National Radio Astronomy Observatory Tucson, Arizona

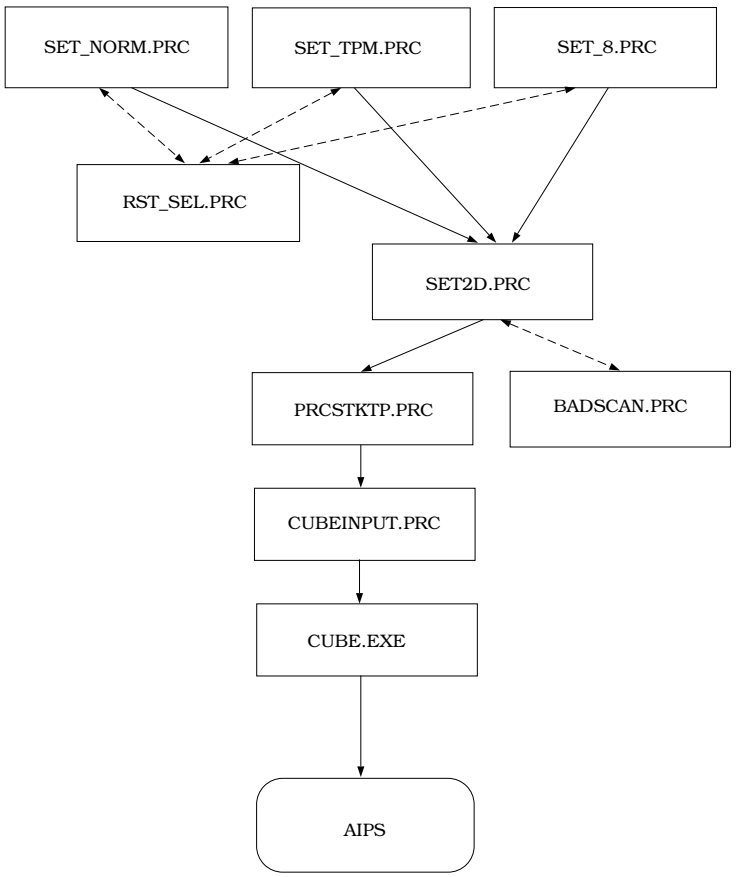
To: Observers

From: Jeff Mangum

Subject: Processing Map Data in UniPOPS

Date: February 18, 1994

A number of general UniPOPS procedures designed to aid observers with the reduction of map data are now available on bohemia. In the following I describe how to use these procedures. **NOTE: These procedures have been tested, though not extensively. If you have trouble using them, let me know (jmangum@as.arizona.edu).** The diagram below shows the general flow pattern through these procedures. In this diagram, solid arrows represent the primary execution route through the procedures, while the dashed arrows indicate conditional branches.



Some Hints Before You Begin

By definition, the processing of map data is tedious, involving a number of UniPOPS variables which can have far-reaching implications for your other data processing needs. Because of these potential pitfalls, I list below a number of hints designed to make your travel through the map processing procedures a little smoother...

1. When processing TPM data, you should pay attention to the online display of your data and note any bad off (TPMF) or on (TPMO) scans. This will speed up the processing of this kind of data by minimizing the number of times you must use SET_TPM.

2. Once you have finished processing a map data set (when you have successfully run CUBEINPUT), your SAVE file is left as TPSAV. It is probably best if you use CHNGFILE to change this back to its default value of LSAVE.

Batching the Procedures Into UniPOPS

In order to have access to these map procedures, you must first BATCH them in. To first see if you have enough space in memory to BATCH in these procedures, use the CORE command to check your current memory allocation...

```
LineF> core
```

The map procedures will occupy approximately 30% of the available program space, 20% of the available variable space, and 40% of the available source space. If you do not have at least this much space available, you must SCRATCH some of the procedures you currently have loaded.

To BATCH all of the procedures necessary for map processing, issue the following command in UniPOPS...

```
LineF> batch mapprocs.plib
```

Processing "Normal" Map Data:

In this memo, "normal" data are those obtained using any mode other than TPM, 8-beam, or frequency switching. The reduction of normal map scans proceeds as follows within UniPOPS...

```
LineF> set_norm
```

This procedure will setup the SELECT parameters necessary to put all of your mapping scans on the stack. If you are running this procedure for the first time or wanting to reset the SELECT adverbs, SET_NORM will do it for you by running the procedure RST_SEL. It will then run the main procedure SET2D, which will ask you a series of questions designed to help you edit-out bad scans and put all of the scans you want to process onto the stack.

```
LineF> prcstktp(xtol,ytol,dscans)
```

This procedure will add all scans on the stack using *xtol* and *ytol* as positional tolerances for the summations. For example, if your map was obtained using a spacing between positions of 15″, then set *xtol* and *ytol* to 15. The parameter *dscans* is a global adverb which tells PRCSTKTP that you want to GET scans from the online data file. **NOTE:** PRCSTKTP uses a scratch SAVE file called TPSAV to hold the reduced scans. Each successive run of PRCSTKTP deletes any existing copies of TPSAV. When PRCSTKTP finishes, it will change your SAVE and KEEP files back to their default values of LSAVE and LKEEP, respectively.

At this point, you can run the tileing procedure TILESAVES to make a tile plot of your map (**NOTE:** the tileing procedures are not part of the mapprocs library). But, since UniPOPS is extremely limited in its ability to process map data, a port to the AIPS reduction package has been setup. For directions on how to transfer your reduced map to the AIPS program for further processing, see the section entitled **Porting Your UniPOPS Cube to AIPS**.

