

Simulating the Short Baseline Array of the ngVLA

Rachel Parziale with Dr. Brian Mason

UWYO Mentor: Dr. Daniel Dale

Abstract:

The Next Generation Very Large Array (ngVLA) is a centimeter-millimeter interferometer consisting of a main array with 214, 18 meter diameter antennas. The largest scale structures the ngVLA can observe are determined by its shortest baselines, which are limited by the diameter of the antennas. The main array by itself achieves insufficient resolution for all science projects planned for the ngVLA as 20% -- 30% of cases require shorter baselines [3]. In order to observe the largest scale structures, the ngVLA will also contain a compact, centrally-located array of antennas in a short baseline configuration referred to as the Short Baseline Array (SBA). The SBA will contain 19, 6 meter diameter antennas to provide supporting data to the ngVLA. Two different configurations for the SBA were investigated: one configuration with regularly spaced antennas and one configuration with randomly spaced antennas. The configurations were investigated by simulating observations of sky images and simulated disks to investigate the PSF deconvolution of the observations and the flux recovery at different scales. The results show that although it produces more prominent sidelobes, the regularly spaced SBA configuration recovers more flux density than the randomly spaced SBA configuration on scales larger than a diameter of ten arcseconds.

Background/Introduction:

The resolution a telescope can achieve is roughly equal to the wavelength being observed over the diameter of the antenna. When observing radio signals in the centimeter to millimeter wavelengths, large dishes are required to achieve high angular resolution, often larger than can be feasibly built. To achieve these resolutions while observing at longer wavelengths, an array of antennas called an interferometer are used instead.

Interferometers function by observing the same source with an array of antennas and correlating the source received by each antenna to simulate observing with a single, larger dish. Instead of the dish diameter of the antennas themselves determining the resolution, the baseline distance between the antennas is what determines the resolution. This is illustrated in Figure 1 below. A larger selection of baselines provides higher UV coverage in the observations.

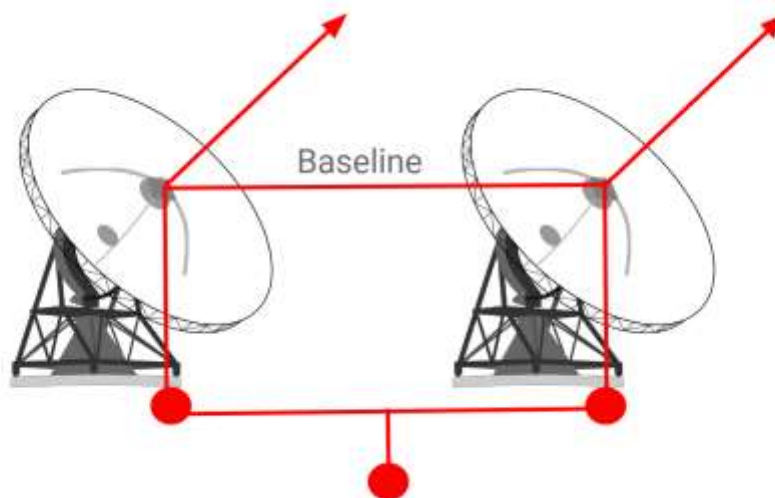


Figure 1: Two antennas of an interferometer pointing at the same source as indicated by the red arrows. Their signals are correlated to simulate observing with a single dish of a diameter equal to the baseline between the antennas.

The Next Generation Very Large Array (ngVLA) is a centimeter-millimeter interferometer planned to be used for imaging thermal emissions. The antennas will be located across New Mexico, Texas, and Mexico, and the main array will feature 214, 8 meter diameter antennas. Of these 214 antennas, the core will consist of the central 94 antennas. The main array will have baselines of up to about 1000 km, determining the highest angular resolution that can be observed. This design was determined by the Science Use Cases seen in the ngVLA Memos 18 and 19 and cost constraints [3, 4].

It is the longest baselines that determine the highest angular resolution interferometers can observe; however, flux from large scale structures will be missed with long baselines. It is the shortest baselines in an interferometer that determines the largest scale structure than can be observed. The largest scale that can be observed is proportional to the wavelength being observed at over the minimum baseline of the antennas that is well sampled. 20% -- 30% of the science use cases of the ngVLA cannot be achieved with the resolution reached with the baselines of the main array of the ngVLA [3]. It's because of this that the ngVLA will also contain a central, compact array of 19 antennas called the Short Baseline Array (SBA). The shortest baselines in an interferometer and in the SBA are limited by the diameter of the antennas which, for the SBA of the ngVLA, is 6 meters. It is also important that the SBA can provide sufficient sensitivities with integration times feasible for the ngVLA. The configuration for the SBA was investigated to determine the configuration that can achieve the resolution needed for the science projects planned for the ngVLA while still producing images that have PSF's that can be deconvolved.

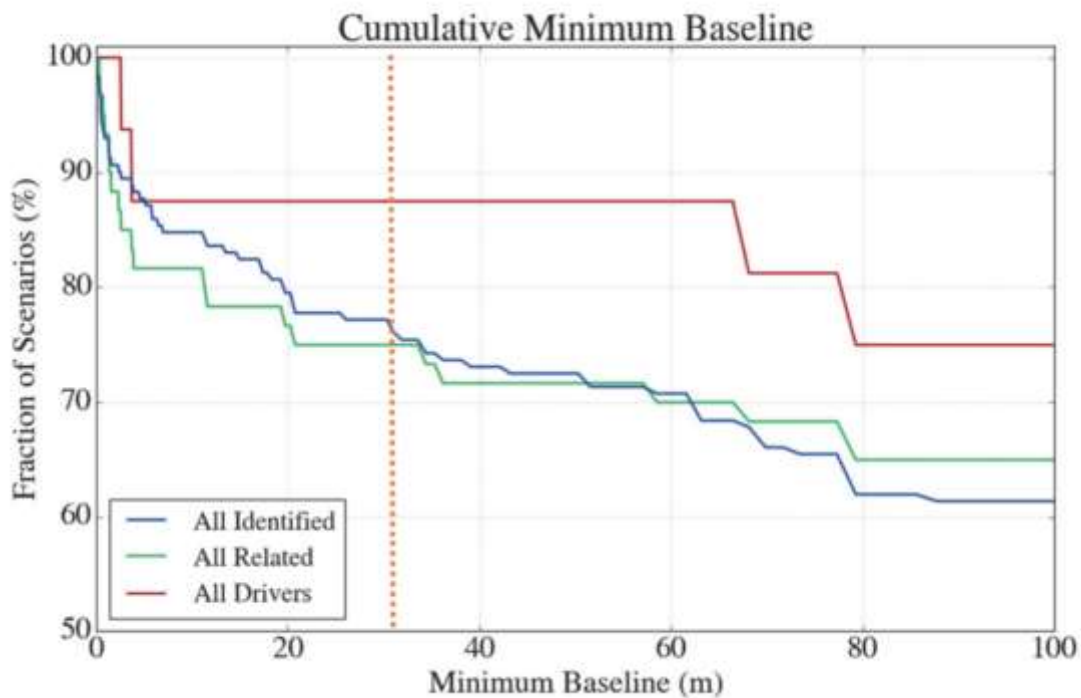


Figure 2: A plot showing the cumulative minimum baseline required for the science use cases of the ngVLA [3]. The dotted orange line refers to the minimum baseline of the ngVLA main array.

