September 1, 2021
NRAO Research Experiences for Undergraduates Program

SiRIUS: Simulation of Radio Interferometry from Unique Sources

Michael Barth,[1, 2] Jan-Willem Steeb,[1] and Andrew McNichols[1]

[1] *National Radio Astronomy Observatory*
[2] *University of Pittsburgh*

ABSTRACT

We present SiRIUS, a protoype software package for the simulation of radio interferometry visibilities. This package is capable of simulating the distortions introduced by the interferometer's beam. The mathematical formalism we use to simulate these effects is the Radio Interferometry Measurement Equation (RIME). The capabilities we have currently developed include several interferometer beam models, mosaic and single-pointing observations, and multiple point sources. This package is written entirely in compiled and parallel Python, which enables straightforward extensibility, cross-platform ease of installation, and efficient performance. The source code can be found at SiRIUS.

Keywords: Radio interferometry (1346) — Astronomy software (1855) — Astronomical simulations (1857) — Astronomy data modeling (1859)

1. INTRODUCTION

The current standard radio interferometry simulation software is the Common Astronomy Software Applications (CASA) package, which has been widely used by the radio astronomy community since 2007 (McMullin et al. 2007). This package has two options for simulating visibilities: component list and model image. The component list simulation operates by adding each source individually as a component of the sky. The SiRIUS package aims to replicate and extend the component list simulation functionality of CASA as part of the larger effort to develop a next-generation CASA prototype.

1.1. *Next Generation CASA prototype: Software Infrastructure*

While CASA has been and continues to be very effective software for radio astronomy, there are several motivations for the current effort to develop a next-generation CASA prototype. The primary goal of this effort is to create a computationally-efficient codebase that can be easily extended with new functionality, can be installed on most platforms, and is written in a language familiar to the broad community of astronomers. The Python language has been adopted by a large portion of the astronomy community for data reduction and analysis, which is why we have chosen Python as the programming language for the next generation CASA prototype. Furthermore, Python has strong cross-platform capabilities. The Numba package allows Python to attain runtime speeds similar to C++ while facilitating rapid development (Lam et al. 2015). Additionally, the Dask package provides a powerful framework for handling the parallelization of Python code on both single machines and large clusters (Rocklin 2015). We combine Numba and Dask to enable Python to achieve similar performance to C++ while simplifying the codebase.

The full software stack for the next-generation CASA prototype is shown in Figure 1. In addition to Numba and Dask, we plan on incorporating CuPy, a package for compiling code to run on Graphics Processing Units (GPUs), rather than Central Processing Units (CPUs), since GPUs are able to execute a larger number of tasks simultaneously (Okuta et al. 2017). For mathematical calculations and vectorization, our code uses Astropy (Astropy Collaboration et al. 2018) and NumPy (Harris et al. 2020). The data structure used to store visibilities generated by SiRIUS is vis.zarr, which is a dataset of labeled n-dimensional arrays and is equivalent to the measurement set in CASA (Kemball & Wieringa 2001). Reading and manipulation of vis.zarr datasets is handled by xarray, which is designed to use an n-dimensional labeled array data structure. A dedicated application-level user interface and a cloud deployment model are under development.

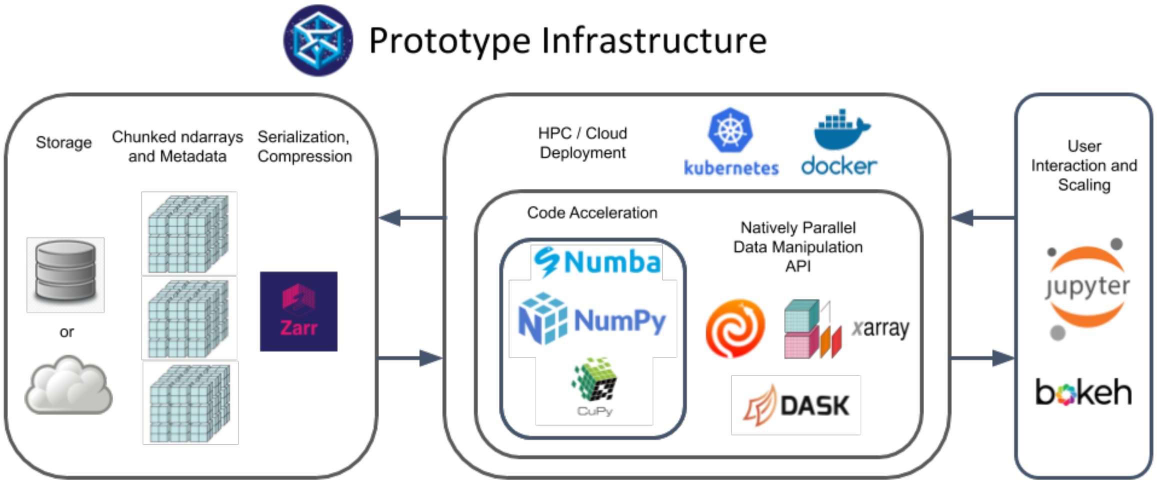

Figure 1: The next generation CASA prototype infrastructure and software stack. This infrastructure is compatible with cloud deployment using Kubernetes and Docker as well as with running on a local machine. N-dimensional arrays are stored in vis.zarr datasets while n-dimensional array labeling is handled by xarray. Astropy and NumPy are used for mathematical computation and vectorization. Numba, Dask and CuPy are used for code optimization. Jupyter Notebooks are used to provide user interaction, and Bokeh is used to display parallel process progress when using Dask. Image source: CNGI Team (2021).

