NOEMA tutorials: III. GRB 04768249

by

Victor de Souza Magalhaes & Jérôme PETY

IRAM

The datasets are not public as of Oct. 2018

⇒ We tampered with the datasets so that it's impossible to do quantitative science from them.

Scripts for this tutorixal are available in the following files

`pro/grb04768249-beginner.map`, `pro/grb04768249-uvfit.map`

The scripts assume that you use oct18 or a more recent version of `GILDAS`

IRAM Millimeter Interferometry Summer School

Oct. 1 - 5 2018, St Martin d'Hères

Size of the problem: I. UV plane

From this slide on, please look into file pro/grb04768249-beginner.map

```
[imiss@reducv2 grb04768249]$ ls -lh school-2mm-01-uo.uvt
-rw-r----- 1 imiss project 160K Oct  2 14:20 school-2mm-01-uo.uvt
MAPPING> header school-2mm-01-uo.uvt
File : school-2mm-01-uo.uvt                                    REAL*4
Size        Reference Pixel          Value                  Increment
        10   1.00000000000000       153774.713281644       1.00000000000000
      3960   0.00000000000000       0.00000000000000       1.00000000000000
Blanking value and tolerance      1.23455997E+34   0.0000000
Source name          SN04768249
Map unit             Jy
Axis type            UV-DATA      RANDOM
Coordinate system    EQUATORIAL          Velocity    LSR
Right Ascension    11:06:29.80158        Declination      -08:37:31.0357
Lii         0.000000000000000            Bii       0.000000000000000
Equinox             2000.0000
Projection type      AZIMUTHAL           Angle     0.000000000000000
Axis 0     A0     11:06:29.80158        Axis 0     D0    -08:37:31.0357
Baselines                0.0       0.0
Axis 1 Line Name     cont2mm             Rest Frequency   153774.7132816444
Resolution in Velocity   -7123.6787      in Frequency          3654.0000
Offset in Velocity        0.0000000      Doppler Velocity      22.643791
Beam                   32.8                 0.00                  0.00
NO Noise level
NO Proper motion
Tel: NOEMA           05:54:28.5   44:38:02.0 Alt.    2560.0 Diam   15.0
UV Data    Channels:       1, Stokes: 1 None         Visibilities:        3960
```

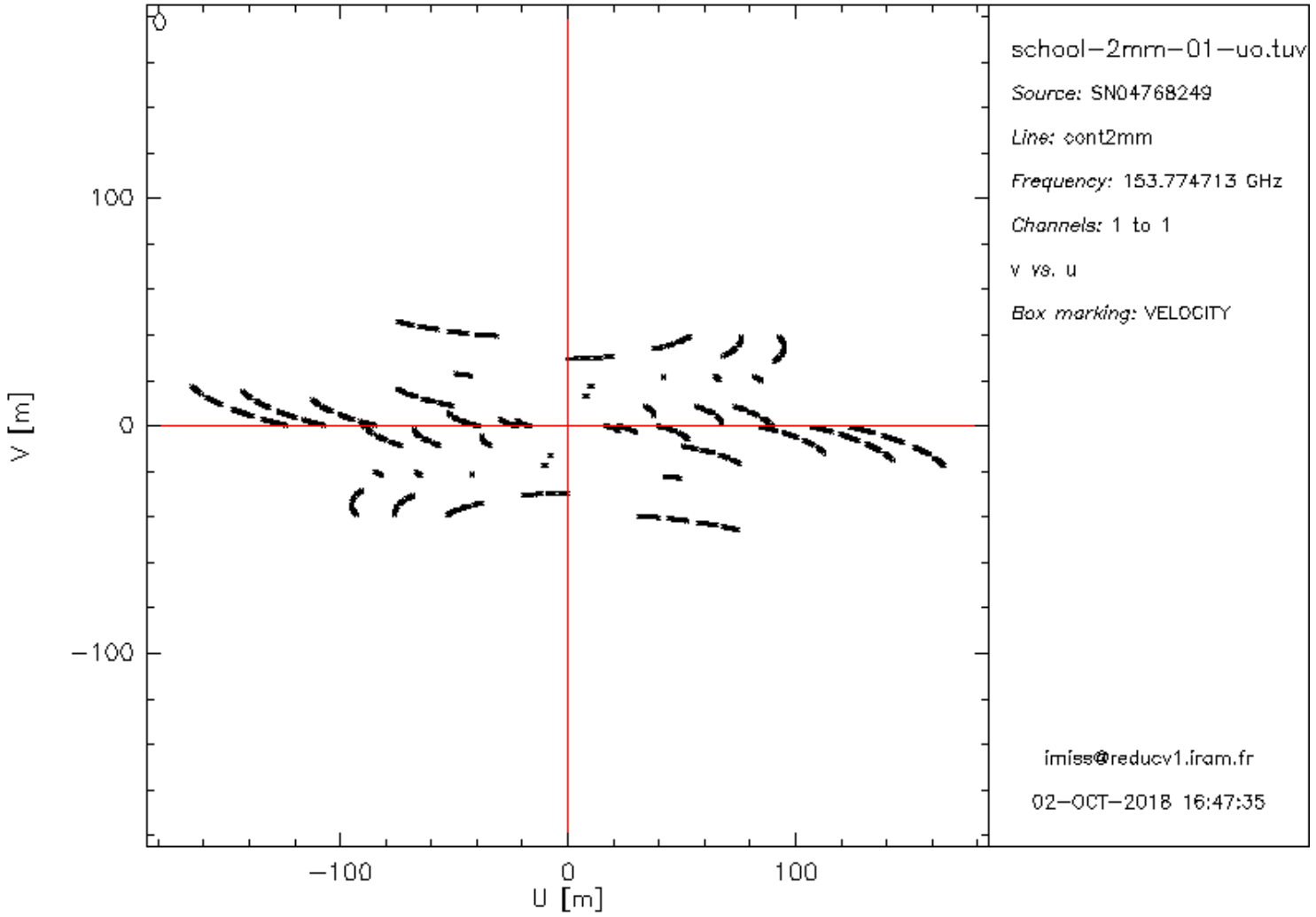
Size of the problem: II. Image plane

```
MAPPING> let name school-2mm-01-uo
MAPPING> go setup
Input file: Interferometer (15m)  school-2mm-01-uo.uvt

   Single field observation (3960 visibilities)

   Observed rest frequency       153.77471328164 GHz
   Half power primary beam       32.8 arcsec
   Phase center RA and Dec       11:06:29.802 -08:37:31.036
   Field of view / Largest Scale 32.8 x 32.8 arcsec

                                 Recommended        Used
   Spectral axis size               1            1 channels
   Map size                       128 x 128       128 x 128 pixels
   Map cell                      0.48 x 0.48     0.48 x 0.48 arcsec
   Image size                   61.9 x 61.9   61.9 x 61.9 arcsec
   Cube size                        64            64 kiB

   Still to be imaged
   Still to be cleaned
```

⇒ Extremely small dataset!

