MomsData_Demo

November 22, 2018

1 Data analysis demo for H-core-M25 stellar hydro project

Last update: Nov 16, 2018.

This notebook contains a demonstration how to analyse the 3D filtered *moms* data and the 1D radial profile *rprof* data from *PPMstar* 3D hydrodynamic simulations.

1.0.1 Data for this demo

The examples are for the project `H-core-M25` (this is the project identifier), the H-core convection simulations of a $25\mathrm{M}_\odot$ star.

Two runs are used: * `M29`: 768^3 grid * `M35`: 1536^3 grid

`M29`, `M35` are the run identifier. Keep run and project identifier attached to all derived data products.

Both runs have 1000x heating which increases their convective velocities by a factor of 10.

For each run there are two types of data to be read for this demo: * *moms* data is the spatially filtered data (2-byte data on reduced grid by factor four in each direction) in 3D * *rprofs* data are spherically averaged radial profiles

1.0.2 Location of data

The data is staged on the UVic Astrophysics Simulation Data Repository (ASDR) mounted in `/data/ASDR`. The repository contains the project folder `H-core-M25`.

1.0.3 Python asumptions

The server defaults each notebook to `%pylab ipympl`

```
In [46]: ## use this for final run to export with images to pdf, markdown or html
         %pylab inline
```

```
DEBUG:matplotlib.pyplot:Loaded backend module://ipykernel.pylab.backend_inline version unknown


Populating the interactive namespace from numpy and matplotlib


/usr/local/lib/python3.6/dist-packages/IPython/core/magics/pylab.py:160: UserWarning: pylab im
`%matplotlib` prevents importing * from pylab and numpy
  "\n`%matplotlib` prevents importing * from pylab and numpy"
```

```
In [47]: import numpy as np
         import matplotlib as mpl
         import matplotlib.pyplot as plt
         import matplotlib.colors as color
         import nugridpy.utils as utils
         import sys, os, time

         # if you make changes to the ppmpy module (e.g. add your analysis methods via a pull
         # request) in the https://github.com/PPMstar/PyPPM repo you may want use that
         # updated version
         #sys.path.insert(0,'/user/david/PyPPM/')
         #sys.path.insert(0,'/user/PyPPM/')
         from ppmpy import ppm

         cb = utils.linestylecb # colours
```

```
In [48]: %%bash
         ls /data/ASDR/H-core-M25/
```

```
M29-768
M35-1536
```

```
In [49]: dir_repo    = '/data/ASDR'
         dir_project = 'H-core-M25'
         rprof = {}; moms = {}         # initialize dictionaries to hold rprof and moms instan

         moms_dumps = {}

         runs                = ['M29-768'] # select runs
         moms_dumps[runs[0]] = 650  # select dump numbers for moms

         add_highres = False
         if add_highres:
             runs.append('M35-1536')
             moms_dumps[runs[1]] = 375

         # rprof instance holds radial profiles for all dumps
         # moms instance holds only one dump at a time
         for run in runs:
             path = os.path.join(dir_repo,dir_project,run)
             # radial profile:
             rprof[run] = ppm.RprofSet(os.path.join(path,'rprofs'))
             moms[run] = ppm.MomsDataSet(os.path.join(path,'myavsbq'),moms_dumps[run])
         print("moms and rprof dictionary created")
```

```
748 rprof files found in '/data/ASDR/H-core-M25/M29-768/rprofs/.
Dump numbers range from 0 to 747.
Reading history file '/data/ASDR/H-core-M25/M29-768/rprofs/HcoreE00768-0000.hstry'.
```

```
748 .aaa files found in '/data/ASDR/H-core-M25/M29-768/myavsbq/.
Dump numbers range from 0 to 747.
The PPMstar grid is being constructed, this can take a moment
moms and rprof dictionary created
```

```
In [50]: runid = 'M29-768'    # select run id for the rest of the notebook
         #runid = 'M35-1536'
```

```
In [51]: # get info about moms instance
         # help(moms[runid])
```

