This document has some of our answers for the extra questions posed in “homework_ext.pdf”. There are probably other, or better ways to arrive at some of the same solutions.

1. SDcal

Most of the hard work is done by SDcal ; which does the calibration, etc. The gridding up of the cube is done by sdimaging as for the script. The viewer is pretty dull - it’s only a star!. but the spectrum is a good way to confirm it’s all been done correctly:

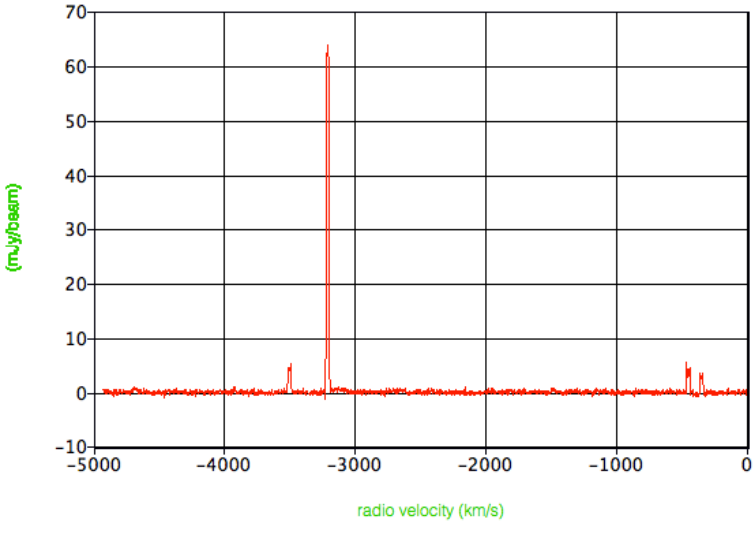
```
asap_init()
default(sdcal)
infile          =  "X44.PM02.asap.nowvr"
calmode         =  "ps"
iflist          =  [1]
average         =  False
maskmode        =  "auto"
blfunc          =  "poly"
order           =  1
outfile         =  "X44.PM02.ms.calbl"
outform         =  "MS2"
sdcal()

default(sdimaging)
infile          = 'X44.PM02.ms.calbl'
specunit        =  'channel'
spw             = 1
stokes          = 'XXYY'
gridfunction= 'SF
outfile         = 'imaging.cube'
overwrite       = True
imsize          = [8, 8]
cell            = ['0.5arcmin', '0.5arcmin']
dochannelmap = True
 nchan          = 3840
phasecenter= 'J2000 09:47:57.406 +13.16.43.561'
sdimaging()
```

After completion, we can view the product:

```
viewer(infile= 'imaging.cube')
```

then tools > spectral profile
clicking on the target button “positioning” will reveal the spectrum.

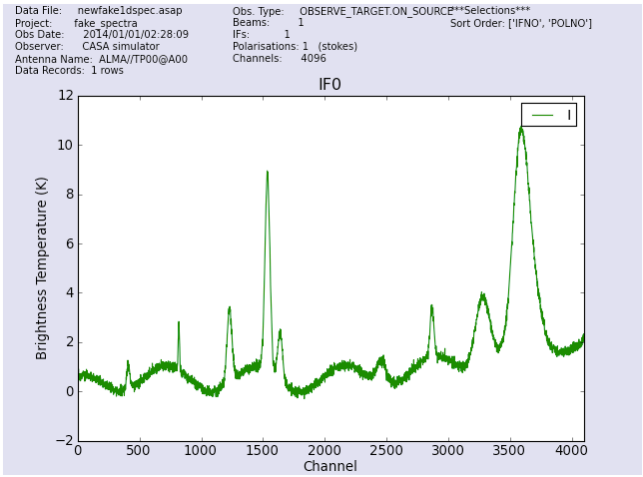


2. SDbaseline.
Look at ‘before’
asap_init()sdplot(infile='fake1dspec.asap')

Get stddev with SDSTAT - on uncorrected data:

sdstat(infile='fake1dspec.asap',outfile='stats.1',overwrite=True,masklist= [[0, 368], [460, 800], [840, 1160], [1300, 1460], [1700, 2380], [2520, 2822], [2931, 3077], [4000, 4096]])
yields:
stddev= 0.47

sdplot(infile='fake1dspec.bl2.asap')
produces the image to the right:



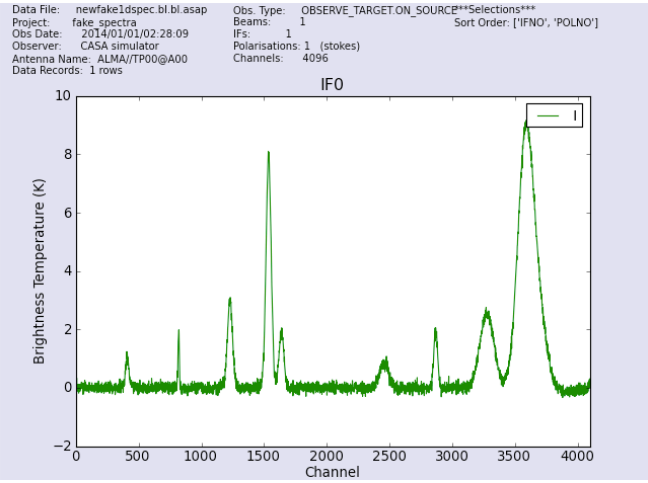
Our best attempt for baselining is via either a one or two-step process.
As two-steps: First to remove a low-order polynomial:

```
asap_init()
default(sdbaseline)
infile      = "fake1dspec.asap"
masklist    = [[0, 368], [460, 800], [840, 1160], [1300, 1460], [1700, 2380], [2520, 2822], [2931, 3077], [4000, 4096]]
maskmode    = "list"
blfunc      = "poly"
order       = 5
outfile     = "fake1dspec.bl.asap"
sdbaseline()
```

Then remove a sinusoid (e.g. a ripple):

```
infile      = "fake1dspec.bl.asap"
masklist    = [[0, 368], [460, 800], [840, 1160], [1300, 1460], [1700, 2380], [2520, 2822], [2931, 3077], [4000, 4096]]
maskmode    = "list"
blfunc      = "sinusoid"
fftmethod   = "fft"
fftthresh   = 3.0
clipthresh  = 3.0
outfile     = "fake1dspec.bl2.asap"
sdbaseline()
```

SDSTAT - on baselined data with inputs similar to above yields:
stddev= 0.101. The *actual* stddev is 0.1

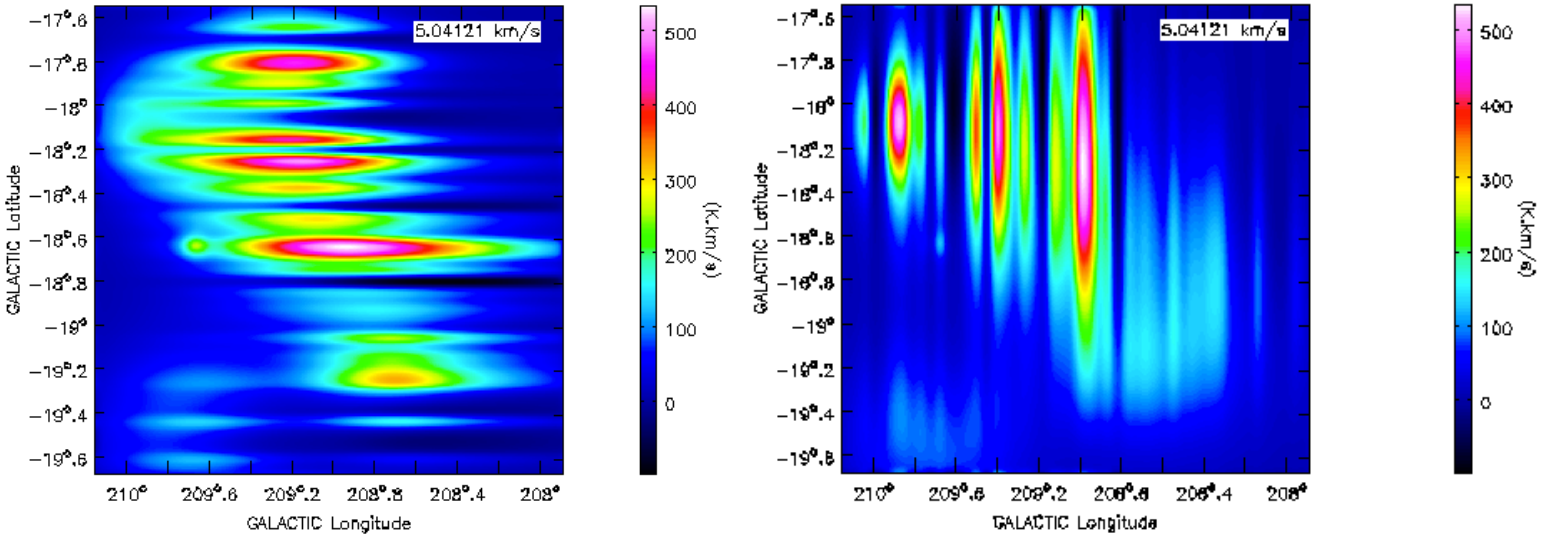


Baselines can also be removed by fitting a spline with a large number of pieces (roughly equivalent to the number of inflections in the baseline+1, the ripple appears to have about 12 “peaks” and “troughs” and an additional rising inflection). In this case, a piece number greater than 14 achieves a good rms for the residual.

```
infile      = "fake1dspec.bl.asap"
maskmode    =  'list'
blfunc      = 'cspline'
npiece      =      13
clipthresh  =     3.0
clipniter   =       0
```