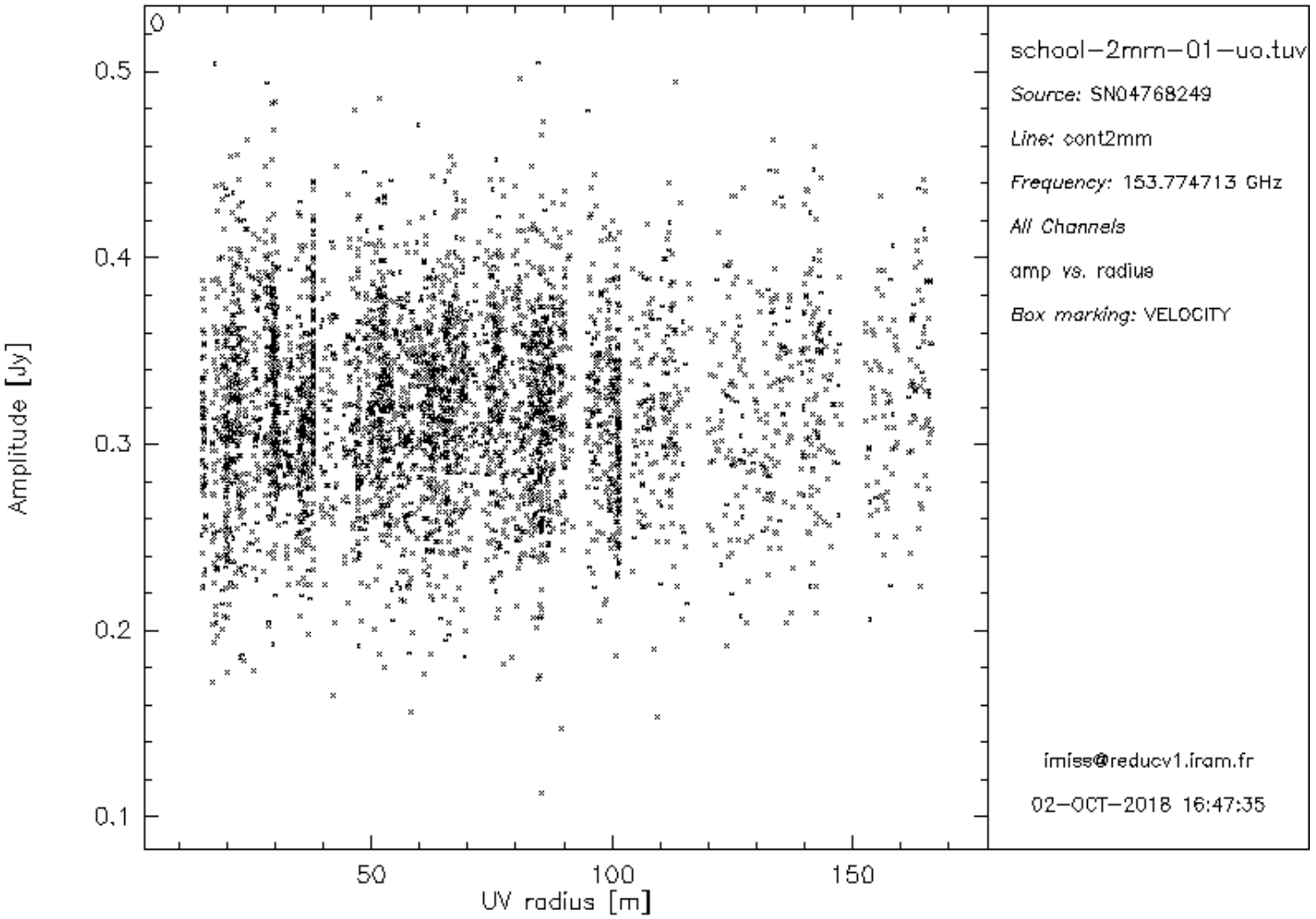
Looking at the data: I. uv coverage

```
MAPPING> let name school-2mm-01-uo
MAPPING> go uvcov
```