1.2. *Next Generation CASA prototype: Design Philosophy*

We have chosen several principles to guide the development of the next generation CASA prototype codebase and simplify user interactions. The first principle is functional programming, the advantage of which is that it is well-suited to mathematical computations, where every equation can be treated as a function; this also simplifies debugging and improves code legibility by breaking down the code base into small, testable functions. The second principle is variable complexity: the user can choose between running the code with full complexity, minimal complexity, or any level in between those two. In SiRIUS, the interferometry model's complexity is reduced by passing *singleton* arguments to the simulation. Singleton arguments have unary dimensions whose values are treated as constants in the model simulation.

1.3. *SiRIUS: New Features*

We incorporate several simulation features in SiRIUS that are not available or not easily accessible in CASA's component list simulator. The new features include full Mueller matrix beam models and compensation for parallactic rotation; additional features are currently being tested. The full Mueller matrix beam models and compensation for parallactic rotation are described in detail in Sections 2 and 3; these features enable more precise simulation of visibilities obtained by large interferometers such as ALMA, VLA, MeerKAT, SKA-Mid, and ngVLA. This precision is especially important for writing observing proposals for current large arrays, validating high-precision observations and algorithms, and planning the design of future large arrays.

2. BACKGROUND

In order to introduce the new simulation features in SiRIUS, we must first introduce the mathematical formalism used by SiRIUS to handle the simulation of radio visibilities. This section is organized as follows: in 2.1, we describe the coordinate system used by SiRIUS; in 2.2, we describe the radio interferometry formalism used by SiRIUS; and in 2.3, we describe the interferometer beam models that can be used in SiRIUS.

2.1. *Coordinate System*

The interferometry coordinate system used by SiRIUS is depicted in Figure 2. The 3-dimensional terrestrial coordinate system ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$) describes the positions of the individual antennae: the z axis points to the north celestial pole and the $x - y$ plane is the equatorial plane of the earth with the x axis oriented to the east. In order to describe the positions of radio antenna baselines, we use a 3-dimensional coordinate system with axes ($\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$) where $\mathbf{w}$ is pointed in the direction of the array's phase center, which is typically at the center of the source. The direction of the **u** and **v** axes is chosen by convention; for SiRIUS, we have chosen to use the following convention: **u** always points in the direction of increasing angular longitudinal coordinate (α), and **v** always points in the direction of increasing angular latitudinal coordinate (δ). This is the same convention that is used by the measurement set in CASA (Kemball & Wieringa 2001). Additionally, we use the coordinate system (**l**, **m**, **n**) to describe the positions of the sources, where **n** is in the direction of the phase center and related to **w** by the relationship

$$\boldsymbol{n} = \boldsymbol{w} - \boldsymbol{S_0} \tag{1}$$

where $\boldsymbol{S_0}$ is the vector between the center of the radio interferometer and the phase center's location in the source plane (sources are assumed to lie in a distant plane, and their light is assumed to arrive at the interferometer as a plane wave). By convention, we set **l** and **m** to point in the same direction as **u** and **v**, respectively. The sky coordinates (l, m, n) are measured along the (**l**, **m**, **n**) axes (which have the same directionality as the (**u**, **v**, **w**) axes). The conversion from right ascension and declination to l and m is based on the following definition: (l, m, n) are the directional cosines of the angles between a given source's position vector and the (**l**, **m**, **n**) axes (Figure 3).