Methods:

Two configurations for the SBA were investigated. One configuration, referred to as the regular spacing SBA, was created by Dr. Brian Mason. It contained 19, 6 meter antennas regularly spaced with semi-randomized positons. The other configuration, referred to as the random spacing SBA, was created using Python code to randomly pick antenna positions within certain constraints for the SBA. The configuration was constrained to have the antennas be within a 60 meter radius disk; have 19, 6 meter antennas total; and each antenna could not be placed within 11 meters of another antenna. A ngVLA core configuration without the SBA was also used, taken from the tools page of the ngVLA website. The core configuration contains 94, 18 meter antennas. A display of these configurations can be seen in Figure 2 below.

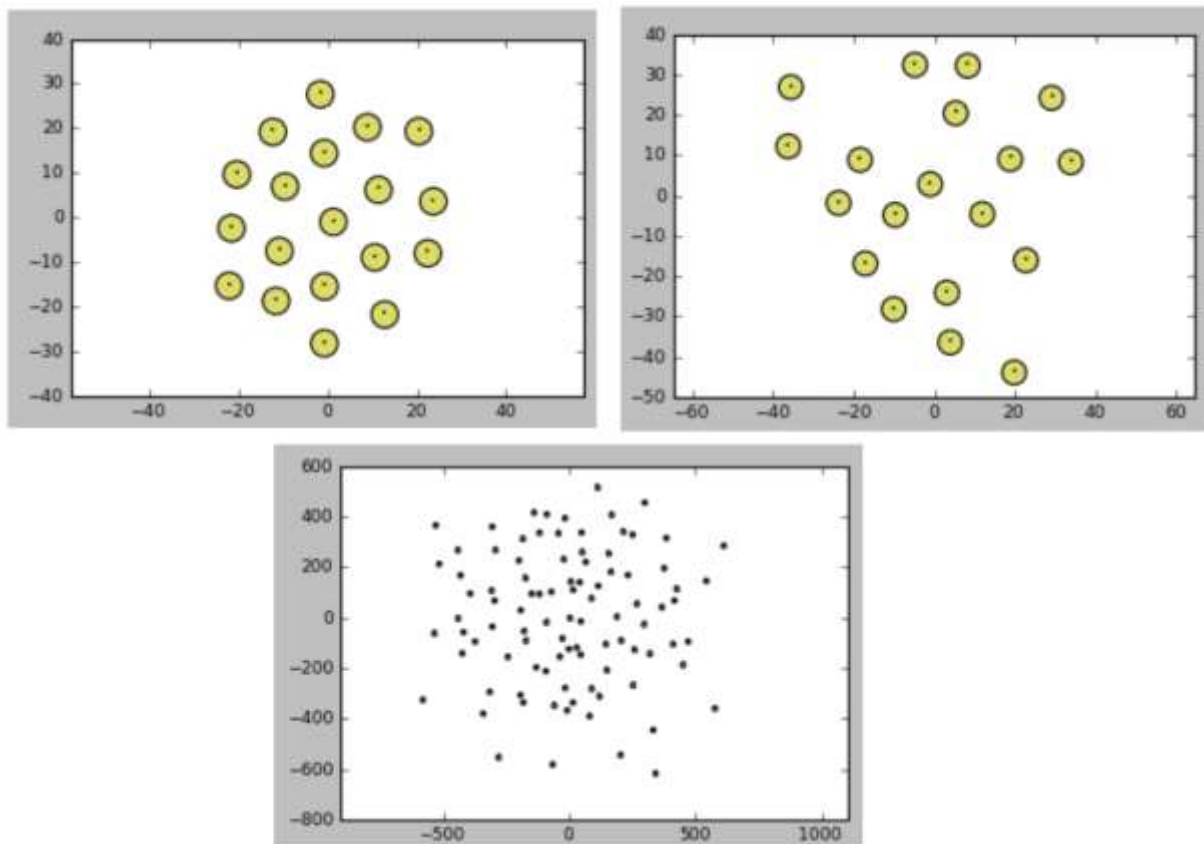


Figure 3: Upper left: Configuration for the regular spacing SBA as displayed by CASA. Upper right: Configuration for the random spacing SBA as displayed by CASA. Bottom middle: Configuration for the ngVLA core without the SBA as displayed by CASA.

Observation simulations were done first by using CASA version 5.1.1-5. First, simulated observations were performed on a blank FITS file using the regular spacing SBA configuration, the random spacing SBA configuration, and the ngVLA core configuration. CASA's simobserve task was used for this. The parameters and scripts used to run this can be seen in the Appendix section below.

Next, simulations were completed to investigate adding the appropriate level of noise to each observation for each configuration. The appropriate level of noise was calculated by using the equation

$$1. \quad \sigma = \frac{SEFD}{\sqrt{2 \delta\nu \tau}}$$

where $\delta\nu$ is the frequency bandwidth being observed and τ is the total observation time. The SEFD value was taken to be 1817.5 for the core, using the SEFD value found in the ngVLA Memo #17 [1]. The SEFD value used for the SBA configurations was 16357.5, found by scaling the 1817.5 value for 18 m antennas to 6 m diameter antennas. 100 second observations at 100 GHz with a 50 MHz bandwidth were originally used for simplicity for every configuration. With 10 second integration times, the calculated level of noise for every observation was 0.163575 Jy for the regular spacing SBA, 0.163575 Jy for the random spacing SBA, and 0.018175 Jy for the core.

Noise was attempted to be added manually using CASA's `sm` commands. The first attempt was made by using the `tsys-manual` parameter of CASA's `sm.setnoise` tool. The values for the `sm.setnoise` parameters were taken from the ngVLA webpage and memos. However, the noise levels achieved with this were significantly off from the calculated noise levels, and an empirical method of adding noise was used instead.

A description of the empirical method of adding noise can be found on the ngVLA tools page [2]. The method started by using a blank FITS file then using the `sm.setnoise` and `sm.corrupt` tool to add simple noise of 1 Jy to the FITS file. The new, corrupted FITS file was then observed, and the level of noise was measured by using the CASA viewer.

Next, simulated observations of sky images were completed using the regular spacing SBA, random spacing SBA, and the core configurations. The scripts for these observations can be found in the Appendix selection below. Instead of observing for 100 seconds total for every configuration, the total observation time was scaled for each configuration. Images of M51 were used for observations, taken from the CASA guides. The image observed can be seen in Figure 3 below. Observations were simulated both with and without adding noise. An example of a simulated observation of M51 for each configuration without noise can be seen in Figure 4 below.

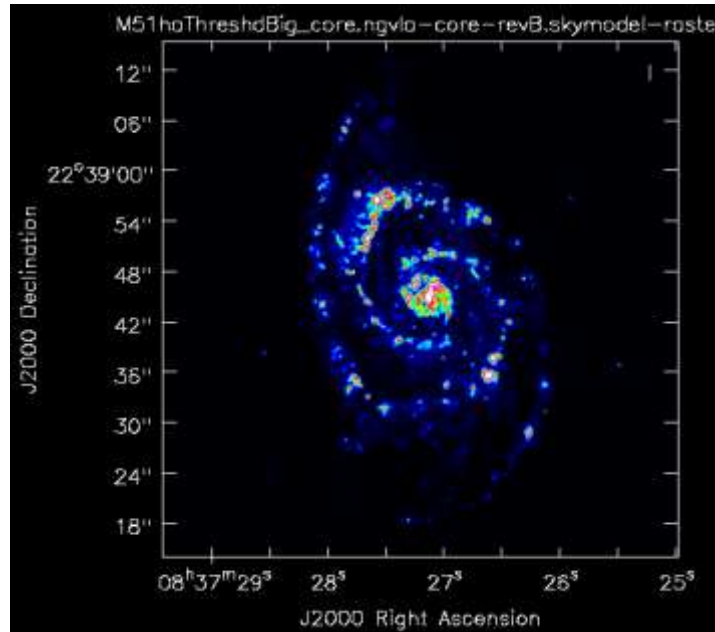


Figure 4: Sky model image of M51 used for the simulated observation for the regular spacing SBA, random spacing SBA, and core configuration.