Looking at the data: II. Amplitude vs uv radius

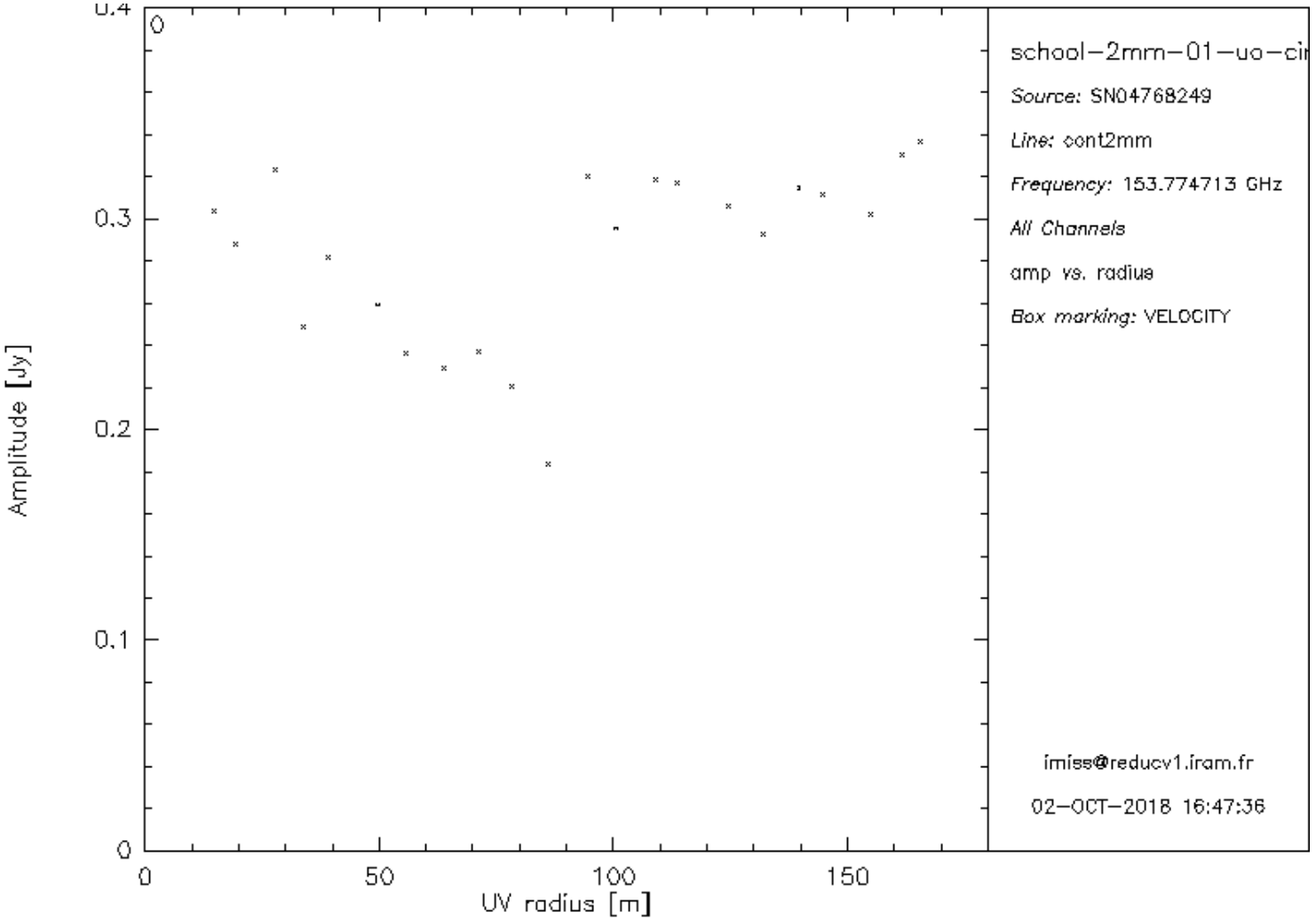
```
MAPPING> let name school-2mm-01-uo
MAPPING> go uvshow
```



⇒ Very noisy. Let's average at constant uv distance.

Looking at the data: II. Amplitude vs uv radius

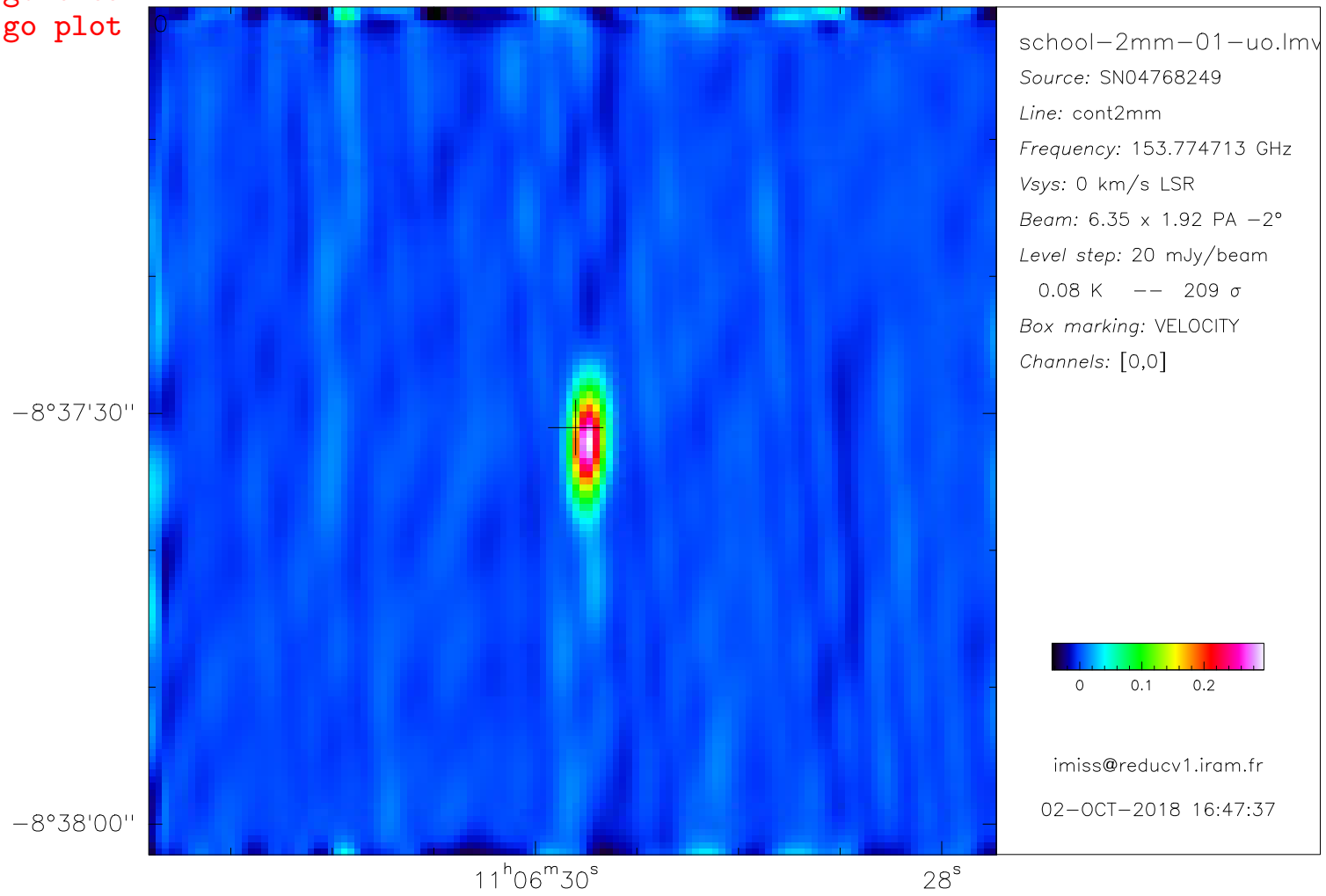
```
MAPPING> let name school-2mm-01-uo
MAPPING> go uvcirc
MAPPING> go uvshow
```



⇒ Could be compatible with a point source. Let's image it.