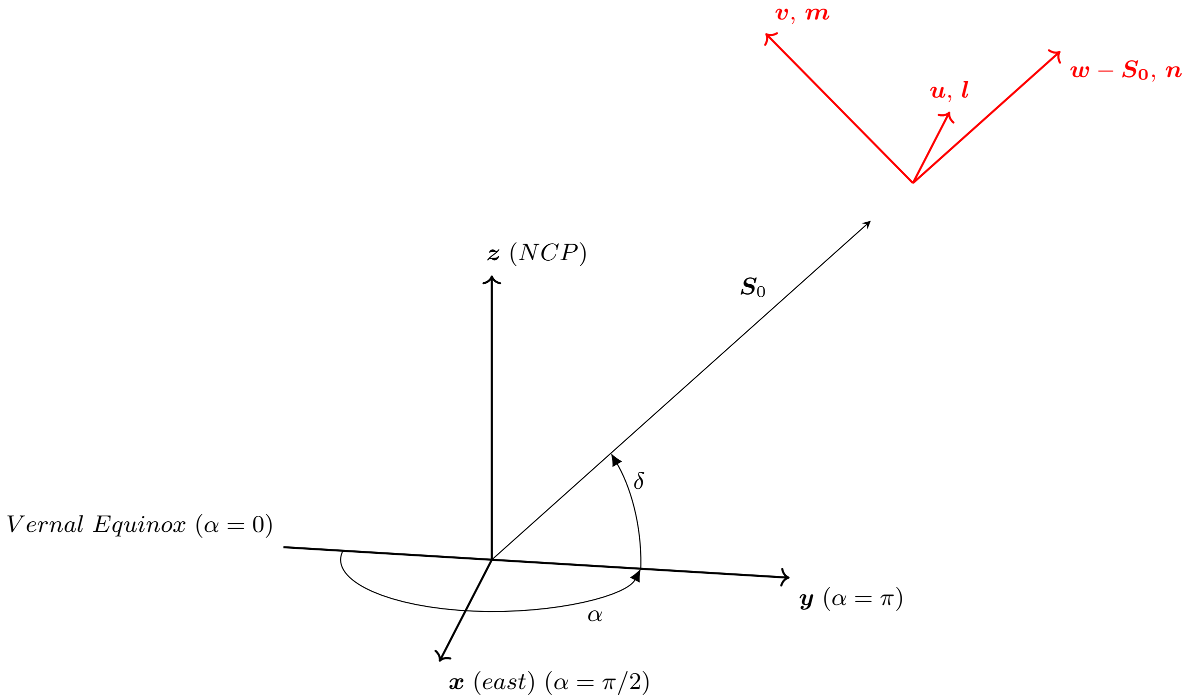
$\boldsymbol{v}$, $\boldsymbol{m}$
$\boldsymbol{w} - \boldsymbol{S_0}$, $\boldsymbol{n}$
$\boldsymbol{u}$, $\boldsymbol{l}$
$\boldsymbol{z}$ (NCP)
$\boldsymbol{S}_0$
δ
$Vernal\ Equinox$ $(\alpha = 0)$
$\boldsymbol{y}$ $(\alpha = \pi)$
α
$\boldsymbol{x}$ $(east)$ $(\alpha = \pi/2)$

Figure 2: Coordinate system at midnight on autumnal equinox; all angles in radians. We represent Right Ascension by α and Declination by δ. In this diagram, the phase reference direction ($\boldsymbol{S_0}$) is $\alpha = \pi$, $\delta = \pi/4$, so $\boldsymbol{S_0}$ is coplanar with the $\boldsymbol{y} - \boldsymbol{z}$ plane. $\boldsymbol{w}$ and $\boldsymbol{n}$ always point in the direction of $\boldsymbol{S_0}$. $\boldsymbol{u}$ and $\boldsymbol{l}$ always point in the direction of increasing longitudinal coordinate (α). In this case, $\boldsymbol{u}$ and $\boldsymbol{l}$ point into the page. $\boldsymbol{v}$ and $\boldsymbol{m}$ point in the direction of increasing latitudinal coordinate (δ).

2.2. *The Radio Interferometry Measurement Equation*

The Radio Interferometry Measurement Equation (RIME) (Jagannathan 2018) is a model of the transformation of sky brightness into visibilities (V_{ij}) using a radio interferometer

$$\boldsymbol{V_{ij}} = M_{ij}^{DI} \int M_{ij}^{DD} \boldsymbol{I}(\boldsymbol{s}) e^{2\pi i \boldsymbol{s} \cdot \boldsymbol{b_{ij}}} \boldsymbol{ds}. \tag{2}$$

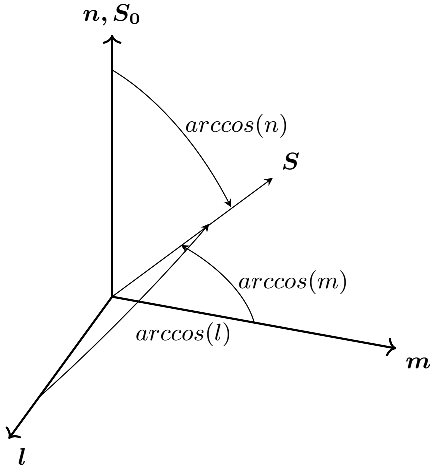

Figure 3: Directional cosines. (l, m, n) are measured along the axes (**l**, **m**, **n**) by taking the cosine of the direction angle from the axis to the source.

The sky brightness is represented by the four polarization sky vector $\boldsymbol{I}(\boldsymbol{s})$. The polarization basis used by interferometers is

$$[PP, PQ, QP, QQ], \tag{3}$$

where PP is the correlated value of the P polarization of antenna 1 with the P polarization of antenna 2, PQ is the correlated value of the P polarization of antenna 1 with the Q polarization of antenna 2, etc (Hamaker et al. 1996). P and Q can be either (X, Y) or (R, L). The four polarization sky vector can be transformed into the Stokes parameters basis using a linear transformation (Thompson et al. 2001); this is typically done for imaging algorithms such as Akiyama et al. (2017).