1.1 Basic grid properties

```
In [52]: x,y,z,r=moms[runid].get_grid()
```

```
In [53]: print(192**3,len(r))
```

```
7077888 7077888
```

```
In [54]: print("Distance center of grid to max x value of domain: %6.4f Mm" % moms[runid].get_
```

```
Distance center of grid to max x value of domain: 2486.9792 Mm
```

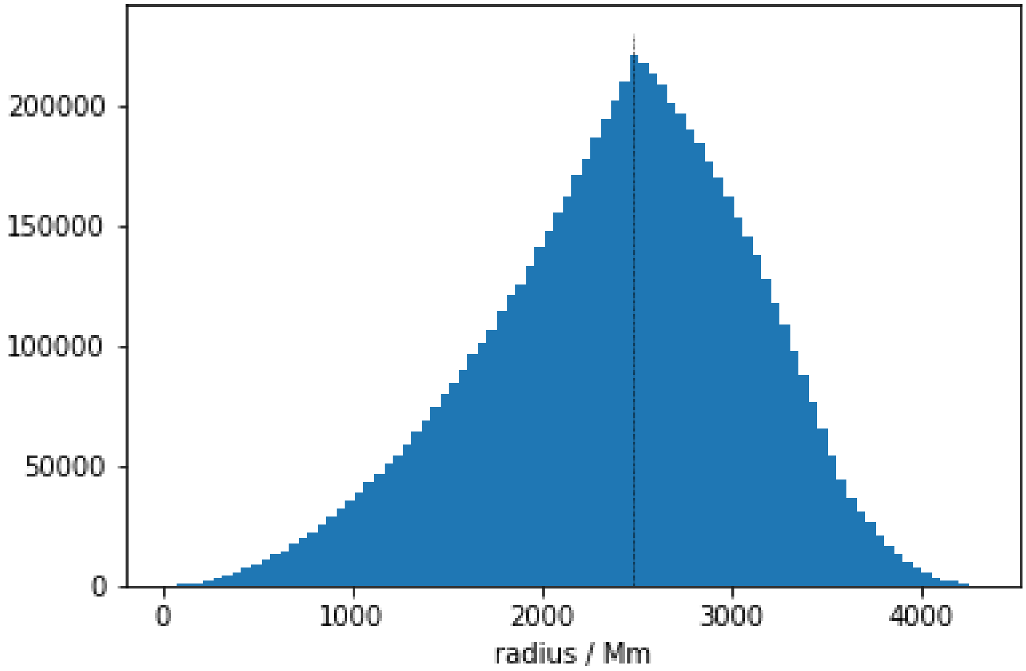
1.1.1 Histogram of radii

- increasing to 1/2 length of grid, then decreasing as only fraction of shell in box

```
In [55]: ifig=0
         ifig += 1; plt.close(ifig);  plt.figure(ifig)
         hist(r,86)
         xmax = moms[runid].get_grid()[0][-1]
         vlines(xmax,0,2.3e5,linestyles='--',lw=0.5)
         xlabel('radius / Mm')
         ylabel('')
```

```
Out[55]: Text(0, 0.5, '')
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



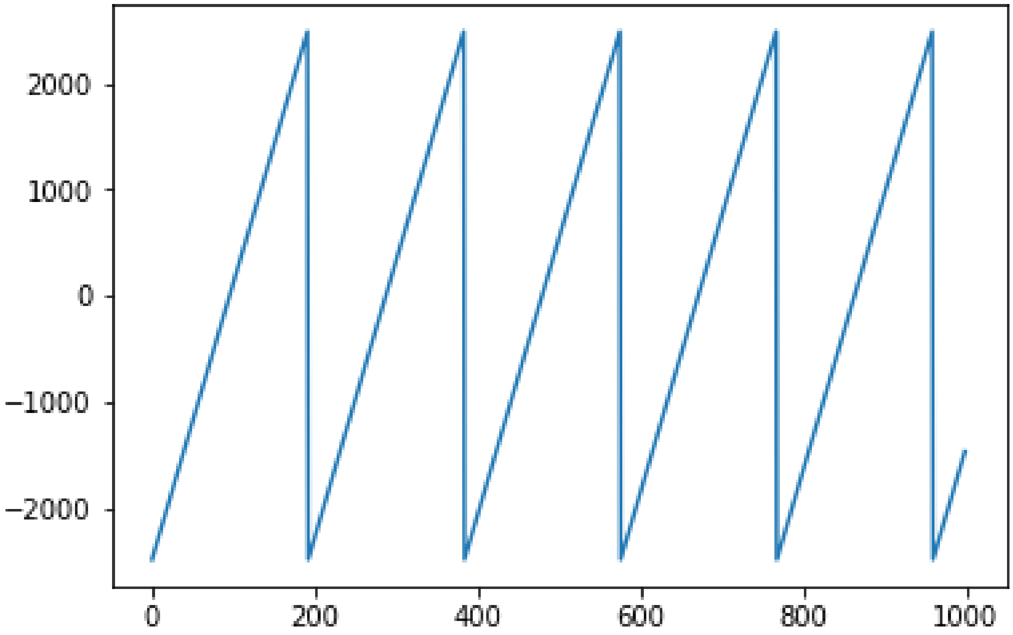
1.1.2 Some more experiments with coordinates

