



Introduction to gd2p.pl (GPS Data 2 Position)

Primary User Interface to GIPSY/OASIS

Jet Propulsion Laboratory, California Institute of Technology



Introduction to GIPSY-OASIS (GOA)



- GIPSY is a set of computer programs to analyze radio-metric data, with an emphasis on GPS.
- Includes many programs and scripts.
 - Experts glue them together for various applications.
 - Typically positioning applications
- Online GIPSY training for registered users:
 - <https://gipsy-oasis.jpl.nasa.gov/gipsy/docs>
- Online GIPSY video training:
 - https://gipsy-oasis.jpl.nasa.gov/gipsy/gipsy_admin/media/gotrain/streamintro.html
- Online GIPSY user group forum monitored by JPL.
 - <https://gipsy-oasis.jpl.nasa.gov/ubb-cgi-bin/ultimatebb.cgi>



Typical Sequence for Point Positioning with GPS Data



- **ninja**
 - Edit data.
 - Cycle slip detection.
- **goa_prod_ftp.pl, igs_orbits_clks_fetch.pl**
 - Fetch pre-determined GPS orbit and clock products.
- **qregres**
 - Model the edited data and compute partials with respect to parameters.
- **wash**
 - Fit the parameters, clean the data.
 - Includes **filter**, **smapper**, **postfit**, and **postbreak**.



Current JPL Orbit and Clock Products



- Orbit products identified in gd2p as:
 - ultra: Ultra-Rapid
 - qlR: Quick-look/Rapid
 - flinnR (Final): Flinn/Final Fiducial
 - flinnR_nf (Final): Flinn/Final Fiducial Free
- **goa_prod_ftp.pl** – script to download JPL orbit/clock products from:
 - `ftp://sideshow.jpl.nasa.gov/pub/JPL_GPS_Products`
- JPL orbit/clock products use absolute antenna calibrations
 - Receivers and transmitters.



Files (\$GOA/file_formats/ JPL_GPS_Products)



YYYY-MM-DD.pos.gz	Orbits of all satellites in pos_goa format
YYYY-MM-DD.tdp.gz	5-min Clocks and Yaw Rates of all satellites in tdp format
YYYY-MM-DD.eo.gz	Earth orientation tpmml and earth_orientation namelists
YYYY-MM-DD.shad.gz	Shadow events
YYYY-MM-DD.ant.gz	Identifies transmitter and receiver antenna calibration files used to generate orbit and clock products
YYYY-MM-DD.wlpb.gz	Global set of wide lane and phase biases, which may be used in smapper bias resolution
YYYY-MM-DD.frame.gz	Defines the Frame in which the orbits are provided
YYYY-MM-DD_nf.pos.gz	Orbits of all satellites in pos_goa format for fiducial free solution
YYYY-MM-DD_nf.tdp.gz	5-min Clocks and Yaw Rates of all satellites in tdp format for fiducial free solution
YYYY-MM-DD_nf.eo.gz	Earth orientation tpmml and earth_orientation namelists for fiducial free solution
YYYY-MM-DD_nf.wlpb.gz	Global set of wide lane and phase biases, which maybe used in smapper bias resolution for fiducial free solution
YYYY-MM-DD.x.gz	X-file providing 7 parameter transformation to convert fiducial free solution into the frame defined in the .frame file.
YYYY-MM-DD_hr.tdp.gz	30-sec Clocks and Yaw Rates of all satellites in tdp format
YYYY-MM-DD_nf_hr.tdp.gz	30-sec Clocks and Yaw Rates of all satellites in tdp format for fiducial free solution



Current JPL Orbit and Clock Products



- YYYY-MM-DD.ant.gz (\$GOA/file_formats/antenna_cal)
- Example 2010-01-10.ant.gz (just two lines)
SAT igs05_1567.xyz
ftp://sideshow.jpl.nasa.gov/pub/gipsy_files/gipsy_params/antenna_cals_xmit/igs05_1567.xyz
- GRN igs05_1567.atx
ftp://sideshow.jpl.nasa.gov/pub/gipsy_files/gipsy_params/antenna_cals_xmit/igs05_1567.atx
- User controls receiver antenna calibration used by **gd2p.pl**, including no calibration (default)
- **gd2p.pl** defaults for the transmitters to value in *.ant
- Best consistency achieved by using receiver antenna calibrations with JPL products.
- YYYY-MM-DD.wlpb.gz - \$GOA/file_formats/wlpblist, used by **gd2p.pl** for bias resolution



Introduction to gd2p.pl

- gd2p - GPS data 2 Position
- High Level Gipsy interface for processing data from a single GPS receiver (or GLONASS).
 - Other automated Gipsy point positioning systems (e.g. **point_rnx**) are not updated with modern features.
 - **gd2p.pl** is a more modern code written in a modern programming language (perl)
 - **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
- **To copy paste commands use: Intro_to_gd2p.txt**
 - **Will fix in a future release of this ppt**



Introduction to gd2p.pl

- Precision Orbit Determination (POD)
 - Dual Frequency
 - Single Frequency
 - Single Antenna
 - Multiple Antennas
- Currently used in production mode for:
 - Low Earth Orbiters:
 - Topex, JASON, GRACE, CHAMP, SACC, COSMIC, Jason-2/OSTM
 - Precise Point Positioning
 - Network solutions.
 - With ambigon2 or just gd2p.pl –amb_res 1
 - ambizap, Geoff Blewitt, Univ. of Nevada
 - Troposphere estimation.
 - APPS (Automatic Precise Positioning Service)
 - <http://apps.gdgps.net>
- Past Applications Included
 - SRTM (Shuttle Radar Topography Mission)
 - Ice Flow Studies in the Antarctic (kinematic)



Point Positioning Example from gd2p.pl -h_examples

- `gd2p.pl -h_examples | less`
 - yields many more examples
- 5 minute data, static position, nominal position obtained from one of the standard data bases, troposphere, clock, and position adjusted.

```
( gd2p.pl -i gol20100.10o -n GOL2 -r 300 -type s -d 2010-01-10 >  
gd2p.log ) > & gd2p.err &
```

- This was one of the motivating reasons. Point positioning should be simple. If the default database did not know about this station, you could use `-p X Y Z` for the nominal position. No need to make a database. `-env_km` for monument to ARP or Phase Center.



