Static and Kinematic Precise Point Positioning with gd2p.pl (GNSS Data 2 Position)

Shailen Desai and Willy Bertiger

Near Earth Tracking Systems and Applications Groups
Jet Propulsion Laboratory, California Institute of Technology

March 4, 2014

Jet Propulsion Laboratory
California Institute of Technology

Introduction to gd2p.pl

- gd2p - GNSS data 2 Position
- High Level Gipsy interface for processing data from a single GPS receiver (or GLONASS).
  - gd2p.pl is the preferred black box for single receiver data
  - Easy to make changes to existing run, (e.g. run_again)
  - Better error detection
    - Small output to STDERR with early detection of errors.
  - Static Point Positioning
  - Kinematic Point Positioning
    - Use care in initialization

Examples of gd2p Applications

- Precise point positioning: static and kinematic
  - Single receiver ambiguity resolution (gd2p.pl with wlpb files.)
    - Resolves integer biases in phase measurements to improve accuracy.
  - Network solutions using smapper and resolvemerged.
    - Described in Release Notes for version 6.1.2.
- Troposphere estimation.
- Automatic Precise Point Positioning Service (APPS).
  - http://apps.gdgps.net
- Precise orbit determination.
  - Low Earth Orbiters
    - Used operationally for Jason-1, GRACE, CHAMP, SACC, COSMIC, Jason-2.
- Past applications also include:
  - Shuttle Radar Topography Mission.
  - Antarctic Ice Flow Studies

Help for gd2p Flags

- gd2p.pl –h | more
  - Gives the entire help/man; written in POD (Plain Old Documentation)
- gd2p.pl –h_sub
  - Gives a listing of subtopics.

  -h_global – Help on flags that apply to multiple steps

  -h_edit – Help on editing data

  -h_model – Help on qregres/measurement model

  -h_wash – Help on filter/smoother

  -h_orbit – Help on orbiter solution options

  -h_examples – Example runs

  -h_files – Input/output file information

  -h_env – Help on environment variables
- **Beginners should start with –h_examples.**

Point Positioning Examples from gd2p.pl –h_examples

- gd2p.pl –h_examples | more
  - Yields many examples

- 5-minute data, static position, nominal position from the station database ($GOA_VAR/sta_info), troposphere, clock, and position adjusted.
  - **Same example from GIPSY Overview package.**

  (gd2p.pl -i $GOA/class_gd2p/gol20100.10o -n GOL2 -r 300 -type s –d 2010-01-10 > gd2p.log) > & gd2p.err &

  - Simple command to automatically invoke the necessary GIPSY modules and to generate the required input namelist files.
  - Motivating reasons for gd2p.
    - Point positioning should be simple.
    - Input RINEX and output precise point position solution.

gd2p flag Details

- Details on flags are easy to find in the help.
  - **gd2p.pl –h | more**
  - Search for "-*flag*" where *flag* is the appropriate string.

  - For example, if you look for –tides:

-tides [WahrK1 PolTid FreqDepLove OctTid]

Adds solid tide models to the station qregres namelist.

Default tide model includes WahrK1 and PolTid. WahrK1 is a subset of FreqDepLove. Any input will override the default values with the input list. To turn off all the solid tides just use the option with an empty list (e.g., -tides).

gd2p Files (-h_files)

- gd2p produces many files.

  gd2p.pl –h_files | more
- **Also refer to GIPSY Overview package for description of many of these files.**

Defining a Base Working Directory for Examples

- In this package, examples assume a working directory defined by the user with **absolute** path name: ***WORKDIR***.
- Examples will be relative to this user-defined working directory.

mkdir *WORKDIR*

cd *WORKDIR*

Fetch JPL Orbit/Clock Products for Examples

- gd2p.pl by default will fetch orbit/clock products from JPL's host server.
  - See –orb_clk flag.
- To avoid ftp fetch for same products, download files needed for examples one time only.

mkdir jplorbclk

cd jplorbclk

goa_prod_ftp.pl -d 2010-01-10 -s flinnR –hr

– Or copy output from example from $GOA:
cp $GOA/class_gd2p/orbits_forclass/* .

ls

2010-01-10.ant.gz 2010-01-10.frame.gz 2010-01-10.pos.gz 2010-01-10.tdp.gz
2010-01-10.eo.gz 2010-01-10_hr.tdp.gz 2010-01-10.shad.gz 2010-01-10.wlpb.gz

Example 1: Receiver Position and Clock Only

- Basic example, adjusting position and clock only.
  - **WARNING: NOT advisable to ignore troposphere.**
    - **This example is just a starting point for other examples.**

cd ..

mkdir spp1

cd spp1

(gd2p.pl –i $GOA/class_gd2p/gol20100.10o -n GOL2 -d 2010-01-10 -r 300 -type s -w_elmin 15 -e "-a 20 -PC –LC -F” -pb_min_slip 1.0e-3 -pb_min_elev 30 -amb_res 2 -dwght 1.0e-5 1.0e-3 -post_wind 5.0e-3 5.0e-5 -trop_z_rw -1.0 –tides –orb_clk “flinnR *WORKDIR*/jplorbclk” > gd2p.log) >& gd2p.err

  - Could also use defaults for many of these options.
    - Defaults intentionally not used to highlight options.

Example 1: Summary of options used.

Option	Description (Required options in bold, others have defaults)
-i	**Input RINEX file**
-n	**Name of Receiver**
-d	**Day to process. Used to define daily orbit/clock product to use.**
-r	Measurement data rate to output from data editor for the filter.
-type	Type of positioning (s for static, k for kinematic)
-w_elmin	Minimum elevation angle cutoff. (usually 7 or 15 degrees)
-e	Quoted string of flags to pass to data editor. This example says used minimum arc length of 20 minutes, LC and PC data
-dwght	**Phase and range** data weights in km. Usually LC=1 cm and PC=1 meter.
-post_wind	Final postfit **range and phase** window in km for edit point cycle. (5-sigma)
-pb_min_slip	Minimum slip for inserting phase break in postbreak (km)
-pb_min_elev	Minimum elevation angle at which postbreak criteria are applied.
-amb_res	Number of iterations for ambiguity resolution with wlpb file.
-trop_z_rw	Random walk parameter for zenith troposphere. Negative means not estimated.
-tides	Tide models to apply to station model. Empty list means no tides modeled.
-orb_clk	Defines type and location of orbit and clock products for transmitting satellites.

Be Careful of Reverse Order

Example 1: Which Orbit/Clock Products

- gd2p defaults to automatically downloading and using JPL's final (flinnR) orbit/clock products.
  - Equivalent to using –orb_clk flinnR option.
  - In this example, pointed to a local directory containing required "flinnR" orbit/clock products.
- Will automatically apply transmitter antenna calibrations identified in YYYY-MM-DD.ant.gz file in product package for that day.
- Other possibilities:
  - Use JPL's Rapid (qlR) or Ultra-Rapid products (ultra)
  - Explicitly point to a local archive of products with the same naming convention of the product type.
  - See –orb_clk option.

Example 1: Useful output from gd2p

- *gd2p.log* provides summary of GIPSY modules executed.
- gd2p.err provides summary of any error or warning messages.
- *logs* directory provides output logs from some of the executed GIPSY modules.
  - **Refer to GIPSY Overview for more details about executed modules.**
- *run_again* script created by gd2p.pl.
  - Can be used to duplicate this run exactly with all options used, e.g.,

    cat run_again
  - Could either edit this file or make a copy, modify options, and re-run.
    - Will do this in following gd2p examples.

Example 1: First look at run

- First sign of trouble is existence of file "EDIT_POINT_FAILURE"
  - The edit point "wash" cycle failed to converge.
  - See also gd2p.err

  cat gd2p.err

    - Indicates edit point failed to converge.

- Potential sign of trouble is large phase (LC) post-fit residuals and large number of phase (LC) outliers.
  - Typical LC RMS of residuals is < 1 cm.
  - Typical PC RMS of residuals is 20-80 cm.

  cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	1.5847E-05	1938	230
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.8335E-04	2168	0

Jet Propulsion Laboratory
California Institute of Technology

Example 1: Check postfit residuals

- Generate tables of LC and PC postfit residuals.

  residuals.pl

  cat residuals.txt | awk '{if ($4 == 120) print $0}' | cl h1 5 > spp1_lc_res.txt

  cat residuals.txt | awk '{if ($4 == 110) print $0}' | cl h1 5 > spp1_pc_res.txt

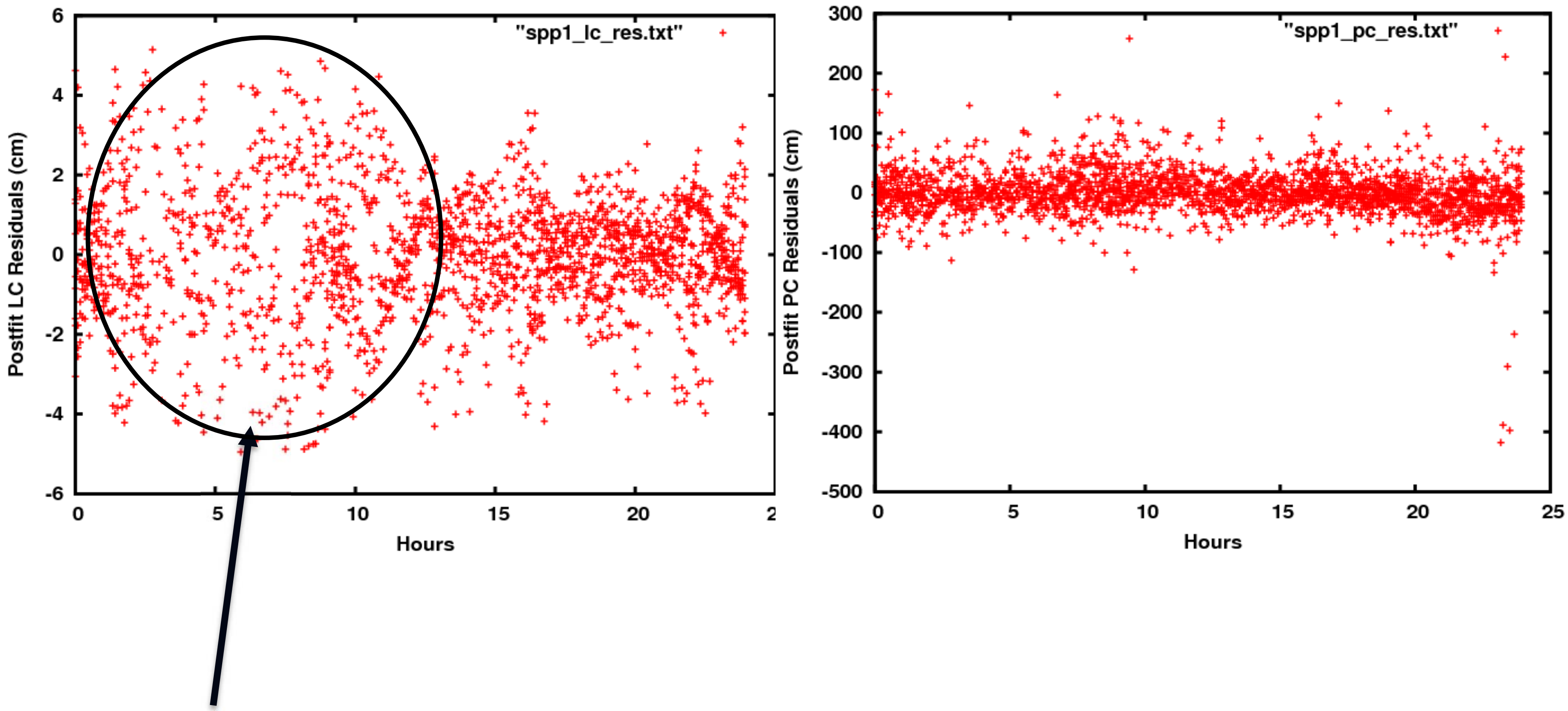
  - Same approach used to generate sppN_XX_res.txt files for future examples.
  - NOTE: **cl h1 5** equivalent to **cl '(c1-f1)/3600' 5**
    - hN means use column N to determine time in hours with respect to time on first line.

- Generate plots of LC and PC postfit residuals.

  gnup –p spp1_lc_res.txt –xl 'Hours' –yl 'Postfit LC Residuals (cm)'

  gnup –p spp1_pc_res.txt –xl 'Hours' –yl 'Postfit PC Residuals (cm)'

Example 1: Plots of postfit residuals



- Higher residuals in first half of the day are suspicious.
- cm-level LC residuals is higher than expected.

Example 1: Where is the solution

- Note: Two solutions actually generated by gd2p because non-zero –amb_res option used.
  - 1: Pre-ambiguity resolution solution.
    - Post-fit RMS of residuals in Postfit.sum, and solution in tdp_final_pre_amb
  - 2: Post-ambiguity resolution solution.
    - Post-fit RMS of residuals in Postfit_amb.sum, and solution in tdp_final
- TDP files (tdp_final, tdp_final_pre_amb) provide all parameters that were adjusted (estimated) from the provided nominals.
  - Nominal (provided first guess) in column 2.
  - Solution (sum total of nominal and estimated adjustment) in column 3.

Example 1: Content of TDP file

- TDP files indicate which parameters were estimated.
  - Minimum set of parameters in static PPP gd2p run:
    - Receiver clock defined by **STA BIAS (km)**
    - Phase biases defined by **PB GPSXX (km)**
    - Station position in **STA X, STA Y, STA Z (km)**
  - Anything else?

  grep -v "STA BIAS" tdp_final | grep -v "PB GPS" | grep -v "STA [XYZ]"
  - Only dummy parameter left, "DUM", used by filter.

Example 2: Add zenith troposphere estimation

- Add estimation of zenith troposphere.

  cd ..

  mkdir spp2

  cd spp2

  cp ../spp1/run_again r2 # Copy run_again, and modify as needed.

  – Edit script r2, and change zenith troposphere line to:

  -trop_z_rw 5.0e-8 \

  – Experience has shown recommended random walk value of 5.0e-8 km/sqrt(sec) for static positioning.