Making an image to check the spatial distribution of the source brightness

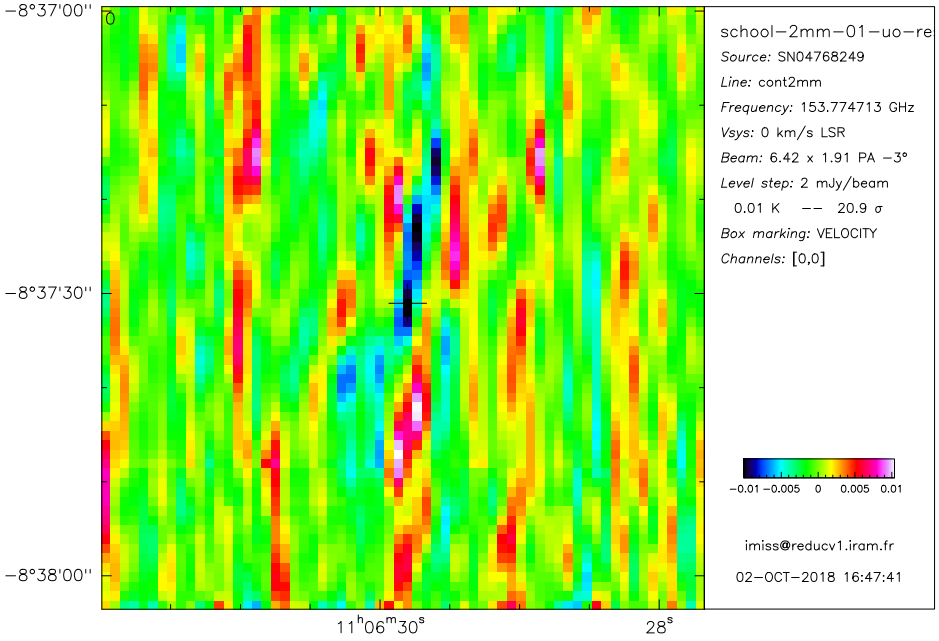
```
MAPPING> let name school-2mm-01-uo
MAPPING> go uvmap
I-CLEAN,  Beam is     6.35" by     1.92" at PA    -2.35 deg.
I-CLEAN,  Errors (    0.03)   (    0.01)      (    0.12)
MAPPING> go clean
MAPPING> go plot
```



⇒ The source seems point-like. Let's check it.

Is it a point source? ⇒ Fitting in the uv plane.
I. Point source

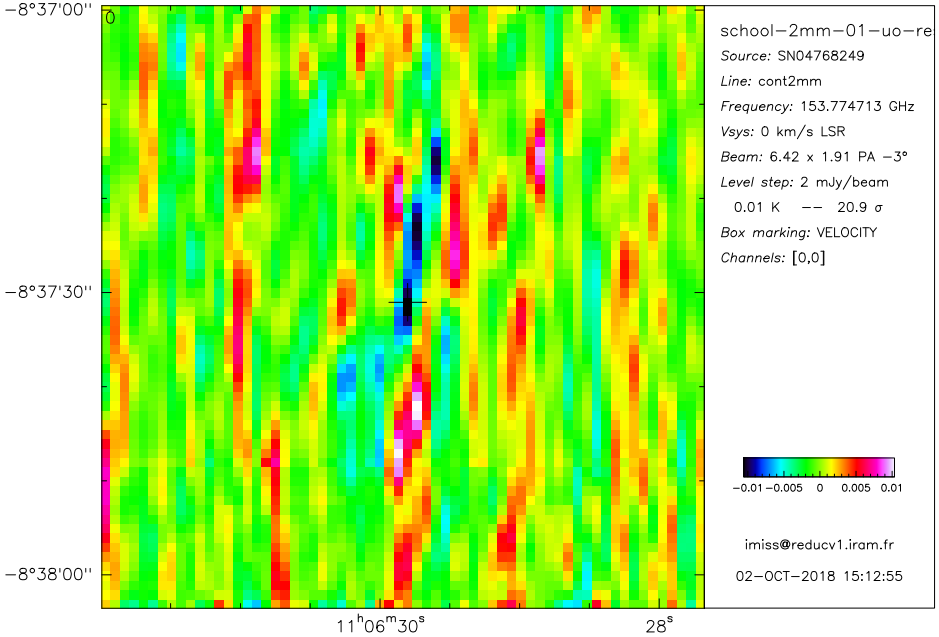
```
MAPPING> let name school-2mm-01-uo ! Start fitting the visibilities
MAPPING> let uvfit%subsf01 yes ! => Subtracting the fitted function
MAPPING> let uvfit%funct01 "point"
MAPPING> go uvfit
r.m.s=     0.1296 Jy.
POINT    R.A.          =    -0.80926 (    .00281)   11:06:29.74701
POINT    DEC.          =    -1.07539 (    .00850)  -08:37:32.1111
POINT    FLUX          =   312.95362 (    .98413)  milliJy
MAPPING> pause
MAPPING> let name school-2mm-01-res ! Start imaging and deconvolving the residuals
MAPPING> go uvmap
MAPPING> go clean
MAPPING> go plot clean
```



⇒ Residuals looks like noise! Let's check.

Is it a point source? ⇒ Fitting in the uv plane.
II. Circular Gaussian

```
MAPPING> let name school-2mm-01
MAPPING> let uvfit%subsf01 yes
MAPPING> let uvfit%funct01 "c_gauss"
MAPPING> let uvfit%range01 0 0 0 1 0 0 0
MAPPING> let uvfit%start01 0 0 0 2 0 0 0
MAPPING> go uvfit
r.m.s.=     0.1296 Jy.
C_GAUSS  R.A.          =    -0.80930 (    .00282)   11:06:29.74701
C_GAUSS  Dec.          =    -1.07531 (    .00850)  -08:37:32.1110
C_GAUSS  Flux          =   313.36129 (   1.40441)  milliJy
C_GAUSS  F.W.H.P.      =     0.10189 (    .12519)
Imaging and deconvolving the residuals...
```



Same flux. Size is compatible with 0''. Residual did not improve. ⇒ It's an unresolved source!