CASA's `tclean` task was used to clean all observation simulations. The parameters used for `tclean` can be found in the scripts in the Appendix section below. An example of a cleaned image of M51 for each configuration without noise can be seen in Figure 5 below.

The resulting cleaned images were compared to their dirty images by eye to examine how well the PSF was deconvolved from each configuration and how much flux was recovered from each configuration. The core and SBA cleaned images were also feathered together to examine how well the core would perform with the SBA. An example of feathered images of M51 without noise can be seen in Figure 6 below.

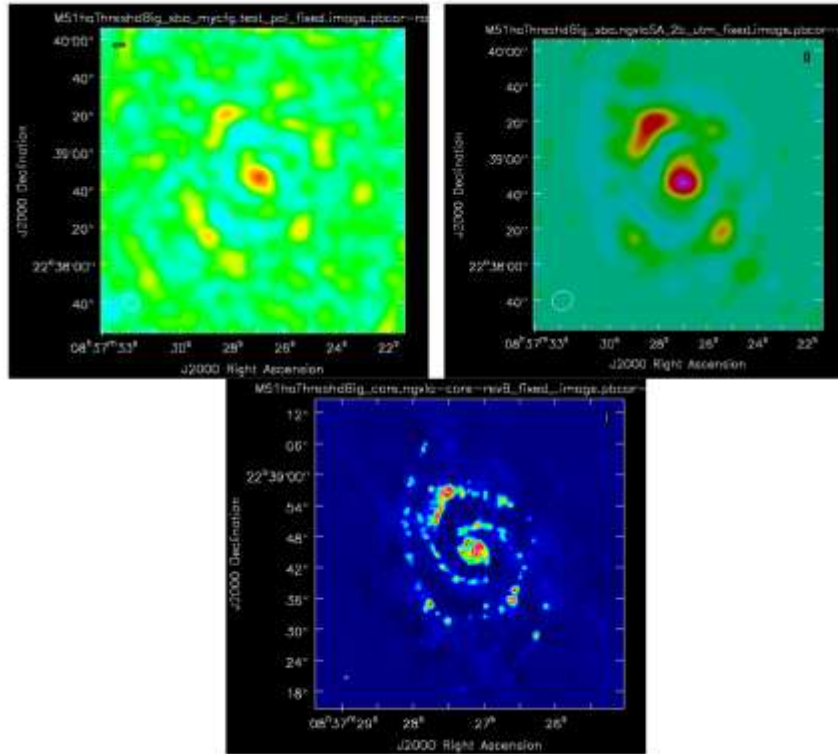


Figure 5: Upper left: Simulated observation of M51 taken with the random spacing SBA configuration without added noise. Upper right: Simulated observation of M51 taken with the regular spacing SBA configuration without added noise. Bottom middle: Simulated observation of M51 taken with the core configuration without added noise.

Next, a quantitative investigation of the largest recoverable scales was performed using CASA version 5.4.0-62. This was achieved by creating FITS files that only contained a single, centered disk of known size and flux. The flux of the disk in this case was one Jy. For the random spacing SBA and regular spacing SBA configurations, the disk size started at a diameter of five arcseconds across and ended at a diameter of 50 arcseconds across. A FWHM value of 11.5 was chosen, and the disk diameter was incremented by 1.25 arcseconds every iteration.

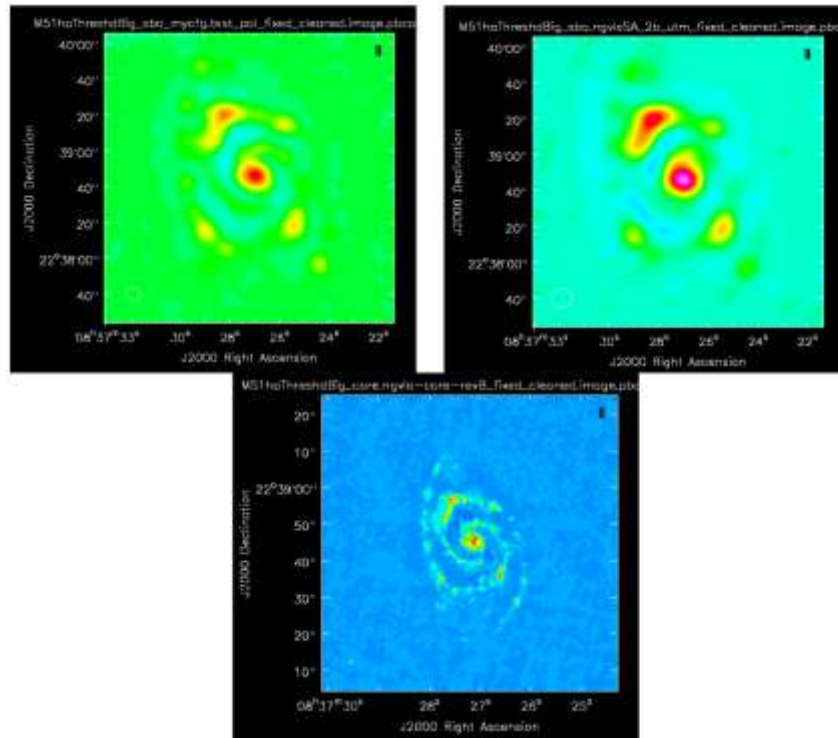


Figure 6: Upper left: Simulated, cleaned observation of M51 taken with the random spacing SBA configuration without added noise. Upper right: Simulated, cleaned observation of M51 taken with the regular spacing SBA configuration without added noise. Bottom middle: Simulated, cleaned observation of M51 taken with the core configuration without added noise.

At every iteration, CASA’s simobserve task was used to simulate observations with the regular spacing and random spacing SBA configurations. These scripts can be seen in the Appendix section below. No noise was added to the FITS files before observation.

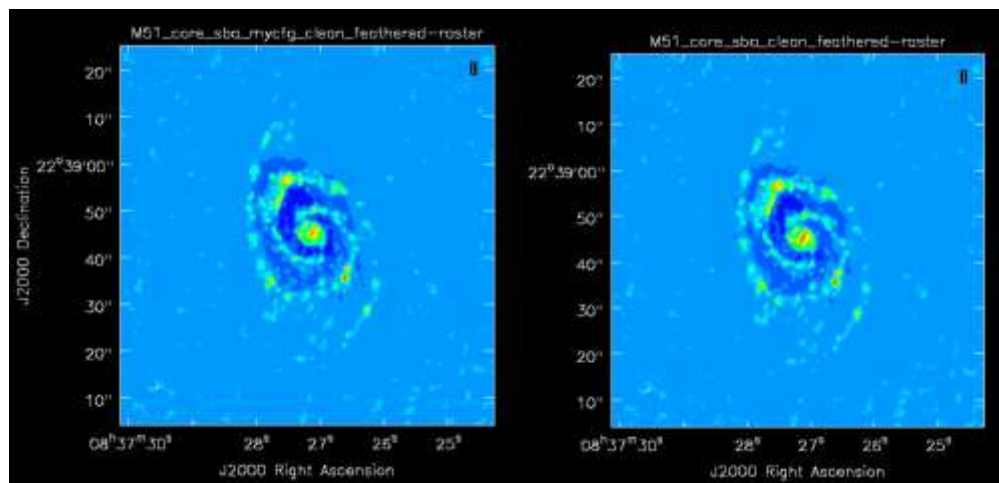


Figure 7: Left: Cleaned image of M51 without created by feathering the images produced by the core and random spacing SBA. Right: Cleaned image of M51 without created by feathering the images produced by the core and regular spacing SBA.