SDSTAT with the above inputs yields: stddev= 0.119
SDSTAT with npiece = 15 yields stddev=0.104

3. SDimprocess
Examine the two scans first:
asap_init()
viewer(infile='vscan4d.image') and viewer(infile='vscan4d.image')
produces:



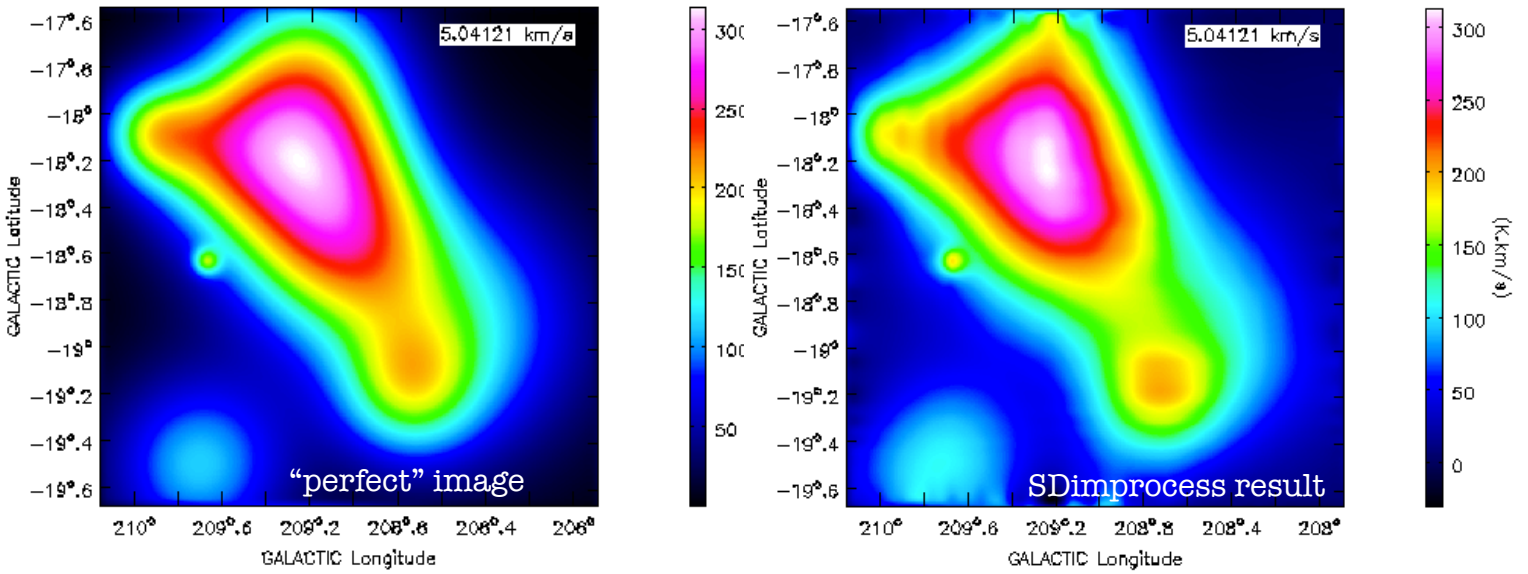
Using these contaminated datasets, we can produce a clean image with sdimprocess:

```
asap_init()
default(sdimprocess)
infiles     = ['vscan4d.image', 'hscan4d.image']
mode        = "basket"
numpoly     = 2
beamsize    = 0.0
smoothsize  = 2.0
direction   = [90.0, 0.0]
masklist    = 6.25
tmax        = 0.0
tmin        = 0.0
outfile     = "basket4d.image"
sdimprocess()
```

After completion, we can view the product:

viewer(infile='basket4d.image')

Which reveals the structure of the synthetic cloud



5. SDtpimaging

Don’t forget to copy the input into a new measurement set
shutil.copytree('moon_atf.ms','moon_atf.bl.ms')

```
default(sdtpimaging)
infile='moon_atf.ms'
calmode='none'
antenna='0'
createimage=True
outfile='moon_atf.ms.Ant0.im'
imsize=[200,200]
cell=['0.2arcmin','0.2arcmin']
phasecenter="AZEL 187d54m22s 41d03m0s"
ephemsrcname='Moon'
gridfunction='SF'
sdtpimaging()
```

```
default(sdtpimaging)
infile='moon_atf.bl.ms'
calmode='baseline'
backup=False
masklist=[50,50]
bpoly=1
antenna='0'
createimage=True
outfile='moon_atf.ms.Ant0.BL.im'
imsize=[200,200]
cell=['0.2arcmin','0.2arcmin']
phasecenter="AZEL 187d54m22s 41d03m0s"
ephemsrcname='Moon'
gridfunction='SF'
plotlevel=1
sdtpimaging()
```

Viewing the products with:
viewer(infile='moon_atf.ms.Ant0.im')
and
viewer(infile='moon_atf.ms.Ant0.BL.im')