Orbit Determination

Example from gd2p.pl -h_examples



- Basic orbit determination.
- 60-second SunSat data, processed as GENRC satellite.

```
( gd2p.pl -n GENRC -r 60 -i ../data/sunsat.rnx  
  -gen_eci 'cp ../data/sunsat_5min_nom.eci leo_nom.eci'  
  -e "-a 0 -LC -PC -nosane" -d 1999-09-30 > gd2p.log )  
>& gd2p.err &
```

- NOTE: -nosane option on the default **ninja** editor. The polynomial fit for tracks below the horizon in orbit to L1-L2 is probably not good enough.



Help for gd2p flags

- **gd2p.pl -h | less**
 - Gives the entire help/man; written in POD (Plain Old Documentation)
- **gd2p.pl -h_sub**
 - Gives a listing of subtopics.
 - h_global
 - h_edit
 - h_model
 - h_wash
 - h_orbit
 - h_examples
 - h_files
- Beginners should always start with **-h_examples**



Detailed Precise Point Positioning



- From **gd2p.pl -h_examples**:

5 minute precise static positioning, using flinnR orbits and ground station antenna map

- First note from the flinnR file: 2010-01-10.ant.gz
zcat 2010-01-10.ant.gz

SAT igs05_1567.xyz ftp://sideshow.jpl.nasa.gov/pub/gipsy_files/gipsy_params/antenna_cals_xmit/igs05_1567.xyz
GRN igs05_1567.atx ftp://sideshow.jpl.nasa.gov/pub/gipsy_files/gipsy_params/antenna_cals_xmit/igs05_1567.atx

- SAT line indicates transmitter calibration file used to generate orbit/clock product, and location on JPL server.
- GRN line indicates receiver calibration file used to generate orbit/clock product, and location on JPL server.
 - In this example, flinnR used an IGS antex file for the ground station antenna calibration.



Creating a Receiver Antenna Calibration File (*.xyz)

- You can produce an xyz file consistent with the antex file with the utility **sta2antxyz.py**, and then use that file with **gd2p.pl**.
- In addition to the antex file, which has calibrations by antenna type and radome, you will need information about the antenna and radome for your site. If it is an IGS site, the site logs are parsed daily for antenna and radome information and converted to sinex format.

curl ftp://igscb.jpl.nasa.gov/igscb/station/general/igs.snx > /tmp/igs.snx
will return information about the current igs site

- Example:

```
( sta2antxyz.py -s GOL2 -antex $GOA_ETC/antenna_cals_xmit/igs05_1567.atx -sinex /tmp/igs.snx -o GOL2.xyz -d 2010-01-10 > sta2antxyz.log ) >& sta2antxyz.err &
```

If you already know the antenna type and radome, **antex2xyz.py**, will make an antenna xyz file without the need of the sinex file.



gd2p.pl Example with flinnR Orbit Products.



- Create a directory `static_position`, with the rinex file for GOL2 on 2010-01-10 and receiver calibration file (GOL2.xyz) both in the directory one level higher.
mkdir static_position
cd static_position
- Run **gd2p.pl** command:

```
( gd2p.pl -i ../gol20100.10o -n GOL2 -r 300 -type s -d 2010-01-10 \  
-add_ocnId -OcnIdCpn -tides WahrK1 PolTid FreqDepLove OctTid \  
-trop_z_rw 5E-8 -wetzgrad 5E-9 -w_elmin 7 \  
-post_wind 5.0E-3 5.0E-5 -AntCal GOL2.xyz -orb_clk flinnR \  
-arp \  
> gd2p.log ) > & gd2p.err &
```
- Could possibly add `-pb_min_slip 10e-5`
- Note that this example is using flinnR products:
 - ftp://sideshow.jpl.nasa.gov/pub/JPL_GPS_Products/Final
- So a consistent ground antenna map is used (GOL2.xyz).
- See the `-AntCal` and `-arp` flags and the programs **sta2antxyz.py** and/or **antex2xyz.py**.



Antenna calibration options in gd2p.pl example



- -orb_clk flinnR
 - Uses flinnR orbit products.
 - Automatically downloaded by **gd2p.pl** if local directory not specified or default local directory does not exist, see help for default local directories.
ftp://sideshow.jpl.nasa.gov/pub/JPL_GPS_Products/Final
 - Automatically applies transmitter antenna calibration identified in yyyy-mm-dd.ant file for that day.
- -AntCal GOL2.xyz
 - Use the receiver antenna calibration in GOL2.xyz
- -arp
 - When using sta_info database sets vector from antenna reference point (ARP) to phase center to be zero.
 - GOL2.xyz file contains phase center variation w.r.t. ARP.



gd2p flag details

- Details on all these flags are easy to find in the help, just do:

gd2p.pl -h |less

and search for "-flag" where flag is the appropriate string

for instance if you looked for -tides:

-tides [WahrK1 PolTid FreqDepLove OctTid]

Adds additional 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.



How to check a gd2p.pl run: gd2p.err

- Example above was run in: `./static_position`
`cd ./static_position`
- Many errors are caught in STDERR, (gd2p.err in this case).
`cat gd2p.err`
 - Returns the following content in gd2p.err

Don't forget that this wash script modifies the rgfile

- rgfile warning from gd2p, which is useful if you use one of the `–start_at` features.
 - Postfit outliers are marked in the regres file
(restore with **rgclean**)
- No obvious errors: Could still be garbage though.



gd2p.err

- For older data, the following widelane phase bias (used for single receiver ambiguity resolution) warning may be in gd2p.err, when the *.wlpb file does not exist in the JPL orbit/clock products:

WARNING: File, /home/ftp/pub/JPL_GPS_Products/Final/
2008/2008-02-15.wlpb.gz not found. JPL did not start making the files till
April 2009 at /GIPSY_source/build/gd2p/gd2p_pm/gd2p.pm line 465