```
In [56]: xx=reshape(x,[192,192,192])
```

```
In [57]: ifig += 1; plt.close(ifig);  plt.figure(ifig)
         #plot(xx[0][0])
         plot(x[0:1000])
```

```
Out[57]: [<matplotlib.lines.Line2D at 0x7fce66ff10b8>]
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



1.1.3 Planar slice image

```
In [58]: # grab the grid
         x,y,z,r = moms[runid].get_grid()

         # they are flattened arrays, rearrange
         resolution = moms[runid].momsdata.resolution
         r_matrix = np.reshape(r,(resolution,resolution,resolution))

         # extent x,y
         extent=[min(x),max(x),min(y),max(y)]

         # slice number
         slice_num = int(resolution/2) # central slice
```

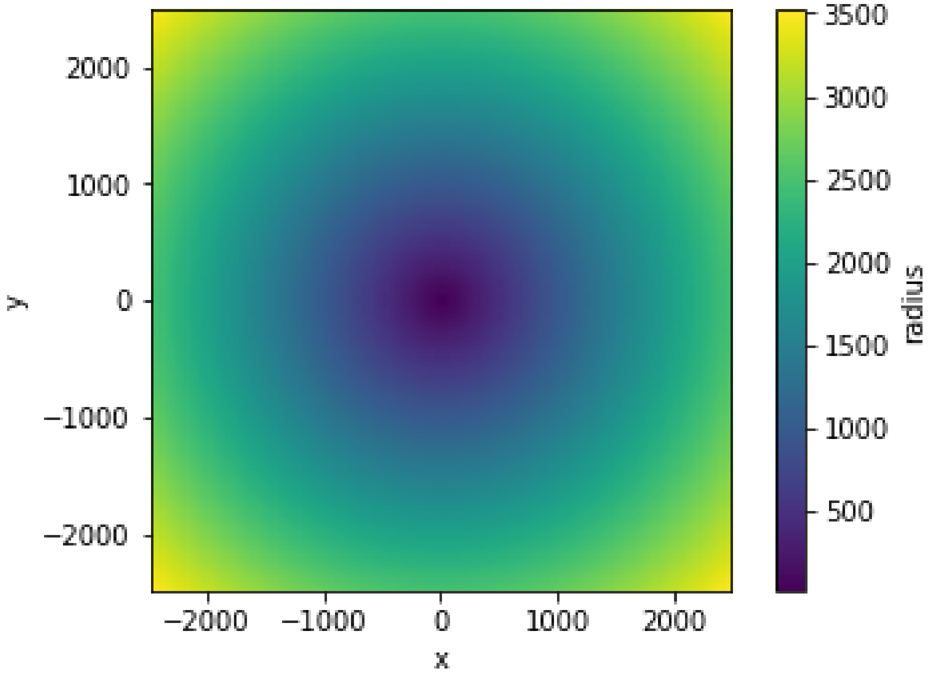
Radius

```
In [59]: ifig += 1; plt.close(ifig); plt.figure(ifig)
         plt.imshow(r_matrix[:][:][slice_num],extent=extent)
         plt.ylabel('y')
         plt.xlabel('x')
         cbar = plt.colorbar()

         # label colorbar
         cbar.ax.set_ylabel('radius')
```

```
DEBUG:matplotlib.colorbar:locator: <matplotlib.colorbar._ColorbarAutoLocator object at 0x7fce8c
DEBUG:matplotlib.colorbar:Using auto colorbar locator on colorbar
DEBUG:matplotlib.colorbar:locator: <matplotlib.colorbar._ColorbarAutoLocator object at 0x7fce8c
DEBUG:matplotlib.colorbar:Setting pcolormesh
```

```
Out[59]: Text(0, 0.5, 'radius')
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



1.1.4 Properties of the grid and time resolution