  – Execute script

    ./r2

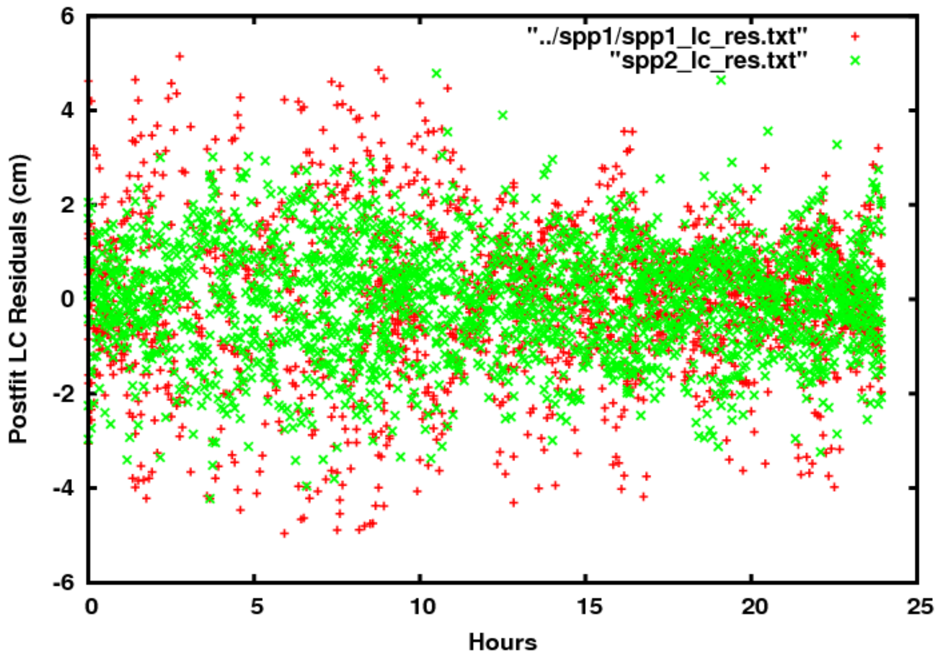
Example 2: First look at run

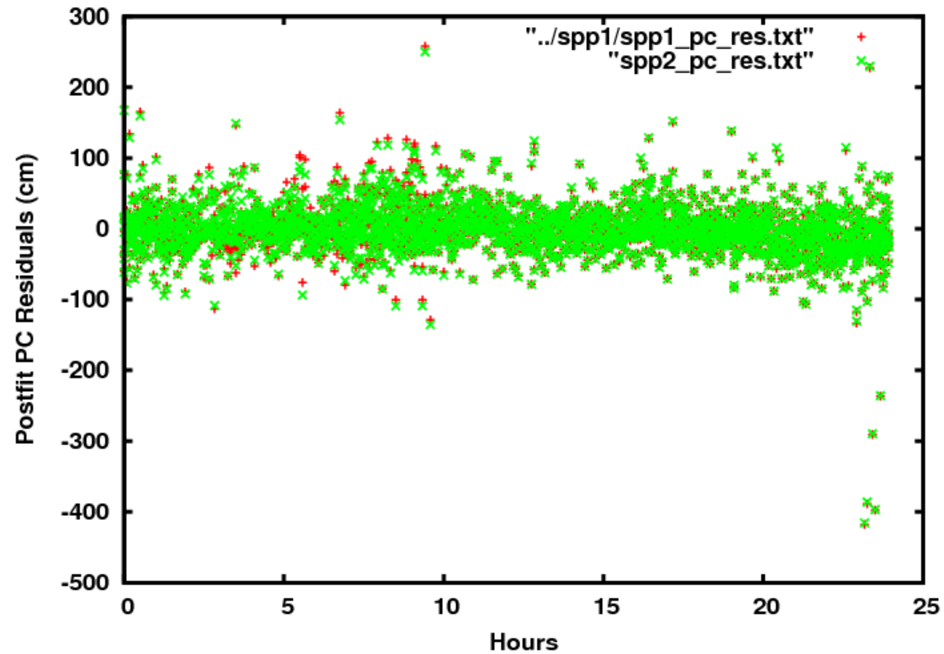
- Edit point cycle seems to have converged – Good news.
  - No "EDIT_POINT_FAILURE" file.
  - No errors in gd2p.err file.
    - Just some warnings that the regres file (rgfile) was modified from its original qregres output.
      - Expected with outlier editing and ambiguity resolution.
- Postfit phase (LC) residuals are significantly smaller AND many fewer outliers.
  - Estimating more parameters (troposphere) might explain smaller residuals, but still good news.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	1.1738E-05	2157	11
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7924E-04	2168	0

Example 2: Plots of postfit residuals





gnup -p ../spp1/spp1_lc_res.txt spp2_lc_res.txt -xl 'Hours' -yl 'Postfit LC Residuals (cm)’

gnup -p ../spp1/spp1_pc_res.txt spp2_pc_res.txt -xl 'Hours' -yl 'Postfit PC Residuals (cm)'

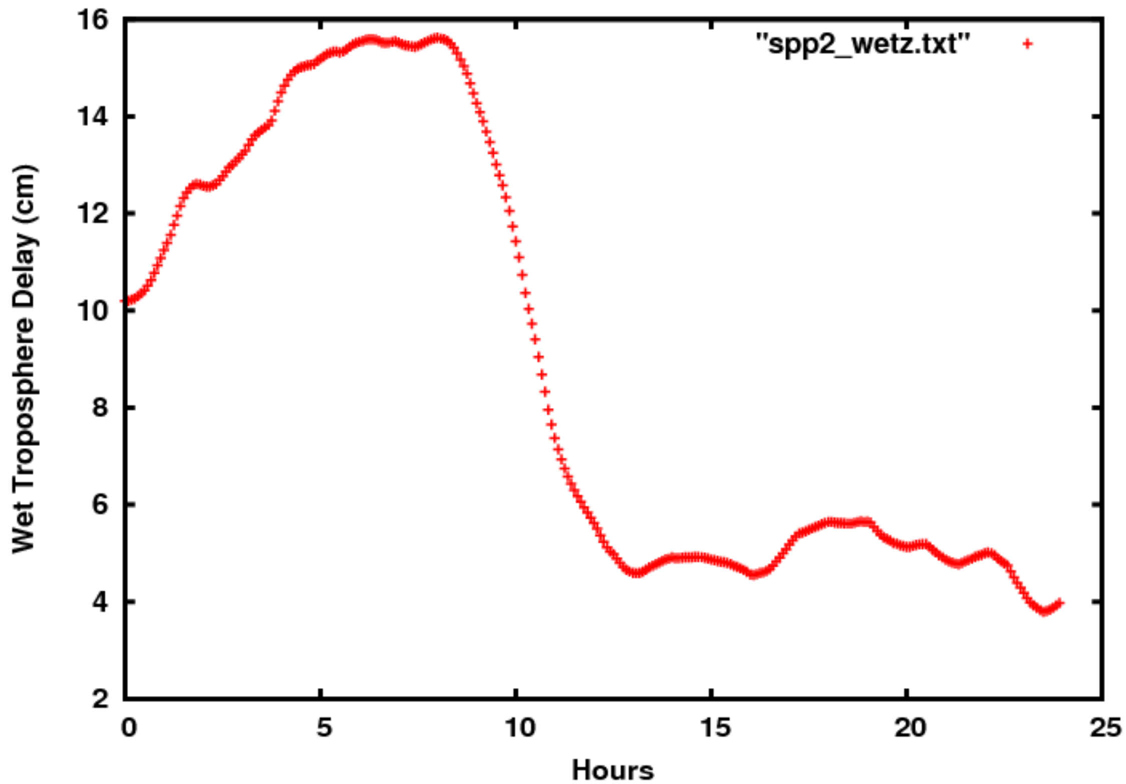
- Scatter of residuals improved in first half of day.
  - Probably have wet troposphere conditions.

Example 2: Plot Troposphere Solution

Units of troposphere parameters in tdp file is m. All other parameters use km.

grep WETZ tdp_final | cl h1 '1.0e2*c3' > spp2_wetz.txt

gnup -p spp2_wetz.txt -xl 'Hours' -yl 'Wet Troposphere Delay (cm)'



Example 2: What changed?

- Compare qregres namelists from Examples 1 and 2.

  diff qregres.nml ../spp1/qregres.nml

  – No changes, except generation time.

  – Means that none of the models changed.

- Compare wash namelists from Examples 1 and 2.

  diff wash.nml ../spp1/wash.nml

```
<      WETZTROP  = .TRUE.
<      TROPDRIFT = 5.0e-8
---
>      WETZTROP  = .FALSE.
>      TROPDRIFT = -1.0
```

  – Zenith troposphere estimation turned on in example 2.

Example 2: First look at solution

- What were the additional parameters estimated?

  grep -v "STA BIAS" tdp_final | grep -v "PB GPS" | grep -v "STA [XYZ]"

  – Zenith troposphere estimates, **WETZTROP**, now in TDP file.

  – WETZTROP has units of meters.

  – **NOTE: This parameter includes zenith wet troposphere and any error in the nominal value of the dry troposphere.**

  – Nominal value of dry troposphere for this run is 2.0507 meters.

    - More on this later.

  grep DRY qregres.nml

  NMLTROP_DRY(1) =   2.0507

  – NOTE: Default NIELL troposphere mapping function used in run.

    - Maps line-of-sight to zenith.

  grep NMLTropMap qregres.nml

  NMLTropMap(1) = 'NIELL'

Example 3: Use Tide Models

- Add solid Earth tide models for station position.

  cd ..

  mkdir spp3

  cd spp3

  cp ../spp2/run_again r3 # Copy run_again, and modify as needed.

  – Edit script r3, and change tide models line.

  -tides WahrK1 FreqDepLove OctTid PolTid \

  – WahrK1, FreqDepLove, and OctTid invoke complete solid Earth tide models from IERS standards.

    - Note: WahrK1 is just a subset of FreqDepLove

  – PolTide invokes solid Earth pole tide from IERS standards.

  – Execute script

    ./r3

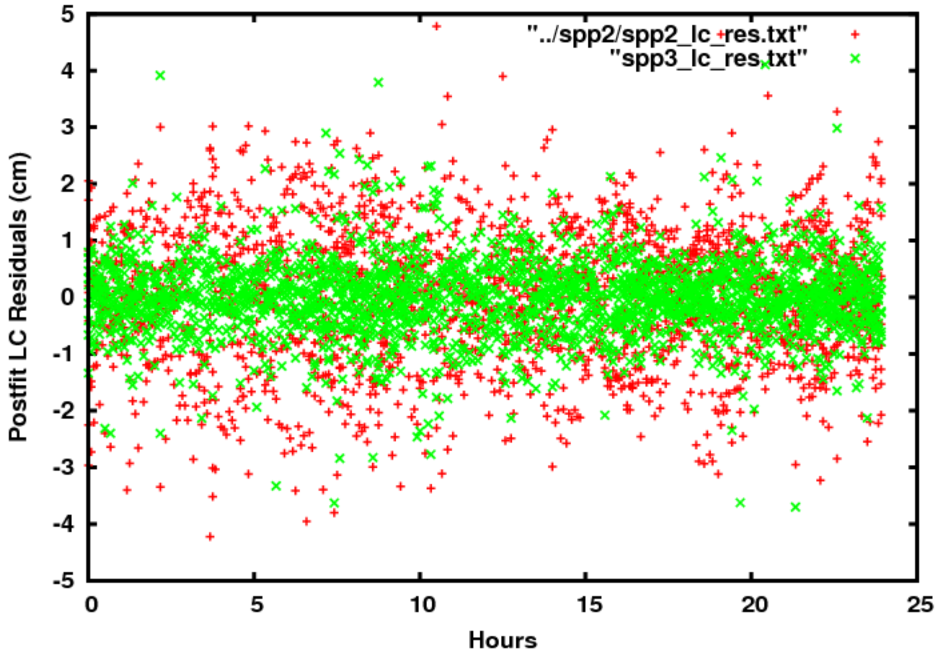
Example 3: Looking at run

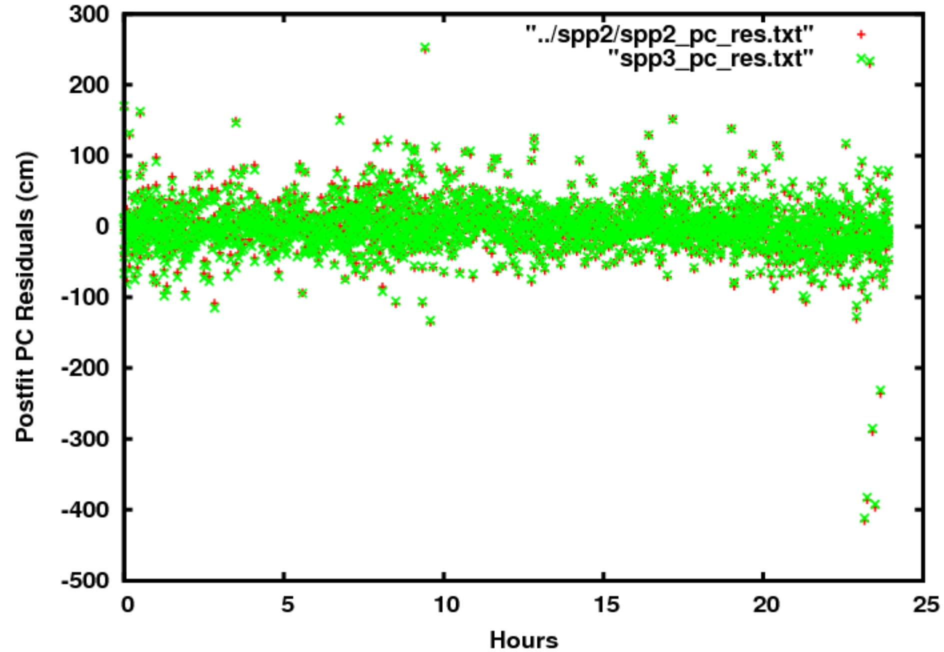
- Edit point cycle converged – Good news.
  - No “EDIT_POINT_FAILURE” file.
  - No errors in gd2p.err.
- Postfit phase (LC) residuals are significantly smaller AND fewer outliers.
  - No additional parameters estimated, so smaller residuals and fewer outliers indicates tide models are helping.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.8493E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7494E-04	2168	0

Example 3: Plots of postfit residuals





gnup -p ../spp2/spp2_lc_res.txt spp3_lc_res.txt -xl 'Hours' -yl 'Postfit LC Residuals (cm)'

gnup -p ../spp2/spp2_pc_res.txt spp3_pc_res.txt -xl 'Hours' -yl 'Postfit PC Residuals (cm)'

- Reduced (improved) scatter of LC residuals.

Example 3: What changed?

- Compare qregres namelists from Examples 2 and 3.

  diff qregres.nml ../spp2/qregres.nml

  ```
  <  NMLFreqDepLove = .TRUE.
  <  NMLOctTid = .TRUE.
  <  NMLPolTid = .TRUE.
  ---
  >  NMLSolidTideOn( 1 ) = .FALSE.
  ```

  – Solid earth tide models turned on in example 3.
    - WahrK1 not explicitly found because it is a subset of FreqDepLove.

- Compare wash namelists from Examples 2 and 3.

  diff wash.nml ../spp2/wash.nml

  – No differences.
    - No change to estimation strategy.

Example 3: First look at solution.

- Verify no additional parameters estimated:

  grep -v "STA BIAS" tdp_final | grep -v "PB GPS" | grep -v "STA [XYZ]” | grep –v “WETZ”

  – Only dummy parameter left, “DUM”, used by filter.

Example 4: Add ocean load tide model

- Add ocean load tide model for station position.

cd ..

mkdir spp4

cd spp4

cp ../spp3/run_again r4 # Copy run_again, and modify as needed.

– Edit script r4, and add two lines.

-add_ocnld \

-OcnldCpn \

– Execute script

./r4

Example 4: Ocean load tide flags

- -addocnld adds ocean load tide coefficients to qregres.nml.
  - 11 tidal frequencies, M2, S2, K2, N2, K1, O1, P1, Q1, Mf, Mm, Ssa.
  - NOTE: Default uses ocean load tide coefficients from:
    - $GOA_VAR/sta_info/ocnld_coeff_cm_got48ac_wtpxo8ofunc
  - Could point to your own database by using the following option:
    - **-add_ocnld "-c your_own_ocean_loading_coefficient_file"**
    - **E.g. use http://holt.oso.chalmers.se/loading/**
  - NOTE: When using JPL's orbit and clock products be sure to use load tide coefficients with respect to center of mass of solid Earth + oceans.
    - JPL's orbit/clock products are delivered with respect to center of mass of total Earth system.
- -OcnldCpn uses 11 tidal frequencies to infer other frequencies (hardisp.f from IERS standards)

Example 4: First look at run.