The RIME multiplies the sky brightness ($\boldsymbol{I}(\boldsymbol{s})$) with direction-dependent effects (M_{ij}^{DD}) of the interferometer's beam and the atmosphere while also performing a Fourier transform over the l and m coordinates ($\boldsymbol{ds} = \boldsymbol{dldm}$). The exponential term contains the dot product between the baseline vector $\boldsymbol{b_{ij}}$ and the source direction $\boldsymbol{s}$; the integral over the differential direction on the sky $\boldsymbol{ds}$ transforms the sky vector into a visibility vector. Non-direction-dependent effects (M_{ij}^{DI}) are then added after the Fourier transform. SiRIUS uses the Mueller matrix formulation of this equation, where M_{ij}^{DD} and M_{ij}^{DI} are Mueller matrices.

A Mueller matrix is obtained by taking the outer product of two Jones matrices; in interferometry, a Mueller matrix is obtained by taking the outer product of the Jones matrices for two antennae (Smirnov 2011). This Mueller matrix represents the transformation of the four polarization sky vector due to various effects, including the interferometer's beam and the atmosphere, and it can be direction-dependent or direction-independent (Jagannathan 2018). A direction-dependent Mueller matrix can be written in the visibility (u-v) plane or in the sky plane. Given a model in the u-v plane, the direction-dependent Mueller matrix in the sky plane M_{ij}^{DD} can be obtained from the Fourier transform of the direction-dependent Mueller matrix in the u-v plane A_{ij}^{DD}, also known as the Aperture Mueller matrix (Jagannathan 2018). The direction-dependent sky Mueller matrix can also be obtained from two Aperture Jones matrices by Fourier transforming both Aperture Jones matrices and then taking the outer product.

2.3. *Radio Interferometer Beam Models*

2.3.1. *Airy Disk*

A Jones matrix can be constructed to model an individual antenna's beam. The simplest model for an antenna is a circular aperture with a central circular blockage in the u-v plane, which becomes an obscured Airy pattern after undergoing a Fourier transform into the sky plane. The Airy disk is also known as the Bessel function of the first kind; SiRIUS uses this function to first order in order to create the Airy disk beam model for an individual antenna. The

Bessel function is centered on the antenna's phase center, and the value of the beam in the direction of the source is calculated analytically. In this model, it is assumed that there are no leakage terms (the Jones matrix is diagonal).

2.3.2. *Zernike Polynomials*

A more precise model for an antenna can be constructed from Zernike polynomials, which describe the shape of the dish as measured with holography. SiRIUS can calculate the Aperture Jones matrix corresponding to a Zernike polynomial model of a dish. This matrix is then Fourier transformed into a sky Jones matrix and sampled in the direction of the source, which is measured relative to the phase center of the antenna. Sampling is done by rotating the coordinates to compensate for parallactic rotation and then interpolating linearly to determine the Jones values at the exact direction of the point source. Compensating for parallactic rotation is necessary because the beam side-lobes are not spherically symmetric.

3. RESULTS

SiRIUS is currently able to simulate observations of point sources using any combination of beam models for the interferometer's antennae (heterogeneous or homogeneous arrays). Point sources are the simplest sources to simulate, since they allow for the integral in 2.2 to be evaluated analytically. Multiple point sources are treated as a sum of integrals, and multiple observations can be combined into a mosaic. Here we present a demonstration of SiRIUS's ability to simulate a mosaic with multiple point sources using various beam models for the interferometer.

The model sky we have chosen to simulate is a collection of point sources resembling the Pleiades star cluster (Figure 4); this is not an accurate representation of the radio sky in the direction of the Pleiades, but it is useful as an illustration of SiRIUS's capabilities. To image the star cluster, we simulate a 35 pointing mosaic made with the VLA in D configuration. The result of this simulation using Airy disk beam models for the antennae is shown in Figure 5. The top image is the dirty image (no deconvolution applied) generated by SiRIUS using the Airy disk beam models; the bottom image is the same image but generated by CASA instead. Both images are shown in the Stokes I polarization. The images created from SiRIUS and CASA visibilities are quite similar when using the same beam models; Figure 6 demonstrates that the difference between the two images tends to be on the order of 10^{-3} Jy, which is much smaller than the observations, which are on the order of 1 Jy. The main reason for these differences is due to a difference in the way the Airy pattern is sampled. Thus, SiRIUS is able to satisfactorily replicate CASA's component-list simulation of a mosaic of point sources using Airy beam models for the interferometer's antennae.

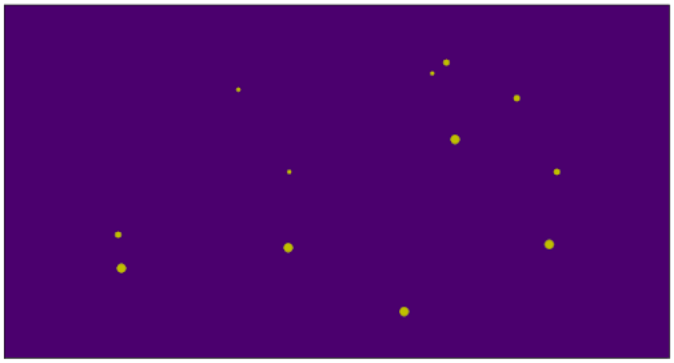

Figure 4: The sky model of the Pleiades Cluster used as a demonstration. The sizes of the point sources are scaled relative to their flux.