WARNING: Can not execute zcat > ../wlpb

JPL did not start making the files till April 2009

probably no WLPB file for this date in source

"flinnR /home/ftp/pub/JPL_GPS_Products/Final/2008"

at /GIPSY_source/build/gd2p/gd2p_pm/gd2p.pm line 465



Basic Checks on gd2p.pl Run Postfit Residuals

- RMS of postfit residuals, in km, are in **Postfit.sum**.
cat Postfit.sum

start time	stop time	type	user	postfit	sigma	npts	outliers
10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	7.1969E-06		2454	1
10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.4065E-04		2455	0

- Probably OK unless you have done something special
- For a closer look at time series of the residuals
residuals.pl
Creates a file **residuals.txt**



Looking closer at the Position Solution in tdp_final



- **tdp_final** contains the position solution.
 - Time dependent parameter file.
 - See \$GOA/file_formats/tdp

- Extracting the position solution.

```
grep 'STA [XYZ]' tdp_final
```

```
316353600.0000 3676.97639800000 3676.97641765436 1.386E-06 STA Z GOL2
316353600.0000 -4641.38532500000 -4641.38529872640 2.080E-06 STA Y GOL2
316353600.0000 -2353.61442000000 -2353.61439881469 1.610E-06 STA X GOL2
```

- How much did the station move from the nominal in mm?

```
grep 'STA [XYZ]' tdp_final | cl '(c2-c3)*1e6'
```

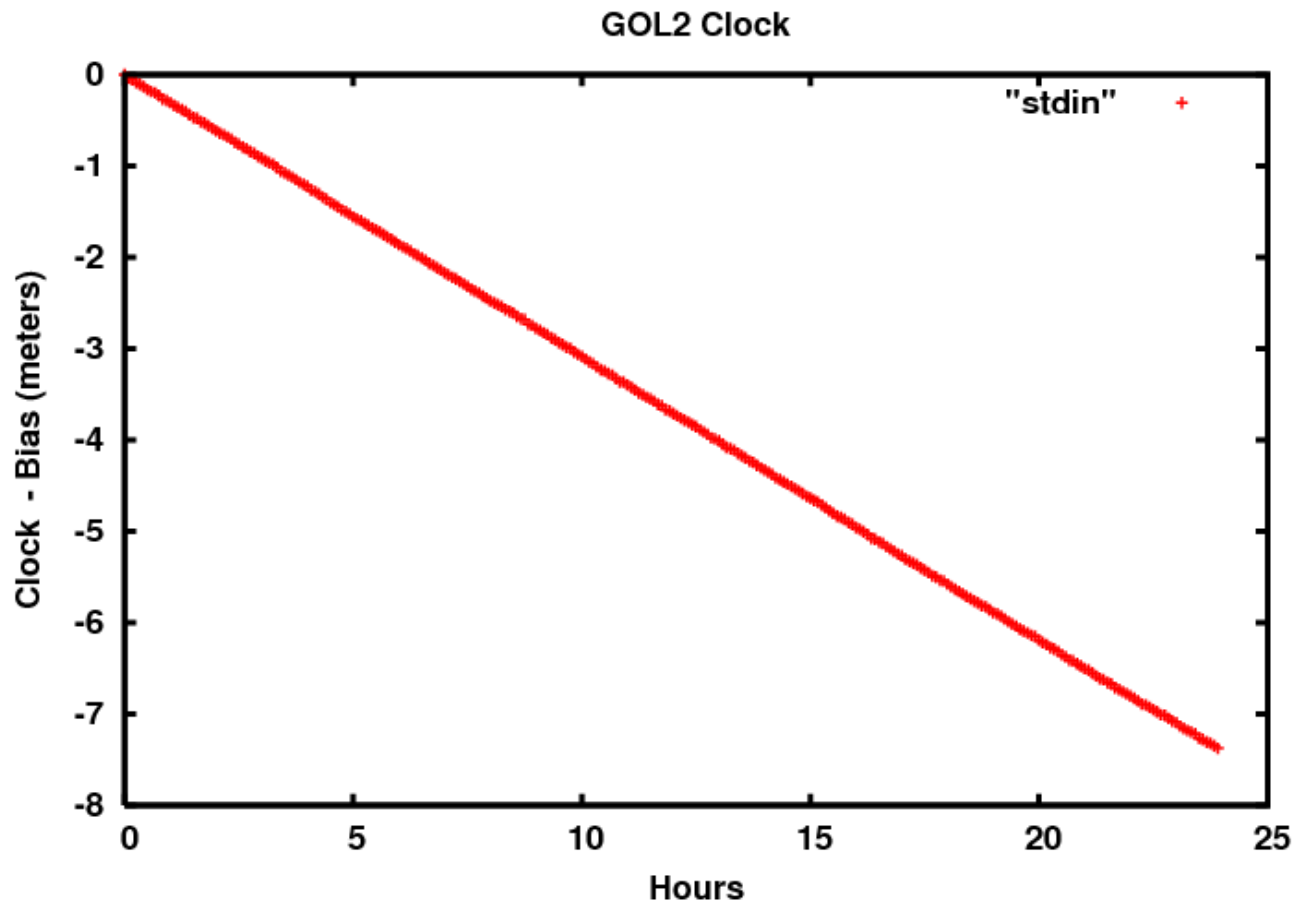
```
-19.6543601305166
-26.2736002696329
-21.1853098335268
```



Looking at the Clock Solution in tdp_final

- What was the clock behavior?

```
grep 'STA BIASGOL2' tdp_final | cl '(c1-f1)/3600' '(c3-f3)*1e3' | gnup  
-t "GOL2 Clock" -xl "Hours" -yl "Clock - Bias (meters)" # -gif  
clk.gif
```





What did gd2p.pl do?