Processing Total Power Map (TPM) Data:

The processing of TPM data proceeds along the same path as that for normal map data, with the exception that the initialization procedure to run is...

```
LineF> set_tpm
```

As with normal map data, SET_TPM will setup the SELECT parameters necessary to edit-out bad on (TPMO) and off (TPMF) scans and put all of the total power ons and offs you wish to process onto the stack. It also allows for reseting through the procedure RST_SEL. **NOTE:** SET_TPM uses scratch SAVE and KEEP files called TPMSAV and TPMKP, respectively, for intermediate processing. The SAVE file is changed back to the default value (LSAVE) when it finishes, but the KEEP file is left as TPMKP. SET_TPM is followed by PRCSTKTP...

```
LineF> prcstktp(xtol,ytol,kscans)
```

...which will add all scans on the stack using *xtol* and *ytol* as positional tolerances for the summations. The parameter *kscans* will allow PRCSTKTP to KGET processed scans from the scratch KEEP file TPMKP. **NOTE:** PRCSTKTP uses a scratch SAVE file called TPSAV to hold the reduced scans. Each successive run of PRCSTKTP deletes any existing copies of TPSAV. When PRCSTKTP finishes, it will change your SAVE and KEEP files back to their default values of LSAVE and LKEEP, respectively.

At this point, one should proceed to AIPS for further processing. For directions on how to transfer your reduced map to the AIPS program for further processing, see the section entitled **Porting Your UniPOPS Cube to AIPS**.

Processing 8-Beam Map Data:

The processing of 8-Beam Receiver data proceeds along the same path as that of normal map data, with the exception that the initialization procedure to run is...

```
LineF> set_8
```

As with normal map data, SET_8 will setup the SELECT parameters necessary to edit-out bad scans and put all of your 8-Beam Receiver scans onto the stack for processing. It also allows for reseting through the procedure RST_SEL. SET_8 is followed by PRCSTKTP...

```
LineF> prcstktp(xtol,ytol,dscans)
```

...which will add all scans on the stack using *xtol* and *ytol* as positional tolerances for the summations. The parameter *dscans* will allow PRCSTKTP to GET the scans you want to process from the online data file.

NOTE: PRCSTKTP uses a scratch SAVE file called TPSAV to hold the reduced scans. Each successive run of PRCSTKTP deletes any existing copies of TPSAV. When PRCSTKTP finishes, it will change your SAVE and KEEP files back to their default values of LSAVE and LKEEP, respectively.

At this point, one should proceed to AIPS for further processing. For directions on how to transfer your reduced map to the AIPS program for further processing, see the section entitled **Porting Your UniPOPS Cube to AIPS**.

Porting Your UniPOPS Cube to AIPS

Transfer of your processed UniPOPS map to the AIPS program is done using the FITS format. The first procedure to run is...

```
LineF> cubeinput(xtol,ytol)
```

...where *xtol* and *ytol* should be the same as those given to PRCSTKTP. CUBEINPUT will setup the inputs necessary for transfering your reduced map SAVE file TPSAV (which, remember, is also the current SAVE file) to a FITS cube file. The type of geometric projection produced in your FITS file depends on the current value for the UniPOPS global parameter *docar*. The default value of *docar* is FALSE, which will give you a map grid in the GLS (global-sinusoidal projection, which is equivalent to the standard cos(dec) correction for RA offsets from the RA reference position of the map) projection. If *docar* is set to TRUE, you will get a grid with no projection (a cartesian map with linear axes).

NOTE: The output FITS cube file must be unique.

You will be asked the following questions by CUBEINPUT:

Input sdd file (save file) : TPSAV

Enter name of cube file : *enter unique FITS cube file name* CUBEINPUT writes a text file called cube.input which contains all of the inputs necessary for the cube writing program CUBE.EXE. Therefore, the next step is to use the SYSTEM verb to run CUBE.EXE on your TPSAV file...

```
LineF> system cube.exe < cube.input
```

At this point you have a FITS cube of your map data which can be read into AIPS.

Processing Map Data in AIPS

Before entering the AIPS program, you should copy your FITS cube to the AIPS FITS subdirectory...

```
% cp fitscube $FITS/FITSCUBE
```

This will allow the AIPS task IMLOD to find the FITS cube. Note that AIPS prefers file names to be upper case, so that the name fitscube becomes FITSCUBE when copied to the AIPS FITS area.

Now enter the AIPS program from any window...

```
% aips
```

You will be asked a number of questions by the AIPS startup. When you are asked for a user number, you can either enter your own personal AIPS user number (if you have one) or you can enter 10, which we use as a general user number.

Now that you are in the AIPS program, you can transfer the FITS map cube into AIPS for further processing using the task IMLOD...

```
> task 'imlod'

> inputs

> outdisk 1

> infile 'FITSCUBE'

> go
```

IMLOD will create an image cube on AIPS disk 1. Note that this cube will have an axis order of (velocity,RA,DEC). Further processing can now be done to make a tile plot (task PLCUB), make a contour plot (task CNTR), make a channel map contour plot (task KNTR), fit baselines to the individual map spectra (task XBASL), etc. For example, to use PLCUB to make a tile plot of your spectra...

```
> task 'plcub'

> indisk 1

> getn 1

> go
```

This will make a plot file of your cube which is attached to the image file via the PL extension. You can display this PL extension file using the tasks TKPL (for a Tektronics plot on your console) or LWPLA (to make a hardcopy).