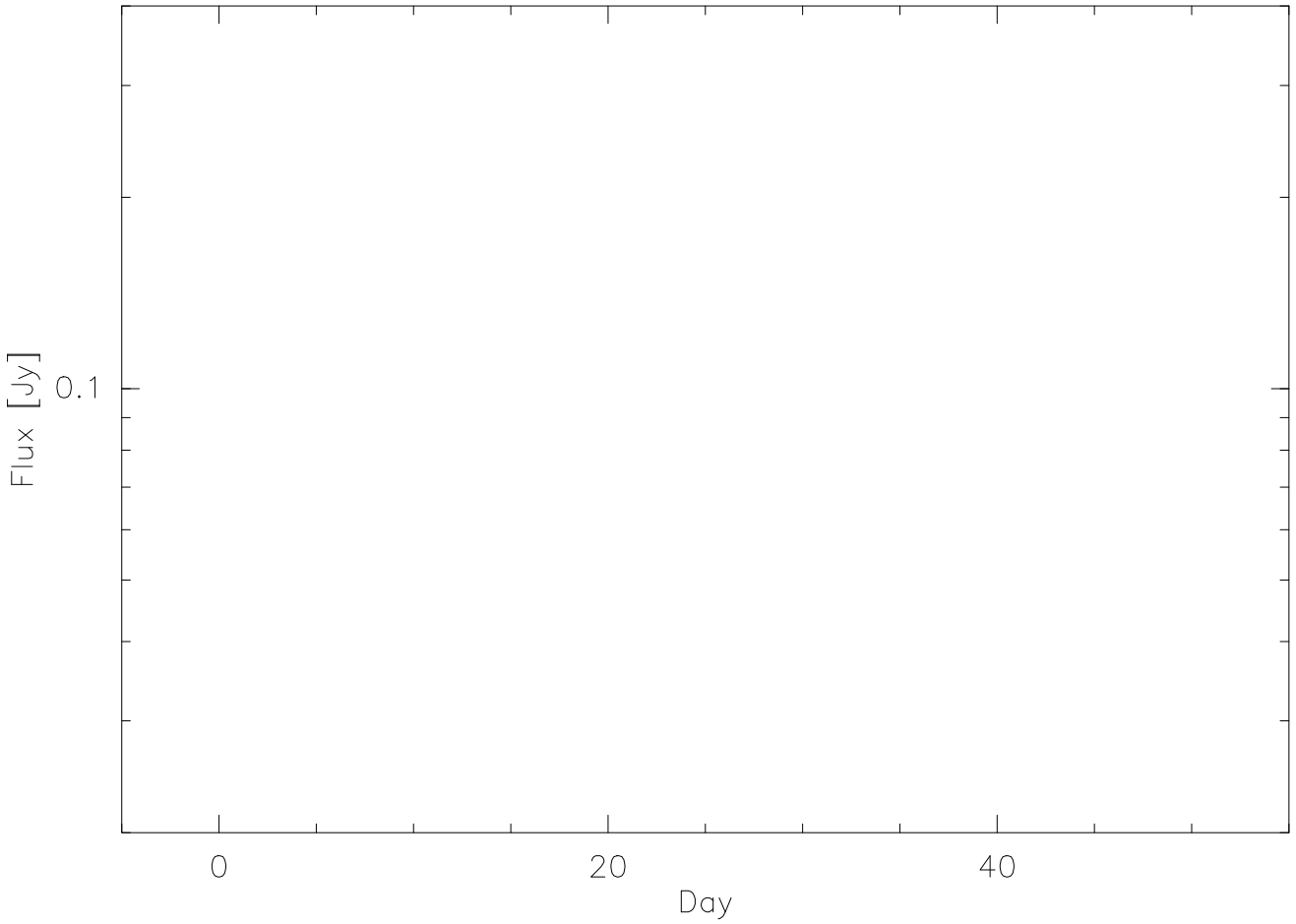
Drawing light curves: I. Manually writing a file with the flux results

```
MAPPING> type 3mm-uo.dat
!  Day    Flux [Jy]      Error  [Jy]
     1    0.2610E+00    0.5210E-03
     2    0.3482E+00    0.2231E-02
    14    0.1052E+00    0.8058E-03
    16    0.1412E+00    0.5557E-03
    19    0.1340E+00    0.5258E-03
    41    0.6897E-01    0.4325E-03
```

A bit long ⇒ See the end of the tutorial to automate.

