

A CASA Python module to generate synthetic radio-astronomical image cubes.
Originally created for a study of interferometric imaging with CASA by L. Cortese, R. Galvan-Madrid, P. Klaassen, S. Longmore, and D. Petry in 2012.
Updated for use with CASA 6 by D. Petry in 2021.

The module gensynthcube enables you to combine various geometric spatial shapes (thick or thin disks or rings with an optional tilt, Gaussians, spheres) with different spectral shapes (continuum, powerlaw, line) and place them into a synthetic image cube as it is needed to create simulated ALMA datasets.

The design of the cube properties is done by filling in a number of Python dictionaries as demonstrated in the script gensynthcube-test.py .



gensynthcube.py

- the gensynthcube module



gensynthcube-test.py

- demo script

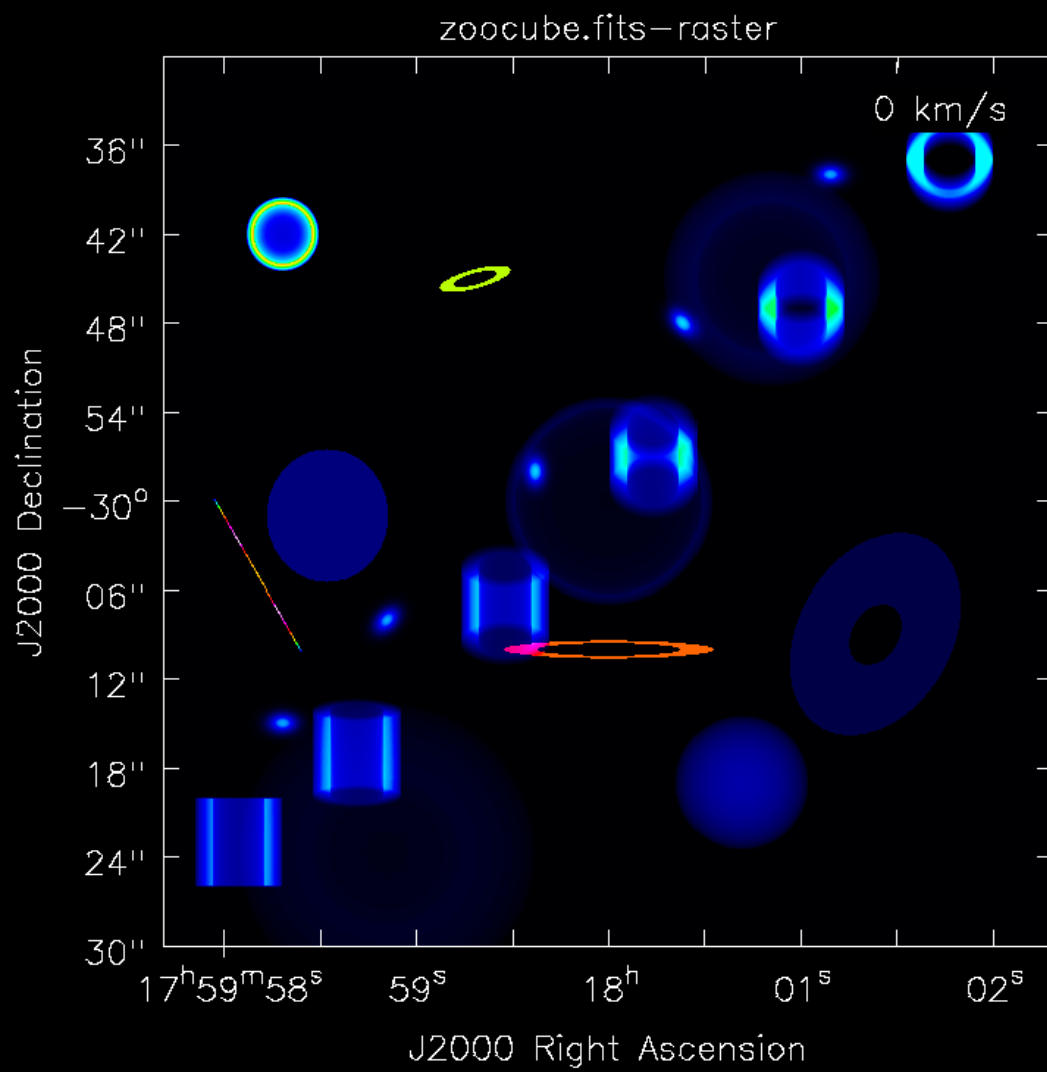
Demo script usage (run in CASA 6):

```
sys.path.append(<directory where you placed gensynthcube.py>)  
execfile('gensynthcube-test.py')
```

Will create the images "zoocube.im" and "zoocube.fits".



zoocube.fits



Sample dictionary content:

```

imdict
{ 'DEC': -30.0,
  'RA': 270.0,
  'channel_width_ghz': 0.015625,
  'freq_first_chan_ghz': 226.0,
  'num_freq_pixels': 1,
  'num_x_pixels': 600,
  'num_y_pixels': 600,
  'pixsize': 0.1,
  'restfreq_ghz': 226.0}

gaussiandict
{ 'arot': array([0.]),
  'integ_flux_first_plane': array([1.]),
  'line_peak_flux': array([0.]),
  'number': 0,
  'spectral_index': array([0.]),
  'title': '2D Gaussian',
  'x_center': array([300.]),
  'x_fwhm': array([20.]),
  'y_center': array([300.]),
  'y_fwhm': array([10.]),
  'z_center': array([0.]),
  'z_fwhm': array([1.])}

spheredict
{ 'integ_flux_first_plane': array([100.]),
  'line_peak_flux': array([0.]),
  'number': 0,
  'rmax': array([100.]),
  'spectral_index': array([0.]),
  'title': 'Spherical Shell',
  'x_center': array([300.]),
  'y_center': array([300.]),
  'z_center': array([0.]),
  'z_fwhm': array([1.]),
  'zeta': array([0.1])}

thindiskdict
{ 'arot': array([0.]),
  'atilt': array([45.]),
  'integ_flux_first_plane': array([100.]),
  'line_peak_flux': array([0.]),
  'number': 0,
  'rmax': array([100.]),
  'spectral_index': array([0.]),
  'title': 'Thin Disk',
  'x_center': array([300.]),
  'y_center': array([300.]),
  'z_center': array([0.]),
  'z_fwhm': array([1.]),
  'zeta': array([0.1])}

thickdiskdict
{ 'arot': array([0.]),
  'atilt': array([45.]),
  'd': array([10.]),
  'integ_flux_first_plane': array([100.]),
  'line_peak_flux': array([0.]),
  'number': 0,
  'rmax': array([100.]),
  'spectral_index': array([0.]),
  'title': 'Thick Disk',
  'x_center': array([300.]),
  'y_center': array([300.]),
  'z_center': array([0.]),
  'z_fwhm': array([1.]),
  'zeta': array([0.5])}

```

Set 'number' to the desired number of elements of the type.

'title' must be different in each dictionary.

The z coordinate is the spectral coordinate.

Coordinates are in pixels or channels resp., arot and atilt are in degrees,

pixsize is in arcsecs, fluxes are in Jy.