- No errors in gd2p.err.
- Additional reduction in RMS of postfit phase (LC) residuals, without estimating any new parameters.
  - Load tide model seems to add value.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.6587E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7503E-04	2168	0

Example 4: What changed?

- Compare qregres namelists from Examples 3 and 4.

  diff qregres.nml ../spp3/qregres.nml

  – -add_ocnld adds the amplitude and phase of ocean load tides at 11 frequencies:

  NMLOceanLoad(1) = .true.

  NMLOcnLoad_Amp(1, *, 1)

  NMLOcnLoad_Phs(1, *, 1)

  – -OcnldCpn adds the following:

  NMLOcnldCpn = .TRUE.

  – Presently invoking IERS hardisp subroutine.

- Compare wash namelists from Examples 3 and 4.

  diff wash.nml ../spp3/wash.nml

  – No differences.

    - No change to estimation strategy.

Jet Propulsion Laboratory
California Institute of Technology

Example 4: First look at solution

- Verify no additional parameters estimated:

  grep -v "STA BIAS" tdp_final | grep -v "PB GPS" | grep -v "STA [XYZ]" | grep –v "WETZ"

  – Only dummy parameter left, "DUM", used by filter.

Example 5: Add ocean load pole tide model

- Add ocean load pole tide model for station position.
  - Displacement of crust due to load of the pole tide in the oceans.
    - Primarily has 12 and 14 month periods.

cd ..

mkdir spp5

cd spp5

cp ../spp4/run_again r5 # Copy run_again, and modify as needed.

- Edit script r5, and add two lines.

-add_ocnldpoltid \

- Execute script

./r5

Example 5: First look at run.

- No errors in gd2p.err.
- Very small increase in post-fit LC residuals at level of 1e-4 mm.
  - May not be statistically significant.
  - Expected because ocean load pole tide effect is < 1 mm and has periods of 12 and 14 months.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.6588E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7503E-04	2168	0

Example 5: What changed?

- Compare qregres namelists from Examples 4 and 5.

  diff qregres.nml ../spp4/qregres.nml

  – -add_ocnldpoltid turns on the model, and adds a path to a file containing a map of the coefficients for the model.

  – See $GOA/file_formats/ocnld_pole_tide_map for description of ocean load pole tide coefficient file.

  NMLOcnldPolTid = .TRUE.

  NMLOcnldPolTidMap = '/GIPSY_source/goa-var/sta_info/oceanpoletide_load_0.5deg.txt'

- Compare wash namelists from Examples 4 and 5.

  diff wash.nml ../spp4/wash.nml

  – No differences.

  - No change to estimation strategy.

Example 5: First look at solution

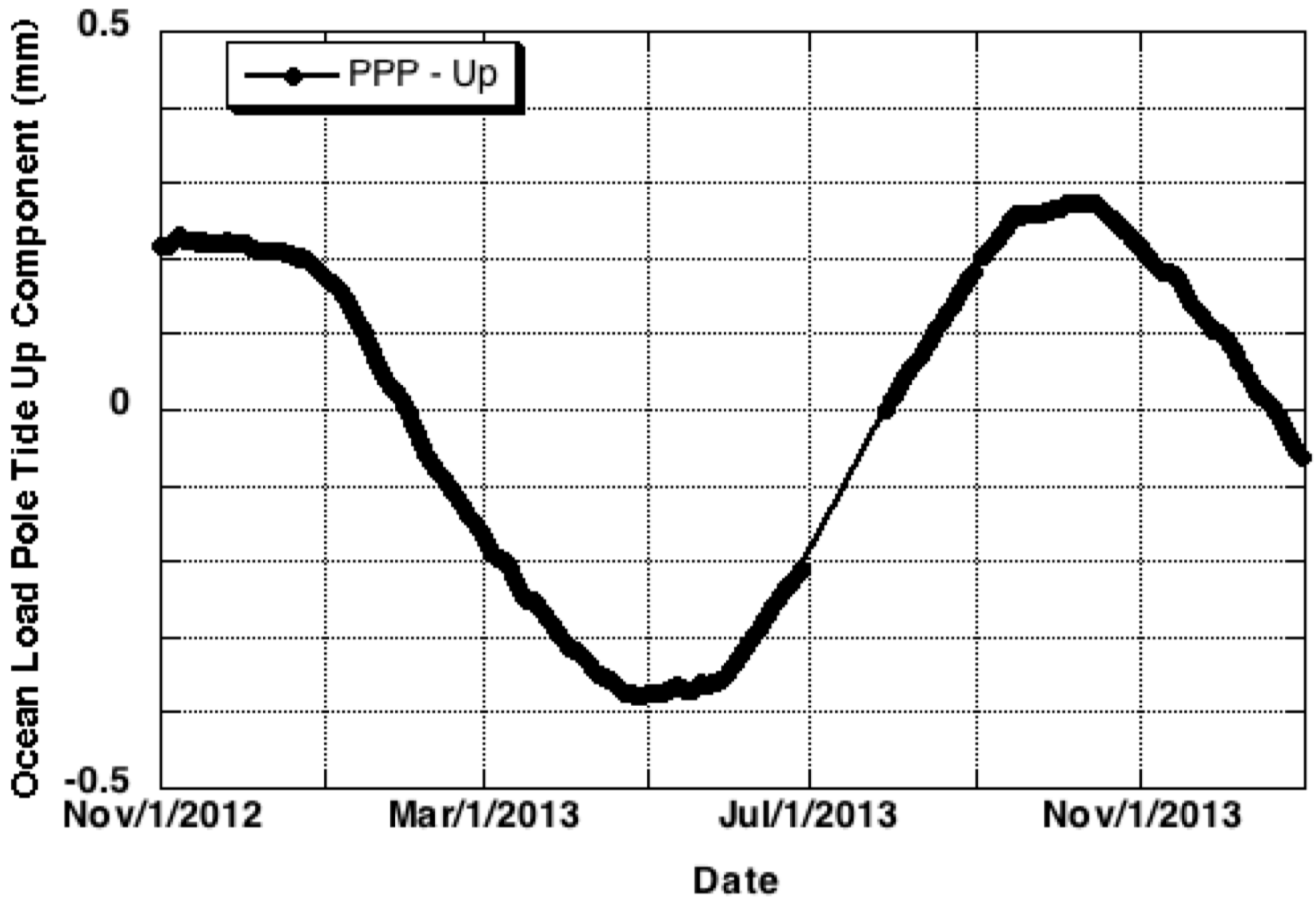
- Verify no additional parameters estimated:

  grep -v "STA BIAS" tdp_final | grep -v "PB GPS" | grep -v "STA [XYZ]" | grep –v "WETZ"

  – Only dummy parameter left, "DUM", used by filter.

Example 5: Effect of ocean load pole tide

- Effect is typically at the sub-mm level, primarily with periods of 12 and 14 months, e.g., Easter Island (ISPA).



Example 6: Add troposphere gradients

- Add estimation of troposphere gradients.

  cd ..

  mkdir spp6

  cd spp6

  cp ../spp5/run_again r6   # Copy run_again, and modify as needed.

  – Edit script r6, and add one line.

  -wetzgrad 5.0e-9 \

  – Experience has shown recommended random walk value of 5.0e-9 km/sqrt(sec) for static positioning.

  – Execute script

    ./r6

Example 6: First look at run

- No errors in gd2p.err.
- Additional improvement to postfit phase (LC) residuals.
  - Expected when estimating additional parameters.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.1667E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7481E-04	2168	0

Example 6: What changed?

- Compare qregres namelists from Examples 5 and 6.

  diff qregres.nml ../spp5/qregres.nml

  – No changes to any models.

- Compare wash namelists from Examples 5 and 6.

  diff wash.nml ../spp5/wash.nml

```
<   WETZGRAD  = .TRUE
---
>   WETZGRAD  = .FALSE.
```

  – Estimation of troposphere gradients turned on in example 6.

Example 6: First look at solution

- Check for additional parameters estimated:

  grep -v "STA BIAS" tdp_final | grep -v "PB GPS" | grep -v "STA [XYZ]” | grep –v “WETZ”

  – Troposphere gradient parameters are in TRPAZSIN and TRPAZCOS.

    - Some consider these to be parameters that absorb other error sources, others consider them to have scientific value.

      – Depends on application, processing approach, and station.

Example 7: Change troposphere mapping function

- Change troposphere mapping function.

  cd ..

  mkdir spp7

  cd spp7

  cp ../spp6/run_again r7   # Copy run_again, and modify as needed.

  – Edit script r7, and add one line.

  -trop_map GPT2\

  – GPT2 = Global Pressure and Temperature Mapping Function.
    - Shown to be better than NIELL.
    - Predecessor is GMF (Global Mapping Function), part of GPT (Version 1).
    - A better option is VMF1GRID (Vienna Mapping Function) or VM1SITE (site-specific Vienna Mapping Function).
      – Requires local database of VMF grid or site files, maintained by user.
      – Path to database specified by **–vmf1dir** option.

  – Execute script

  ./r7

Example 7: First look at run

- No errors in gd2p.err.
- Small reduction in postfit residuals for this station/day of 4e-4 mm.
  - May not be statistically significant.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.1663E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7481E-04	2168	0

Example 7: What changed?

- Compare qregres namelists from Examples 6 and 7.

  diff qregres.nml ../spp6/qregres.nml

  < NMLTropMap(1) = ’GPT2'

  > NMLTropMap(1) = 'NIELL'

  – Just the mapping function model changed.
- Compare wash namelists from Examples 6 and 7.

  diff wash.nml ../spp6/wash.nml

  – No differences.
    - No change to estimation strategy.
- Save zenith troposphere solution for comparison to next example.

  grep WETZ tdp_final | cl h1 '1.0e2*c3' > spp7_wetz.txt

Nominal Values of Zenith Troposphere

- Default troposphere nominals in qregres are:
  - Zenith dry troposphere (NMLTROP_DRY) = 2.0 m
  - Zenith wet troposphere (NMLTROP_WET) = 0.1 m
- If NMLStochMonment is .TRUE.:
  - Zenith dry troposphere: 1.013*2.27*exp(-0.116e-03*ht) meters.
  - Nominal station position height used to compute dry.
- If a TDP file with station position(s) is provided (STA [XYZ]):
  - Zenith dry troposphere: 1.013*2.27*exp(-0.116e-03*ht) meters.
  - Zenith wet troposphere: 0.1*exp(-ht/2000e0) meters.
- Nominal model can be improved and provided to gd2p with an input TDP file.
  - See: **tropnominal**

tropnominal – Models of nominal troposphere delay

- **tropnominal** generates times series of zenith dry and/or wet troposphere models:
  - STATIC: Dry based on exponential function
    - 1.013 * 2.27 * exp(-.116e-3 * h) meters
    - No wet.
  - GPT: Dry based on Global Pressure and Temperature climatological model (mean, annual, and 20-degree horizontal resolution)
    - http://ggosatm.hg.tuwien.ac.at/delay.html
    - No wet.
  - GPT2: Dry and wet using the GPT2 climatological model.
    - Mean, annual, semiannual, and 5-degree horizontal resolution.
  - VMF1GRID: Wet and dry based on Vienna mapping function:
    - http://ggosatm.hg.tuwien.ac.at/delay.html
    - Based on ECMWF meteorological data (post-processed)
    - User needs to maintain own database.
- **tropnominal** can accept time series of station positions.

tropnominal – Example of getting required inputs

- Determine appropriate inputs to cover observation window.
  - UTC start and stop times required.

cd ..

mkdir tropnominal

cd tropnominal

grep WETZ ../spp7/tdp_final | head -1 | cl 1 | gps2utc –

316353585.00000000

grep WETZ ../spp7/tdp_final | tail -1 | cl 1 | gps2utc –

316439685.00000000

  - Station lat, lon, and height required.
    - Grab from qreges namelist, for example from ../spp7/qregres.nml

```
!    lat           long          elev (m)
!    35.4251561   -116.8892511   987.0973
```

tropnominal - Example

- Execute tropnominal:

  tropnominal -n GOL2 -m GPT2 -latdeg 35.4251561 -londeg -116.8892511 -h_m 987.0973 -stsec 316353585.0 -endsec 316439685.0 -samp 300

  - By default the output files are defined according to the station name (-n option).
  - In this case, output files are GOL2.TDPdry and GOL2.TDPwet

- Make a combined file, and rename output files.

  cat GOL2.TDPdry GOL2.TDPwet > GOL2_GPT2.TDPdryandwet

  mv GOL2.TDPdry GOL2_GPT2.TDPdry
  mv GOL2.TDPwet GOL2_GPT2.TDPwet

- Execute tropnominal using the default static model applied in GIPSY, and rename output file.

  tropnominal -n GOL2 –m STATIC -latdeg 35.4251561 -londeg -116.8892511 -h_m 987.0973 -stsec 316353585.0 -endsec 316439685.0 -samp 300

  mv GOL2.TDPdry GOL2_STATIC.TDPdry

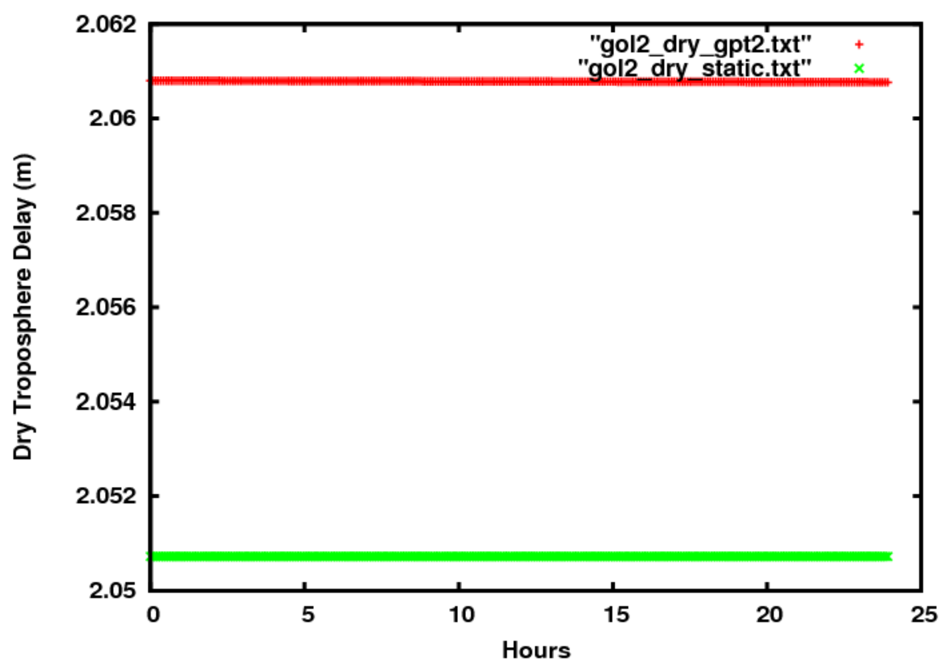
Comparison of tropnominal models

cat GOL2_GPT2.TDPdry | cl h1 3 > gol2_dry_gpt2.txt

cat GOL2_STATIC.TDPdry | cl h1 3 > gol2_dry_static.txt

gnup -p gol2_dry_gpt2.txt gol2_dry_static.txt -xl 'Hours' -yl 'Dry Troposphere Delay (m)'

- GPT2 model is ~ 1.0cm higher than GIPSY's default.
  - GPT2 is time-varying (although not obvious on this scale, and for this particular station-day)



Example 8: Provide apriori dry troposphere model (GPT2)

- Use GPT2 model for nominal dry troposphere delay.

  cd ..

  mkdir spp8

  cd spp8

  cp ../spp7/run_again r8 # Copy run_again, and modify as needed.

