
         -f dif.sel -x4 -y13 -s. -c '($6==17)' --cl g -l "Cross"
    --xn 55000  --xl "GPS seconds of day" --yl "metres" -t "PRN17: Orbit error"
```

B3. Orbit error: Differential positioning
Range Domain Orbit Error



B.3.2.- Absolute **Range error** computation for satellite PRN17 from **EBRE:**

1. Using the original orbits **brdc0770.10n**, compute the geometric range between EBRE and satellite PRN17 (i.e **rang.ebre** file).

   1.1- Compute the satellite coordinates with the original orbits and generate a file with the following content:

```
  1     2 3 4
[time X Y Z]
```

```
gLAB_linux -input:nav brdc0770.10n -pre:dec 30 | grep SATPVT |
                              gawk '{if ($6==17) print $4,$7,$8,$9}' > xyz.brdc
```

   1.2.- Using the coordinates **EBRE=[4833520.1197 41537.2015 4147461.6263]**, compute the geometric range between EBRE and satellite PRN17:

```
cat xyz.brdc | gawk 'BEGIN{x0=4833520.1197;y0=41537.2015;z0=4147461.6263}
                    {dx=$2-x0;dy=$3-y0;dz=$4-z0;rho=sqrt(dx**2+dy**2+dz**2);
                                printf "%s %16.6f \n", $1,rho}' > rang.ebre
```

B3. Orbit error: Differential positioning
Range Domain Orbit Error

2. Repeat the previous computation using the corrupted orbits file **brdc0770.10nERR** (i.e. generate the **rang.ebreERR** file).

   2.1- Compute the satellite coordinates with the corrupted orbits and generate a file with the following content:

   ```
    1    2 3 4
   [time X Y Z]
   ```

```
gLAB_linux -input:nav brdc0770.10nERR -pre:dec 30 | grep SATPVT |
                   gawk '{if ($6==17) print $4,$7,$8,$9}' > xyz.brdcERR
```

   2.2.- Using the coordinates **EBRE=[4833520.1197 41537.2015 4147461.6263]**, compute the geometric range between EBRE and satellite PRN17:

```
cat xyz.brdcERR|gawk 'BEGIN{x0=4833520.1197;y0=41537.2015;z0=4147461.6263}
                    {dx=$2-x0;dy=$3-y0;dz=$4-z0;rho=sqrt(dx**2+dy**2+dz**2);