After each observation was made, CASA’s tclean task was used to clean the images using the automasking parameter. The parameters used for this can be seen in the scripts in the Appendix section below.

CASA’s imstat task was then used to measure the flux recovered by each configuration at each iteration. A disk region with a diameter one FWHM value greater than the disk diameter was defined to and centered around the disk. The flux density within this disk was then measured and written out by imstat using the parameters seen in the scripts in the Appendix below. These values were also corroborated by using the same defined region and the CASA viewer to manually read the flux density values. The value read by the CASA viewer followed the same trend by those read by imstat at smaller disk diameters, but did not produce the same exact values. These values had differences on the order of 0.0001 Jy to 0.1 Jy. An example of this process can be seen in Figure 7 below.

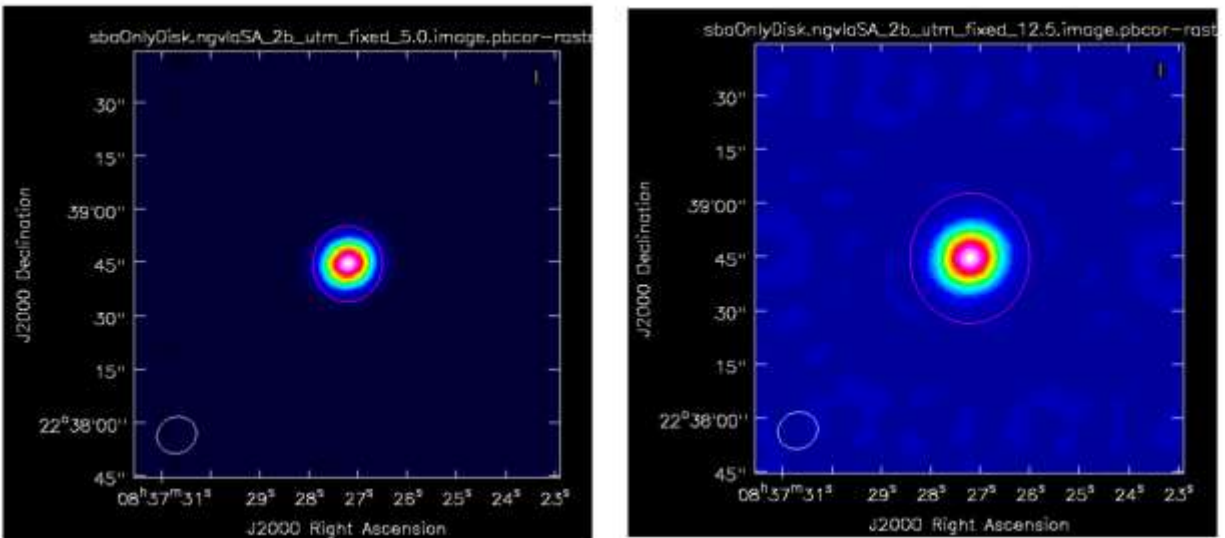


Figure 8: Left: 5 arcsecond diameter disk imaged by the regular spacing SBA. The pink disk surrounding it is the region defined to measure the recovered flux density within. Left: Disk incremented to a diameter of 12.5 arcsecond images by the regular spacing SBA. The pink disk surrounding it is the region defined to measure the recovered flux density within.

A quantitative investigation of the largest recoverable scales of the core was also performed using a similar method as the one described above. This was achieved by creating a FITS files that only contained a single, centered disk of known size and flux. The flux of the disk was one Jy. The disk size started at a diameter of one arcsecond and ended at a diameter of 50 arcseconds across. A FWHM value of 0.63 arcseconds was chosen, and the disk diameter was incremented by 0.5 arcseconds for every iteration. Due to time constraints, the regular spacing

and random spacing SBA configurations could not be rerun to start at a disk diameter of one arcsecond with 0.5 arcsecond incrementations as well.

At every iteration, CASA's simobserve task was used to simulate observations with the core configuration. No noise was added to the FITS files before observation. After each observation was made, CASA's telean task was used with the automasking parameter to clean the images.

CASA's imstat task was then used to measure the flux recovered by each configuration at each iteration. A disk region with a diameter one FWHM value greater than the disk diameter was defined to and centered around the disk. The flux density within this disk was then measured and written out by imstat. The scripts for this process and the parameters used can be found in the Appendix section below.

The flux densities taken at every iteration for each configuration were then plotted and compared. This is discussed in more detail in the Results section below.

Results:

M51 Qualitative Observations:

Due to time constraints, a quantitative analysis of the observations of M51 were not performed. However, a qualitative analysis was performed by investigating the dirty and cleaned simulated observations from each configuration by eye. It was seen that the core configuration by itself misses a significant amount of flux from larger scale structures and needs to be supplemented by the SBA configurations. It was also seen that the regular spacing SBA configuration produced more prominent sidelobes than the random spacing SBA configuration.

Largest Recoverable Scales:

The flux recovered at every disk size iteration from the core, regular spacing SBA, and random spacing SBA can be found in the Appendix section below. A plot of the results can be seen in Figure 8 below.

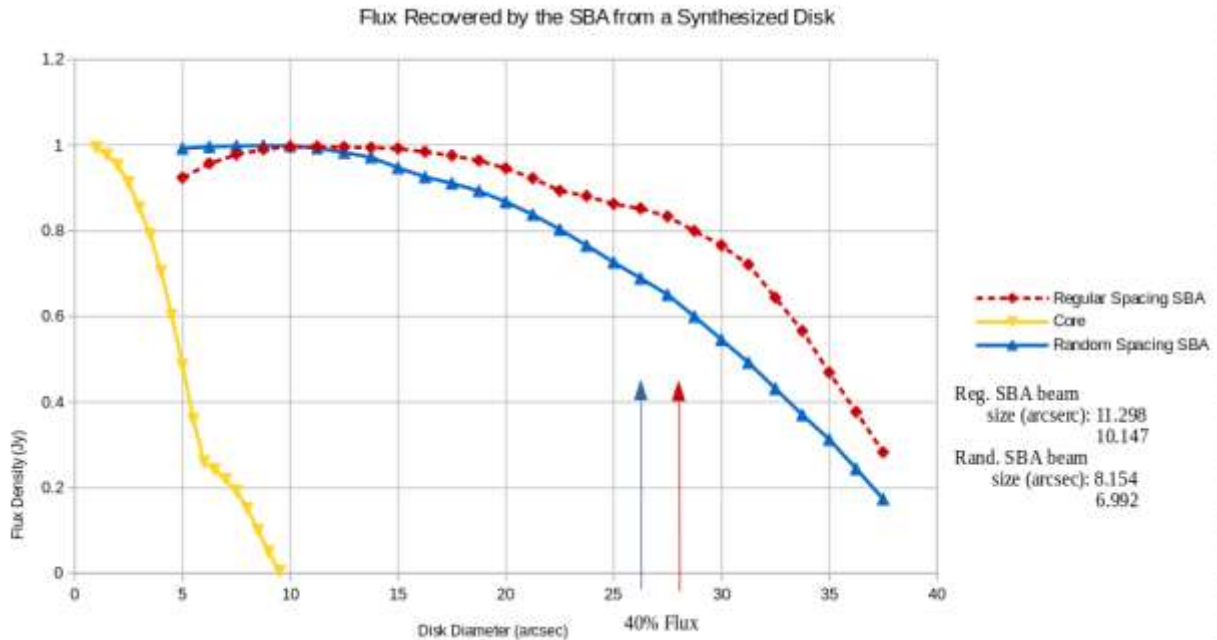


Figure 9: Plot of the flux density recovered from each configuration at every iteration of disk size. The regular spacing SBA line is marked with red squares, the random spacing SBA line is marked with blue triangles, and the core line is marked with yellow triangles.