- In GOL2 example, STDOUT sent to **gd2p.log**.
 - Provides a log of each of the GIPSY programs that were executed.
 - How they were executed (ie. options used).
- In GOL2 example, STDERR sent to **gd2p.err**.
 - Catches errors from run.
- An executable script, **run_again**, is created that reproduces the **gd2p.pl** run with all options used.
 - Either edit this file or copy it to make small modifications and re-run.
- Examine gd2p.log and run_again with your favorite editor/pager



Files Generated by gd2p.pl

gd2p.pl -h | less

- Search for “**FILES**”
- Files generated by **gd2p.pl** are described.
- Below is listing from GOL2 example run directory.

accume.nio	postbreak_postEdit.log.0	smsol.nio
batch.out.0	postfit.nio	Sum_postbreak_logs
batch.txt	Postfit.sum	tdp_clk_yaw
gd2p.err	prefilter.txt	tdp_final
gd2p.log	qmfile	tpnml
GOL2.xyz	qregres.nml	wash
GPS_antcal_ref	rgfile	wash.nml
logs/	run_again	wlpb
pos	shadow	
postbreak.log.0	smcov.nio	
postbreak.nml	smooth.nio	



stacov and Station Models

- Where are stacov files?
 - Stacov is needed for ambizap runs.
 - **gd2p.pl -h | less**
 - Look for -stacov.
 - If -stacov option invoked, for ground stations only, **gd2p.pl** will generate a stacov_final file in the same reference frame as the orbits.
- What station models were used?
 - See run_again, **gd2p.pl -h**, and qregres.nml.
 - Look for:
 - -trop_map
 - Troposphere mapping functions.
 - -ion_2nd
 - Uses second order ionosphere correction.



Kinematic Positioning with gd2p.pl



- Start by looking at an example:
gd2p.pl -h_examples | less
- 30 sec kinematic positioning, orbits and clocks from flinnR, troposphere modeled with scale height only

```
( gd2p.pl -i ../gol20100.10o -n GOL2 -r 30 -type k -d 2010-01-10  
-AntCal ../GOL2.xyz -orb_clk flinnR -arp > gd2p.log ) > & gd2p.err  
&
```

- NOTE: By specifying a higher rate than 5 minutes (ie. -r 30), **gd2p.pl** automatically fetches and uses JPL's high rate clocks.
 - See for example, high rate (_hr) tdp product used in gd2p.log
2010-01-10_hr.tdp.gz



Kinematic Positioning

Apriori Positions and Troposphere



- Kinematic positioning more complicated than static positioning.
 - Rapid movement or slow movement?
 - If moving rapidly, should have a nominal XYZ in an input tdp file (e.g. receiver navigation solution)
 - **Default quadratic interpolation**
 - Separating troposphere from position is more difficult.
 - Apriori troposphere from weather models (e.g. NCEP), can also be input through tdp file. (NCEP is not part of GIPSY)
 - If there is a tdp input file with XYZ in it and no troposphere GIPSY uses:
 - Zenith wet troposphere:
 - $0.1D_0 \cdot \exp(-ht/2000D_0)$ meters.
 - Zenith dry troposphere:
 - $1.013 \cdot 2.27 \cdot \exp(-0.116d-03 \cdot ht)$ meters.



Kinematic Positioning Suggestions

- Troposphere and clock are more correlated with position
 - Apriori trop
 - Atomic clock
- Be more careful of bad data.
- Default **gd2p.pl** model is a random walk model on XYZ (see flags **-type**, **-kin_sta_xyz**)
- Random walk correlates position in time.
- If the random walk (or white noise) update rate is different than the data rate and you iterate with a tdp file, the XYZ position at some measurements will be interpolated with a polynomial.
- Iterate unless you have a good nominal (rename tdp_final and input using **-tdp_in**).
- APPS tries to handle some of the complications automatically. <http://apps.gdgps.net>



Another Kinematic Example

- Modify previous example to use lower rate data and other options from static example:
 - Only change from static (-type s) is to use “**-type k**” and appropriate path to antenna calibration, GOL2.xyz.

```
mkdir kinematic  
cd kinematic
```

```
( gd2p.pl -i ../gol20100.10o -n GOL2 -r 300 -type k -d 2010-01-10  
-add_ocnld -OcnldCpn -tides WahrK1 PolTid FreqDepLove OctTid  
-trop_z_rw 5E-8 -wetzgrad 5E-9 -w_elmin 7 -post_wind 5.0E-3 5.0E-5  
-AntCal ../GOL2.xyz -orb_clk flinnR -arp > gd2p.log ) > & gd2p.err &
```



Sanity checks on Kinematic Solution

- Look for obvious errors in gd2p.err, e.g.:
cat gd2p.err
Don't forget that this wash script modifies the rgfile
- Compare postfit residuals to static case (e.g., **cat ../static_position/Postfit.sum**).
cat Postfit.sum

start time	stop time	type	user	postfit	sigma	npts	outliers
------------	-----------	------	------	---------	-------	------	----------

10JAN09 23:59:45	10JAN10 23:54:45	LC	GOL2	5.3090E-06	2455	0
------------------	------------------	----	------	------------	------	---

10JAN09 23:59:45	10JAN10 23:54:45	PC	GOL2	4.3985E-04	2455	0
------------------	------------------	----	------	------------	------	---

