ALMA Winter school
CASA hands-on session

Czech ARC node, 28/29 February 2012

Prerequisites

13:30 hands-on part 1
15:00 coffee
15:30 hands-on part 2

Prerequisites
1) at least 60 GB empty disk space
2) Installation of CASA 3.3
3) files
M100line-orig.mask.tgz
m100-tutorial-msdata.tgz
reduce-m100-with-problems-light.py
from
http://almascience.eso.org/arcdistribution/casa-tutorials/7a76f411abc50757f2daa53399d20fe0

Prerequisites

The data

47 pointings mosaic in ALMA Band 3, CO(1-0), of NGC4321 (M100),
the brightest Spiral Galaxy in the Virgo Cluster located at a distance of 14-20 Mpc.
One arcsec corresponds to ~70-100 pc.

Two MSs:

X220-line.ms, X54-line.ms

observed on 10 Sept 2011 with 13 antennas

Already applied calibrations/corrections:

Tsys, WVR, Antenna Positions, Delay Errors

Calibrator source observations contained in the dataset:

Flux calibrator:	Titan
Bandpass calibrator:	3C 273
Phase calibrators:	QSO 1224+213 (primary) 3C 273 (secondary)
Pointing calibrator:	3C 273

Hands-on session - overview

Your task:

Download

reduce-m100-with-problems-light.py

(see link “script” in the program point for this session on the school home page)

Script contains a complete analysis of the M100 data **in 21 steps**
up to imaging the CO(1-0) line emission and some image analysis.

We introduced 9 **PROBLEMS** in the script.

Try to solve them.

Hands-on session - overview

Analysis steps in reduce-m100-with-problems-light.py:

0: 'Flagging',

1: 'Rebin to a reduced resolution of approx. 10 km/s',

2: 'Fast phase-only gaincal for bandpass',

3: 'Bandpass',

4: 'Setjy',

5: 'Fast phase-only gaincal',

6: 'Slow phase-only gaincal',

7: 'Slow amp and phase gaincal',

8: 'Fluxscale',

9: 'Applycal',

→ after ca. 50 min net processing time: **X220-line-vs.ms, X54-line-vs.ms**

Hands-on session - overview

Analysis steps in reduce-m100-with-problems-light.py (ctnd.):

10: 'Test image of the secondary phase cal',

11: 'Test image of the primary phase cal',

12: 'Test image of Titan',

13: 'Split out corrected data and time average',

14: 'Concatenate data', → **M100all_lores.ms**

15: 'Adjust fluxscale',

16: 'Split out the corrected M100 data',

17: 'Continuum image of M100',

18: 'Determine and subtract continuum',

19: 'Test image of central field',

20: 'Clean CO(1-0) line cube mosaic',

21: 'Make moment maps'

The infrastructure

Actual analysis begins after

```
# Begin analysis
```

The Python variable

```
mysteps
```

will control which steps are executed when you start the script using

```
execfile('reduce-m100-with-problems-light.py')
```

e.g. typing

```
mysteps = [2,3,4]
execfile('reduce-m100-with-problems-light.py')
```

will execute only steps 2, 3, and 4

Setting mysteps = [ ] (empty list) will make it execute all steps.

The `timing()` function tells you about the execution times of the steps (good diagnostic).

Hands-on session - the problems

The problems

Look for the string ‘PROBLEM”in the script. List is given at the beginning:

```
# List of problems in the individual analysis steps of this script:
# Step  0: Write a flagdata2 command to flag the channels
#              239, 447/448, 720/721 and 2847/2848 in all SPWs
# Step  5: Determine the solint parameter in gaincal
# Step  7: Determine the gaintable parameter in gaincal
# Step  9: Complete the last applycal command
# Step 13: Determine the missing parameters in split
# Step 17: Determine the mode, imagermode, and mask parameters in clean
# Step 18: Determine the fitspw parameter in uvcontsub
# Step 19: The mask test-M100line-orig.mask is missing. Generate it.
# Step 21: Determine the axis and includepix parameters in immoments
```

Hands-on session - the problems step by step

The problem in step 0

```
mystep = 0
if(mystep in thesteps):
  print 'Step ', mystep, step_title[mystep]

  for name in basename:

    flagmanager(vis=name+'-line.ms', mode='restore',
      versionname='apriori')
    # Edge channels
    flagdata2(vis=name+'-line.ms', selectdata=T,
              field='',  manualflag=T,
              mf_spw='0~3:0~10;3800~3839',
              flagbackup = F)
    # Channels 239, 447/448, 720/721 and 2847/2848 are off in all SPWs
    # PROBLEM: write a flagdata2 command to flag these channels
```

Hands-on session - the problems step by step

The problem in step 5: Determine the solint parameter in gaincal

```
# Fast phase-only gaincal
mystep = 5
if(mystep in thesteps):
  print 'Step ', mystep, step_title[mystep]

  for name in basename:
    os.system('rm -rf cal-'+name+'-int.Gp')
    gaincal(vis=name+'-line-vs.ms',
            caltable='cal-'+name+'-int.Gp',
            spw='*:25~455',
            field='*Phase*,*Band*,Titan',
            gaintable='cal-'+name+'.B1',
            selectdata=F, solint='PROBLEM',
            refant=therefant, calmode='p')
```

Hands-on session - the problems step by step

The problem in step 7: Determine the gaintable parameter in gaincal

```
mystep = 7
if(mystep in thesteps):
  print 'Step ', mystep, step_title[mystep]

  for name in basename:
    os.system('rm -rf cal-'+name+'-scan.Gap')
    gaincal(vis=name+'-line-vs.ms',
            caltable='cal-'+name+'-scan.Gap',
            spw='*:25~455',
            field='*Phase*,*Band*,Titan',
            gaintable=['PROBLEM'],
            selectdata=F, solint='inf',
            refant=therefant, calmode='ap')
```