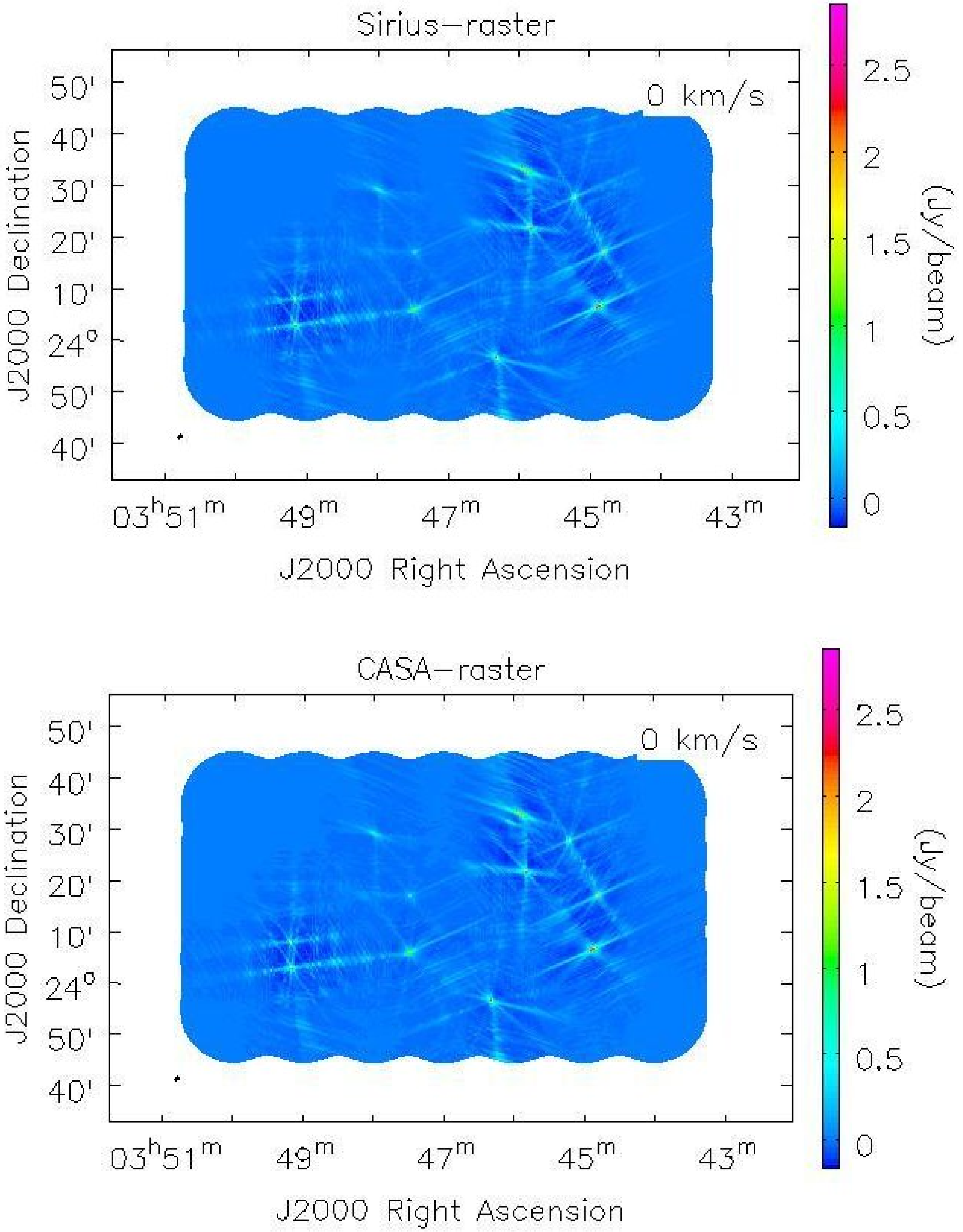

Figure 5: The dirty image of the Pleiades Cluster using Airy disk beams in SiRIUS (above) and CASA (bottom).