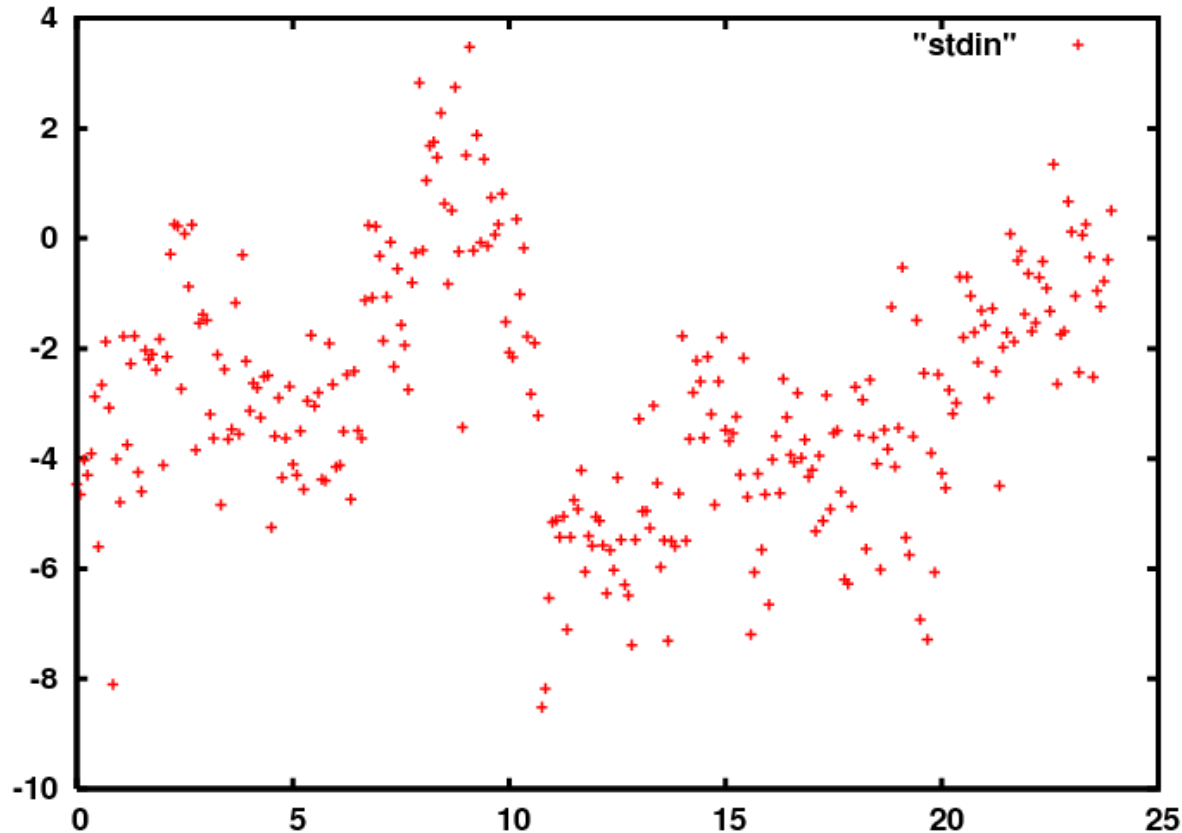
- As expected, residuals from kinematic solution are smaller than from static solution.
 - More degrees of freedom.



Looking at the Kinematic Solution w.r.t. Nominal

- Difference relative to nominal, taken from sta_info database (in cm)

```
grep 'STA Y' tdp_final | cl '(c1-f1)/3600' '(c2-c3)*1e5' | gnup
```

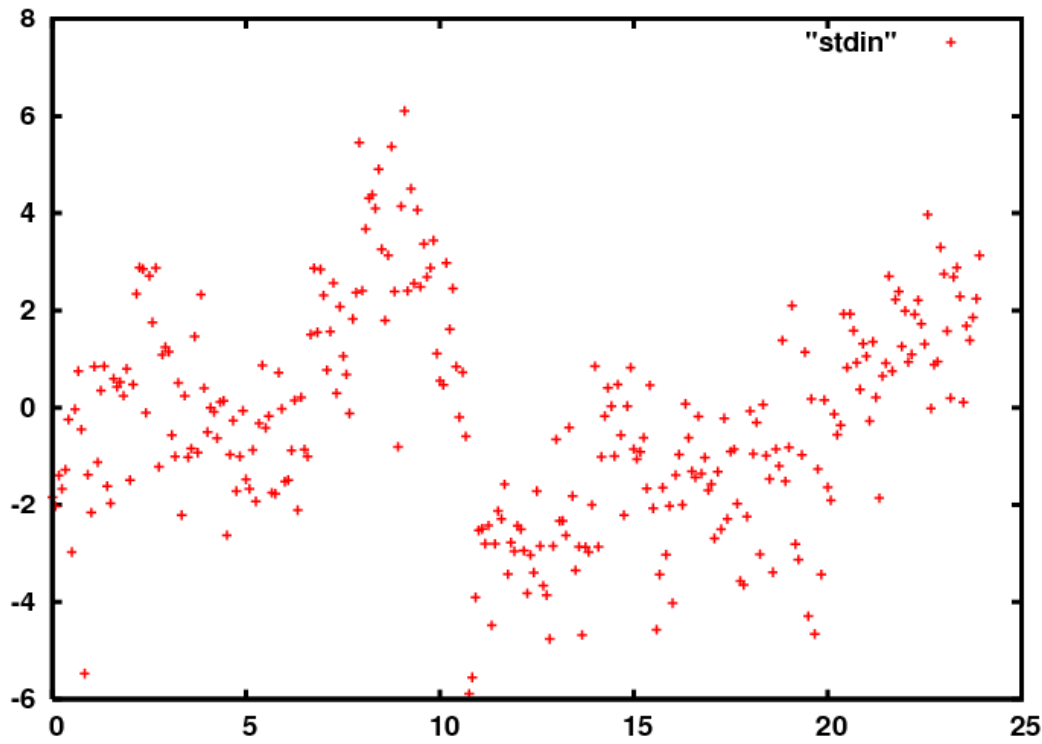


- Nominal appears to be biased from estimate.



Looking at the Kinematic Solution w.r.t. static solution

- Difference relative to our static position solution (in cm).
 - (e.g., `grep 'STA Y' ../static_position/tdp_final`)
- ```
grep 'STA Y' tdp_final | cl '(c1-f1)/3600' '(-4641.38529872640-c3)*1e5' | gnup
```



- Not biased w.r.t. static solution.