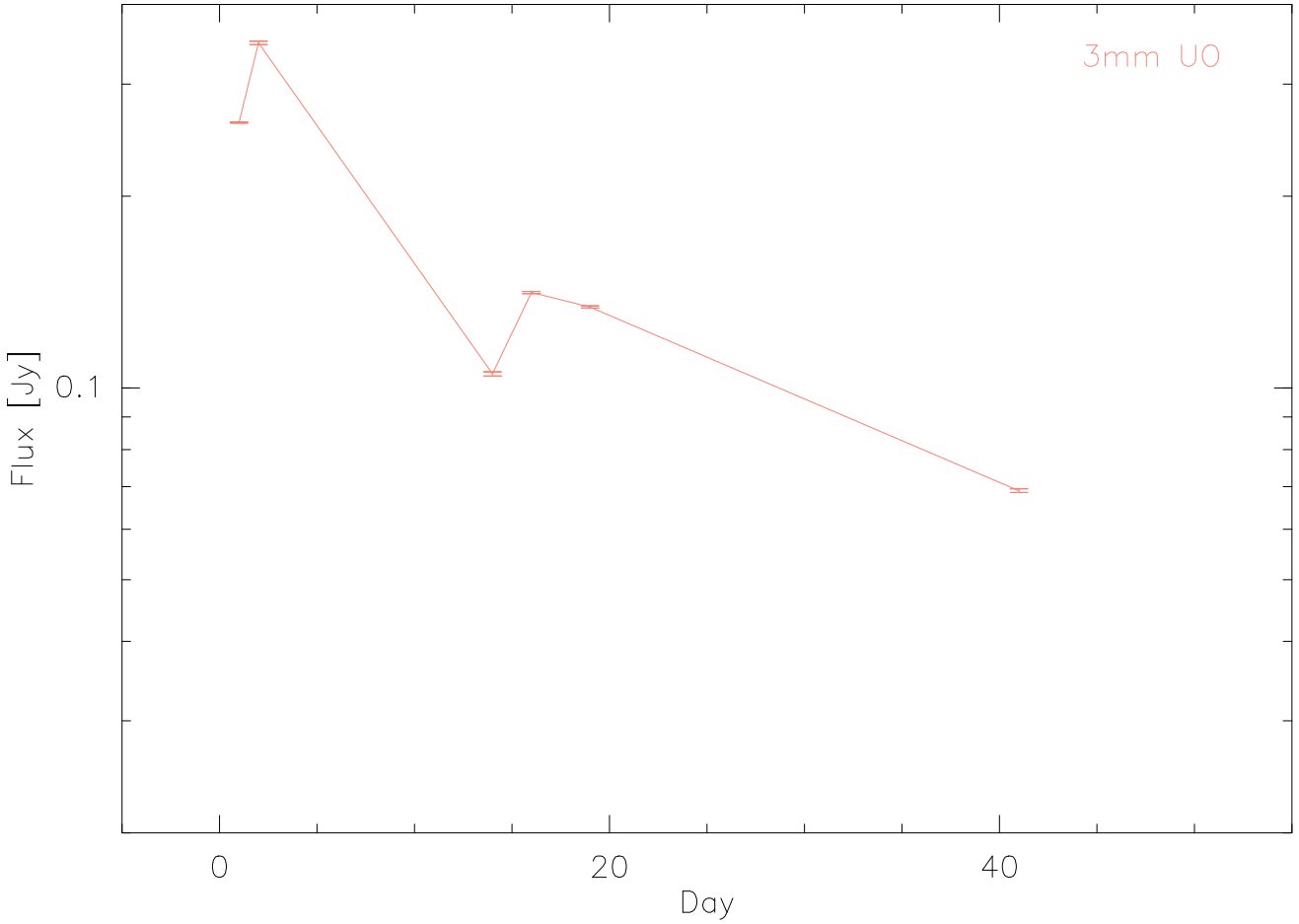
Drawing light curves: II. Starting the plot

```
MAPPING> clear                          ! Clear the board
MAPPING> set /default                   ! Reset the defaults
MAPPING> set expand 1.2                 ! Multiply default character size by 1.2
MAPPING> limits -5 55 0.02 0.4 /ylog ! Define plot limits (y axis will display the logarithm)
MAPPING> box                            ! Draw the box
MAPPING> label "Day" /x
MAPPING> label "Flux [Jy]" /y 3.5
```



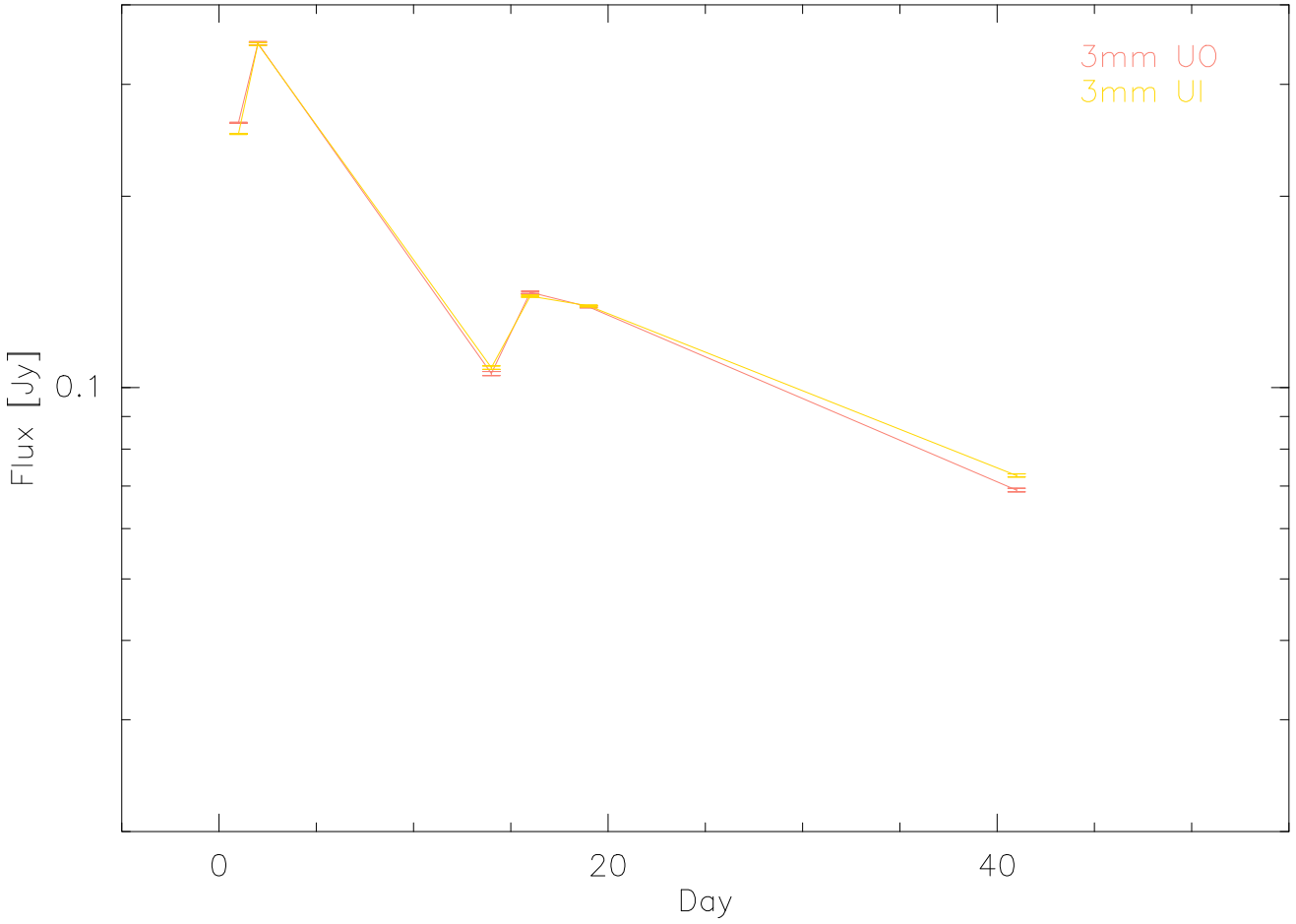
Drawing light curves: II. Drawing one light curve

```
MAPPING> pen /col salmon                          ! Select a pen of color salmon
MAPPING> column x 1 y 2 z 3 /file 3mm-uo.dat      ! Read columns #1 in x, #2 in y, and #3 in z variable
MAPPING> connect                                  ! Connect (x,y) points
MAPPING> error y                                  ! Draw z error bars along the y axis around each (x,y) poin
MAPPING> draw text -6 -1 "3mm UO" 3 /char 9       ! Caption in the top right corner
```