  – Edit script r8, and add one line.

  -tdp_in ../tropnominal/GOL2_GPT2.TDPdry \

  – Input TDP file provides nominal time series of dry troposphere delay.

  – Execute script

    ./r8

Example 8: First look at run

- No errors in gd2p.err.
- Small reduction to postfit residuals.
  - GPT2 has positive impact, since no change to estimation strategy

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.1648E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7481E-04	2168	0

Example 8: What changed?

- Compare qregres namelists from Examples 7 and 8.

  diff qregres.nml ../spp7/qregres.nml

  – No change to internal models.

  – **However, qregres command now ingests information from tdp file.**

- Compare wash namelists from Examples 7 and 8.

  diff wash.nml ../spp7/wash.nml

  – No differences.

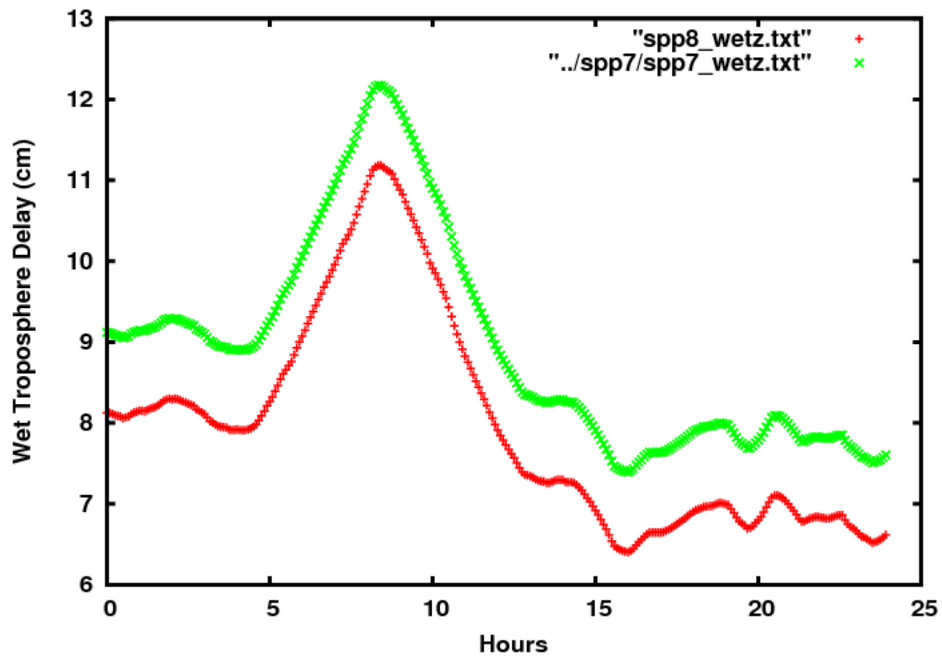
    - No change to estimation strategy.

Example 8: Plot of Zenith Wet Troposphere Solution

grep WETZ tdp_final | cl h1 '1.0e2*c3' > spp8_wetz.txt

gnup -p spp8_wetz.txt ../spp7/spp7_wetz.txt -xl 'Hours' -yl 'Wet Troposphere Delay (cm)'

- Total troposphere delay remains about the same.
  - Input nominal dry increased by 1 cm, so estimated "wet" decreased by 1 cm.
  - But, improved residuals suggests we have a better nominal dry.



Example 9: Provide apriori wet troposphere model (GPT2)

- Use GPT2 model for nominal wet and dry troposphere delay.

  cd ..

  mkdir spp9

  cd spp9

  cp ../spp8/run_again r9 # Copy run_again, and modify as needed.

  – Edit script r9, and change one line.

  -tdp_in ../tropnominal/GOL2_GPT2.TDPdryandwet \

  – Input TDP file provides nominal time series of dry and wet troposphere delay.

  – Execute script

    ./r9

Example 9: First look at run

- No errors in gd2p.err.
- No change to postfit residuals.
  - Providing nominal wet troposphere, but also estimating wet troposphere.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.1648E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7481E-04	2168	0

Example 9: What changed?

- Compare qregres namelists from Examples 8 and 9.

  diff qregres.nml ../spp8/qregres.nml

  - No change to internal models.
  - **However, qregres is ingesting information from tdp file.**

- Compare wash namelists from Examples 8 and 9.

  diff wash.nml ../spp8/wash.nml

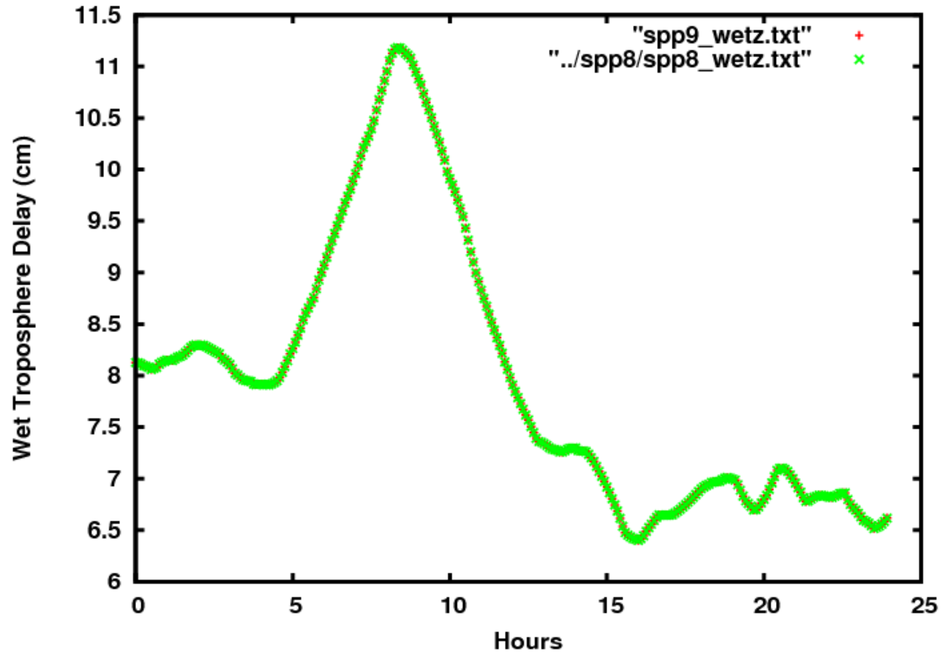
  - No differences.
    - No change to estimation strategy.

Example 9: Plot of Zenith Wet Troposphere Solution

grep WETZ tdp_final | cl h1 '1.0e2*c3' > spp9_wetz.txt

gnup -p spp9_wetz.txt ../spp8/spp8_wetz.txt -xl 'Hours' -yl 'Wet Troposphere Delay (cm)'

- Estimated wet troposphere delay remains about the same.
  - Usually not necessary to provide wet troposphere nominal when estimating wet troposphere.



Example 9: Station position estimate

- However, providing a nominal wet delay even when estimating wet delay can influence the estimated station position.
  - Some prefer to provide a nominal wet delay.

paste tdp_final ../spp8/tdp_final | grep "STA [XYZ]" | cl '1e6*(c3-c10)'

0.000350155460182577

-0.000470208760816604

-0.000240106601268053

- Depends on application, but typically no need to use nominal wet delays when estimating wet delay. Providing nominal dry delay is useful.

Example 10: Use Second Order Ionosphere Correction

- Use IRI2012 model of total electron content to derive second order ionosphere correction to data.

  cd ..

  mkdir spp10

  cd spp10

  cp ../spp9/run_again r10 # Copy run_again, and modify as needed.

  – Edit script r10, and add following lines.

  -ion_2nd \

  -shell_height 600 \

  -tec_mdl iri \

  – See –iri_dir option for path to IRI2012 coefficient files.
    - Default is $GOA_VAR/etc/iri

  – Execute script

    ./r10

Example 10: First look at run

- No errors in gd2p.err.
- Small reduction to postfit residuals.
  - Second order ionosphere correction has positive impact, since no change to estimation strategy

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.1619E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7467E-04	2168	0

Example 10: What changed?

- Compare qregres namelists from Examples 9 and 10.

  diff qregres.nml ../spp9/qregres.nml

```
< $ionosphere_mdl
<     Ion_2nd = .true.
<     mag_mdl = 'IGRF'
<     tec_mdl ='IRI'
<     Ion_2nd_shell_hgt = 600
< $end
```

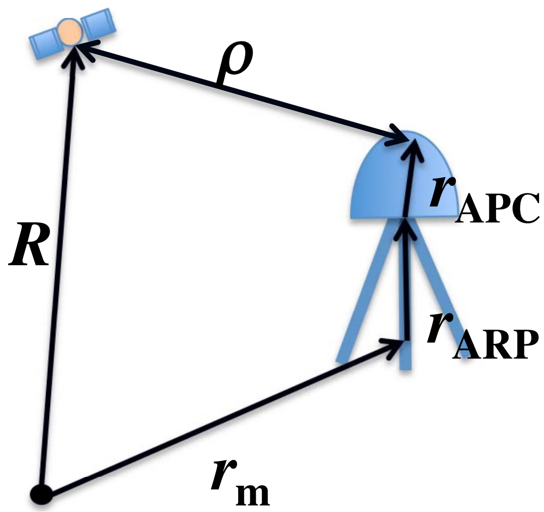
  – Ionosphere models are now defined.

- Compare wash namelists from Examples 9 and 10.

  diff wash.nml ../spp9/wash.nml

  – No differences.
    - No change to estimation strategy.

Monument, ARP, and APC



- **M = Monument**
- **ARP = Antenna Reference Point**
- **APC = Antenna Phase Center**

- Phase and range measurements, ρ, are between transmitter and receiver APCs.
  - Orbit products are associated with a model of transmitter APC.
    - e.g., specified in YYYY-MM-DD.ant file

$\mathbf{r}_{APC}$	$\mathbf{r}_{ARP}$	**What position is being determined?**
Not Modeled	Not Modeled	APC
Modeled	Not Modeled	ARP = APC - $\mathbf{r}_{APC}$
Not Modeled	Modeled	APC-$\mathbf{r}_{ARP}$
Modeled	**Modeled**	**Monument = APC – $\mathbf{r}_{APC}$ - $\mathbf{r}_{ARP}$**

Where does GIPSY/gd2p Define and Get ARP and APC Information?

- Monument, ARP, and APC information defined in qregres namelist.
- gd2p defaults to using s2nml which uses the station database:
  - $GOA_VAR/sta_info (See $GOA/file_formats/sta_info)
- NMLMonument in qregres namelist defines point being positioned (usually the monument):
  - s2nml uses site position/velocity from sta_pos
    - Station dependent
- NMLSite in qregres namelist defines sum total of $\mathbf{r}_{ARP}$ and $\mathbf{r}_{APC}$.
  - s2nml defines $\mathbf{r}_{ARP}$ from sta_svec
    - Station dependent
  - s2nml defines $\mathbf{r}_{APC}$ from pcenter.
    - Antenna model dependent.

Example 11: Use –arp option

- Change reference for estimation to Antenna Reference Point.

  cd ..

  mkdir spp11

  cd spp11

  cp ../spp10/run_again r11 # Copy run_again, and modify as needed.

  – Edit script r11, and add one line.

  – Add one line:

  -arp

    - Effectively sets $\mathbf{r}_{\mathbf{APC}}$ to zero (i.e. ignores information in pcenter file).
    - Recommended when using antenna calibration file (next example).

  – Execute script

    ./r11

Example 11: Effect of -arp

- **In general, would expect a different result if pcenter file had $\mathbf{r}_{APC} \neq 0$.**
- For this particular example there was no impact from using the –arp option.

  diff tdp_final ../spp10/tdp_final

  – Explained by fact that GOL2's antenna type is AOAD/M_T, which has $\mathbf{r}_{APC}$**=0** in the pcenter file.

  grep "AOAD/M_T" $GOA_VAR/sta_info/pcenter
- All examples so far have effectively been positioning the point represented by APC-$\mathbf{r}_{ARP}$ since ARP-APC vector was unrealistically set to zero.
  – We need to define a realistic model for the station's APC relative to the ARP in order to better position the monument.

IGS Antenna Calibrations – ANTEX files

- IGS provides receiver calibrations in file called "ANTEX".
- JPL orbit/clock products identify the adopted transmitter and receiver calibrations.

zcat ../jplorbclk/2010-01-10.ant.gz

```
SAT igs08_1604.xyz    ftp://sideshow.jpl.nasa.gov/pub/gipsy_files/gipsy_params/antenna_cals_xmit/igs08_1604.xyz.gz
GRN igs08_1604.atx    ftp://sideshow.jpl.nasa.gov/pub/gipsy_files/gipsy_params/antenna_cals_xmit/igs08_1604.atx.gz
```

  - SAT line indicates transmitter calibration file (in GIPSY format) used to generate orbit/clock product, and location on JPL server.
    - **gd2p uses SAT line to add transmitter calibrations to qregres namelist.**
  - GRN line indicates receiver calibration file used to generate orbit/clock product, and location on JPL server.
  - In this example, an IGS antex file was used for the antenna calibrations of the ground station network used to generate the orbit/clock products.
- Best to use consistent set of calibrations.
  - Main consideration for consistency is reference frame identifier (e.g. igs08).

antex2xyz.py - Generating a receiver antenna calibration file

- GOL2 RINEX file identifies antenna type AOAD/M_T and radome type NONE.
  - All of the igs08 antex files have the same calibration for this antenna.
  - Calibration changes usually occur when reference frames change.
- Native GIPSY format for antenna calibrations is "xyz" format.
  - See $GOA/file_formats/antenna_cal
- Generate xyz file from ANTEX file:

cd ..

mkdir antcal

cd antcal

antex2xyz.py -antexfile $GOA_VAR/etc/antenna_cals_xmit/igs08_1604.atx -xyzfile GOL2_antex.xyz -anttype AOAD/M_T -recname GOL2 -radcode NONE -fel 0 –del 5 -daz 5 –extrap

  - NOTE: Effective phase center offset, $\mathbf{r}_{APC}$, in mm is (see GOL2_antex.xyz):
    - NORTH = 1.600180 : EAST = -0.029940 : UP = 47.796758

sta2antxyz.py – Generating a receiver antenna calibration file from sinex file

- If station is part of IGS network, could use IGS SINEX file to determine antenna and radome type.
  - Accounts for any recorded changes to antenna
- Generate antenna calibration from ANTEX and SINEX files.

  sta2antxyz.py -s GOL2 -sinex $GOA_VAR/sta_info/igs.snx -antex $GOA_VAR/etc/antenna_cals_xmit/igs08_1604.atx -o GOL2_antex_sinex.xyz -d 2010-01-10
- Two files are identical:

  diff GOL2_antex.xyz GOL2_antex_sinex.xyz
  - sta2antxyz.py and antex2xyz.py use same library.
  - antex2xyz.py is probably more flexible but means user must keep track of station equipment.

Jet Propulsion Laboratory
California Institute of Technology

Example 12: Add receiver antenna calibration

- Add receiver antenna calibration.

  cd ..

  mkdir spp12

  cd spp12

  cp ../spp11/run_again r12 # Copy run_again, and modify as needed.

  – Edit script r12, and add one line:

  -AntCal ../antcal/GOL2_antex.xyz \

  – Execute script

  ./r12

Example 12: First look at run

- No errors in gd2p.err.
- Reduction of postfit residuals.
  - Receiver antenna calibration has positive impact, since no change to estimation strategy

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	5.9454E-06	2167	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	3.7453E-04	2168	0

Example 12: What changed?