It was found that the regular spacing SBA recovered 40% of the total flux at a disk diameter of about 36 arcsec. This was compared to the value of the wavelength observed over the minimum baseline value. The wavelength observed was 0.003 m. The minimum baseline for the regular spacing SBA was 12.432 m. This produced a value of 49.770 m, 0.723 times the measured value. It was also found that the random spacing SBA configuration recovered 40% of the total flux at a disk diameter of about 33 arcseconds. This was compared to the value of the wavelength observed over the minimum baseline value. The wavelength observed was 0.003 m. The minimum baseline for the random spacing SBA was 18.696 m. This produced a value of 33.096 m, 1.002 times the measured value. The core configuration recovered 40% of the total flux at a disk diameter of about 5 arcseconds. This was compared to the value of the wavelength observed over the minimum baseline value. The wavelength observed was 0.003 m. The minimum baseline for the core was 148.685 m. This produced a value of 4.161 m, 0.832 times the measured value.

The beam size of the regular spacing and random spacing SBA configurations were also measured in this process, and these values are noted in Figure 8 above. The regular spacing SBA beam had a semi-major axis of length 11.298 arcseconds and a semi-minor axis length of 10.147 arcseconds. The random spacing SBA beam had a semi-major axis of length 8.154 arcseconds and a semi-minor axis length of 6.992 arcseconds.

Discussion:

M51 Qualitative Observations:

The M51 observation qualitatively confirmed that the core configuration misses a significant amount of flux from larger scale structure without the SBA. It also demonstrated that a random configuration for the SBA produces weaker sidelobes than a regular spaced configuration does.

Largest Recoverable Scales:

Overall, it was seen that the flux recovery of the core configuration by itself drops off much more steeply than the SBA configurations. The core stops recovering a measurable amount of flux by CASA's imstat task below a disk diameter of ten arcseconds.

It was also seen that, in general, for disk diameters above ten arcseconds, the regular spacing SBA configuration was able to recover more flux than the random spacing SBA configuration.

But, as can be seen in Figure 8 above, although the regular spacing SBA recovers more flux than the random spacing SBA at larger disk sizes, it exhibits a dip in flux recovery at smaller disk sizes, from a diameter of 5 arcseconds to 10 arcseconds. The reason for this dip has yet to be investigated, though one possibility is that the simulated observations were not cleaned deeply enough for the regular spacing SBA configuration. Further simulations would need to be run with a lower tclean threshold and higher niter tclean value to achieve a deeper clean to investigate this possibility.

Also seen in Figure 8 above is the lack of data for the regular spacing and random spacing SBA configurations at disk diameters below five arcseconds. This is due to time constraints. Further investigation could include running the SBA configuration on a larger range of disk diameters. Also of note is the difference between the disk size iterations for the SBA configurations and the core configuration. The disk diameter for the SBA configurations incremented by 1.25 arcseconds every run while the disk diameter for the core configuration incremented by 0.5 arcseconds every run. This was again due to time constraints, and further investigation could include decreasing the incrementation in the SBA configuration runs.

In order to continue investigating the best choice for the ngVLA SBA configuration, more SBA configurations could be examined, including the configuration of a single, large dish rather than an array of antennas. Further work could also include simulating more images in addition to M51 to reproduce the results and comparing all results to the science projects planned for the ngVLA.

Conclusions:

Overall, it was seen that the core configuration drops off steeply in flux density recovery for larger scale structure, especially those above five arcseconds in diameter. The SBA is needed to recover flux for the largest scale structures, and it was seen that, despite having more prominent sidelobes, the regular spacing SBA configuration recovered more flux at larger scales than the random spacing SBA configuration, especially for structure more extensive than ten arcseconds in diameter.

Since the completion of this project, the regular spacing SBA has been presented as the design for the short baseline array that will support the ngVLA. It can match the sensitivity of the main array with 2.78 times the integration time [5]. However, it was determined that 4 total power antennas are also needed in order to fill in baselines smaller than even the SBA configuration would achieve. These antennas will be 4, 18 meter diameter dishes that can match the sensitivity of the ngVLA in 1.77 times the SBA integration time or 4.29 times the integration time of the ngVLA main array [5].

Acknowledgments:

This project was funded by NRAO's Summer Student Research Assistantship program. Many thanks also to Dr. Jim Braatz for leading the summer student program and Dr. Brian Mason for being a mentor to this project!

References:

- [1] R. Selina, E. Murphy, April 30th, 2018. *ngVLA Memo #17: ngVLA Reference Design Development & Performance Estimates*.
https://library.nrao.edu/public/memos/ngvla/NGVLA_17.pdf
- [2] C.L. Carilli & R. Indebetouw, September 2017, *Instructions for using CASA simulator for the ngVLA*.
https://science.nrao.edu/futures/ngvla/tech_materials/CONFIG.CASA.pdf
- [3] R. Selina, E. Murphy, July 26th, 2017. *ngVLA Memo #18: Summary of the Science Use Case Analysis*.
- [4] Bolatto Alberto, et al., August 2nd, 2017, *ngVLA Memo #19: Key Science Goals for the Next Generation Very Large Array (ngVLA): Report from the ngVLA Science Advisory Council*.
- [5] Mason Brian S., et al., April 18th, 2018, *ngVLA Memo #43: The ngVLA Short Baseline Array*.

Appendix:

Python code for observing blank FITs and M51 image files with the core configuration:

```
# _fixed is imaging (only) with the tarball kumar gave me from cas-9004 (on my ticket, cas-11271)
```

```
vp.reset()
```

```
# make sure you have the right dec. sign
```

```
# and that it matches what's in the pointing file-
```

```
in_map_center="J2000 08h37m27.2s 22d38m44.7s"
```

```
phase_center=in_map_center
```

```
# little mosaic
#myproj="coreOnly_mos2"

# single ptg-
#myproj = "blankrun_coreonly"
myproj="M51haThreshdBig_core"
#myproj="coreOnly"

# not really blank - has a source 50" to north from ctr-
#skymod="blank3500.fits"

#skmod="reallyBlank3500.fits"
skmod="M51haThreshdBig.fits"

#totalt = "100s"
#totalt = "7200s"

mytotaltime = "900s" # so (totalt)/(number of pointings) ~ (integer)*(int time) and I think
number of pointings = 45

#my_cfg = "nudged-30m-SWcore"
my_cfg = "ngvla-core-revB"
out_vis = myproj+'/' + myproj+'.' + my_cfg
real_cfg=my_cfg+'.cfg'

vp.reset()

vp.setpbairy(telescope='NGVLA',dishdiam=18.0,blockagediam=0.0,maxrad='8.5deg',reffreq='1.
0GHz',dopb=True) # ???

vp.saveastable('core_new.tab')
core_table = vp.getuserdefault('NGVLA')
```

```
vp.setuserdefault(core_table,'NGVLA','coreNgvla')
```

```
default("simobserve")
```

```
project      = myproj
```

```
indirection=in_map_center
```

```
skymodel     = skmod
```

```
incell="0.06arcsec" # should this be the same?
```

```
inbright="1Jy/pixel"
```

```
incenter="100.0GHz"
```

```
inwidth="50MHz"
```

```
setpointings = True
```

```
mapsize = "2.0arcmin"
```

```
hourangle = "transit"
```

```
integration  = "10s"
```

```
obsmode      = "int"
```

```
graphics     = "both"
```

```
refdate = "2017/12/01"
```

```
antennalist  = real_cfg
```

```
totaltime   = mytotaltime
```

```
thermalnoise = ""
```

```
simobserve()
```

```
# dirty image
```

```
default('tclean')
```

```
vis=[out_vis+'.ms']
```

```
imagename = out_vis+'_fixed'
```

```
vptable = 'sba.tab'
```



```
gridder = 'mosaic'
pblimit= 0.005
niter= 0
phasecenter = in_map_center
interactive = False
pbcor= True
imsize = [5400,5400]
cell      = '0.06arcsec'
threshold = 0.00012959187079218703
usemask='auto-multithresh'
sidelobethreshold= 3.0
noisethreshold= 5.0
lownoisethreshold= 1.5
minbeamfrac= 0.3
growiterations= 75
negativethreshold= 7.0
tclean()

noise = '0.012215898517992244Jy' #core

os.system('cp -r ' + out_vis + '.ms ' + out_vis + '_noise.ms')
sm.openfromms(out_vis + '_noise.ms')
sm.setnoise(mode = 'simplenoise', simplenoise = noise)
sm.corrupt()
sm.done()

vis=[out_vis+'_noise.ms']
```

```
imagename = out_vis+'_noise_fixed'  
tclean()
```

Python code for observing blank FITs and M51 image files with the SBA configurations:

```
# Simulate NGVLA Short Baseline Array observations with and without noise  
# Uses automasking for TCLEAN  
# Ryan Parziale, 7-30-2018
```

```
vp.reset()
```

```
in_map_center="J2000 08h37m27.2s 22d38m44.7s"  
phase_center=in_map_center
```

```
#myproj="blankrun_sbaonly_mycfg"  
myproj="M51haThreshdBig_sba_mycfg"
```

```
#skmod="reallyBlank3500.fits"  
skmod="M51haThreshdBig.fits"
```

```
#totalt ="1218s" #scaled by the 900s from the core observartion for the regular configuration and  
the number of pointings (7)
```

```
totalt ="1673s" #scaled by the 900s from the core observartion for my configuration and the  
number of pointings (7)
```

```
#my_cfg = "ngvlaSA_2b_utm"
```

```
my_cfg = "test_pol"
```

```
out_vis = myproj+'/'+myproj+'.'+my_cfg
```

```
real_cfg=my_cfg+'.cfg'
```

```
vp.reset()
```

```
vp.setpbairy(telescope='NGVLA',dishdiam=6.0,blockagediam=0.0,maxrad='3.5deg',reffreq='1.0  
GHz',dopb=True)
```

```
vp.saveastable('sba.tab')
```

```
sba_table = vp.getuserdefault('NGVLA')
```

```
vp.setuserdefault(sba_table,'NGVLA','sbaNgvla')
```

```
default("simobserve")
```

```
project      = myproj
```

```
indirection = in_map_center
```

```
skymodel     = skmod
```

```
incell="0.18arcsec"
```

```
inbright="0.001Jy/pixel"
```

```
incenter="100.0GHz"
```

```
inwidth="50MHz"
```

```
setpointings = True
```

```
hourangle = "transit"
```

```
integration  = "10s"
```

```
mapsize = "2.0arcmin"
```

```
obsmode      = "int"
```

```
graphics      = "both"
refdate = "2017/12/01"
antennalist   = real_cfg
totaltime    = totalt
thermalnoise  = ""
simobserve()

vp.reset()

vp.setpbairy(telescope='NGVLA',dishdiam=6.0,blockagediam=0.0,maxrad='3.5deg',reffreq='1.0
GHz',dopb=True)

sba_table = vp.getuserdefault('NGVLA')
vp.setuserdefault(sba_table,'NGVLA','sbaNgvla')

# TCLEAN
default('tclean')
vis=[out_vis+'.ms']
imagenam = out_vis+'_fixed'
vptable = 'sba.tab'
gridder = 'mosaic'
pblimit= 0.005
niter= 0
phasecenter = in_map_center
interactive = False
pbcor= True
imsize = [5400,5400]
cell      = '0.06arcsec'
```

```
#threshold = '0.0050527020000000001Jy' #reg  
threshold = '0.0050735340000000002Jy' #mycfg
```

```
usemask='auto-multithresh'
```

```
sidelobethreshold=1.25
```

```
noisethreshold=5.0
```

```
lownoisethreshold=2.0
```

```
minbeamfrac=0.1
```

```
growiterations=75
```

```
negativethreshold=0.0
```

```
tclean()
```

```
#noise = '0.25451190232784393Jy' #SBA
```

```
noise = '0.29883797025062347Jy' #My SBA
```

```
os.system('cp -r ' + out_vis + '.ms ' + out_vis + '_noise.ms')
```

```
sm.openfromms(out_vis + '_noise.ms')
```

```
sm.setnoise(mode = 'simplenoise', simplenoise = noise)
```

```
sm.corrupt()
```

```
sm.done()
```

```
vis=[out_vis+'_noise.ms']
```

```
imagename = out_vis+'_noise_fixed'
```

```
tclean()
```

Python code for observing FITS files to later add a disk to with the core configuration:

Observes a FITS file (Run this once)

```
in_map_center="J2000 08h37m27.2s 22d38m44.7s"
```

```
phase_center=in_map_center
```

```
myproj="coreOnlyDisk_final"
```

this has a source 25" to the north of the map center- 1 jy/pix. But this will be overwritten.

```
skmod="blank3500.fits"
```

I'm not sure if this should be the same or not?? or should I scale it for the different configurations??'

```
mytotaltime="270s" # core
```

```
my_cfg = "ngvla-core-revB"
```

```
out_vis = myproj+'/' + myproj+'.' + my_cfg
```

```
real_cfg=my_cfg+'.cfg'
```

```
vp.reset()
```

```
vp.setpbairy(telescope='NGVLA',dishdiam=18.0,blockagediam=0.0,maxrad='8.5deg',reffreq='1.0GHz',dopb=True) # ????
```

```
vp.saveastable('core_new.tab')
```

```
core_table = vp.getuserdefault('NGVLA')
```

```
vp.setuserdefault(core_table,'NGVLA','coreNgvla')
```

```
default("simobserve")
```

```
project      = myproj
```

```
indirection=in_map_center
```

```
skymodel      = skmod
incell="0.06arcsec" # should this be the same?
inbright="1Jy/pixel"
incenter="100.0GHz"
inwidth="50MHz"
setpointings  = True
mapsize      = "1.5arcmin"
hourangle    = "transit"
integration   = "10s"
obsmode      = "int"
graphics     = "both"
refdate      = "2017/12/01"
antennalist  = real_cfg
totaltime    = mytotaltime
thermalnoise  = ""
simobserve()
```