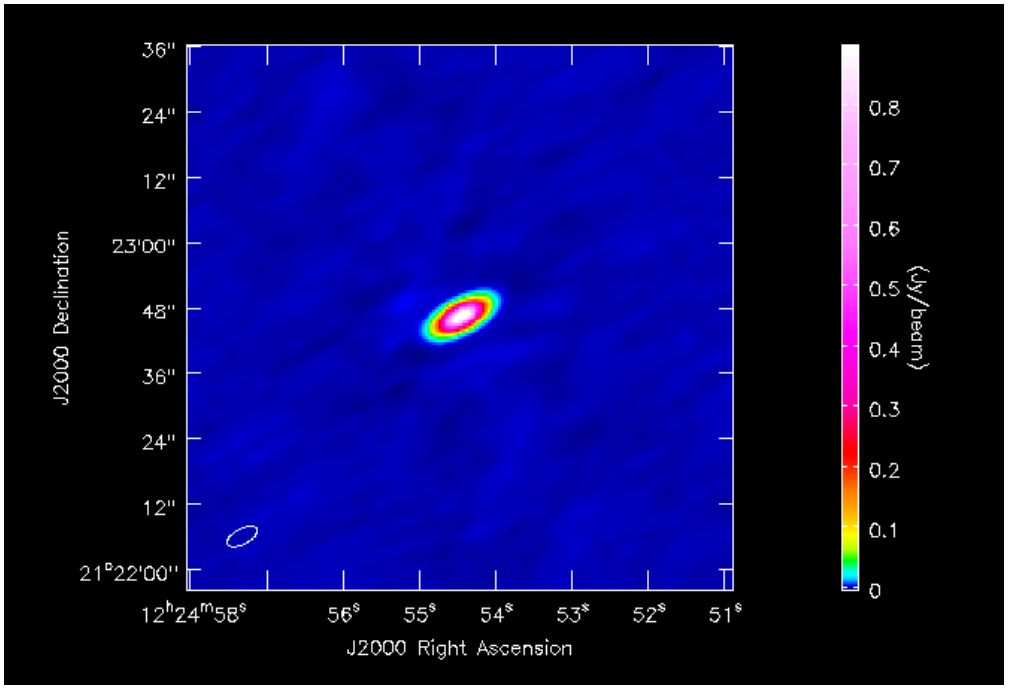
Hands-on session - the problems step by step

The problem in step 9: Complete the last applycal command

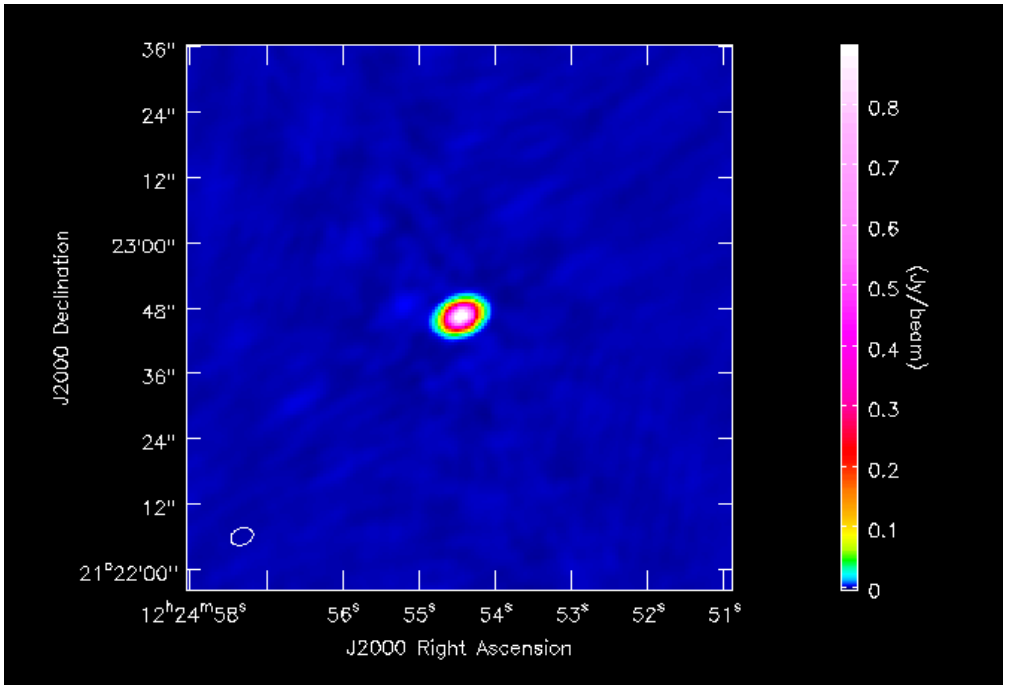
```
mystep = 9
  ...
  # to Titan
  applycal(vis=name+'-line-vs.ms',field='Titan',
           gaintable=['cal-'+name+'.B1',
                      'cal-'+name+'-int.Gp','cal-'+name+'.flux'],
           interp=['nearest','nearest','nearest'],
           gainfield=['*Band*','*Band*','*Band*'],
           calwt=F,
           flagbackup=T)

  # to M100
  applycal(PROBLEM)
```

Hands-on session - sanity check

The test images of the phase calibrators and the flux calibrator from steps 10, 11, and 12:



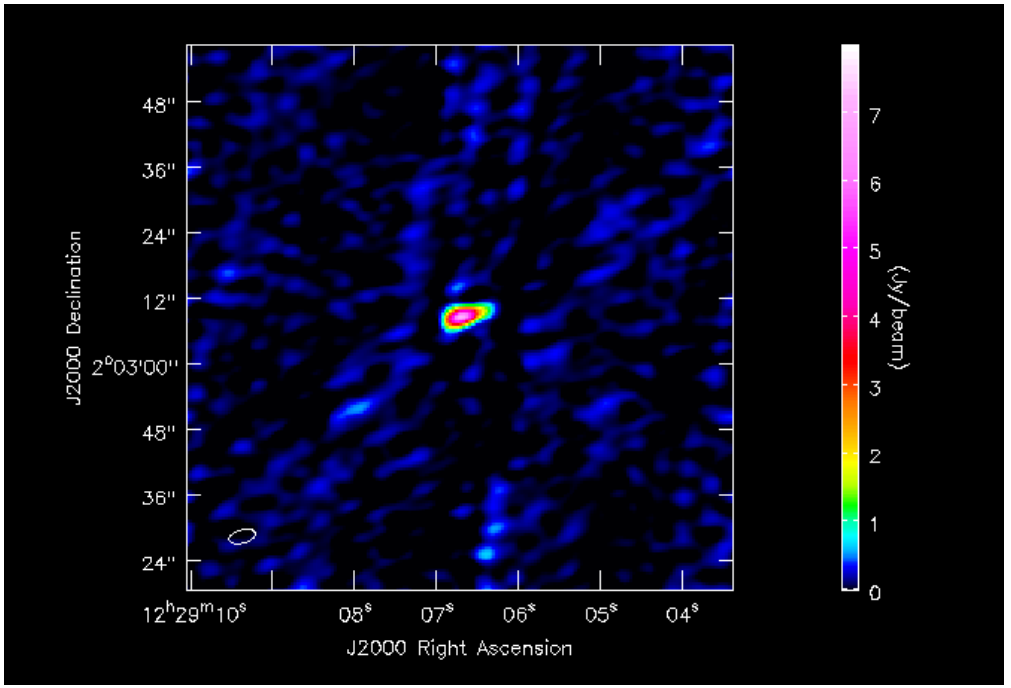
test-X220-prim_phasecal.png



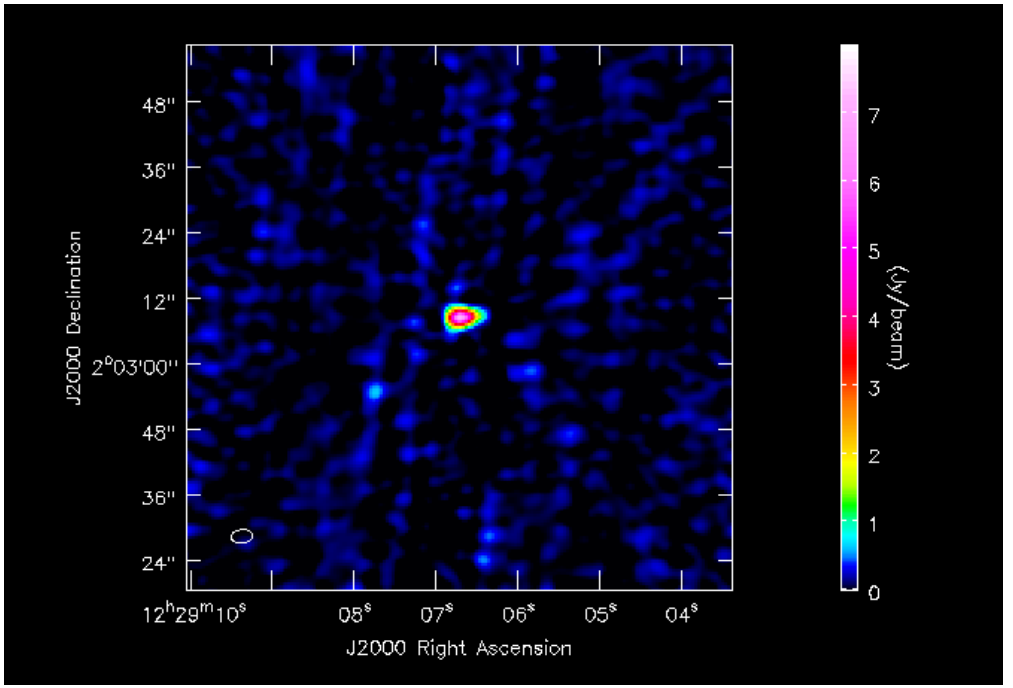
test-X54-prim_phasecal.png

Hands-on session - sanity check

The test images of the phase calibrators and the flux calibrator from steps 10, 11, and 12:



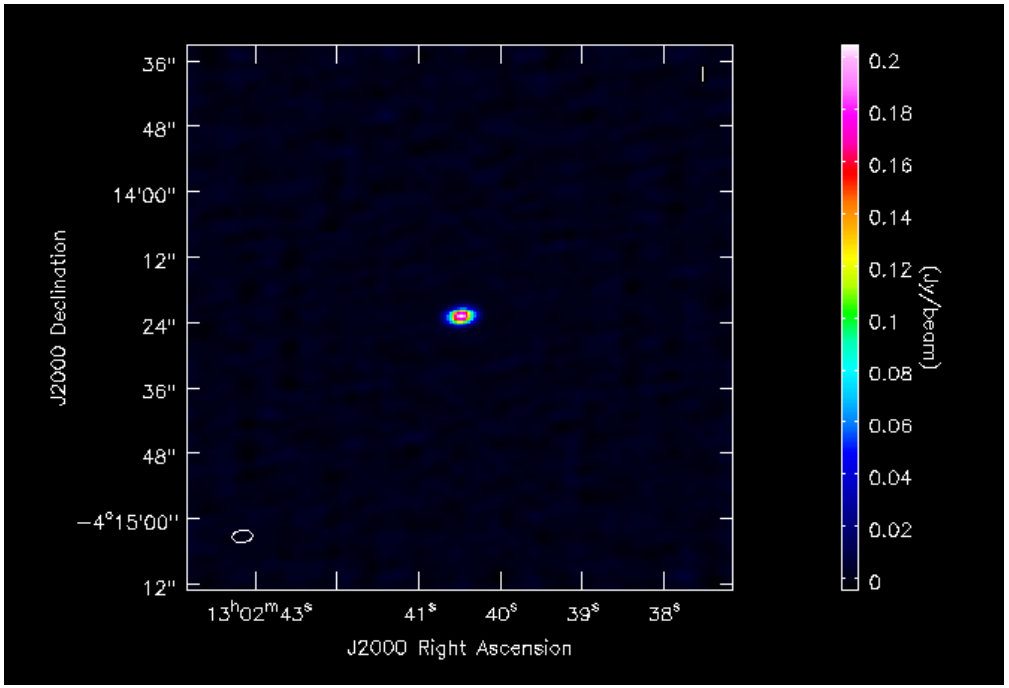
test-X220-sec_phasecal.png



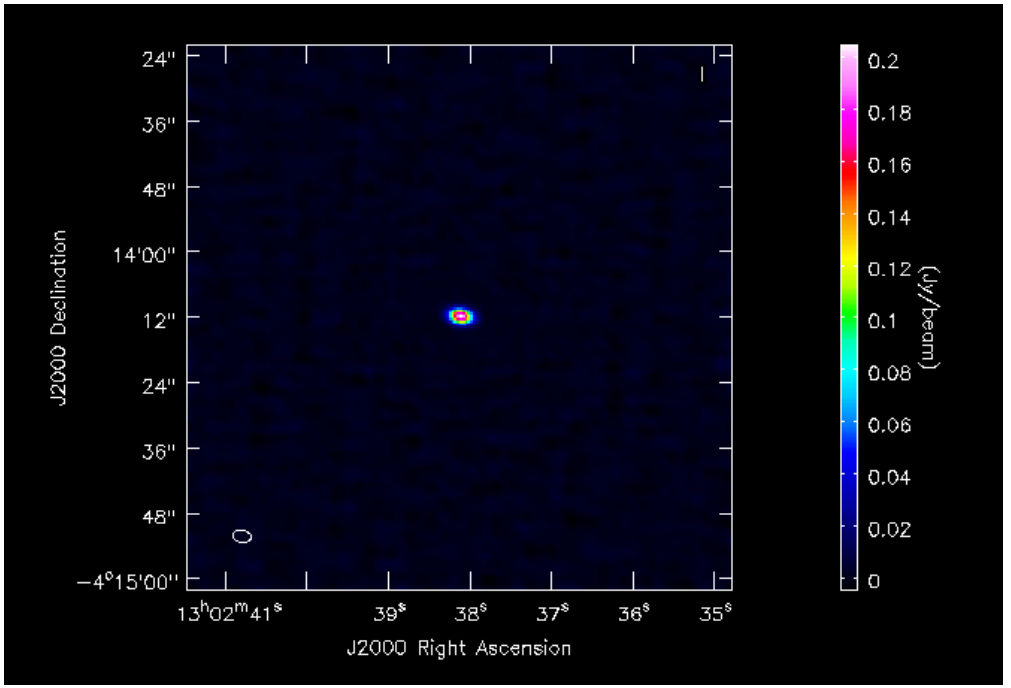
test-X54-sec_phasecal.png

Hands-on session - sanity check

The test images of the phase calibrators and the flux calibrator from steps 10, 11, and 12:



test-X220-Titan.image



test-X54-Titan.image

Hands-on session - the problems step by step

The problem in step 13: Determine the missing parameters in split

```
# Split out corrected data and time average at the same time
#  into 1 minute time bins (makes following steps factor 3 faster!)
mystep =  13
if(mystep in thesteps):
    print 'Step ', mystep, step_title[mystep]

    for name in basename:
        os.system('rm -rf '+name+'-corrected.ms*')
        split(vis=name+'-line-vs.ms', outputvis=name+'-corrected.ms'
              ) # PROBLEM (find the missing parameter)
```

Hands-on session - the problems step by step

The problem in step 17: Determine the mode, imagermode, and mask parameters in clean

```
# Continuum image
mystep = 17
if(mystep in thesteps):
    print 'Step ', mystep, step_title[mystep]
    os.system('rm -rf M100cont.*')
    clean(vis = 'M100all_lores.ms',
          imagename = 'M100cont',
          field='2~47',
          spw='0:10~210;256~440,1~3:10~460', # exclude CO(1-0) line
          mode = 'PROBLEM',
          niter = 1000,
          mask=[0,0,0,0],# PROBLEM
          imagermode = 'PROBLEM',
          interactive = F, # switch to interact. clean to determine mask
          imsize = 200,
          cell = '0.5arcsec',
          phasecenter='J2000 12h22m54.9 +15d49m15')
```

Hands-on session - the problems step by step

The problem in step 18: Determine the fitspw parameter in uvcontsub

```
mystep = 18
if(mystep in thesteps):
   print 'Step ', mystep, step_title[mystep]

   os.system('rm -rf M100all_lores.ms.c*')
   uvcontsub(vis='M100all_lores.ms',field='',
             fitspw='PROBLEM', # use plotms to determine fitspw
             combine='',solint='inf',fitorder=1,spw='0',want_cont=False)
```