- Compare qregres namelists from Examples 11 and 12.

  diff qregres.nml ../spp11/qregres.nml

  – Additional namelist items for receiving antenna calibration.
    - NMLGCen, NMLG*

  – **NOTE: gd2p automatically adds transmitter antenna calibrations to qregres namelist based upon "SAT" information provided in orbit/clock products (.ant file).**
    - NMLOCen, NMLO*

- Compare wash namelists from Examples 11 and 12.

  diff wash.nml ../spp11/wash.nml

  – No changes to solution strategy.

Example 13: Lower Elevation Angle Cutoff

- Modify elevation angle cutoff.

  cd ..

  mkdir spp13

  cd spp13

  cp ../spp12/run_again r13 # Copy run_again, and modify as needed.

  – Edit script r13.

  – Change elevation angle cutoff to 7 degrees.

  -w_elmin 7 \

    - Recommended cutoff with troposphere gradients and antenna calibration.

  – Execute script

    ./r13

Example 13: First look at run

- No errors in gd2p.err.
- Increase in RMS of postfit residuals, but associated with increase in number of measurements.
  - More low elevation data used in solution.
  - Low elevation data usually have higher data noise.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	7.1846E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0513E-04	2429	0

Example 14: Elevation-Dependent Weighting

- De-weight low elevation data since it has higher data noise.

  cd ..

  mkdir spp14

  cd spp14

  cp ../spp13/run_again r14   # Copy run_again, and modify as needed.

  – Edit script r14.

  – Add one line:

  -eldepwght SQRTSIN \

    - Data weight = sqrt(sin(elevation))/sigma.
    - Depending on receiver and application, might also consider “SIN” weighting. (Data weight = sin(elevation)/sigma)
    - Without this option data weight is constant, 1/sigma.

  – Execute script

    ./r14

Example 14: First look at run

- No errors in gd2p.err.
- Increase in RMS of postfit residuals, because lower elevation (higher noise) data has lower impact on the solution.
  - More low elevation data used in solution.
  - Low elevation data usually have higher data noise.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	7.8489E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0521E-04	2429	0

Example 14: What changed?

- Compare qregres namelists from Examples 13 and 14.

  diff qregres.nml ../spp13/qregres.nml

  – No changes to model.

- Compare wash namelists from Examples 13 and 14.

  diff wash.nml ../spp13/wash.nml

  < SIGELV = .TRUE., .TRUE.

  < SQRELP1 = .FALSE.

  – Data weight options to filter are set.

Example 15: Not using station database

- For improved transparency, explicitly provide monument position, monument to ARP vector, and antenna calibrations.

  cd ..

  mkdir spp15

  cd spp15

  cp ../spp14/run_again r15 # Copy run_again, and modify as needed.

  – Edit script r15.

  – Add two lines:

  -p -2353.614421 -4641.385321 3676.976407 \

  -env_km 0.0 0.0 0.0000254 \

  - Chose position (-p) and site vector (-env_km) coordinates from prior qregres namelists only for consistency with previous examples.

  – Remove line with –arp (not needed when using –p option).

  – Execute script

    ./r15

Example 15: First look at run

- No errors in gd2p.err.
- No change to postfit residuals.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	7.8489E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0521E-04	2429	0

Example 15: What changed?

- Compare qregres namelists from Examples 14 and 15.

  diff qregres.nml ../spp14/qregres.nml

  – Some changes in order of parameters and format only.
- Compare wash namelists from Examples 14 and 15.

  diff wash.nml ../spp14/wash.nml

  – No changes to solution strategy.
- Using –p and –env_km options resulted with an identical run.
  - Might have round-off effects.
  - E.g., compare station position lines in TDP file.

  grep "STA [XYZ]" tdp_final

  grep "STA [XYZ] ../spp14/tdp_final

Jet Propulsion Laboratory
California Institute of Technology

Example 16: Generate stacov file

- Add option to generate stacov file.

  cd ..

  mkdir spp16

  cd spp16

  cp ../spp15/run_again r16 # Copy run_again, and modify as needed.

  – Edit script r16, and add one line.

  -stacov \

  – Option results with generation of a stacov file with the position solution and covariance matrix.
    - See $GOA/file_formats/stacov

  – Execute script

    ./r16

Example 16: What changed?

- Compare qregres namelists from Examples 15 and 16.

  diff qregres.nml ../spp15/qregres.nml
- Compare wash namelists from Examples 15 and 16.

  diff wash.nml ../spp15/wash.nml
  - No changes to solution strategy.
- Identical solution but additional output files:
  - stacov_final_pre_amb – Pre-ambiguity resolution solution.
  - stacov_final – Post-ambiguity resolution solution

Example 16: Pre- and Post-amb Residuals

- Ambiguity resolution results with larger post-fit residuals.
  - Expected because of constraints on phase biases.
  - Should be the case for all examples.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	7.8489E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0521E-04	2429	0

cat Postfit_amb.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	7.9731E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0541E-04	2429	0

Example 16: Compare Pre- and Post-Amb Position Solutions

```
grep "STA [XYZ]" tdp_final_pre_amb > pre_amb.pos
grep "STA [XYZ]" tdp_final > post_amb.pos
paste pre_amb.pos post_amb.pos | cl '1.0e6*(c3-c10)'
0.0456102497992106        (Z = 0.0 mm)
0.514080056746025         (Y = 0.5 mm)
-0.851220192998881        (X = -0.9 mm)
```

- Some other things to consider:
  - Pre- vs Post- amb clocks?
  - Pre- vs Post-amb wet?

Evaluating Effectiveness of Ambiguity Resolution

- Ambiguity resolution performance can be evaluated from percent of narrow lanes that are close to an integer.
  - Look at NLCUM statistics in logs/smap_ambigon.log
  - Reflects cumulative distribution of number and percent of narrow lanes that are close to an integer.

grep NLCUM logs/smap_ambigon.log

```
PNLCUM   0   3542  87.2843765402
PNLCUM   5    366   9.0192212913
PNLCUM  10     78   1.9221291276
...
```

  - Column 2 is centicycles from integer.
  - Column 3 is number of narrow lanes within bin.
  - Column 4 is percent of narrow lanes within bin.
    - E.g., 87.3 percent of narrow lanes < 5 centicycles of integer.
    - Larger numbers closer to integer implies better solution prior to ambiguity resolution. (likely means better ambiguity resolution).

Example 16: Station Position Estimate

- How much did the station move relative to the provided nominal value?

grep "STA [XYZ]" tdp_final | cl '1.0e6*(c3-c2)'

8.11517975307652	(Z = 8.1 mm)
32.6875997416209	(Y = 33.7 mm)
26.7990303655097	(X = 26.8 mm)

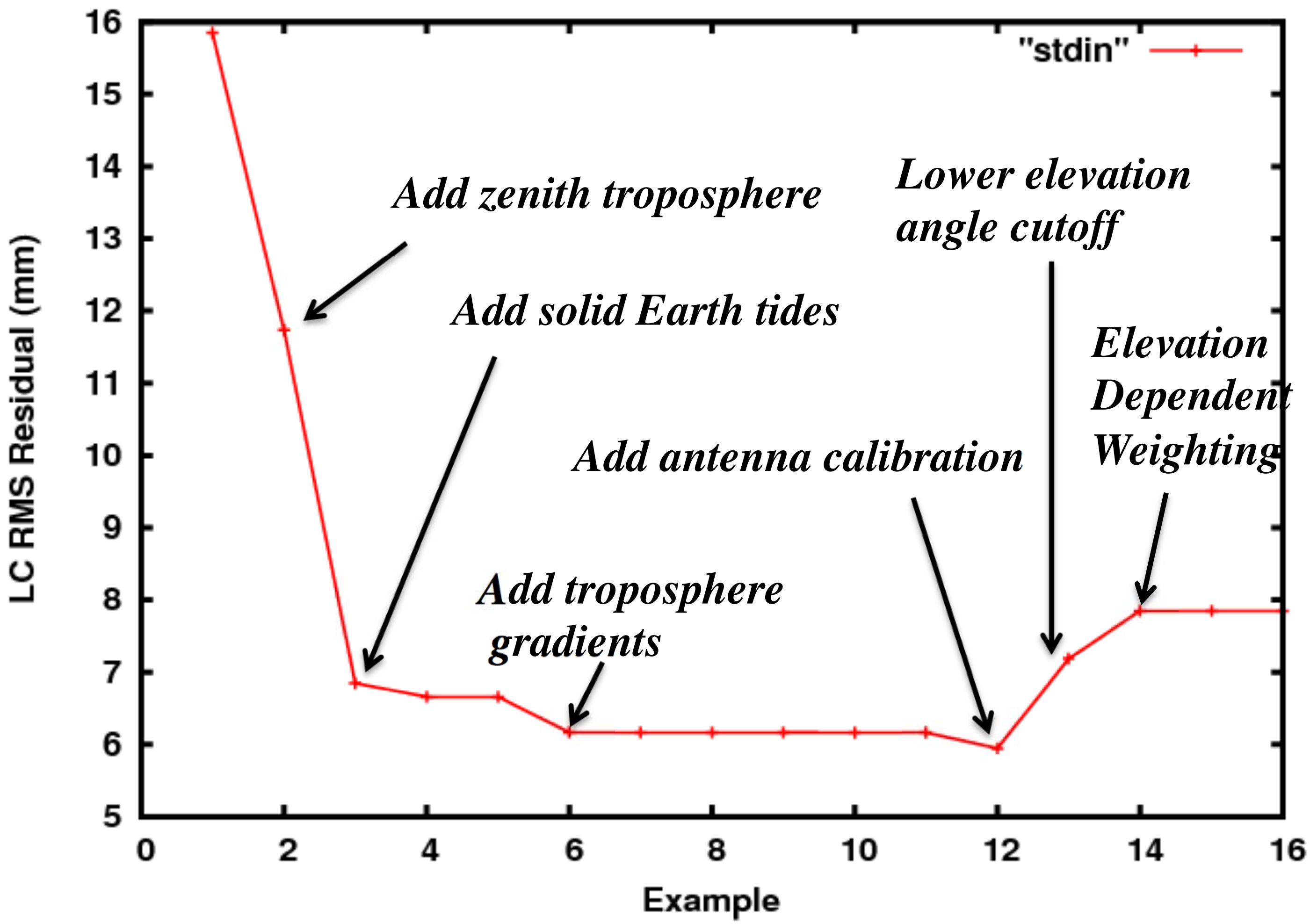
gd2p.pl settings from Example 16

```
( gd2p.pl \
    -i  $GOA/class_gd2p/gol20100.10o \
    -n GOL2 \
    -d 2010-01-10 \
    -r 300 \
    -type s \
    -w_elmin 7 \
   -eldepwght SQRTSIN \
    -e "-a 20 -PC -LC -F" \
    -pb_min_slip 1.0e-3 \
    -pb_min_elev 30 \
    -amb_res 2 \
    -dwght 1.0e-5 1.0e-3 \
    -post_wind 5.0e-3 5.0e-5 \
    -trop_z_rw 5.0e-8 \
    -wetzgrad 5.0e-9 \
    -trop_map GPT2 \
    -tdp_in ../tropnominal/GOL2_GPT.TDPwetanddry \
    -tides WahrK1 FreqDepLove OctTid PolTid \
    -add_ocnld \
    -OcnldCpn \
   -add_ocnldpoltid \
    -ion_2nd \
    -shell_height 600 \
    -tec_mdl iri \
    -orb_clk "flinnR WORKDIR/jplorbclk"
    -AntCal ../antcal/GOL2_antex.xyz \
    -p -2353.614421 -4641.385321 3676.976407 \
    -env_km 0.0 0.0 0.0000254 \
    -stacov \
 > gd2p.log ) |&  sed '/^Skipping namelist/d' > gd2p.err
```

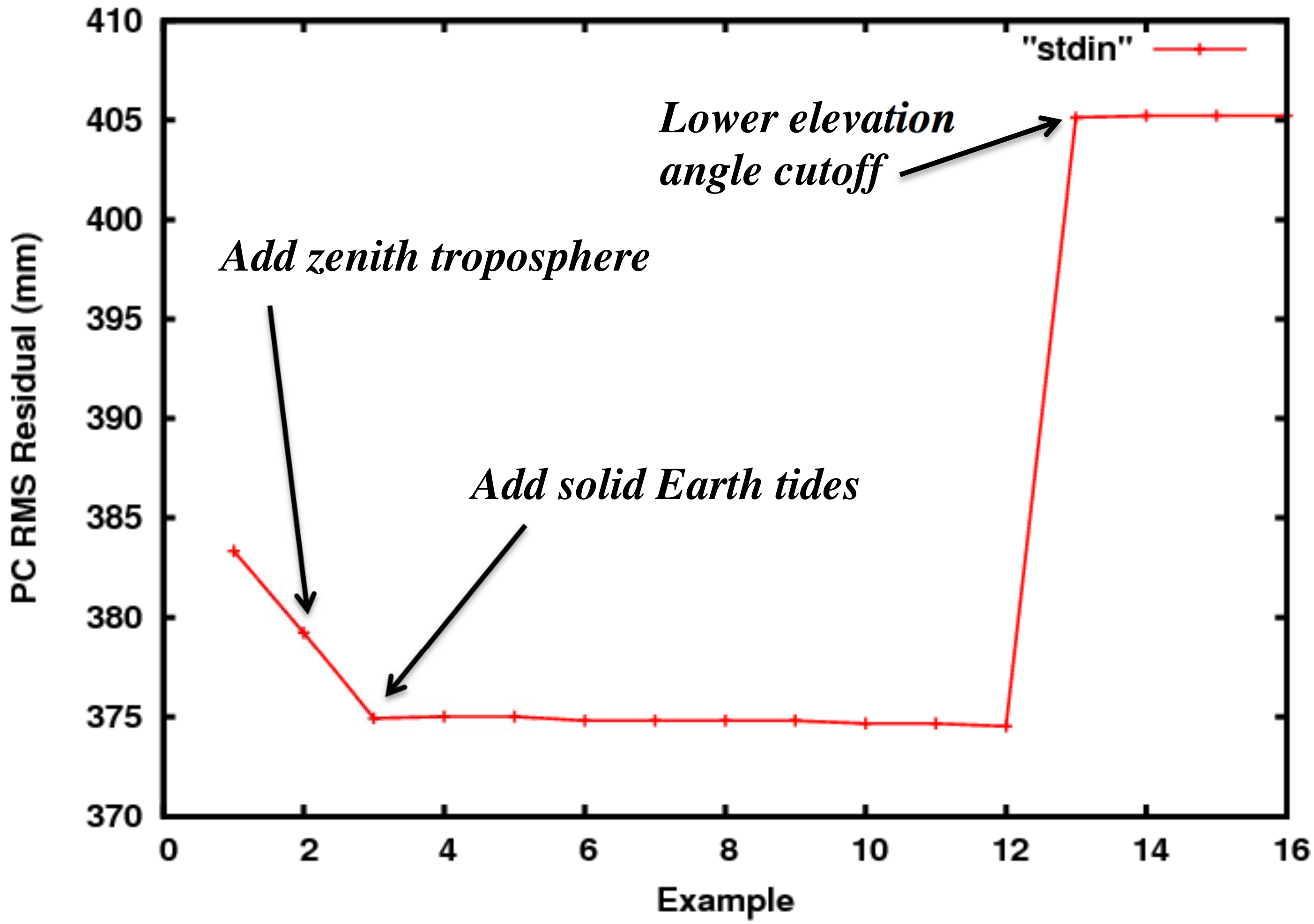
gd2p.pl for typical station

- run_again from Example 16 reflects the gd2p.pl settings for a typical IGS-quality GPS station.
  - Could be improved by:
    - Using VMF1GRID or VMF1SITE (via tropnominal) for input nominal wet and dry troposphere delay.
      - Nominal wet is less important when estimating wet.
    - Using VMF1GRID or VMF1SITE troposphere mapping function
    - Some prefer to set -pb_min_slip to large value so that postbreak is not exercised. Some prefer smaller value. User choice.
      - We have found 1 meter pb_min_slip and 30-degree elevation lower limit (-pb_min_elev) to search for breaks is reasonable for modern receivers.