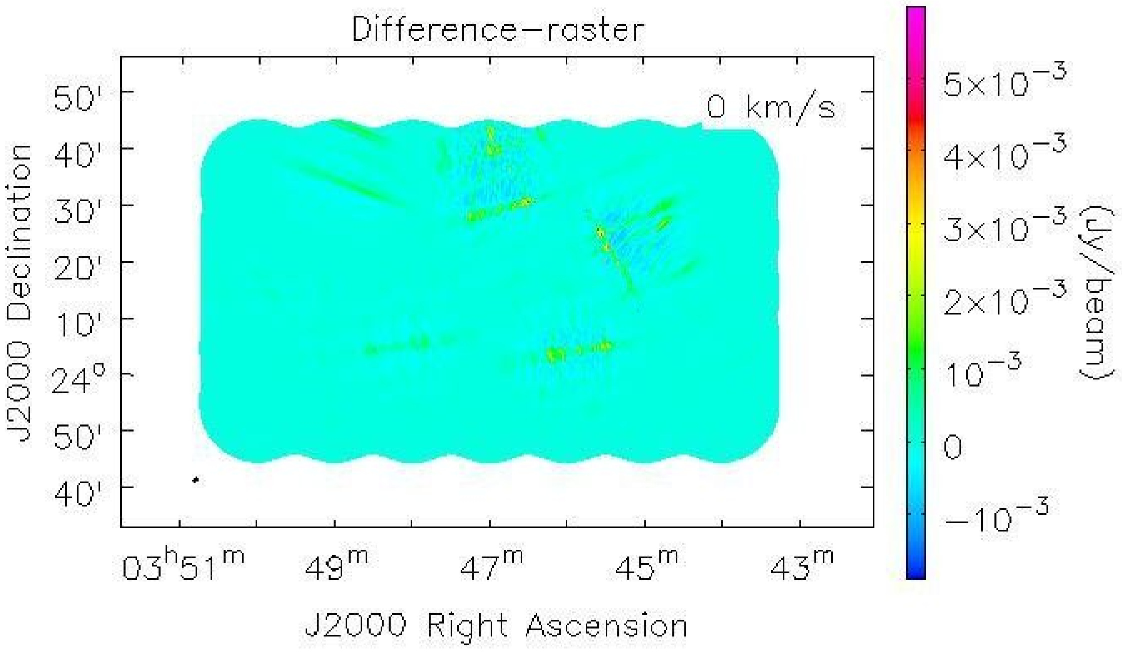

Figure 6: The difference between the dirty images using Airy beam models in both SiRIUS and CASA. The differences are reasonable small (on the order of 10^{-3} Jy).

In addition to the Airy disk beam model simulation, we perform the same simulation with SiRIUS's Zernike polynomial beam model. The Zernike polynomials for VLA antennae have been obtained from holography; the Mueller matrix for this model is shown in Figure 7. It is important to note that each of the images in Figure 7 has been normalized to one in order to make the off-diagonal components more easily visible since the off-diagonal components of the Mueller matrix are very small. The diagonal components of this matrix are similar to Airy disks to first order, but the off-diagonal components are a departure from the diagonal Airy disk model used in Figure 5. All of the beam components have asymmetries, which underscores the importance of incorporating parallactic rotation into the SiRIUS simulation.

The dirty image generated by SiRIUS using the Zernike polynomial model is shown in Figure 8. This image appears broadly similar to the other images produced by SiRIUS, but Figure 9 shows that the differences between SiRIUS's Zernike polynomial model and CASA's Airy disk model are an order of magnitude larger than the differences between SiRIUS and CASA when Airy disk models are used for both. This increase in the difference image underscores the importance of SiRIUS's full Mueller matrix simulation capabilities: the beam model in Figure 7 is much more realistic, and this improvement leads to a significant change in the simulated image.