Hands-on session - the problems step by step

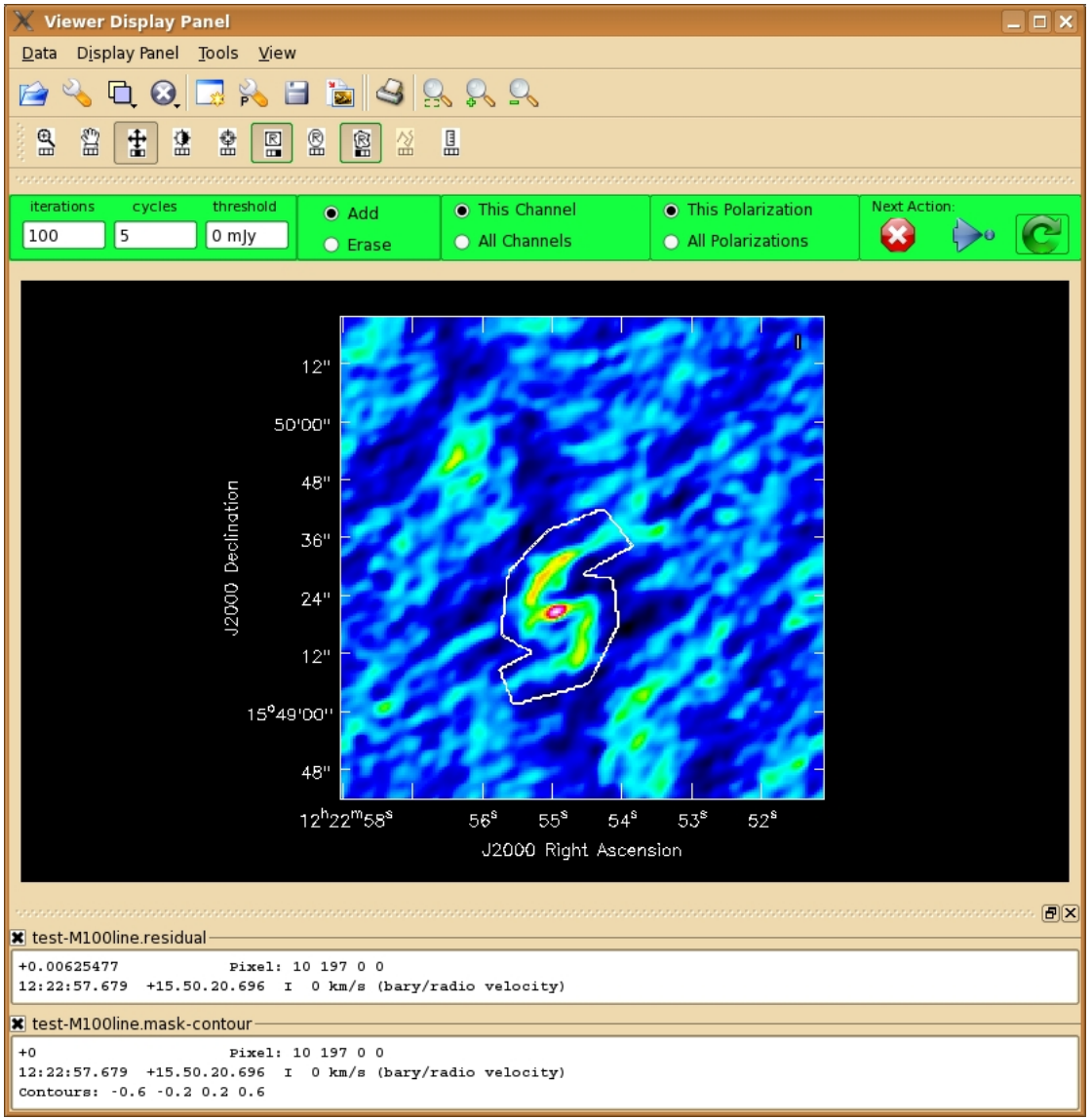
The problem in step 19: The mask test-M100line-orig.mask is missing.
Generate it using interactive clean

```
mystep = 19
if(mystep in thesteps):
    print 'Step ', mystep, step_title[mystep]
    os.system('rm -rf test-M100line.*')
    clean(vis='M100all_lores.ms.contsub', imagename='test-M100line',
          field='26',spw='0:231~248',
          mode='mfs',
          niter=500,gain=0.1,threshold='0.0mJy',
          imagermode='csclean',
          mask='test-M100line-orig.mask', # PROBLEM: mask missing
          interactive=False, # switch to interactive to determine it
          outframe='BARY',veltype='radio',
          imsize=200,cell='0.5arcsec', phasecenter='',
          stokes='I',weighting='briggs',robust=0.5,
          calready=False,
          npercycle=100,cyclefactor=1.5,cyclespeedup=-1)
```

Careful: don't press Ctrl-C during clean!