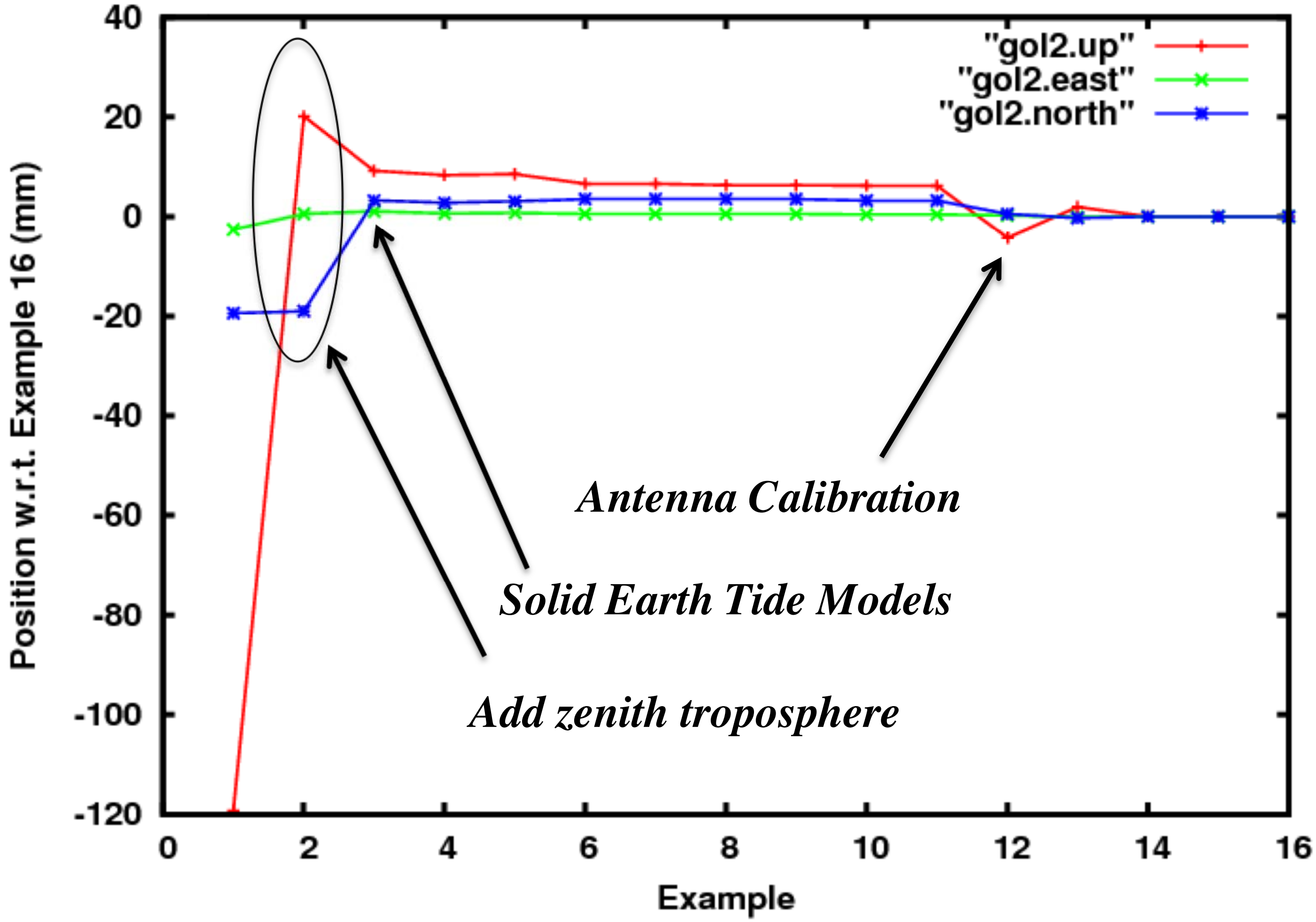
Effect of models/estimation approach on Postfit LC RMS Residual



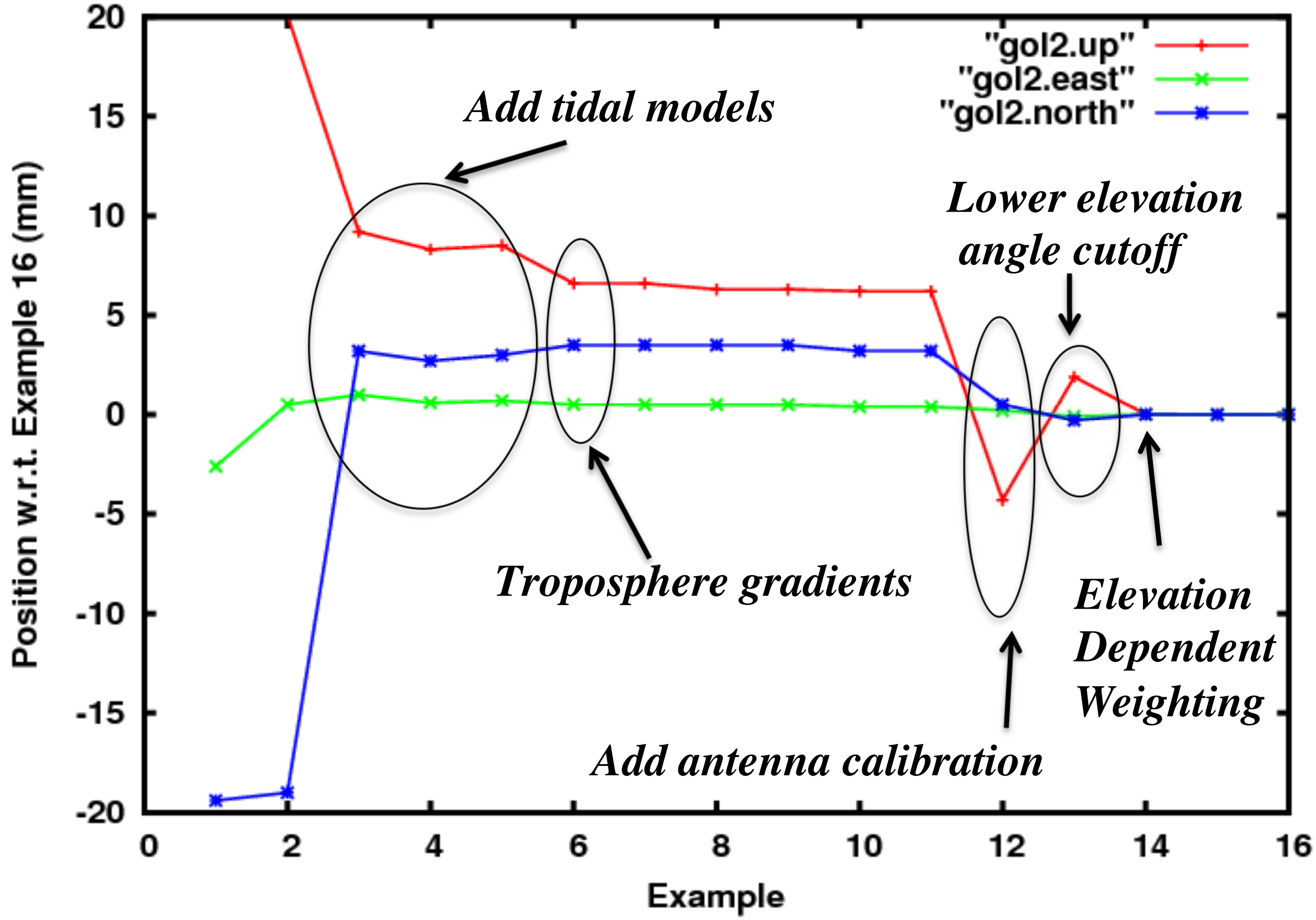
Effect of models/estimation approach on Postfit PC RMS Residual



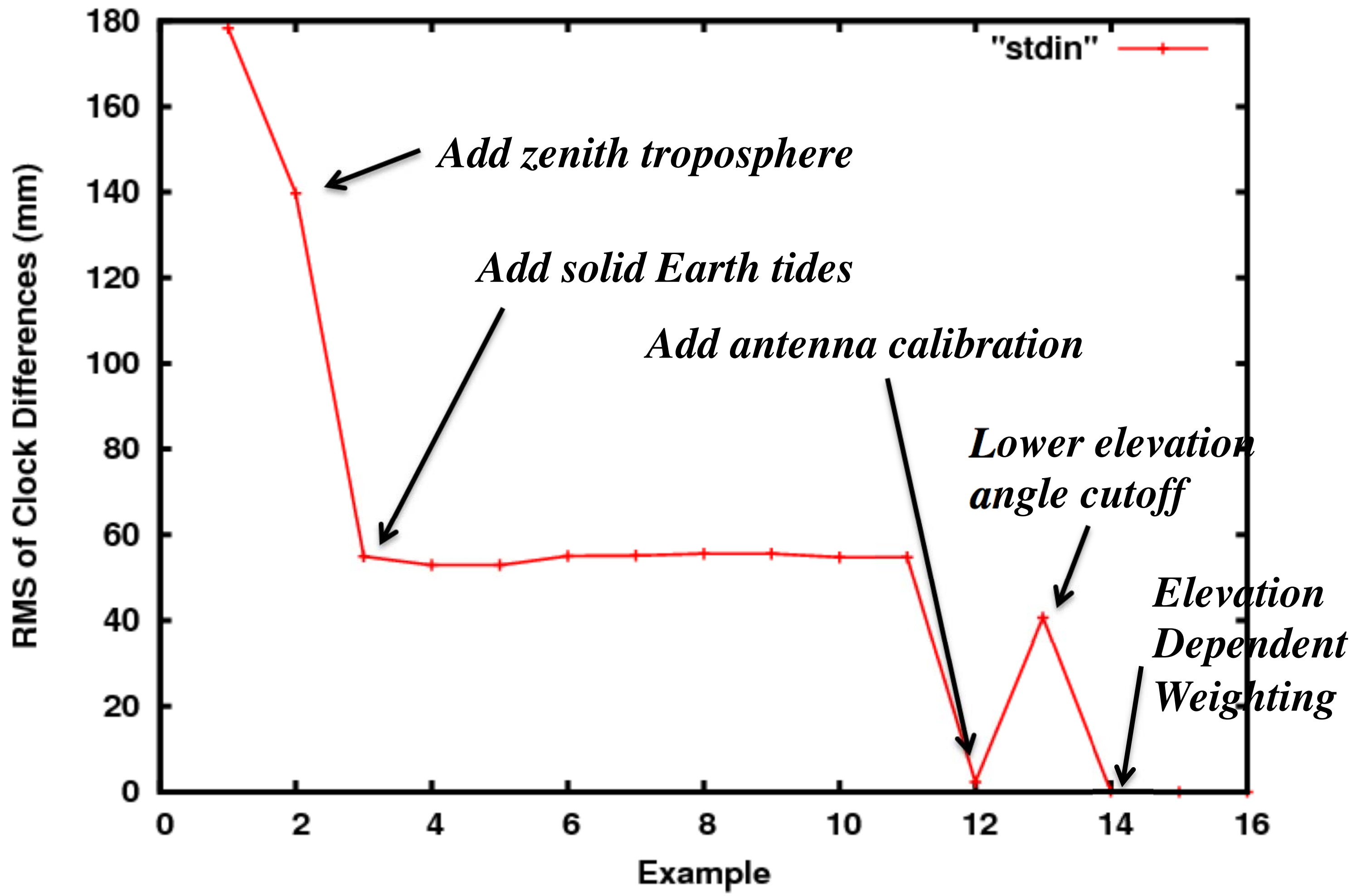
Effect of models/estimation approach on station position (Relative to Example 16)



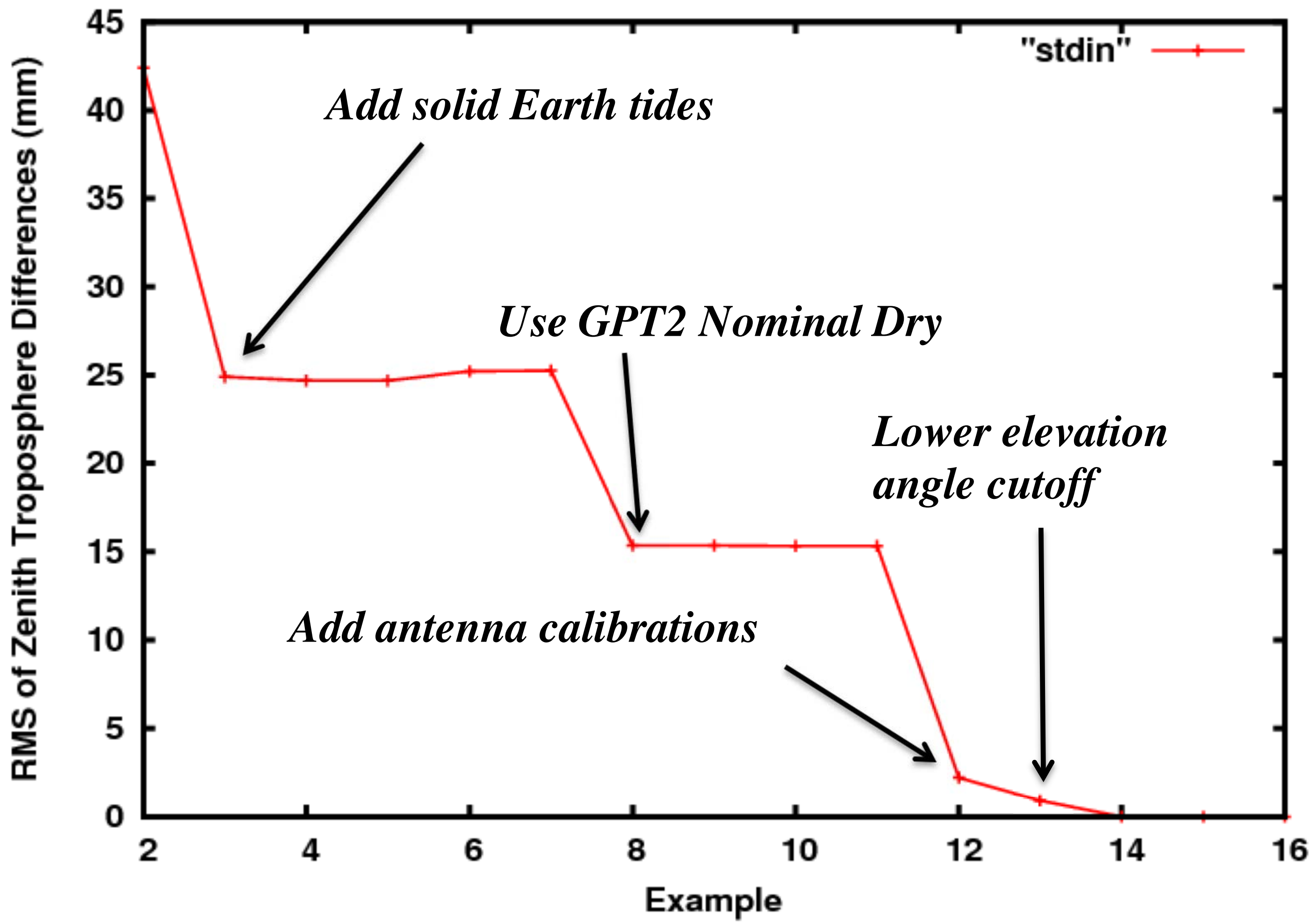
Effect of models/estimation approach on station position (Zoom, Relative to Example 16)