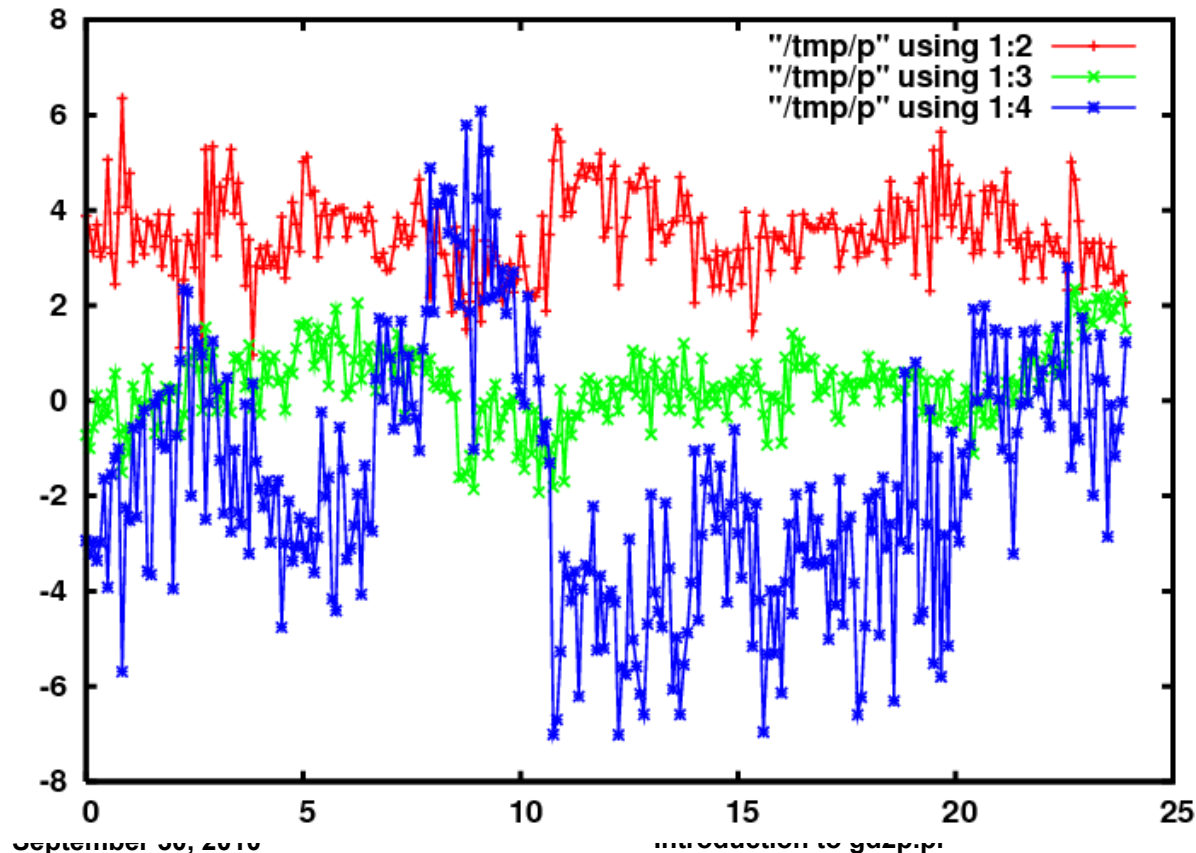


# Looking at the Kinematic Solution as North-East-Vertical (NEV)

- north/east/vertical (nev) relative to nominal in cm.

```
tdp2llh tdp_final GOL2 | grep est | cl '(c1-f1)/3600' 'c2*100' 'c3*100'
'c4*100' > /tmp/p
```

```
gnup -lp1:2 /tmp/p -lp1:3 /tmp/p -lp1:4 /tmp/p
```



- What is the scatter in North, East, Vertical?  
`cl 2 /tmp/p | stats | cl 3`

**0.9 cm North**  
**0.8 cm East**  
**2.6 cm Vertical**



# Modifying the Kinematic Run

- Suppose we know we have something moving slowly (e.g. ice flow).

**cp run\_again r2**

– Edit r2 by inserting:

**-start\_at qregres \**

**-kin\_sta\_xyz 100e-5 5.7e-7 300 RANDOMWALK \**

– DEFAULT: 2.0E3 1.0E-3 data\_rate RANDOMWALK

– NOTE:  $5.7e-7 * \sqrt{300}$  km  $\approx$  10 mm



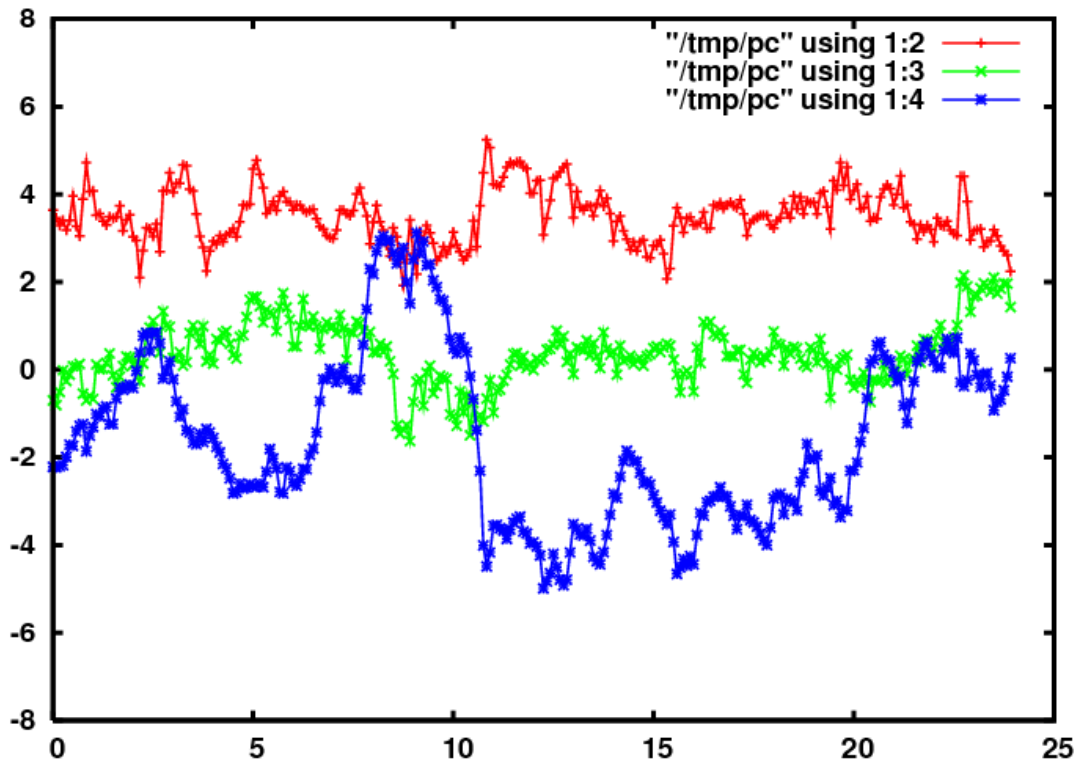
# Modified Kinematic Run

- New script has following **gd2p.pl** command:  
( **gd2p.pl -i ../gol20100.10o -n GOL2 -r 300 -type k -d 2010-01-10  
-add\_ocnld -OcnldCpn -tides WahrK1 PolTid FreqDepLove OctTid  
-trop\_z\_rw 5E-8 -wetzgrad 5E-9 -w\_elmin 7 -post\_wind 5.0E-3  
5.0E-5 -AntCal ../GOL2.xyz -orb\_clk flinnR -arp -start\_at qregres  
-kin\_sta\_xyz 100e-5 5.7e-7 300 RANDOMWALK > gd2p.log ) > &  
gd2p.err &**
  