Python code for observing FITS files to later add a disk to with the SBA configurations:

```
# Observes a FITS file (Run this once)
```

```
in_map_center="J2000 08h37m27.2s 22d38m44.7s"
phase_center=in_map_center
myproj="sbaOnlyDisk_lasttime" # EDIT THIS!!!
```

```
# this has a source 25" to the north of the map center- 1 jy/pix. But this will be overwritten.
```

```
skmod="blank3500.fits"
```

```
# I'm not sure if this should be the same or not?? or should I scale it for the different
configurations??'

# EDIT THIS!!!

mytotaltime="1800s" # regular SBA
#mytotaltime="2480s" # my SBA

# EDIT THIS!!!

my_cfg = "ngvlaSA_2b_utm" # regular SBA
#my_cfg = "test_pol" # my SBA
out_vis = myproj+'/' + myproj+'.' + my_cfg
real_cfg=my_cfg+'.cfg'

# if i run the script without the following line
# the simulator doesnt use the right PB. The imaging
# at least judging by the mask does.

vp.reset()

vp.setpbairy(telescope='NGVLA',dishdiam=6.0,blockagediam=0.0,maxrad='8.5deg',reffreq='1.0
GHz',dopb=True)

vp.saveastable('sba_new.tab')

sba_table = vp.getuserdefault('NGVLA')

vp.setuserdefault(sba_table,'NGVLA','sbaNgvla')

default("simobserve")

project      = myproj
indirection=in_map_center
skymodel     = skmod
incell="0.06arcsec"
```



```
inbright="1Jy/pixel"
incenter="100.0GHz"
inwidth="50MHz"
setpointings = True
mapsize = "1.5arcmin"
hourangle = "transit"
integration = "10s"
obsmode = "int"
graphics = "both"
thermalnoise = ""
refdate = "2017/12/01"
antennalist = real_cfg
totaltime = mytotaltime
thermalnoise = ""
simobserve()
```

Python code for adding a disk to observations and measuring flux recovery with the core configuration:

```
# Adds a disk and cleans the image
# Run this once (and I'll put in a loop)

print "Removing the previous .cl regions\n"
os.system('rm -r *.cl')

# EDIT THIS!!!
```

```
#diskDiam = 0.63 # Core
diskDiam = 1.0
dd = diskDiam /2 # Half of the BMAJ of the pbcor beam
FWHM = 0.63

maxDiam = 50.0 #just for now

print "Opening the output file\n"
#Opening the file where all the flux sums will be written to
try:
    outFile = open("diskFluxSums_Core_Final.txt","w+") # EDIT THIS
    outFile.write('# The flux sums for disks of increasing diameter for the core cfg\n# Starts at a
diameter of ' + str(diskDiam) + ' and increases by ' + str(dd) + ' every iteration.\n')
except OSError as err:
    print("OS error: {0}".format(err))

while diskDiam <= maxDiam:

    print "Creating the disk of diameter " + str(diskDiam) + "\n"
    # Creating the disk
```

```
cl.addcomponent(flux=[1.0,0,0,0], dir=me.direction('J2000', '08h37m27.2', '22d38m44.7'),
majoraxis= str(diskDiam) + 'arcsec', minoraxis= str(diskDiam) + 'arcsec', positionangle='0deg',
shape='Disk')
```

```
cl.rename('mydisk' + str(diskDiam) + '.cl') #disk named based off disk diameter
```

```
cl.done()
```

```
sm.openfromms(out_vis+'.ms')
```

```
# set field id's appropriately for the given ms ... will be different for different
telescopes/mosaic coverages etc.
```

```
for k in range(0,27):
```

```
    sm.setdata(fieldid=k)
```

```
    sm.setvp(True, False, 'core_new.tab') # why doesn't it woook
```

```
    sm.predict(complist='mydisk' + str(diskDiam) + '.cl')
```

```
sm.close()
```

```
# the above may overwrite noise so would need to precede corrupt ...
```

```
# For now, I'm running without noise
```

```
print "tcleaning\n"
```

```
# 7m cleaned image
```

```
default('tclean')
```

```
vis=[out_vis+'.ms']
```

```
imagename=out_vis+'_fixed_' + str(diskDiam)
```

```
vptable='core_new.tab' # ???
```

```
gridder='mosaic'
```

```
pblimit=0.05
```

```
niter=100000
```

```
phasecenter=phase_center
```

```
interactive=False
pbcor=True
imsize      = [4800,4800]
cell        = '0.1arcsec' #this seems like a TYPO!!!
threshold = '0.01Jy'
usemask='auto-multithresh'
sidelobethreshold= 3.0
noisethreshold= 5.0
lownoisethreshold= 1.5
minbeamfrac= 0.3
growiterations= 75
negativethreshold= 7.0
tclean()
```

```
print "Creating a CASA region file\n"
```

```
#Creating a CASA region file
```

```
try:
```

```
    regionFile = open("fluxRegion"+str(diskDiam),"w+")
```

```
    regionFile.write('#CRTFv0 CASA Region Text Format version 0\n')
```

```
    regionFile.write('ellipse [[08:37:27.20000, +022.38.44.7000], [' + str(diskDiam +
FWHM) + 'arcsec, ' + str(diskDiam + FWHM) + 'arcsec], 0.00000000deg] coord=J2000,
corr=[I], linewidth=1, linestyle=-, symsize=1, symthick=1, color=magenta, font="DejaVu Sans",
fontsize=11, fontstyle=normal, usetex=false')
```

```
except OSError as err:
```

```
    print("OS error: {0}".format(err))
```

```

regionFile.close()

print "Calculating flux density using imstat()\n"
#Calculates the sum of the flux and writes it to the output file
default(imstat)

xstat = imstat(imagename = out_vis + '_fixed_' + str(diskDiam) + '.image.pbcor', region =
'fluxRegion' + str(diskDiam), logfile = 'fluxLogs_Core_fix_final.txt') # EDIT THIS??
outFile.write(str(xstat['flux'][0])+'\n')

diskDiam = diskDiam + dd

outFile.close()

```

Python code for adding a disk to observations and measuring flux recovery with the SBA configurations:

```

# Adds a disk and cleans the image
# Run this once (and I'll put in a loop)

print "Removing the previous .cl regions\n"
os.system('rm -r *.cl')

# EDIT THIS!!!

#diskDiam = 11.5 #BMAJ of the beam??, regular SBA