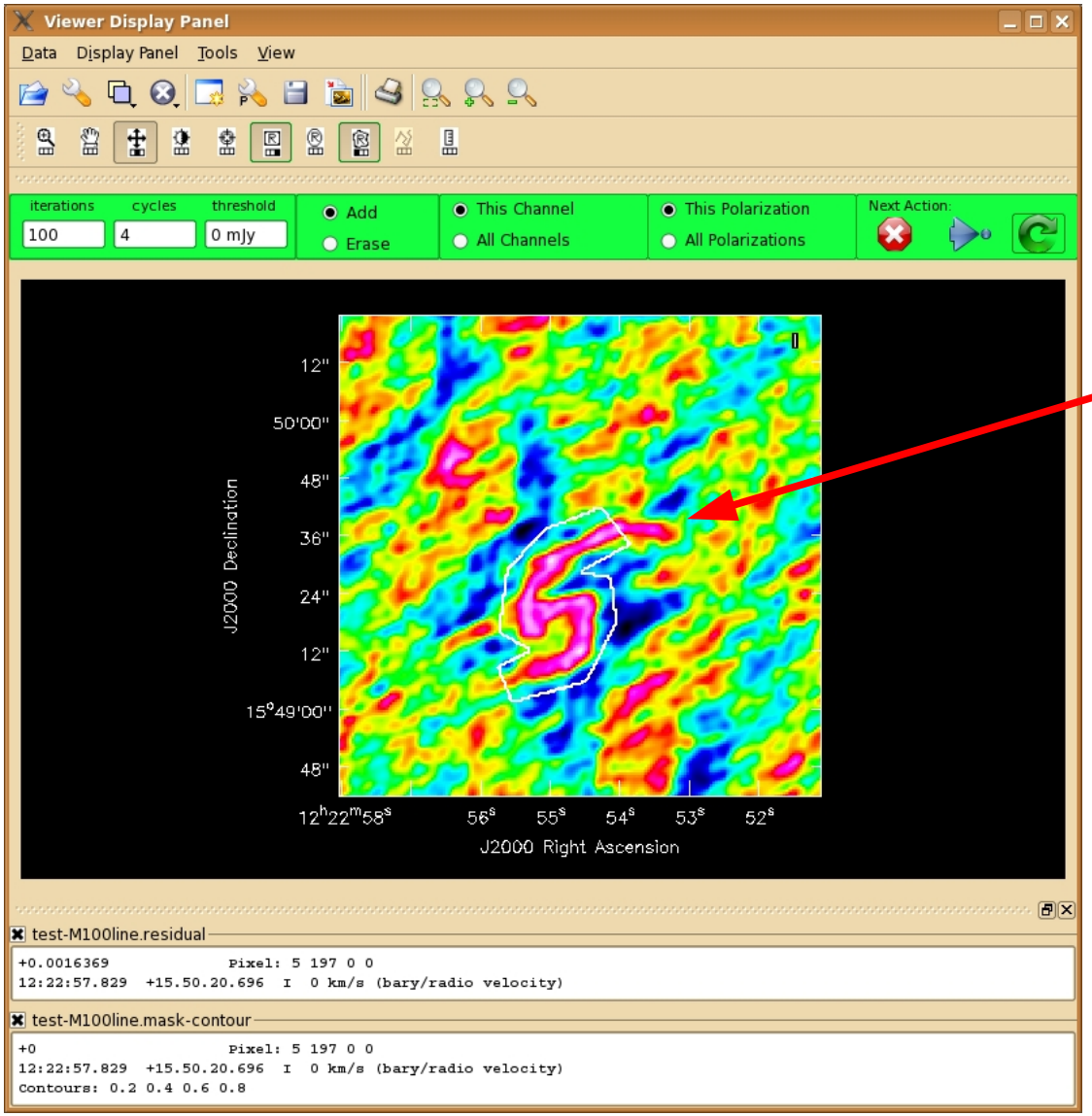
Hands-on session - the problems step by step

The problem in step 19: Generate a mask using interactive clean



Hands-on session - the problems step by step

The problem in step 19: Generate a mask using interactive clean

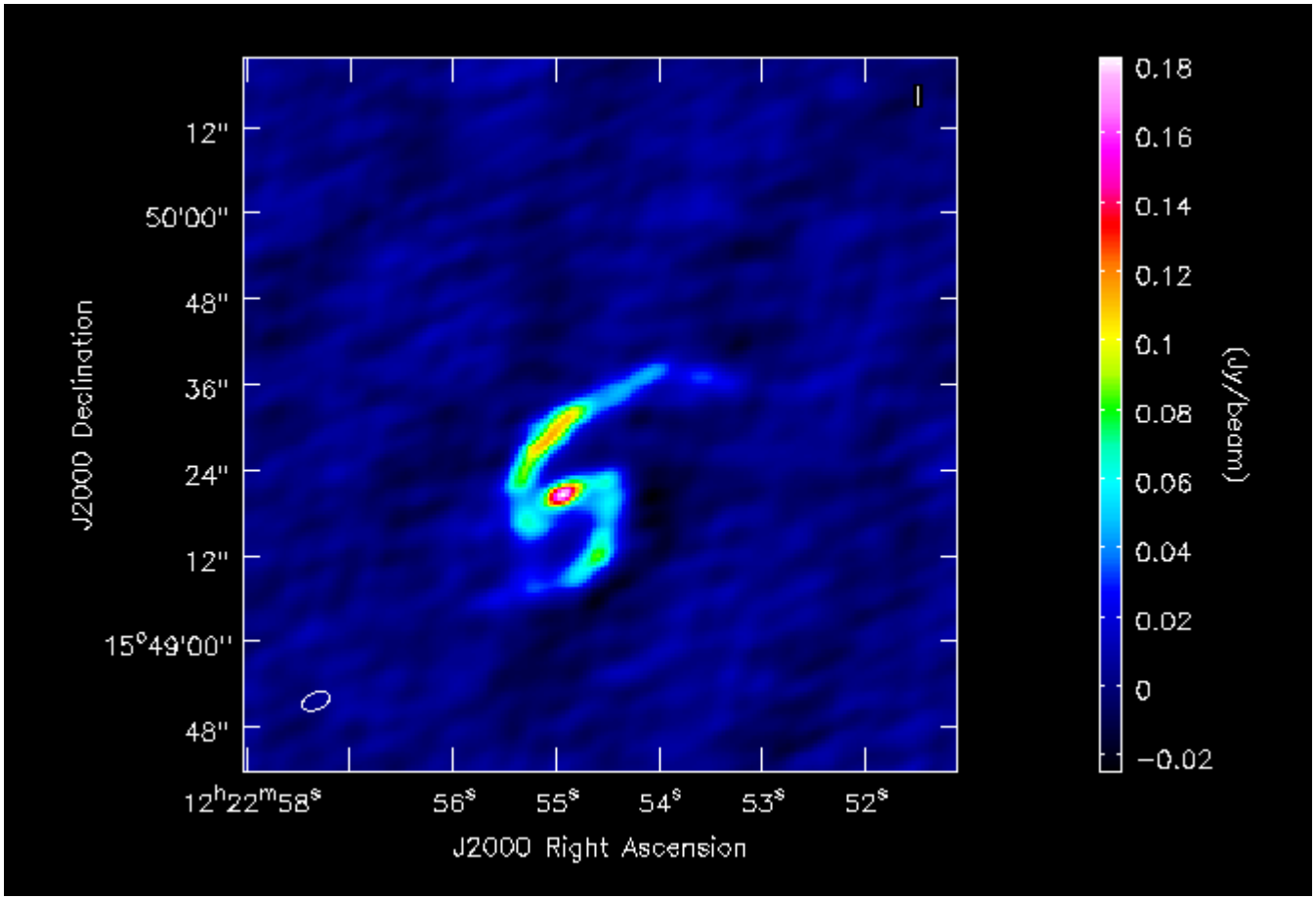


Need to further adjust mask after a few iterations ...