  - Execute new script:
    - Runs gd2p again starting at qregres with new random walk parameters
- ./r2**



# Looking at New Kinematic Run

```
tdp2llh tdp_final GOL2 | grep nev | cl '(c1-f1)/3600' 'c2*100'
'c3*100' 'c4*100' > /tmp/pc
gnup -lp1:2 /tmp/pc -lp1:3 /tmp/pc -lp1:4 /tmp/pc -y -8:8
```



- What is scatter in the constrained solution:  
**cl 2 /tmp/pc | stats | cl 3**

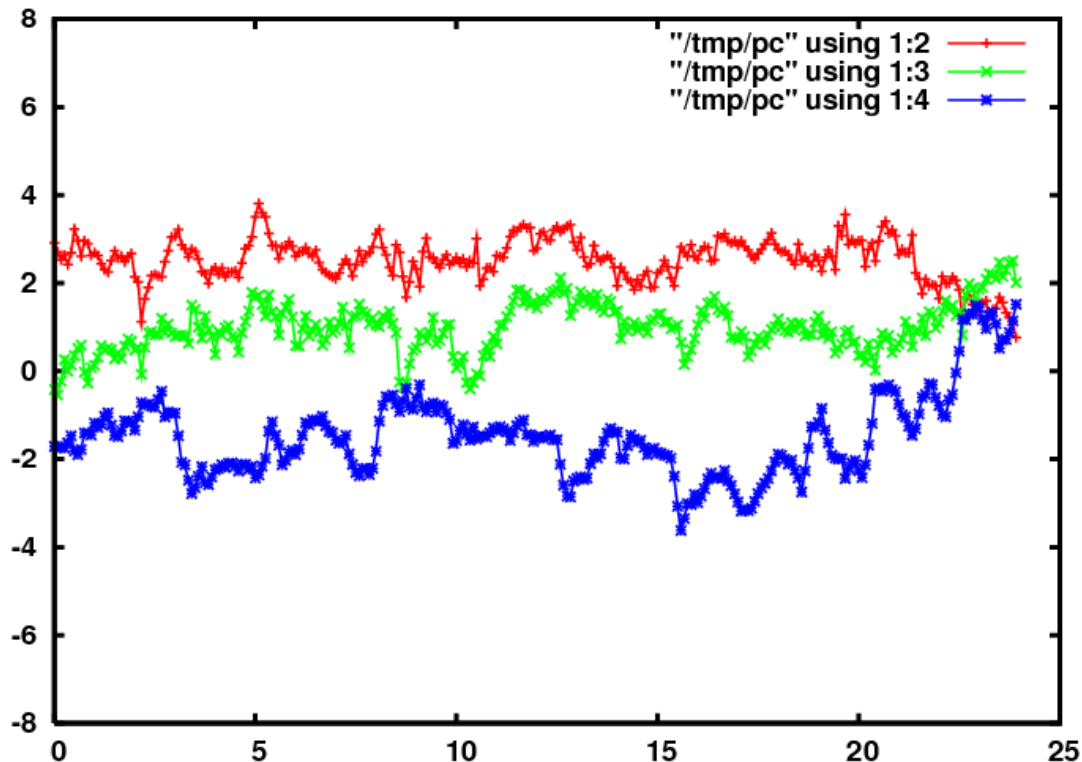
**0.6 cm North**  
**0.7 cm East**  
**1.9 cm Vertical**



# Now if we fix the trop

```
grep -P 'TRP|WETZ' ../static_position/tdp_final > tdp_trop_static
```

Modify run script, add `-tdp_in tdp_trop_static`, comment out trop



- What is the scatter in North, East, Vertical?  
`cl 2 /tmp/p | stats | cl 3`

**0.5 cm North**

**0.6 cm East**

**1.0 cm Vertical**

**Bias is in sta\_info**

**Add `-amb_res 1`**

**NEV Scatter: 0.4,0.4, 0.6**



# Generating GIPSY format products from IGS Orbit/Clocks



- **igs\_orbits\_clks\_fetch.pl -b 2010-01-09 -e 2010-01-11**
  - fetches igs final products  
igs15656.clk.Z igs15656.sp3.Z igs15660.clk.Z  
igs15660.sp3.Z igs15661.clk.Z igs15661.sp3.Z
- **igs2flinn.pl -b 2010-01-10 -e 2010-01-10**
  - Creates GIPSY format orbit and clock products from IGS products in current directory.  
2010-01-10.ant.gz 2010-01-10.eo.gz 2010-01-10.frame.gz  
2010-01-10.pos.gz 2010-01-10.shad.gz 2010-01-10.tdp.gz
- **igs2goa.pl --- more general, any length arc**



# Using IGS Orbit/Clocks with GOA

- **mkdir igs**
- **cd igs**
- **cp ../static\_position/run\_again r**
  - Edit script to point to directory with IGS orbits/clocks in GIPSY format.  
**-orb\_clk "flinnR /full\_path/igs\_orbits"**
- New gd2p.pl command:  
**( gd2p.pl -i ../gol20100.10o -n GOL2 -r 300 -type s -d 2010-01-10  
-add\_ocnld -OcnldCpn -tides WahrK1 PolTid FreqDepLove OctTid  
-trop\_z\_rw 5E-8 -wetzgrad 5E-9 -w\_elmin 7 -post\_wind 5.0E-3 5.0E-5  
-AntCal ../GOL2.xyz -orb\_clk "flinnR /full\_path/igs\_orbits" -arp >  
gd2p.log ) > & gd2p.err &**
- Execute modified script:  
**./r**