```

```
#diskDiam = 8.2 # my SBA
diskDiam = 1.0
dd = diskDiam/2 # Half of the BMAJ of the pbcor beam
FWHM = 11.5

#lamb = (3e8)/(100e9)

# EDIT THIS!!!
#bmin = 12.4329320757 #regSBA
#bmin = 18.6966320941 #mySBA
#maxDiam = lamb/bmin #What the fuck my dude?? Idk how to do the max diameter so I'll just
come back to that
maxDiam = 5.0 #just for now

print "Opening the output file\n"
#Opening the file where all the flux sums will be written to
try:
    outFile = open("diskFluxSums_SBA_mycfg_final.txt","w+") # EDIT THIS
    outFile.write('# The flux sums for disks of increasing diameter for my SBA cfg\n# Starts at
a diameter of ' + str(diskDiam) + ' and increases by ' + str(dd) + ' every iteration.\n')
except OSError as err:
    print("OS error: {0}".format(err))
```

```
while diskDiam <= maxDiam:
```

```
    print "Creating the disk of diameter " + str(diskDiam) + "\n"
```

```
    # Creating the disk
```

```
    cl.addcomponent(flux=[1.0,0,0,0], dir=me.direction('J2000', '08h37m27.2', '22d38m44.7'),  
majoraxis= str(diskDiam) + 'arcsec', minoraxis= str(diskDiam) + 'arcsec', positionangle='0deg',  
shape='Disk')
```

```
    cl.rename('mydisk' + str(diskDiam) + '_SBA_mycfg.cl') #disk named based off disk  
diameter
```

```
    cl.done()
```

```
    sm.openfromms(out_vis+'.ms')
```

```
    # set field id's appropriately for the given ms ... will be different for different  
telescopes/mosaic coverages etc.
```

```
    for k in range(0,3):
```

```
        sm.setdata(fieldid=k)
```

```
        sm.setvp(True, False, 'sba_new.tab')
```

```
        sm.predict(complist='mydisk' + str(diskDiam) + '_SBA_mycfg.cl')
```

```
    sm.close()
```

```
    # the above may overwrite noise so would need to precede corrupt ...
```

```
    # For now, I'm running without noise
```

```
    print "tcleaning\n"
```

```
    # 7m cleaned image
```

```
    default('tclean')
```

```
    vis=[out_vis+'.ms']
```

```
    imagename=out_vis+'_fixed_' + str(diskDiam)
```

```
    vptable='sba_new.tab'
```

```
gridder='mosaic'
pblimit=0.05
niter=100000
phasecenter=phase_center
interactive=False
pbcor=True
imsize      = [800,800]
cell        = '0.6arcsec' #this seems like a TYPO!!!
threshold = '0.01Jy'
usemask='auto-multithresh'
sidelobethreshold= 1.25
noisethreshold= 5.0
lownoisethreshold= 2.0
minbeamfrac= 0.1
growiterations= 75
negativethreshold= 0.0
tclean()
```

```
print "Creating a CASA region file\n"
```

```
#Creating a CASA region file
```

```
try:
```

```
    regionFile = open("fluxRegion_SBA_mycfg_"+str(diskDiam),"w+")
    regionFile.write("#CRTFv0 CASA Region Text Format version 0\n")
    regionFile.write('ellipse [[08:37:27.20000, +022.38.44.7000], [' + str(diskDiam +
FWHM) + 'arcsec, ' + str(diskDiam + FWHM) + 'arcsec], 0.00000000deg] coord=J2000,
```



```
corr=[I], linewidth=1, linestyle=-, symsize=1, symthick=1, color=magenta, font="DejaVu Sans",  
fontsize=11, fontstyle=normal, usetex=false')
```

```
except OSError as err:
```

```
    print("OS error: {0}".format(err))
```

```
regionFile.close()
```

```
print "Calculating flux density using imstat()\n"
```

```
#Calculates the sum of the flux and writes it to the output file
```

```
default(imstat)
```

```
xstat = imstat(imagename = out_vis + '_fixed_' + str(diskDiam) + '.image.pbcor', region =  
'fluxRegion_SBA_mycfg_' + str(diskDiam), logfile = 'fluxLogs_SBA_mycfg_final.txt') # EDIT  
THIS???
```

```
outFile.write(str(xstat['flux'][0])+'\n')
```

```
diskDiam = diskDiam + dd
```

```
outFile.close()
```

Flux density recovered by the core configuration across disk sizes as measure by CASA's imstat task:

	Core
Disk Diameter (arcsec)	Flux Density (Jy)

1	0.994174
1.5	0.976601
2	0.95264
2.5	0.911744
3	0.854093
3.5	0.790119
4	0.704611
4.5	0.601967
5	0.485323
5.5	0.359739
6	0.26028
6.5	0.241749
7	0.219417
7.5	0.19151
8	0.151634
8.5	0.100724
9	0.0505
9.5	0.00478
10	-0.03684
10.5	-0.0765
11	-0.10672
11.5	-0.12666
12	-0.13702
12.5	-0.14218
13	-0.1443
13.5	-0.1442
14	-0.1438
14.5	-0.14063
15	-0.13174
15.5	-0.11966
16	-0.10642
16.5	-0.09267
17	-0.08139
17.5	-0.0712
18	-0.06068
18.5	-0.04938
19	-0.03914
19.5	-0.03235
20	-0.02773

20.5	-0.02429
21	-0.02101
21.5	-0.01778
22	-0.01453
22.5	-0.01013
23	-0.00612
23.5	-0.00363
24	-0.00201
24.5	-0.00048
25	0.000329
25.5	0.000912
26	0.001512
26.5	0.000967
27	-0.00012
27.5	2.80E-05
28	0.001044
28.5	0.001866
29	0.002411
29.5	0.002793
30	0.003058
30.5	0.003715
31	0.004618
31.5	0.005387
32	0.005681
32.5	0.005952
33	0.006309
33.5	0.007189
34	0.00803
34.5	0.008312
35	0.008461
35.5	0.008553
36	0.008634
36.5	0.008592
37	0.008357
37.5	0.007838
38	0.00691
38.5	0.005797
39	0.004602
39.5	0.003517

40	0.00257
40.5	0.00187
41	0.001154
41.5	0.000417
42	-0.0003
42.5	-0.00084
43	-0.00133
43.5	-0.0017
44	-0.002
44.5	-0.00213
45	-0.00202
45.5	-0.00175
46	-0.00139
46.5	-0.00092
47	-0.0003
47.5	0.000347
48	0.000727
48.5	0.000915
49	0.001017
49.5	0.001056
50	0.000973

Flux density recovered by the regular spacing SBA and random spacing SBA configurations across disk sizes as measure by CASA's imstat task:

(Due to being unable to remotely access the files located on NRAO's computer, the flux density for the random spacing SBA for disk diameters above 45 arcseconds cannot be recovered right now.)

	Regular Spacing SBA	Random Spacing SBA
Disk Diameter (arcsec)	Flux Density (Jy)	Flux Density (Jy)
5	0.923621945	0.992193817
6.25	0.956922033	0.99630524
7.5	0.978092913	0.997723714
8.75	0.990008352	0.999009387
10	0.996026465	0.998185676
11.25	0.996258083	0.993246933
12.5	0.995206929	0.982758343

13.75	0.993839009	0.971362016
15	0.991706559	0.947061766
16.25	0.984230395	0.925367428
17.5	0.975144064	0.910741204
18.75	0.963734185	0.892798548
20	0.945509246	0.867414201
21.25	0.922214572	0.837729772
22.5	0.893410209	0.803060006
23.75	0.880609665	0.765535152
25	0.862373839	0.726289321
26.25	0.851853599	0.68923882
27.5	0.833212694	0.650710732
28.75	0.799231547	0.59964904
30	0.765481105	0.545736621
31.25	0.720905281	0.492300161
32.5	0.643584133	0.431589616
33.75	0.56604106	0.370176122
35	0.468583215	0.31269955
36.25	0.376872318	0.244261514
37.5	0.282610566	0.173456818
38.75	-0.432960096	-0.190390914
40	-0.378874302	-0.150234302
41.25	-0.323364805	-0.116243977
42.5	-0.267382005	-0.08723365
43.75	-0.214712665	-0.062955634
45	-0.165766514	-0.044216529
46.25	-0.121279811	
47.5	-0.081536258	
48.75	-0.046357058	
50	-0.018505868	