Effect of models/estimation approach on Clock Solutions (Relative to Example 16)

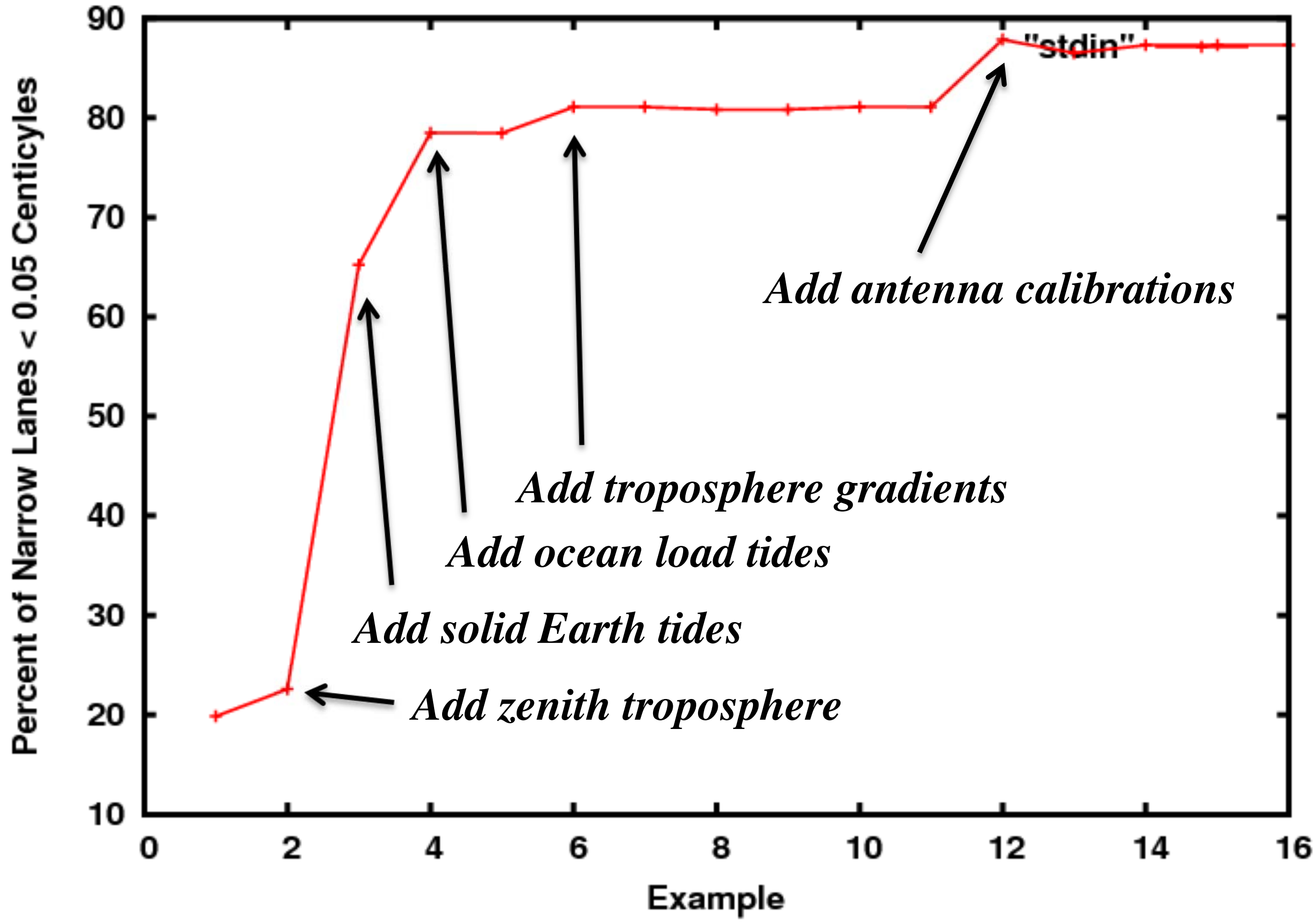


Effect of models/estimation approach on station position on Zenith Troposphere Solutions(Relative to Example 16)



Effect of models/estimation approach on station position on ambiguity resolution

NOTE: Number of narrow lanes < 0.05 centicycle also increased when elevation angle cutoff lowered to 7 degrees (3160 to 3510), and overall percentage increased slightly.



pr2p

- pr2p determines station position and clock from GPS pseudorange data alone.
  - No apriori information or models for receiver.
  - Only provide orbits/clocks of transmitters.
  - Useful when nominal station position not known, e.g. moving platforms.

pr2p –h | more

Jet Propulsion Laboratory
California Institute of Technology

pr2p - Example

NASA

cd ..

mkdir pr2p

cd pr2p

pr2p –i $GOA/class_gd2p/gol20100.10o –n GOL2 –orb_clk "flinnR *WORKDIR*/jplorbclk"

- Provides an average position in the log file (pr2p.log).

Average Position, (x,y,z), in km: -2353.614218 -4641.384766 3676.976173

Position Scatter, (x,y,z), in km: 0.001444 0.002298 0.001526

  – Could be used in follow-up gd2p run (-p option)

- Provides time series of receiver positions and clocks in output tdp_final.

  grep "STA [XYZ]" tdp_final

  grep "STA BIAS" tdp_final

  – Could be used in follow-up gd2p run (-tdp_in)

Kinematic Point Positioning

Kinematic Positioning
Apriori Positions and Troposphere

- Kinematic point positioning is more challenging than static positioning.
  - How fast is the platform moving?
    - For rapid movement, provide a nominal STA [XYZ] through input TDP file.
      - E.g., Solution from pr2p, prevous gd2p, or navigation solution.
      - Default is quadratic interpolation of positions in TDP file.
    - For rapid motion be careful with –amb_res. May be better not to use.
      - Check NLCUM statistics in smap_ambigon.log
  - Separating troposphere from position is more difficult.
    - Use apriori dry and preferably wet troposphere from tropnominal, or weather models (e.g., NCEP, ECMWF)
      - Provide through input TDP file.
  - If a TDP file with station position(s) is provided (STA [XYZ]), default:
    - Zenith dry troposphere: 1.013*2.27*exp(-0.116e-03*ht) meters.
    - Zenith wet troposphere: 0.1*exp(-ht/2000e0) meters.

Example 1

- Start with run_again from static point positioning example 15.

  cd ..

  mkdir kpp1

  cd kpp1

  cp ../spp15/run_again r1   # Copy run_again, and modify as needed.

  – Edit script r1:

  – Change positioning type to kinematic

  -type k \

  – **NOTE: We are already providing a nominal zenith dry and wet troposphere correction.**

    - **-tdp_in ../tropnominal/GOL2_GPT.TDPwetanddry**

  – Execute script

    ./r1

Example 1: First look at run

- No errors in gd2p.err.
- Postfit residuals significantly lower than corresponding static run.
  - Position has more freedom to follow data.

cat Postfit.sum

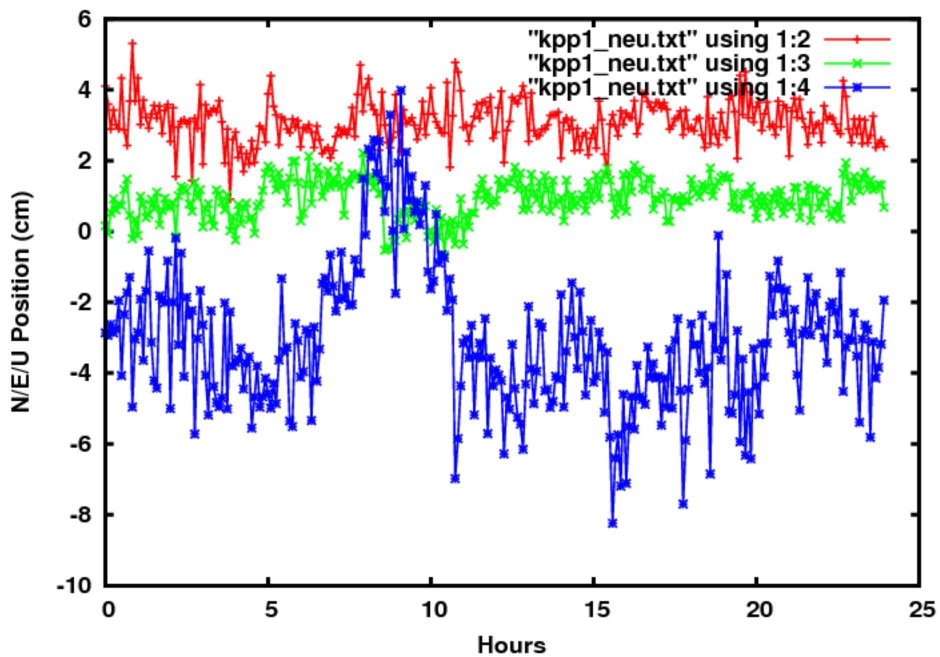
start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	5.9291E-06	2429	0
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0420E-04	2429	0

Example 1: Position solution relative to nominal

- Use tdp2llh to compare the solution to the nominal.

  tdp2llh tdp_final GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp1_neu.txt

  gnup -lp1:2 kpp1_neu.txt -lp1:3 kpp1_neu.txt -lp1:4 kpp1_neu.txt -xl 'Hours' -yl 'N/E/U Position (cm)'



- What is standard deviation of time series?

cat kpp1_neu.txt | cl 2 | stats | cl 3
cat kpp1_neu.txt | cl 3 | stats | cl 3
cat kpp1_neu.txt | cl 4 | stats | cl 3

- North: 0.6 cm
- East: 0.5 cm
- Up: 2.0 cm

Example 2: Provide Wet Troposphere

- Provide wet troposphere from static solution instead of GPT2, and append to already used dry troposphere.

cd ..

mkdir kpp2

cd kpp2

grep -P 'TRP|WETZ' ../spp15/tdp_final > ../tropnominal/GOL2.TDPwetanddry
cat ../tropnominal/GOL2_GPT2.TDPdry >> ../tropnominal/GOL2.TDPwetanddry

cp ../kpp1/run_again r2 # Copy run_again, and modify as needed.

– Edit script r2:

– Change TDP input file:

-tdp_in ../tropnominal/GOL2.TDPwetanddry \

– Remove lines –trop_z_rw, and –wetzgrad (Still need –trop_map).

– Execute script

./r2

Example 2: First look at run

- No errors in gd2p.err.
- Check that troposphere parameters not estimated.

  diff wash.nml ../kpp1/wash.nml

  – WETZTROP and WETZGRAD are now both set to .FALSE.
- Postfit residuals are slightly lower, although 1 outlier.
  – Note: Fewer parameters being estimated, and less freedom for data.

  cat Postfit.sum

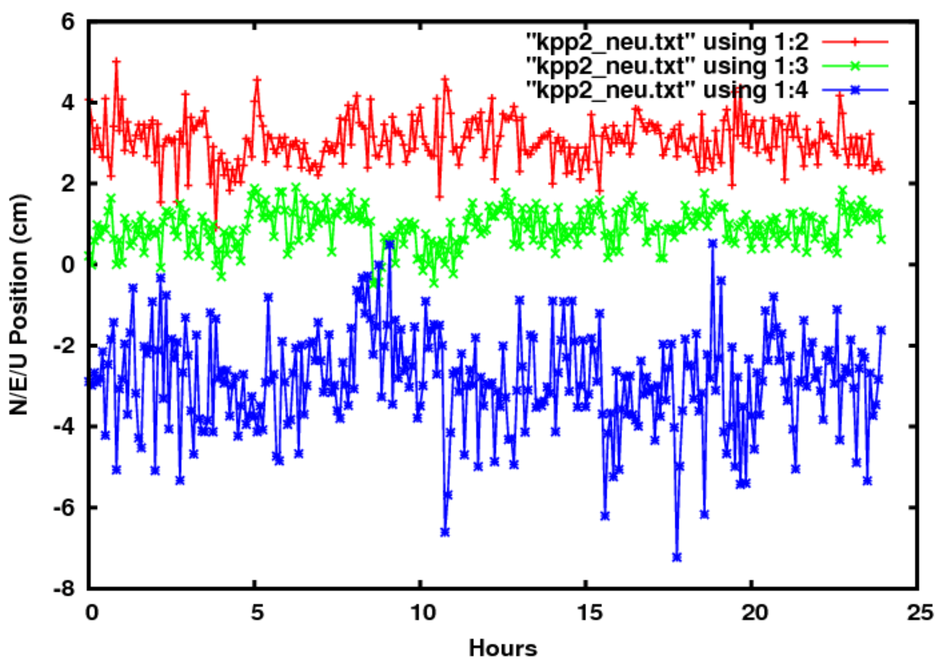
start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	5.6768E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0494E-04	2429	0

Example 2: Position solution relative to nominal

- Use tdp2llh to compare the solution to the nominal.

tdp2llh tdp_final GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp2_neu.txt

gnup -lp1:2 kpp2_neu.txt -lp1:3 kpp2_neu.txt -lp1:4 kpp2_neu.txt -xl 'Hours' -yl 'N/E/U Position (cm)'



- What is standard deviation of time series?

cat kpp2_neu.txt | cl 2 | stats | cl 3
cat kpp2_neu.txt | cl 3 | stats | cl 3
cat kpp2_neu.txt | cl 4 | stats | cl 3

- North: 0.6 cm
- East: 0.5 cm
- Up: 1.2 cm
- This is lower than when troposphere was estimated.
  - Improved result given station is not moving.

Example 3: Changing Random Walk Parameters

cd ..

mkdir kpp3

cd kpp3

cp ../kpp2/run_again r3 # Copy run_again, and modify as needed.