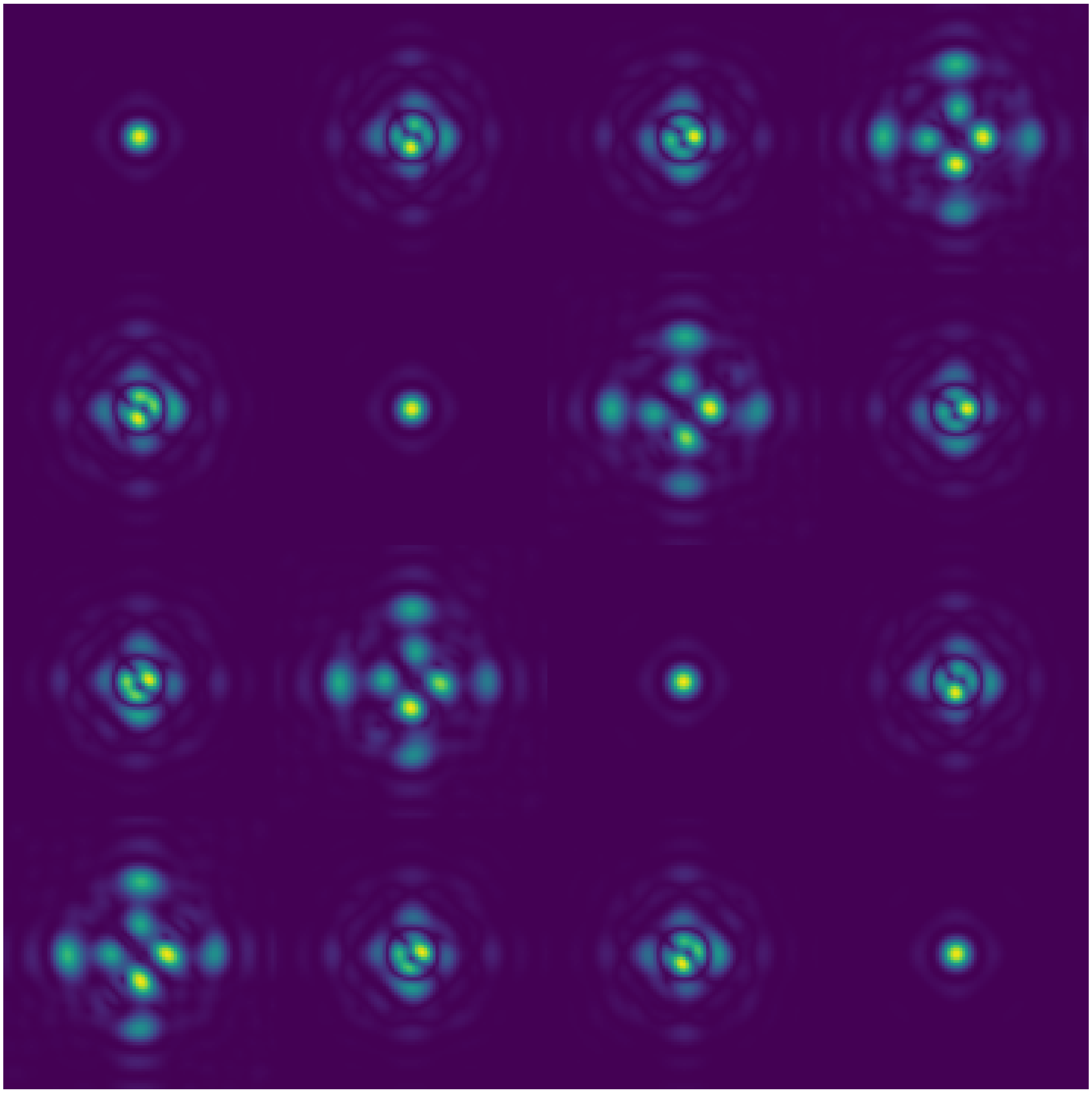

Figure 7: The Mueller matrix constructed from the Zernike polynomial model for a VLA dish. Each component of the matrix in this diagram is an image of that component of the beam across the entire sky. All of the images have been normalized to one to make the off-diagonal components more visible.

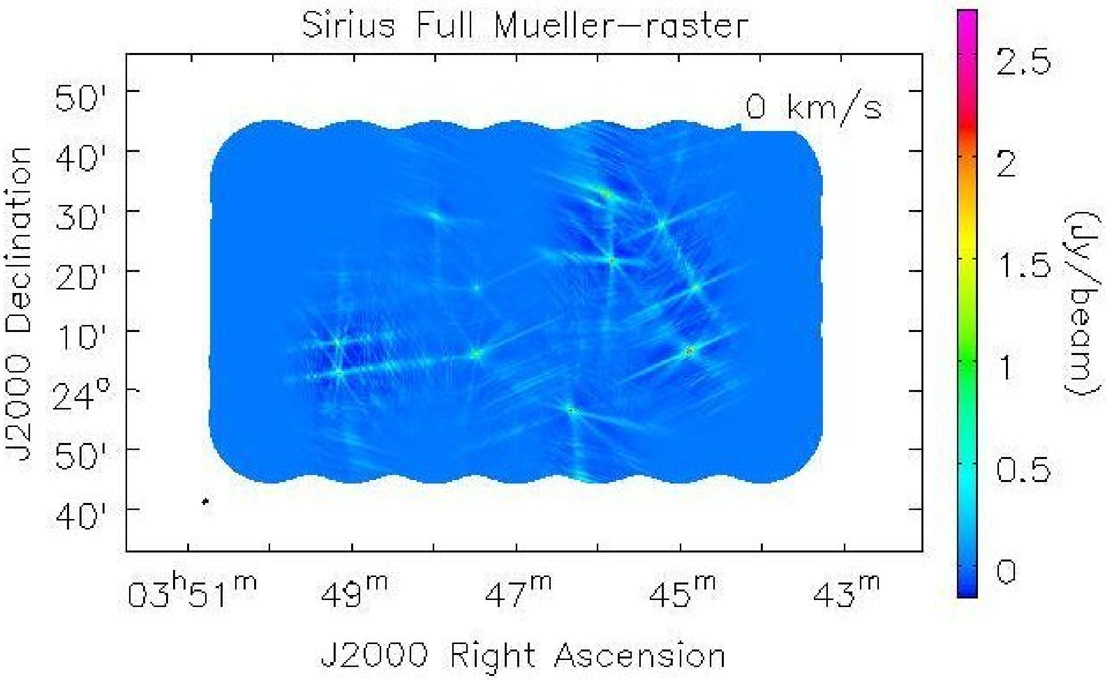

Figure 8: The dirty image of the Pleiades Cluster using Zernike polynomial beams in SiRIUS.