Choose “hot metal” colour scheme in noisy images to better see regions.

Move final mask to a name not starting with same name as image!

The problem in step 19: Generate a mask using interactive clean



Final test image of the central field.

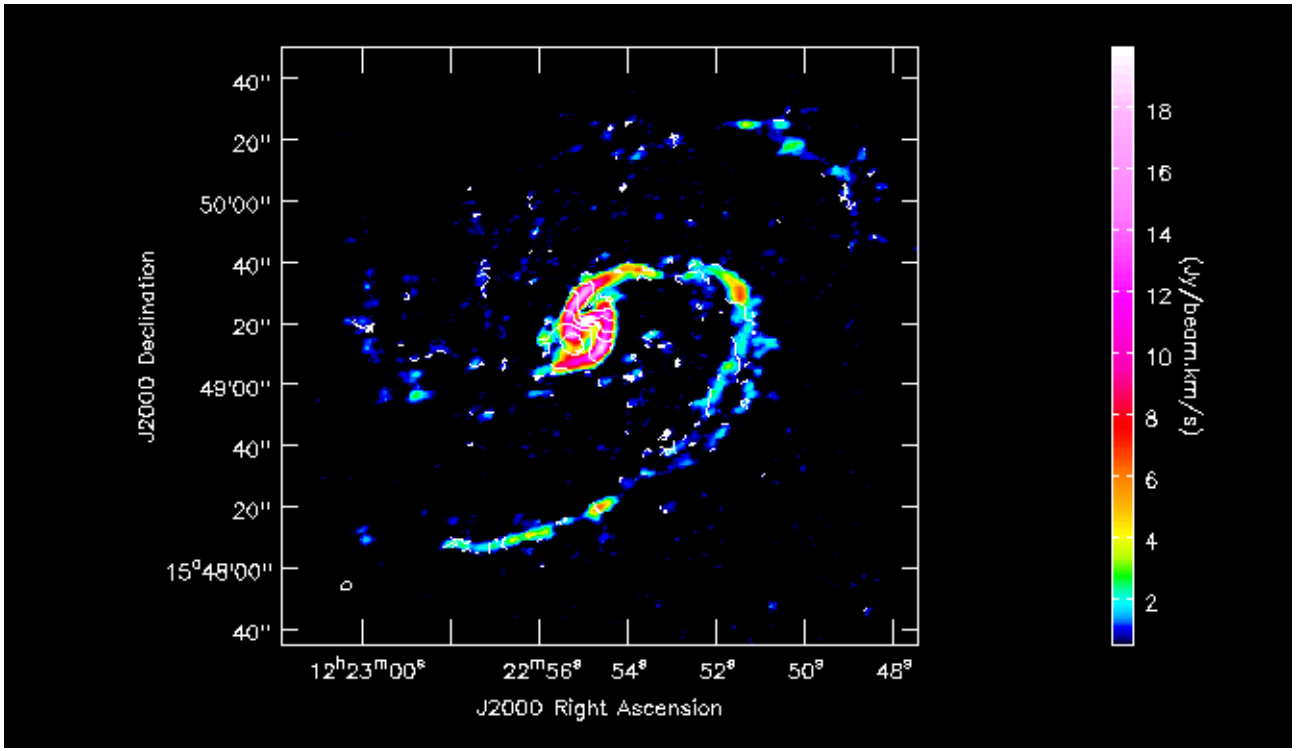
Hands-on session - the problems step by step

The problem in step 21: Determine the axis parameter in immoments

```
mystep = 21
if(mystep in thesteps):
    print 'Step ', mystep, step_title[mystep]
    os.system('rm -rf M100-CO.mom?')
    immoments(imagename='M100line.image',
              moments=[0], # i.e. the integrated spectrum
              axis='PROBLEM',
              region='',box='100,110,515,500',
              chans='7~35',
              mask='',
              outfile='M100-CO.mom0',
              includepix=[0.03, 1000000])
    immoments(imagename='M100line.image',
              moments=[1], # i.e. the velocity field
              axis='PROBLEM',
              region='',box='100,110,515,500',
              chans='7~35',
              mask='',
              outfile='M100-CO.mom1',
              includepix=[0.035, 1000000])
```