– Edit script r3:

– Add the following line:

-kin_sta_xyz 1.0e-3 5.7e-7 300 RANDOMWALK \

– gd2p's defaults are: 2.0e-3 1.0e-3 data_rate RANDOMWALK
  - In this example data rate = 300 seconds.
  - Defaults imply 1.0e-3*sqrt(300) km = 17.3 meters
    – Could just as well be white noise process for this type of station.
  - Changes imply 5.7e-7*sqrt(300) km = 10 mm

– Execute script

./r3

Example 3: First look at run

- No errors in gd2p.err.
- Postfit residuals are higher.
  - Tighter constraints on random walk mean less freedom for positions to follow data.

cat Postfit.sum

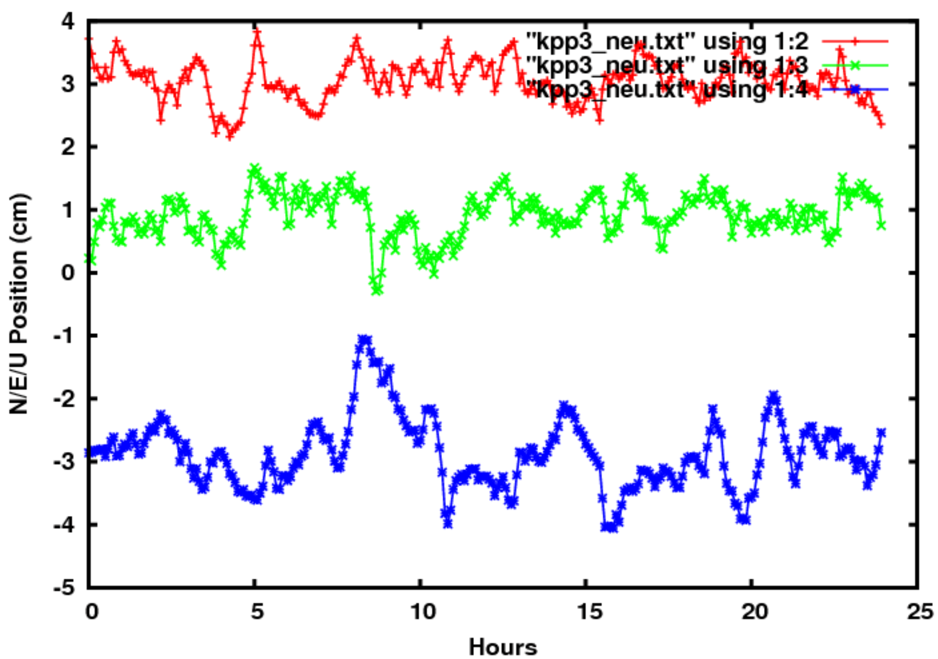
start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.7252E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0493E-04	2429	0

Example 3: Position solution relative to nominal

- Use tdp2llh to compare the solution to the nominal.

tdp2llh tdp_final GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp3_neu.txt

gnup -lp1:2 kpp3_neu.txt -lp1:3 kpp3_neu.txt -lp1:4 kpp3_neu.txt -xl 'Hours' -yl 'N/E/U Position (cm)'



- What is standard deviation of time series?

cat kpp3_neu.txt | cl 2 | stats | cl 3
cat kpp3_neu.txt | cl 3 | stats | cl 3
cat kpp3_neu.txt | cl 4 | stats | cl 3

- North: 0.3 cm
- East: 0.4 cm
- Up: 0.6 cm
- Lower scatter with tighter constraints on random walk.

Example 3: Comparing pre- and post-amb solutions

- Get the pre-amb position solutions:

  tdp2llh tdp_final_pre_amb GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp3_neu_pre_amb.txt

  cat kpp3_neu_pre_amb.txt | cl 2 | stats | cl 3

  cat kpp3_neu_pre_amb.txt | cl 3 | stats | cl 3

  cat kpp3_neu_pre_amb.txt | cl 4 | stats | cl 3

- Standard deviation of pre-amb versus post-amb time series.
  - Post-amb has slightly lower scatter.
    - Probably better since we know station is static.

	Pre-Amb	Post-Amb
North	0.4 cm	0.3 cm
East	0.4 cm	0.4 cm
Up	0.8 cm	0.6 cm

Example 4: Providing Nominal Station Position with TDP input

cd ..

mkdir kpp4

cd kpp4

cp ../kpp3/run_again r4   # Copy run_again, and modify as needed.

– Combine nominal troposphere (WET and DRY) and station positions in TDP input.
    - Use solution from previous example, i.e., iterate

cp ../tropnominal/GOL2.TDPwetanddry ../tropnominal/GOL2.TDPwetanddrysta

grep "STA [XYZ]" ../kpp3/tdp_final >> ../tropnominal/GOL2.TDPwetanddrysta
WARNING: -env_km is ignored when providing STA [XYZ] in TDP input file.

– Edit script r4:

– Change the following line:

-tdp_in ../tropnominal/GOL2.TDPwetanddrysta \

– Execute script

./r4

Example 4: First look at run

- No errors in gd2p.err.
- Postfit residuals are slightly lower.
  - Better nominal solution provided.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	6.4449E-06	2428	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.0494E-04	2429	0

- Verify that the nominal station positions are time-varying in this example.
  - Look at column 2 of tdp_final

grep "STA X" tdp_final

  - In previous examples the nominal was constant.

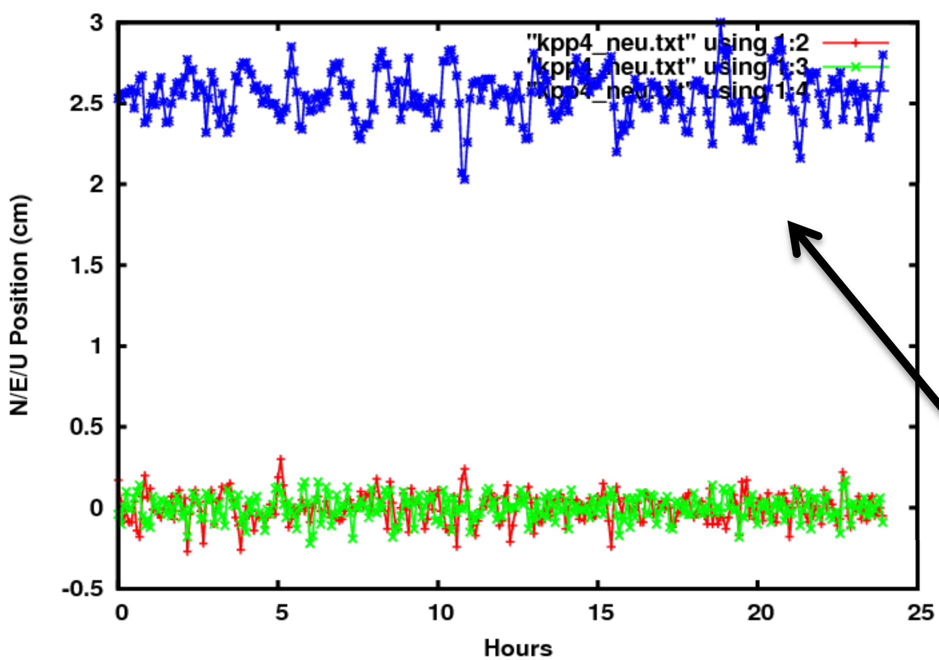
grep "STA X" ../kpp3/tdp_final

Example 4: Position solution relative to nominal

- Use tdp2llh to compare the solution to the nominal.

tdp2llh tdp_final GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp4_neu.txt

gnup -lp1:2 kpp4_neu.txt -lp1:3 kpp4_neu.txt -lp1:4 kpp4_neu.txt -xl 'Hours' -yl 'N/E/U Position (cm)'



- What is scatter of time series?

cat kpp4_neu.txt | cl 2 | stats | cl 3
cat kpp4_neu.txt | cl 3 | stats | cl 3
cat kpp4_neu.txt | cl 4 | stats | cl 3

- North: 0.1 cm
- East: 0.1 cm
- Up: 0.1 cm
- Using "est" from tdp2llh means we are looking at the change relative to the provided time varying nominal.
- **Bias in UP component results from –env_km information being ignored when providing STA [XYZ] in input.**

Jet Propulsion Laboratory
California Institute of Technology

Example 5: Using High Rate Data

cd ..

mkdir kpp5

cd kpp5

cp ../kpp4/run_again r5 # Copy run_again, and modify as needed.

– Edit script r5:

– Change data rate, and update rate of station positions to 30 seconds:

-r 30 \

-kin_sta_xyz 1.0e-3 5.7e-7 30 RANDOMWALK

– Execute script

./r5

NOTE: Takes much longer to run with more data.

Example 5: First look at run

- No errors in gd2p.err.
- Number of data points increased by factor of 10.
- Post-fit PC residuals increase significantly.
  - PC data are not carrier smoothed if –r option is same as original data rate.

cat Postfit.sum

start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:59:15	LC	GOL2	7.2019E-06	23883	14
10JAN09 23:59:45	10JAN10 23:59:15	PC	GOL2	1.0627E-03	23690	207

Example 5 – More checks

- Verify that the input transmitter clocks are high-rate (30-second) clocks.
  - Look at gd2p.log to verify that high-rate clock product was found.

  2010-01-10_hr.tdp.gz

  - Look at column 1 of tdp_clk_yaw

  more tdp_clk_yaw

- If high-rate transmitter clocks not available, could also use low-rate (5-minute) clocks and add –interp_300 option to gd2p command.
  - Expect higher postfit residuals and accuracy of solution to be worse.
- Verify that station position estimates are also at high-rate.
  - See column 1 of tdp_final file.

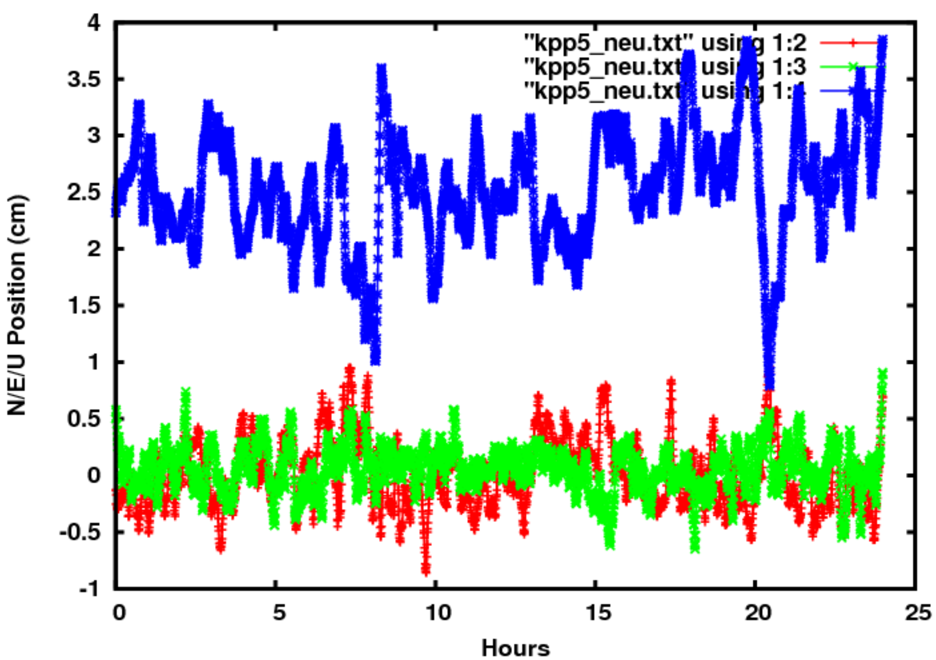
  grep "STA X" tdp_final | more

Example 5: Position solution relative to nominal

- Use tdp2llh to compare the solution to the nominal.

tdp2llh tdp_final GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp5_neu.txt

gnup -lp1:2 kpp5_neu.txt -lp1:3 kpp5_neu.txt -lp1:4 kpp5_neu.txt -xl 'Hours' -yl 'N/E/U Position (cm)'



- What is scatter of time series?

cat kpp5_neu.txt | cl 2 | stats | cl 3
cat kpp5_neu.txt | cl 3 | stats | cl 3
cat kpp5_neu.txt | cl 4 | stats | cl 3

- North: 0.3 cm
- East: 0.2 cm
- Up: 0.5 cm
- Larger scatter than low-rate run, as expected.

Example 6: Using Whitenoise Process

cd ..

mkdir kpp6

cd kpp6

cp ../kpp5/run_again r6 # Copy run_again, and modify as needed.

- Edit script r6:
- Switch to a white noise process to give kinematic solution freedom to follow the data. Useful for following cases:
  - Platform moving rapidly (e.g., airplane)
  - Motion is not known (could use pr2p to provide apriori information).
    - Might use as first attempt before "smoothing" with RANDOMWALK.
  - To capture rapid "jumps" in station position.

-kin_sta_xyz 1.0e3 1.0e3 30 WHITENOISE

- Execute script

  ./r6

Example 6: First look at run

- No errors in gd2p.err.
- Post-fit PC residuals decrease significantly.
  - Positions have significant freedom to follow the data.

cat Postfit.sum

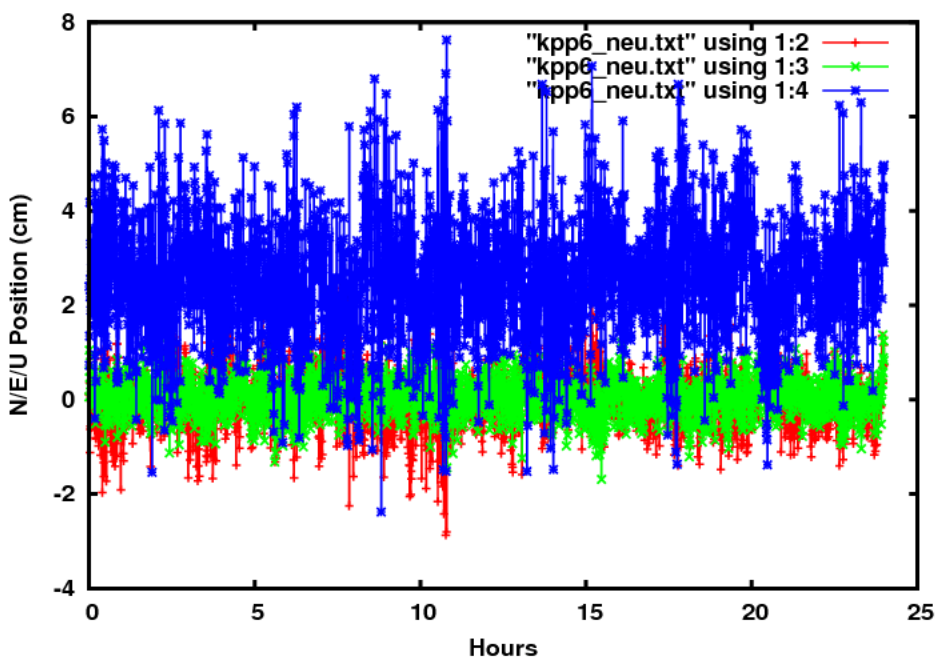
start time	stop time	type	user	postfit sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:59:15	LC	GOL2	5.9111E-06	23896	1
10JAN09 23:59:45	10JAN10 23:59:15	PC	GOL2	1.0622E-03	23689	207

Example 6: Position solution relative to nominal

- Use tdp2llh to compare the solution to the nominal.

  tdp2llh tdp_final GOL2 | grep est | cl h1 '1e2*c2' '1e2*c3' '1e2*c4' > kpp6_neu.txt

  gnup -lp1:2 kpp6_neu.txt -lp1:3 kpp6_neu.txt -lp1:4 kpp6_neu.txt -xl 'Hours' -yl 'N/E/U Position (cm)'



- What is standard deviation of time series?

cat kpp6_neu.txt | cl 2 | stats | cl 3
cat kpp6_neu.txt | cl 3 | stats | cl 3
cat kpp6_neu.txt | cl 4 | stats | cl 3

- North: 0.6 cm
- East: 0.4 cm
- Up: 1.2 cm
- Significantly larger scatter as expected.
- Represents extreme limit of positioning accuracy achievable from available data.
  - Random walk effectively smooths this solution.

Kinematic Positioning Suggestions

- Troposphere and clock are more correlated with position.
  - Provide nominal troposphere.
  - Use an atomic clock.
- Be more careful of bad data.
- Default gd2p.pl model is random walk model on STA [XYZ]
  - See flags –type, -kin_sta_xyz
  - If random walk (or white noise) update rate is different than the data rate, and nominal positions are provided through an input TDP file, positions will be interpolated to measurement times.
- Iterate on solution unless you have a good nominal.
  - Depends on speed of platform.