                                printf "%s %16.6f \n", $1,rho}' > rang.ebreERR
```

B3. Orbit error: Differential positioning
Range Domain Orbit Error

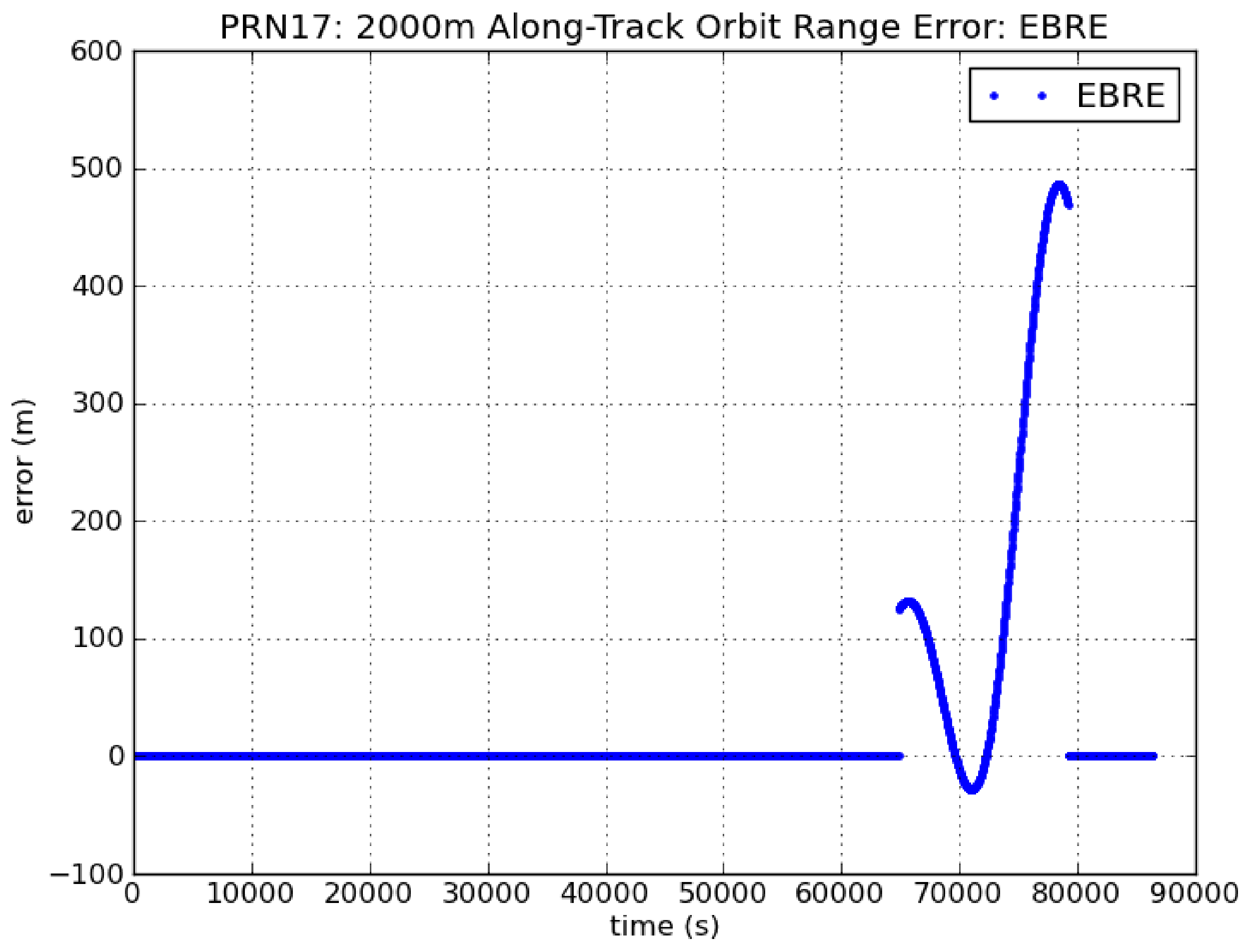
3. Calculate the discrepancy between the geometric ranges computed using the original and the corrupted orbits:

```
cat rang.ebre rang.ebreERR |
   gawk '{i=$1*1;if (length(r[i])==0) {r[i]=$2}else{print i,$2-r[i]}}' > drang.ebre
```

4.- Plot the results

```
graph.py -f drang.ebre  -x1 -y2 -l"EBRE"
         --xl "time (s)" --yl "error (m)" --yx 600
         -t"PRN17: 2000m AT orbit error PRN17: EBRE: Range Error"
```

B3. Orbit error: Differential positioning
Range Domain Orbit Error



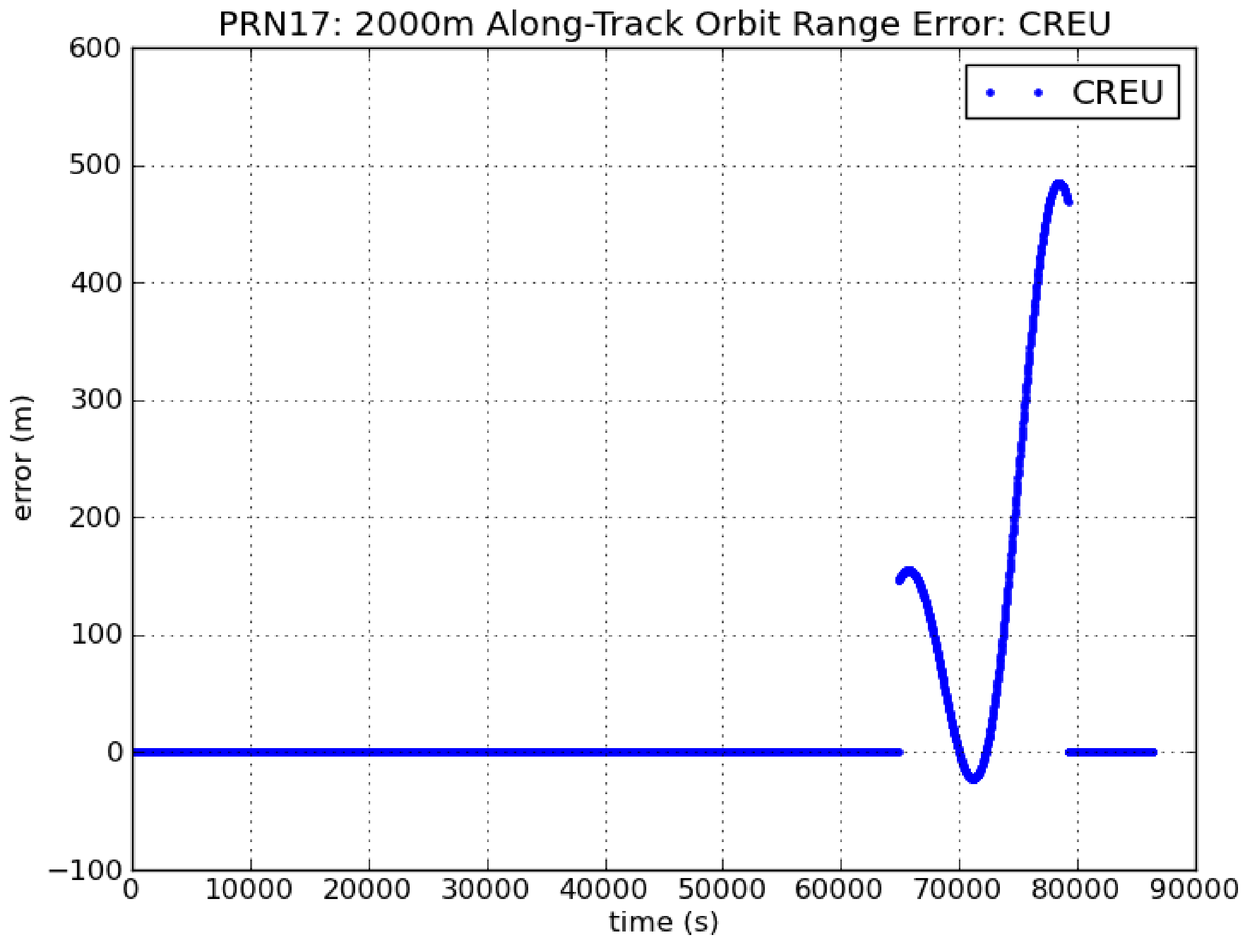
B3. Orbit error: Differential positioning
Range Domain Orbit Error

Absolute **Range error** computation for satellite PRN17 from **CREU:**

a) Repeat the previous computations of section B.3.2 for the receiver CREU.

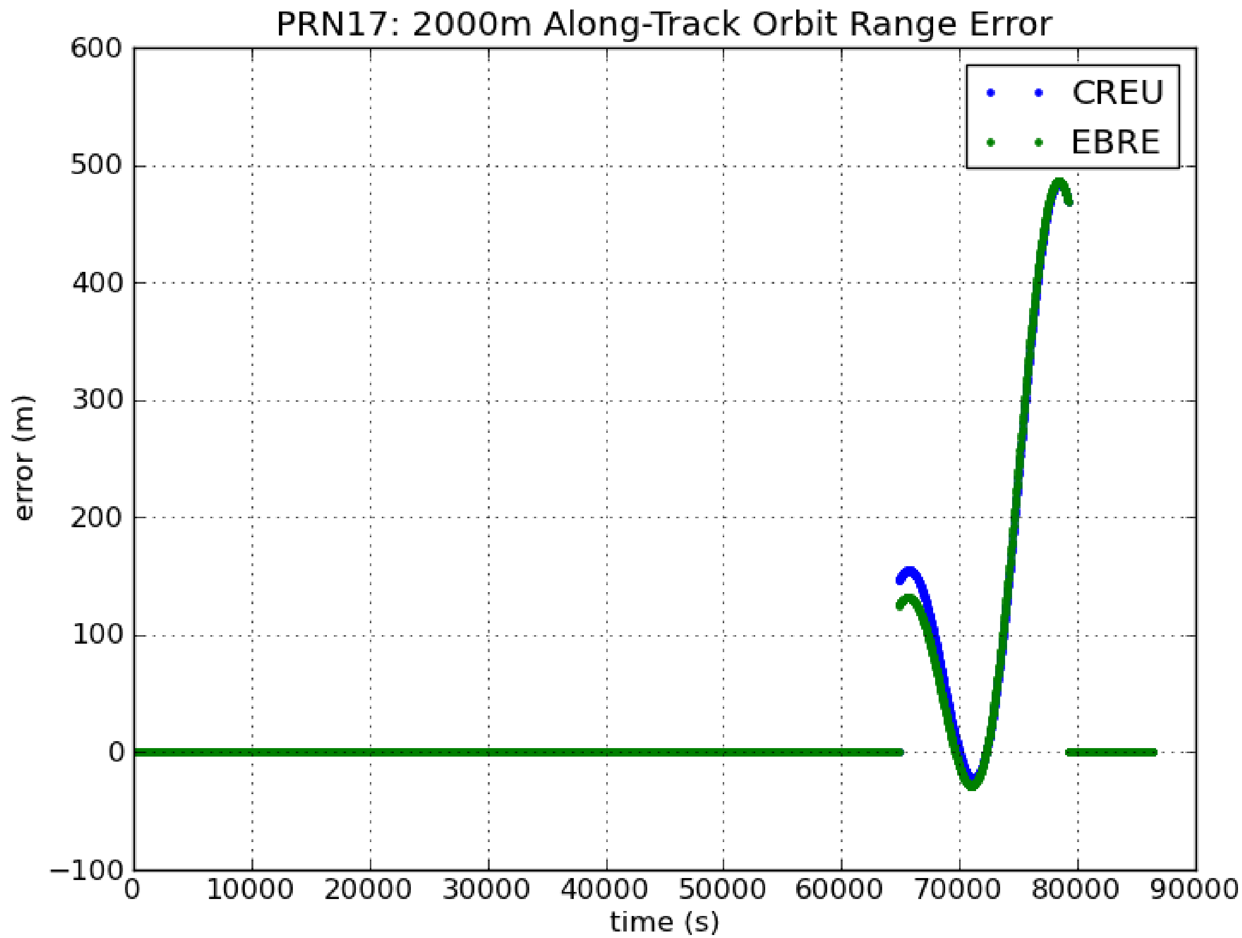
b) Compare in the same plot the results for EBRE and CREU