```
In [60]: resolution*4
```

```
Out[60]: 768
```

```
In [61]: # spatial resolution for momsdata
         print('The spatial resolution of 768 momsdata is {:0.3f}'.format(np.diff(moms[runid].g
         print('While PPMStar 768 has a spatial resolution of {:0.3f}'.format(np.diff(moms[runi

         print('')

         # what is the extent of the simulation?
         print('The extent of the simulation is then {:0.0f}'.format(np.diff(moms[runid].get_g
```

```
The spatial resolution of 768 momsdata is 26.042 Mm
While PPMStar 768 has a spatial resolution of 6.510 Mm

The extent of the simulation is then 2500 Mm
```

```
In [62]: #  for 768 momsdata
         print('The temporal resolution of momsdata is the same as the PPMStar output which ave
               format(np.mean(np.diff(rprof[runid].get_history().get('time(mins)'))),'minutes

         print('')

         print('The run-time temporal resolution of the PPMStar output averages around {:0.2f}
               format(np.mean(rprof[runid].get_history().get('dt(secs)')),'seconds per cycle')
```

```
The temporal resolution of momsdata is the same as the PPMStar output which averages around 16

The run-time temporal resolution of the PPMStar output averages around 5.47 seconds per cycle
```

1.2 Find Times For Dumps

As hinted at in the above section, there is a history file that gives us information about the run. This is located in the rprof files themselves

```
In [63]: # get the simulation time in seconds for dump 100
         print('{:0.1f} seconds '.format(rprof[runid].get_history().get('time(secs)')[moms_dump
         'have passed since the simulation started for the '+runid+' run')
```

```
6507520.0 seconds have passed since the simulation started for the M29-768 run
```

1.3 What quantities have what index?

The following quantities written into the moms data file which can be called with an index:

index	quantity
0	x
1	$\vec{u}_x$

index	quantity
2	$\vec{u}_y$
3	$\vec{u}_z$
4	$\lvert\vec{u}_t\rvert$
5	$\lvert\vec{u}_r\rvert$
6	$\lvert\vec{\omega}\rvert$
7	P
8	rho
9	fv

- Note that these are just 10 out of 32 quantities that can be made available in the moms data.
- fv is the fractional volume of the material initially only outside the convection zone.

Some Helpful Definitions μ = fv × 0.617 + (1 - fv) × 0.669

$T = \frac{P\mu}{\rho R_{gas}}$

$R_{gas} = 8.314462$

$\vec{\omega} = \vec{\nabla} \times \vec{u}$

1.4 Radial profiles

Radial profiles can be taken from the *rprof* data sets. They can also be constructed from the *moms* data. This is demonstrated below.