# Check static solution from IGS Orbit/Clocks



- Compare postfit residuals to static case with JPL orbits/clocks (e.g., `cat ../static_position/Postfit.sum`).

```
cat Postfit.sum
```

```
start time stop time type user postfit sigma npts outliers
```

```
10JAN09 23:59:45 10JAN10 23:54:45 LC GOL2 7.6926E-06 2458 3
```

```
10JAN09 23:59:45 10JAN10 23:54:45 PC GOL2 4.5950E-04 2461 0
```

- Compare static position estimates to static case with JPL orbits/clocks.

```
grep 'STA [XYZ]' tdp_final | cl 3 > /tmp/i; grep 'STA [XYZ]' ../static_position/tdp_final | cl 3 > /tmp/f
```

```
paste /tmp/[if] | cl '(c1-c2)*1e6'
```

```
0.622640072833747
```

```
-3.21125025948277
```

```
0.236309915635502
```



# Ambiguity Resolution with gd2p.pl (-amb\_res)

- see \$GOA/class\_gd2p/amb\_res.ppt for benefits/ comparison ambigon2/network processor
- From **gd2p.pl -h**:
  - amb\_res number\_of\_ iterations

Performs input number of iterations of ambiguity resolution from a global network (flinnR/qIR) bias fixed run using the global networks table of wide lane and phase biases ( \$GOA/file\_formats/wlpblist ). This file is retrieved from the orbit and clock databases automatically and placed in the current directory with the name wlpb.

default: 0 No ambiguity resolution will be attempted.

A value of 1 worked for station positioning.

A value of 10 worked for GRACE.



# Ambiguity Resolution



– `amb_res_nml amb_res_default.nml`

Namelist file to be used for smapper when performing bias resolution.

default: undefined

If it is not defined, a default value for low earth orbiters or stations will be constructed and written to file `amb_res_default.nml`.

- The default may not be appropriate.



# Example - Static Positioning from above with ambiguity resolution

- Just need to add "-amb\_res 1" if flinnR has wlpb file
  - Could use -start\_at amb\_res in the static directory, but let's save that directory.

```
mkdir static_position_amb_res
```

```
cd static_position_amb_res
```

```
cp ../static_position/run_again r
```

- Add ambiguity resolution option to input options, with one iteration  
-amb\_res 1

- New gd2p.pl command:  
( **gd2p.pl -l ../gol20100.10o -n GOL2 -r 300 -type s -d 2010-01-10 -add\_ocnld -OcnldCpn -tides WahrK1 PolTid FreqDepLove OctTid -trop\_z\_rw 5E-8 -wetzgrad 5E-9 -w\_elmin 7 -post\_wind 5.0E-3 5.0E-5 -AntCal ../GOL2.xyz -orb\_clk flinnR -arp -amb\_res 1 > gd2p.log ) > & **gd2p.err &****



## New files with ambiguity resolution

- Execute modified script:  
`./r`
- Check `gd2p.err` for errors.
- New files to note when using ambiguity resolution
  - **ls \*amb\***
    - Postfit\_amb.sum – Postfit residuals after ambiguity resolution  
(Postfit.sum – Postfit residual before ambiguity resolution)
    - tdp\_final\_pre\_amb – Solution before ambiguity resolution
    - tdp\_final – Solution after ambiguity resolution



# Compare pre- and post-ambiguity resolution residuals

- Residuals before ambiguity resolution identical to earlier static example.

## cat Postfit.sum

| start time       | stop time        | type | user | postfit    | sigma | npts | outliers |
|------------------|------------------|------|------|------------|-------|------|----------|
| 10JAN09 23:59:45 | 10JAN10 23:54:45 | LC   | GOL2 | 7.1969E-06 |       | 2454 | 1        |
| 10JAN09 23:59:45 | 10JAN10 23:54:45 | PC   | GOL2 | 4.4065E-04 |       | 2455 | 0        |

- Residual after ambiguity resolution slightly higher (as expected).

## cat Postfit\_amb.sum

| start time       | stop time        | type | user | postfit    | sigma | npts | outliers |
|------------------|------------------|------|------|------------|-------|------|----------|
| 10JAN09 23:59:45 | 10JAN10 23:54:45 | LC   | GOL2 | 7.3455E-06 |       | 2454 | 1        |
| 10JAN09 23:59:45 | 10JAN10 23:54:45 | PC   | GOL2 | 4.4071E-04 |       | 2455 | 0        |



# Compare pre- and post-ambiguity resolution solutions

- Differences in solutions by comparing `tdp_final_pre_amb` and `tdp_final`:  
**grep 'STA [XYZ]' tdp\_final\_pre\_amb > tmp1**  
**grep 'STA [XYZ]' tdp\_final > tmp2**  
**paste tmp1 tmp2 | cl '1.0e6\*(c3-c10)'**  
-0.260390152106993 (X = -0.3 mm)  
0.943149643717334 (Y = 0.9 mm)  
-0.380330220650649 (Z = -0.4 mm)



# Example - Kinematic Positioning from above with ambiguity resolution

- Just need to add "-amb\_res 1" if flinnR has wlpb file
  - Could use -start\_at amb\_res in the kinematic directory, but let's save that directory.

```
mkdir kinematic_position_amb_res
```

```
cd kinematic_position_amb_res
```

```
cp ../kinematic/run_again r
```

- Add ambiguity resolution option to input options, with one iteration  
-amb\_res 1



# Kinematic Position with Ambiguity Resolution



- New gd2p.pl command:  
( gd2p.pl -i ../gol20100.10o -n GOL2 -r 300 -type k -d 2010-01-10  
-add\_ocnId -OcnIdCpn -tides WahrK1 PolTid FreqDepLove OctTid  
-trop\_z\_rw 5E-8 -wetzgrad 5E-9 -w\_elmin 7 -post\_wind 5.0E-3 5.0E-5  
-AntCal ../GOL2.xyz -orb\_clk flinnR -arp -kin\_sta\_xyz 100e-5 5.7e-7  
300 RANDOMWALK -amb\_res 1 > gd2p.log ) > & gd2p.err &
- Execute modified script:  
./r
- Check gd2p.err for errors.



# Questions?