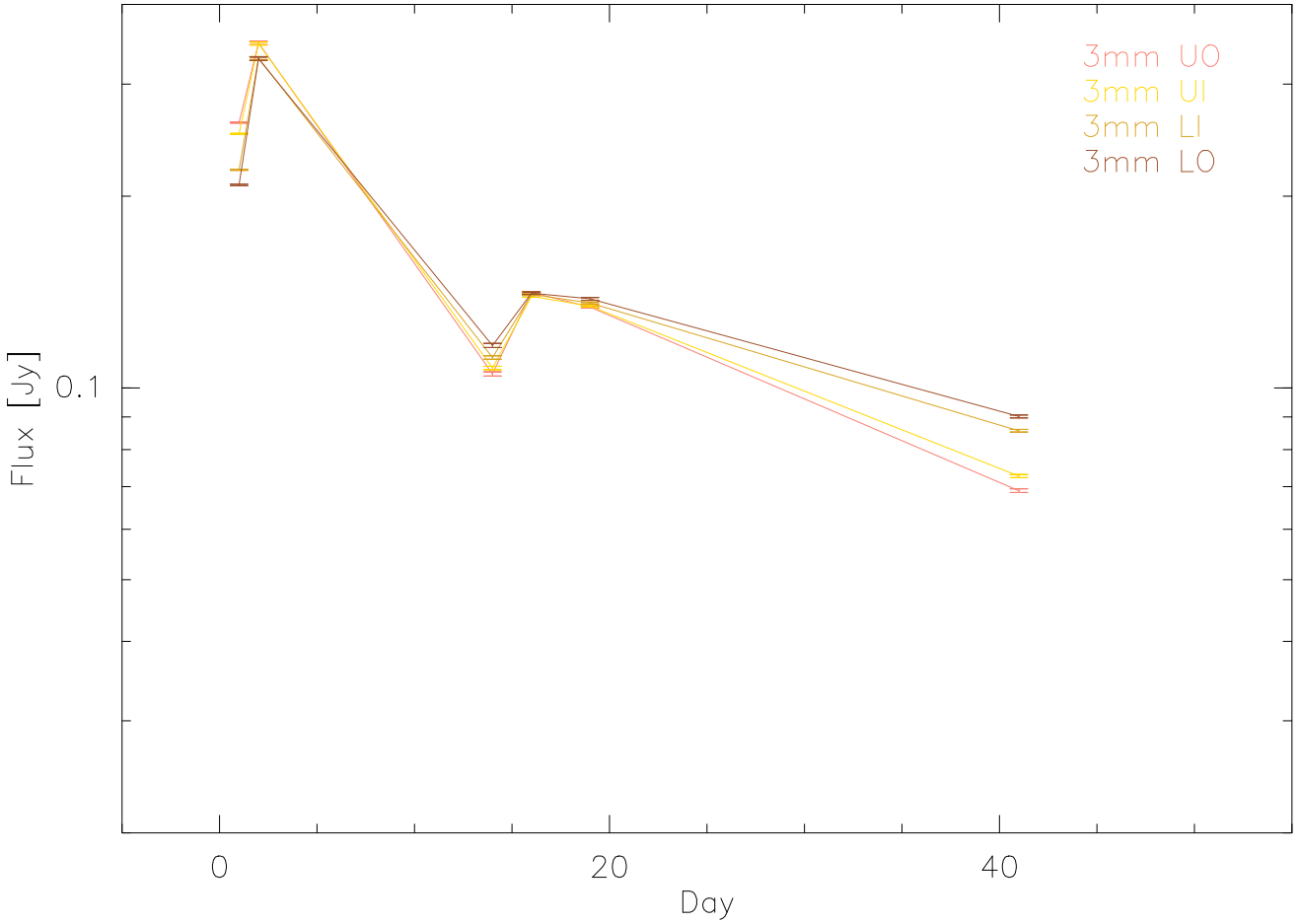
Drawing light curves: II. Drawing a second light curve

```
MAPPING> pen /col gold
MAPPING> column x 1 y 2 z 3 /file 3mm-ui.dat
MAPPING> error y
MAPPING> connect
MAPPING> draw text -6 -2 "3mm UI" 3 /char 9
```