B3. Orbit error: Differential positioning
Range Domain Orbit Error



B.3.2. Absolute Range Orbit error: CREU

B3. Orbit error: Differential positioning
Range Domain Orbit Error



B.3.3 Differential range Error computation

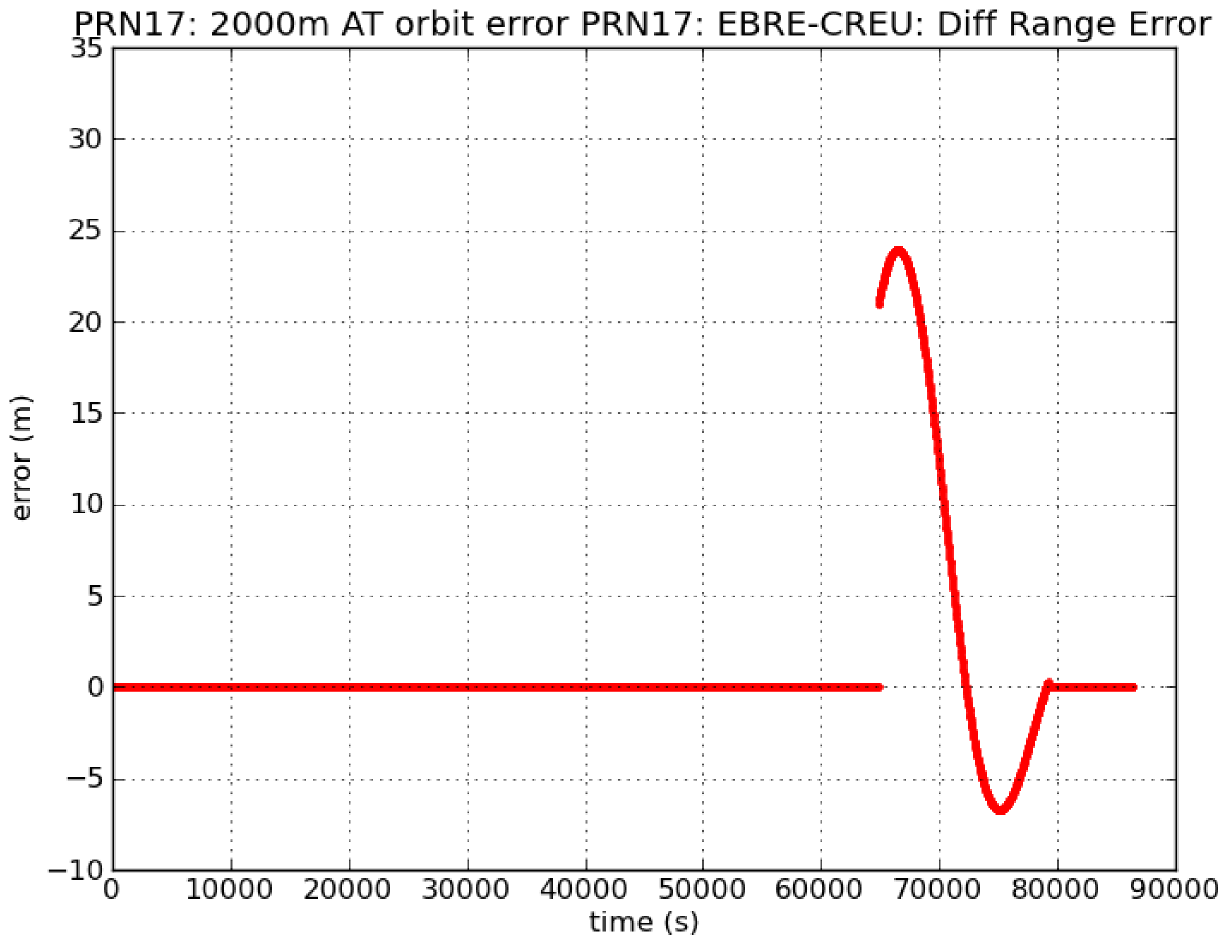
Using the previous files **drang.ebre** and **drang.creu**, compute the differential range orbit error of PRN17 between the receivers EBRE and CREU

```
cat drang.ebre drang.creu | gawk '{i=$1*1;if (length(r[i])==0)
            {r[i]=$2}else{printf "%s %16.6f \n", i,$2-r[i]}}' > ddrang.creu_ebre
```

Plot the results

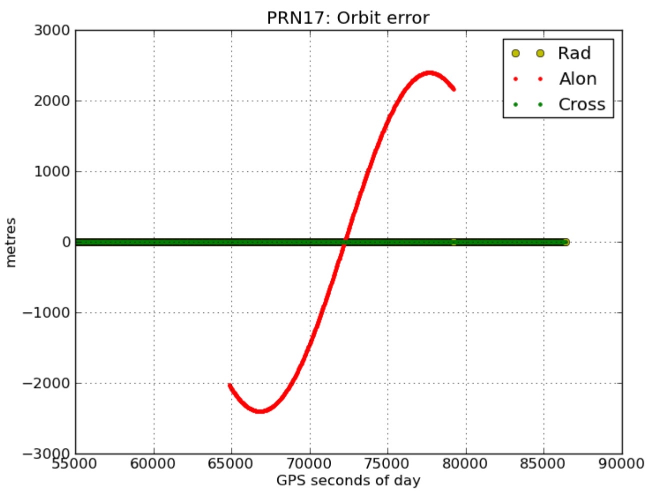
```
graph.py -f ddrang.creu_ebre  -x1 -y2 -s. --cl r --yn -10 --yx 35
         --xl "time (s)" --yl "error (m)"
         -t"PRN17: 2000m AT orbit error PRN17: EBRE-CREU: Diff Range Error"
```

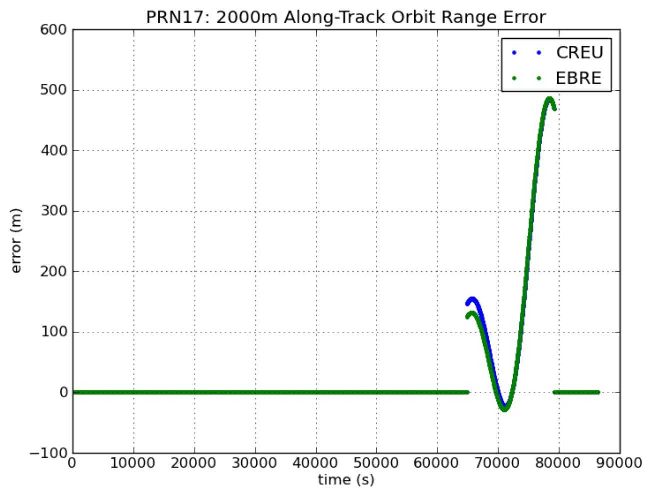
B3. Orbit error: Differential positioning
Range Domain Orbit Error

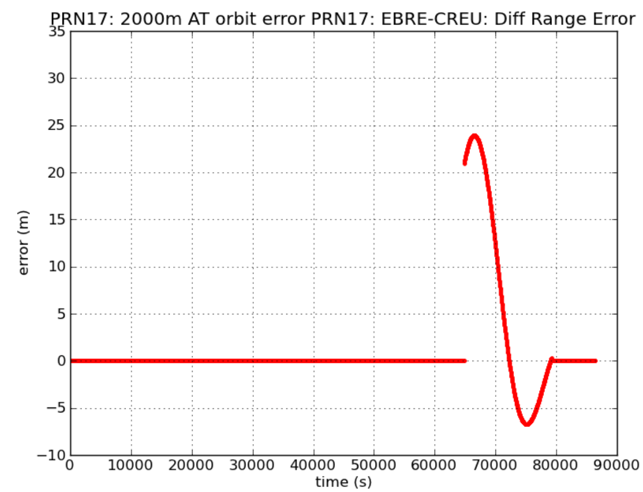