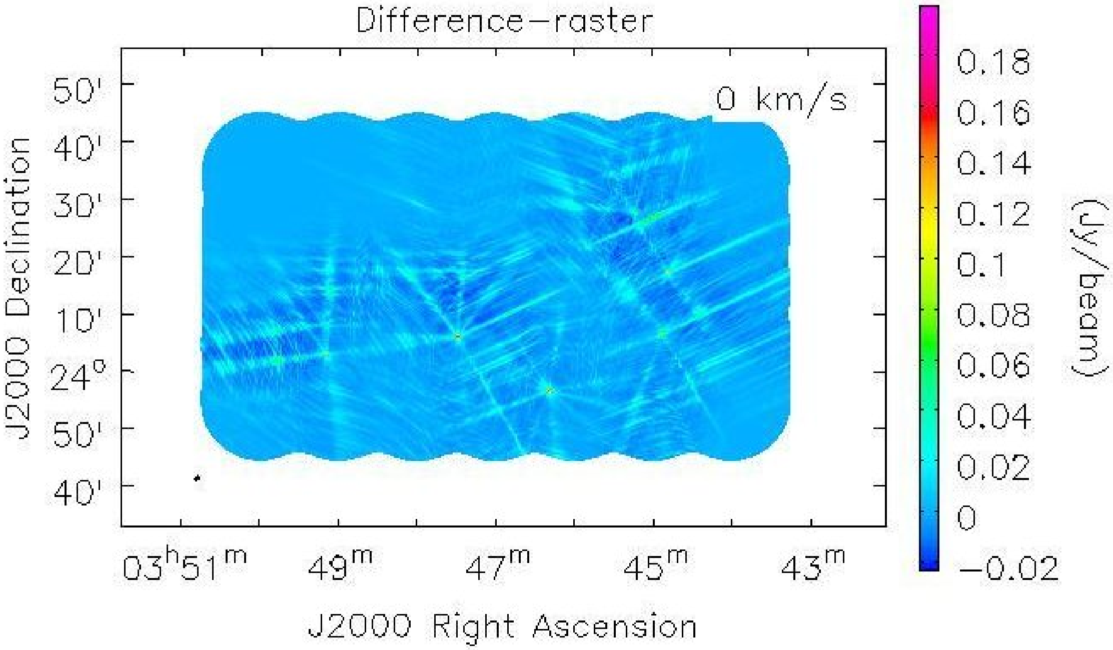

Figure 9: The difference between the dirty images using Zernike beam models in SiRIUS and Airy beam models in CASA. The differences are still fairly small (on the order of 10^{-2} Jy) but not as small as the difference between the two Airy beam models.

4. FUTURE WORK

The application-level interface is currently under construction; this interface will allow users to interact with SIRIUS on a high level. Several additional features for the SiRIUS simulation are planned or currently in testing. We are currently testing the functionality for pointing errors, which occur when interferometer dishes do not all point in precisely the same direction due to mechanical error. We are also performing verification tests on the parallactic rotation functionality as well as the heterogeneous array beam model (where the array's dishes can have different types of beam models). An important next step is to benchmark SiRIUS with large, computationally-intensive datasets such as those that will be generated by the ngVLA. Future work will also include the simulation of direction-independent atmospheric effects, the incorporation of bandwidth and movement smearing, and the addition of extended sources (Gaussians, disks, limb-darkened disks). Since SiRIUS is easily extensible, additional capabilities, such as VLBI, may be incorporated as needed.

5. CONCLUSION

The SiRIUS software package is capable of efficiently simulating the optical distortions introduced by interferometry. This simulation is performed using the Mueller Matrix formulation of the RIME. The current functionality of the simulation includes several interferometer beam models, mosaic and single-pointing observations, and multiple point sources. Additional features are currently being added to the simulation. Potential applications of this software include observing proposals for existing large arrays (VLA, ALMA, MeerKAT), validation of high-precision observations and algorithms (such as mosaic gridder), and planning the design of future large arrays such as the ngVLA. The extensibility of SiRIUS provides the flexibility to adapt to additional applications in the field of radio interferometry.

6. ACKNOWLEDGMENTS

M. Barth thanks J.-W. Steeb and A. McNichols for their kind and committed mentorship. We would like to acknowledge the National Radio Astronomy Observatory and the National Science Foundation's Research Experiences for Undergraduates program for supporting this work. The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

REFERENCES

Akiyama, K., Ikeda, S., Pleau, M., et al. 2017, AJ, 153, 159, doi: 10.3847/1538-3881/aa6302

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123, doi: 10.3847/1538-3881/aabc4f

CNGI Team. 2021, Development¶. https://cngi-prototype.readthedocs.io/en/latest/development.html

Hamaker, J. P., Bregman, J. D., & Sault, R. J. 1996, A&AS, 117, 137

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357, doi: 10.1038/s41586-020-2649-2

Jagannathan, P. 2018, PhD thesis. https://open.uct.ac.za/handle/11427/28421

Kemball, A., & Wieringa, M. 2001, MeasurementSet definition version 2.0. https://casacore.github.io/casacore-notes/229.html

Lam, S. K., Pitrou, A., & Seibert, S. 2015, in Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15 (New York, NY, USA: Association for Computing Machinery), doi: 10.1145/2833157.2833162

McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in Astronomical Society of the Pacific Conference Series, Vol. 376, Astronomical Data Analysis Software and Systems XVI, ed. R. A. Shaw, F. Hill, & D. J. Bell, 127

Okuta, R., Unno, Y., Nishino, D., Hido, S., & Crissman. 2017

Rocklin, M. 2015, in Proceedings of the 14th Python in Science Conference, ed. K. Huff & J. Bergstra, 130 – 136

Smirnov, O. M. 2011, A&A, 527, A106, doi: 10.1051/0004-6361/201016082

Thompson, A. R., Moran, J. M., & Swenson, George W., J. 2001, Interferometry and Synthesis in Radio Astronomy, 2nd Edition