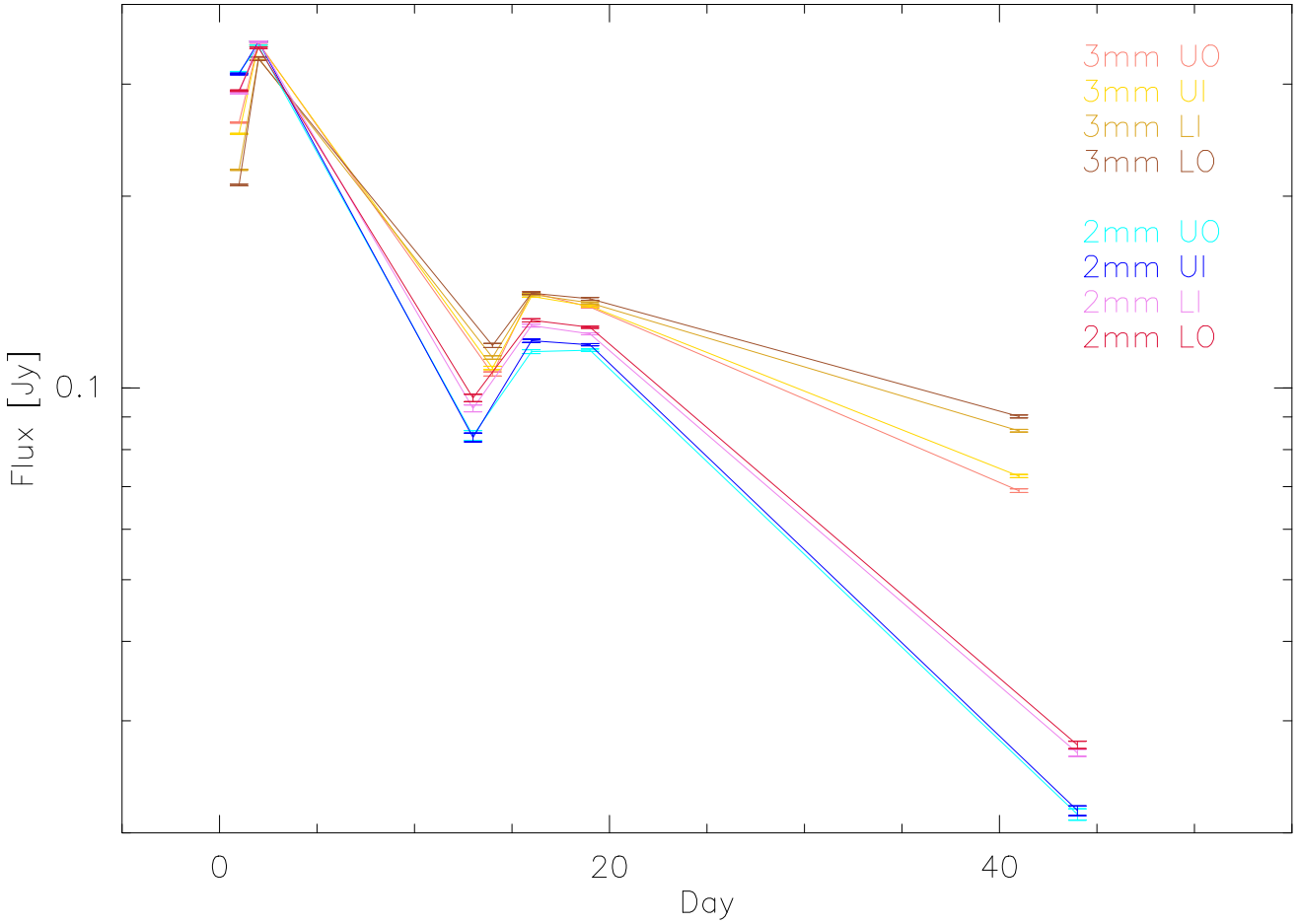
Drawing light curves: III. Iterating at 3 mm

```
MAPPING> pen /col gold
MAPPING> column x 1 y 2 z 3 /file 3mm-XX.dat ! Replace XX by LI or LO
MAPPING> error y
MAPPING> connect
MAPPING> draw text -6 -Y "3mm XX" 3 /char 9 ! Replace Y by an integer and XX by LI or LO
```



Drawing light curves: IV. Iterating at 2 mm

```
MAPPING> pen /col gold
MAPPING> column x 1 y 2 z 3 /file 2mm-XX.dat ! Replace XX by UO, UI, LI or LO
MAPPING> error y
MAPPING> connect
MAPPING> draw text -6 -Y "2mm XX" 3 /char 9 ! Replace Y by an integer and XX by UO, UI, LI or LO
```



Automatically writing a file with the flux results: I. Reading the results in variables

From this slide on, please look into file pro/grb04768249-uvfit.map

```
begin procedure my-uvfit-oneday-onebb
  !-------------------------------------------------------------------------
  ! &1: Wavelength
  ! &2: Baseband
  ! &3: day
  !-------------------------------------------------------------------------
  let name school-&1-&3-&2
  let uvfit%subsf01 yes
  let uvfit%funct01 "point"
  go uvfit
  !
  define double flux error                  ! Define intermediate variables as real*8
  define table uvfit 'name'".uvfit" read    ! Map the UVFIT variable to the binary table where the results
  let flux  uvfit[1,12]                     ! Get the flux into the FLUX variable
  let error uvfit[1,13]                     ! Get the error into the ERROR variable
  delete /var uvfit                         ! Delete the UVFIT variable and thus close the associated file
  say &3 flux error /format i5 e14.4 e14.4 ! Write the result on screen in a formatted way
end procedure my-uvfit-oneday-onebb
```

Automatically writing a file with the flux results: II. Iterating on the days and writing the results in an ASCII file

```
begin procedure my-uvfit-2mmbb
  !-------------------------------------------------------------------------
  ! &1: Baseband
  !-------------------------------------------------------------------------
  sic output 2mm-&1.dat new                    ! Open an ASCII file names 2mm-&1.dat
    say "!  Day   Flux [Jy]     Error  [Jy]"  ! The output of any SAY commands will be printed in the
                                               ! file. Here it's a comment line
    @ my-uvfit-oneday-onebb 2mm &1 01          ! Straightforward iteration
    @ my-uvfit-oneday-onebb 2mm &1 02          ! Straightforward iteration
    @ my-uvfit-oneday-onebb 2mm &1 13          ! Straightforward iteration
    @ my-uvfit-oneday-onebb 2mm &1 16          ! Straightforward iteration
    @ my-uvfit-oneday-onebb 2mm &1 19          ! Straightforward iteration
    @ my-uvfit-oneday-onebb 2mm &1 44          ! Straightforward iteration
  sic output                                   ! Close the output file
end procedure my-uvfit-2mmbb
```

Automatically writing a file with the flux results: III. Iterating on the basebands

```
@ my-uvfit-2mmbb lo
@ my-uvfit-2mmbb li
@ my-uvfit-2mmbb ui
@ my-uvfit-2mmbb uo
```