B3. Orbit error: Differential positioning
Range Domain Orbit Error

Results comparison

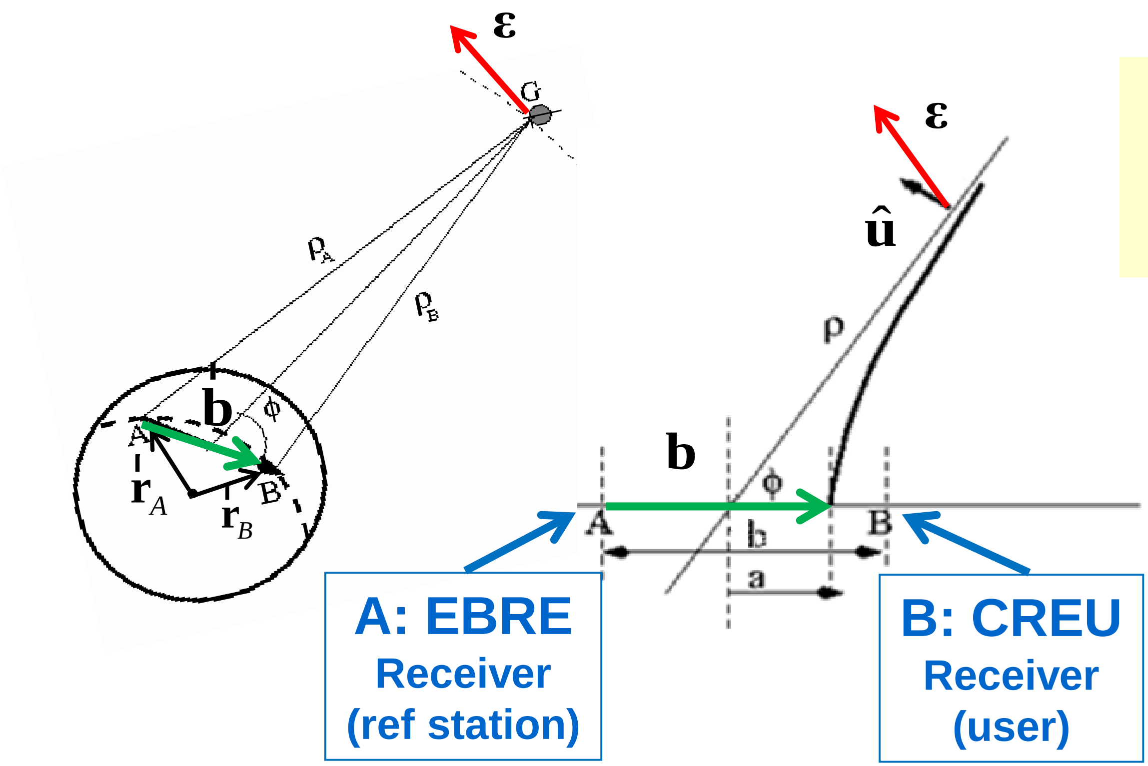






B.3.4. Prediction of the differential range orbit error

Verify the next expression, which relates the orbit error ε with the differential range error $\delta\rho$:



$$\delta\rho = -\frac{b}{\rho}\boldsymbol{\varepsilon}^T \cdot \mathbf{u} = -\frac{b}{\rho}\boldsymbol{\varepsilon}^T \cdot \left[\hat{\mathbf{b}} - \hat{\boldsymbol{\rho}}\left(\hat{\boldsymbol{\rho}}^T \cdot \hat{\mathbf{b}}\right)\right]$$

Baseline vector

$$\mathbf{b} = \mathbf{r}_B - \mathbf{r}_A$$

$$\mathbf{r}_A = [4833520.1197,\ 41537.2015,\ 4147461.6263]$$

$$\mathbf{r}_B = [4715420.3054,\ 273177.8809,\ 4271946.7957]$$

B3. Orbit error: Differential positioning
Range Domain Orbit Error

The following procedure can be applied:

1.- Compute the orbit error ε vector from the original and corrupted orbits:

```
gLAB_linux -input:nav brdc0770.10n -pre:dec 30 |grep SATPVT