Hands-on session - the problems step by step

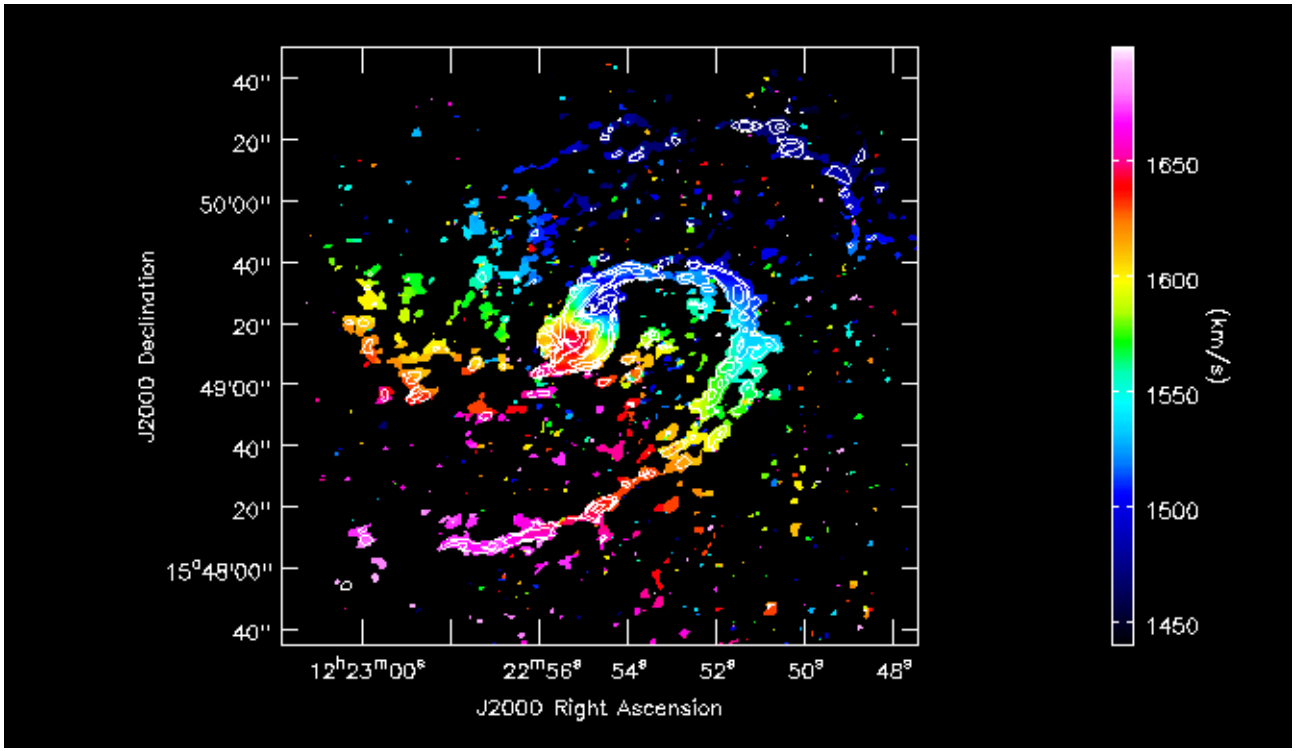
Step 21: immoments results



M100-CO_map.png (colour = mom0, contour = mom1)

Hands-on session - the problems step by step

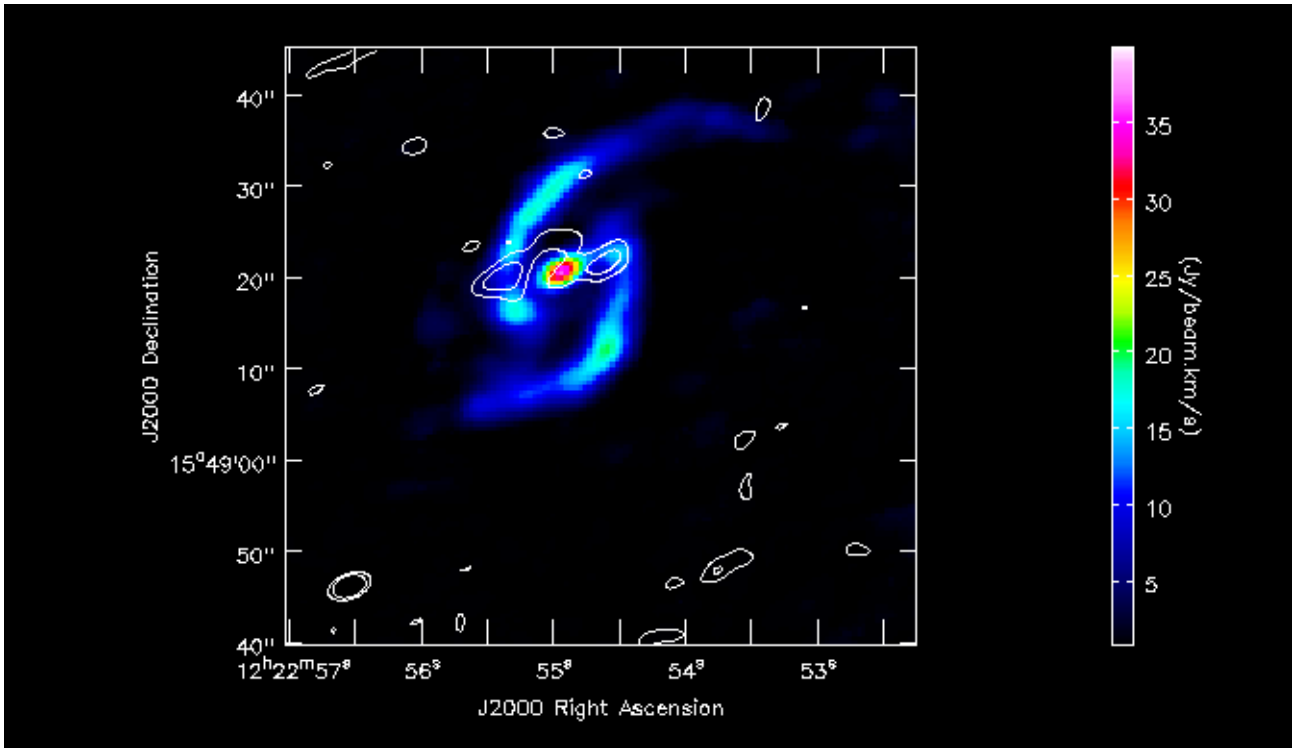
Step 21: immoments results



M100-CO_velfield.png (colour = mom1, contour = mom0)

Hands-on session - the problems step by step

Step 21: immoments results



M100-CO_contmap.png (colour = mom0, contour = cont)