The Pointing Model

Note: The following is a description of the pointing analysis done on VELA. You **must** be logged in as director. Once you have finished the pointing run:

1. Click on `Pointing`
2. Open the *pointing data*.
3. Choose *SEST Pointing* for *Extract Data* and *space* for *field separator*
4. Fill in the dates for your pointing run and select *send the data to* `director` on `PUPPIS`.
   Note: `FTP` the data only to PUPPIS or gemini, if you need to make further analysis.
5. The pointing data will now be saved in `poin_yymmdd-yymmdd.dat` in the director's home directory.
6. `mv poin_yymmdd_yymmdd.dat ./pointing`
   Moves the file to the directory `/aux2/home/daprog/director/pointing`
7. `rm poin_yymmdd_yymmdd.sou`
   This file only contains the names of the sources used in the pointing run and can be safely removed.
8. `cd ./pointing`
9. The following steps is how to extract the data in a proper format for later data-analysis:
   (a) `point_file_sort`
   This will start the data sorting/pruning software. If you need a menu of the commands in `point_file_sort` use `?`.
   (b) `ofp` and select `poin_yymmdd-yymmdd.dat` as the database that you want to work with.
   (c) `wf` to create an initial data file for processing.
   (d) Call it `yymmmdd.wf`.
   (e) Type `x` to exit
10. Now we will try to determine the pointing model from the processed database (`yymmmdd.wf`).
   (a) `point_fit` to run an initial fit on the database.
   (b) For `Data base 1` load the file `yymmmdd.wf`, you just created.
   (c) Just press `Return` for the second data base if not needed (normally only one database is used).

(d) Answer the next question about database format with `1` (Since you are now using a `wf` formatted file).

(e) Then, at the next prompt, enter the file name for the fit output.

(f) Call it `yymmmdd.fit1` for the first fit (at subsequent steps you will increment this number by 1 *ie* in second iteration call the file `yymmmdd.fit2` etc...).

(g) `point_fit` will exit after the fit has been executed.

11. Look at the fit file just created (`more yymmmdd.fit1`)

12. `lp yymmmdd.fit1`
    to print out the file with the results of the fit on `orion`.

13. First you will see the initial rms fitted to the data. Down below, you will see how big the pointing corrections for each source are before and after the fit.

14. Mark down the numbers for those sources, which you believe should be flagged as bad. Keep these numbers and the printout as later manipulating of the dataset will use these original numbers whenever you want to edit the data (editing of the data is always done on the first dataset). This means that if you find a bad point in an edited file then you will have to identify the point in the original printout to find out its number.

15. Go back into `point_file_sort`, where you can now manipulate the data.

    (a) `ofp`

    (b) Load `poin_yymmdd-yymmdd.dat`

    (c) Now type `md` to manipulate data.

    i. Choose `ma` to mask the bad numbers. Using starting and stopping numbers. To mask a single bad number specify the number twice ie `4 4` to mask scan `4` as the flagging needs both a starting and finish number. Repeat this flagging to flag all bad points as necessary.

    ii. `el` to mask all data taken at an elevation above or below certain limits f.ex `80`^{o} `and 25`^{o}

    iii. With `lm` you can list all of the data marked as bad to double-check.

    iv. Type `-` to go back to the main menu

    v. `wf` to write a new file, this time call it `yymmmdd.md`.

    vi. type `x` to exit

16. Now you can run `point_fit` again

    (a) Use the modified data base `yymmmdd.md`

    (b) Write the new fit results into `yymmmdd.fit2`

(c) `point_fit` will exit after the fit has been executed.

17. `lp yymmmdd.fit2`
    To print out the result of the fit.

18. Check the results again. If the flagging of the bad data has been done properly, this should give you a satisfying result.

19. If the fit isn't satisfactory then you need to identify the sources with the large offsets in the file (`yymmmdd.md`, see also point 14).

    (a) Go back to the original data set (`yymmmdd.wf`) to find the correct number for flagging the sources with `point_file_sort` again (see step 14 where you noted down the previously bad points) as you also need to flag again the bad points previously flagged.

    (b) remove the unsuccesful (`.md`) file.

    (c) Run `point_file_sort` and `point_fit` again ie repeat step 19 and add the bad points you found in steps 18 to the bad points in step 14 and 15(c)ii. **Note:** Call the result of the fit `yymmmdd.fit3`

20. If you still are not satisfied with the fit then go back to step 19 (repeat this until you are satisfied with the fit).

21. When finished, print out the last file with the result of the fit:
    `lp yymmmdd.fitX`
    where `X` is 2, 3, or . . . *last fit*

22. Put this printout in the *Pointing binder*

23. On VELA just type: `cd ~`(To go to the `home` directory.

24. `cp .,poin.par .,poin.par.mmmdd_yyyy`
    backing up the old pointing file

25. edit `.,poin.par`.
    Note! values for `el3` need to be negated (sic).

    (a) Find the old pointing corrections and replace them with the new values in *both* columns, since the second column is what PEGASUS will read as `default`.

    (b) Find the variables `absaz`, `absel`, `az`, `el` and put them all to zero in both columns.

    (c) Save the edited file.

26. Logout as `director`

27. Login to VELA as `daprog`

28. You now have to install the pointing file globally.
    (a) Run the script `update_pointing`
    (b) Follow the instructions
    (c) logout
29. login as `director`
30. Check that the new pointing offsets are installed by checking the new pointing offsets in the PEGASUS window $\boxed{\text{Point}}$.
31. Use $\boxed{\text{Files}}$ to write all the pointing spectra to tape. Use the pointing DAT tape for this purpose.
32. Use `archive` to backup the `director`'s account on the `director` DAT tape.
33. Clean the account afterwards!