                |gawk '{if ($6==17) print $4,$7,$8,$9}' > xyz.brdc

gLAB_linux -input:nav brdc0770.10nERR -pre:dec 30 |grep SATPVT
                |gawk '{if ($6==17) print $4,$7,$8,$9}'> xyz.brdcERR
```

2.- From previous results, generate a file **err.dat** with the orbit error (ε) vector and the Line-Of-Sight (ρ) from CREU receiver (with coordinates [4715420.3054, 273177.8809, 4271946.7957]), according to format: **err.dat=[sec** ε_x ε_y ε_z ρ_x ρ_y ρ_z**]**

```
cat xyz.brdc xyz.brdcERR|awk 'BEGIN{xb=4715420.3054;yb=273177.8809;zb= 4271946.7957}
            {i=$1*1;dx=$2-xb;dy=$3-yb;dz=$4-zb;rho=sqrt(dx**2+dy**2+dz**2);
           if (length(x[i])==0){x[i]=$2;y[i]=$3;z[i]=$4}
      else{printf "%6i %16.6f %16.6f %16.6f %16.6f %16.6f %16.6f %16.6f \n",
                     i,$2-x[i],$3-y[i],$4-z[i],dx,dy,dz,rho}}' |sort -n -k +1> err.dat
```

3.- Using the expression:

$$\delta\rho = -\frac{b}{\rho}\boldsymbol{\varepsilon}^T\cdot\mathbf{u} = -\frac{b}{\rho}\boldsymbol{\varepsilon}^T\cdot\left[\hat{\mathbf{b}} - \hat{\boldsymbol{\rho}}\left(\hat{\boldsymbol{\rho}}^T\cdot\hat{\mathbf{b}}\right)\right]$$

Baseline vector

$$\mathbf{b} = \mathbf{r}_B - \mathbf{r}_A$$

and the receivers' coordinates:

EBRE: $\mathbf{r}_A = [4833520.1197,\ 41537.2015,\ 4147461.6263]$

CREU: $\mathbf{r}_B = [4715420.3054,\ 273177.8809,\ 4271946.7957]$

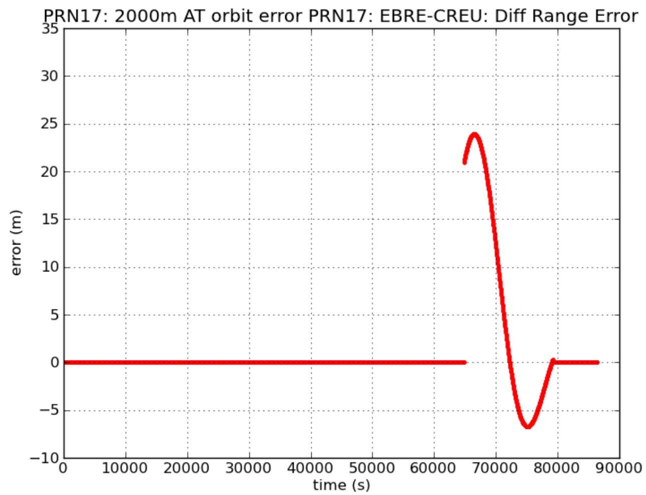
Compute the predicted differential orbit error $d\rho$ for the receiver CREU relative to EBRE station.

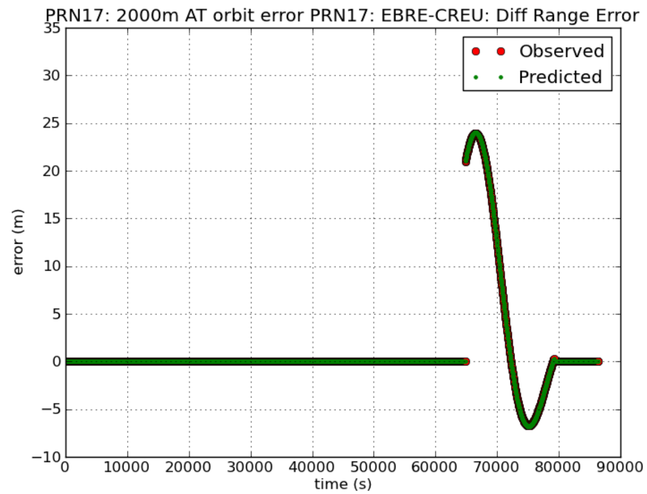
```
cat err.dat|gawk 'BEGIN{xa=4833520.1197;ya=41537.2015;za= 4147461.6263;
    xb=4715420.3054;yb=273177.8809;zb= 4271946.7957; bx=xb-xa; by=yb-ya; bz=zb-za;
    b=sqrt(bx*bx+by*by+bz*cz)} {rtc=($5*bx+$6*by+$7*bz)/$8/b;ux=bx/cb-$5/$8*rtc;
    uy=by/b-$6/$8*rtc;uz=bz/b-$7/$8*rtc;dr=-b/$8*($2*ux+$3*uy+$4*uz);
                                           printf "%6i %16.6f \n", $1,dr}' > rang.err
```

B4. Orbit error: Differential positioning
Range Domain Orbit Error

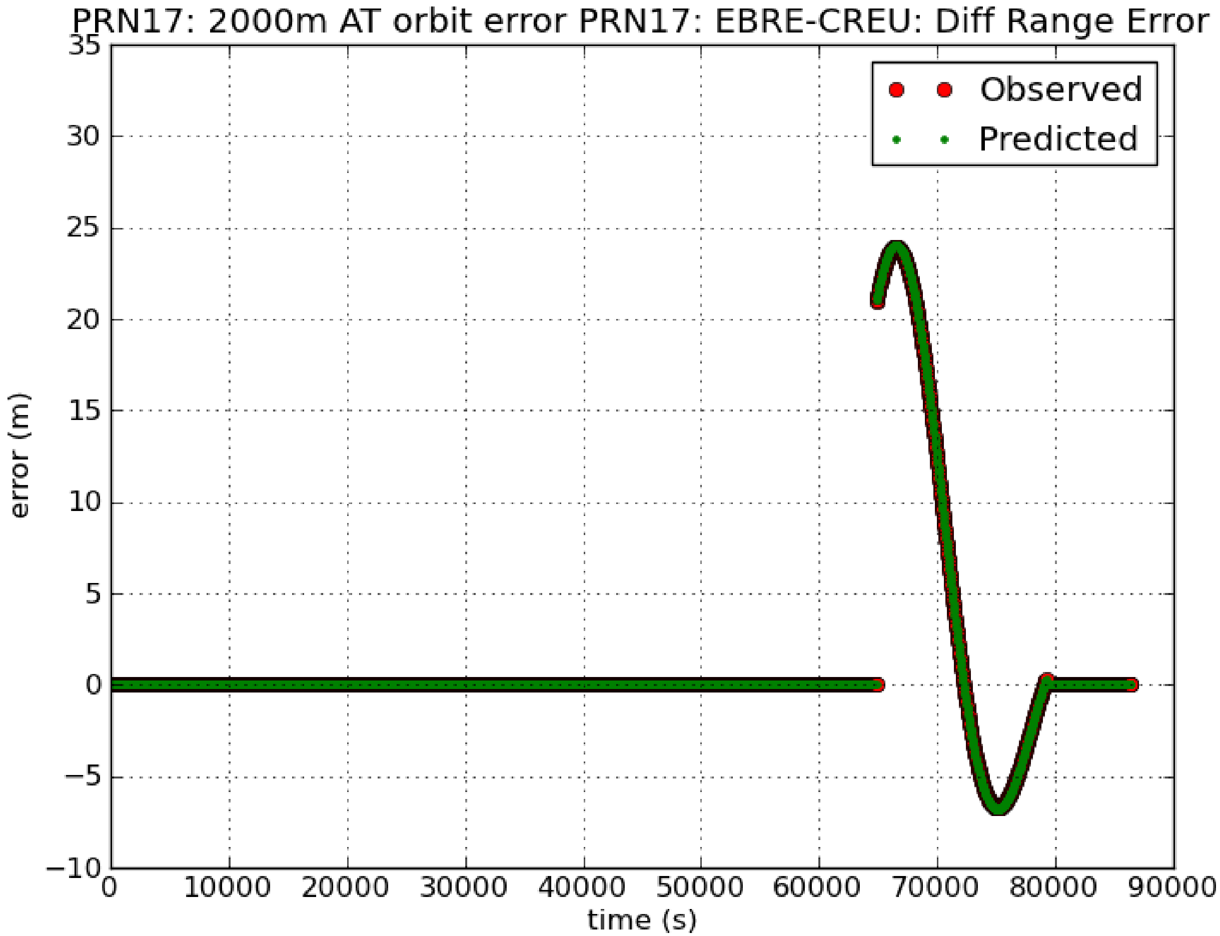
Plot the results and compare with the result found in the previous exercise B.3.3 (i.e. with those of file `ddrang.creu_ebre`):

```
graph.py -f ddrang.creu_ebre -x1 -y2 -so --cl r -l "Observed"
         -f rang.err -x1 -y2  -l "Predicted"
         --xl "time (s)" --yl "error (m)"
         -t"PRN17: 2000m AT orbit error PRN17: EBRE-CREU: Diff Range Error"
         --yn -10 --yx 35
```





B3. Orbit error: Differential positioning
Range Domain Orbit Error



Thanks for your attention

Acknowledgements

- To Institut Cartografic de Catalunya for the data files of receivers EBRE, CREU, GARR and MATA.
- To Adrià Rovira-Garcia for his contribution to the editing of this material and ***gLAB*** updating and integrating this learning material into the **GLUE**.

The ESA/UPC GNSS-Lab Tool suite (*gLAB*) has been developed under the ESA Education Office contract N. P1081434.