```
In [64]: # define variables for dump number, rprof and moms
         thisdump = moms_dumps[runid]
         thisrprof = rprof[runid]
         thismoms = moms[runid]

         # get T9 and Ut
         P_rprof = thisrprof.get('P0',fname=thisdump,resolution='h')[0::2] + thisrprof.get('P1
         rho_rprof = thisrprof.get('Rho0',fname=thisdump,resolution='h')[0::2] + thisrprof.get
         FV_rprof = thisrprof.get('FV',fname=thisdump,resolution='h')[0::2]

         # T9 in rprof class is not correct, calculate directly
         T9_rprof = P_rprof * (0.617*FV_rprof + 0.669*(1-FV_rprof)) / (8.314462 * rho_rprof)

         R_rprof = thisrprof.get('R',fname=thisdump,resolution='l')
         Ut_rprof = thisrprof.get('|Ut|',fname=thisdump)
```

```
In [65]: # make an rprof of temperature and ut
         ut_avg, radial_axis = thismoms.get_rprof(4,thisdump)

         # first we need to construct T from quantities
         mu = 0.617 * thismoms.get(9,fname=thisdump) + (1 - thismoms.get(9,fname=thisdump))*0.
         P = thismoms.get(7,fname=thisdump)
         rho = thismoms.get(8,fname=thisdump)
         Rgas = 8.314462
```

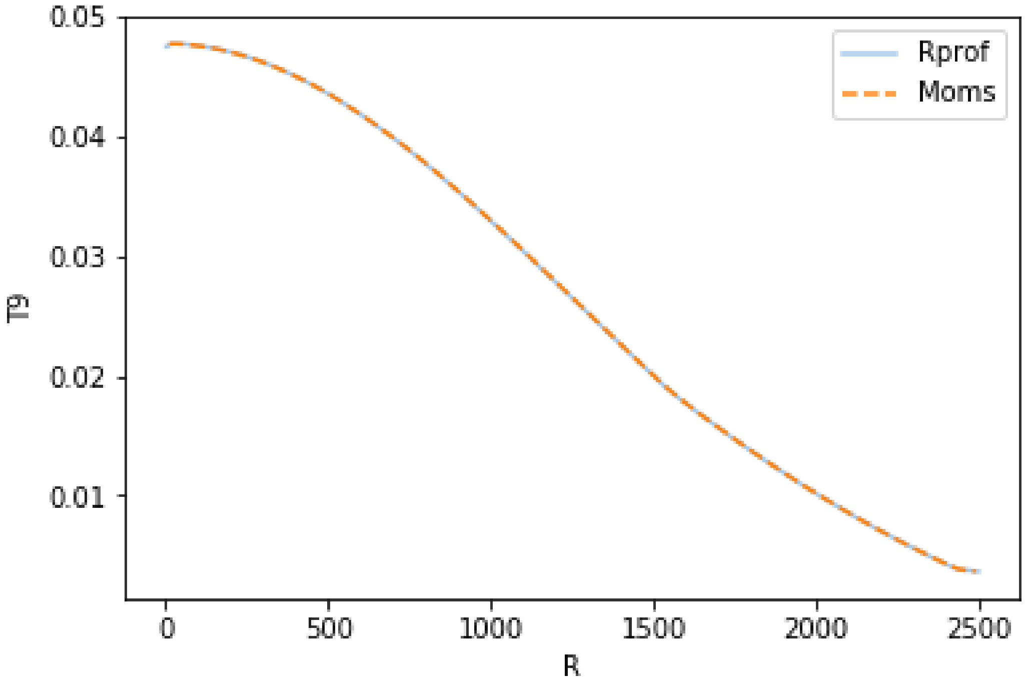
```
        # put it all together
        T = (mu * P) / (Rgas * rho)

        # we can give the rprof method an array to be spherically averaged
        T_avg, radial_axis = thismoms.get_rprof(T,thisdump)
```

```
/usr/local/lib/python3.6/dist-packages/scipy/stats/_binned_statistic.py:607: FutureWarning: Us
  result = result[core]
```

```
In [66]: # plot
         ifig += 1; plt.close(ifig);  plt.figure(ifig)

         plt.plot(R_rprof,T9_rprof,label='Rprof',ls=cb(0)[0],color=cb(0)[2])
         plt.plot(radial_axis,T_avg,label='Moms',ls=cb(1)[0],color=cb(1)[2])
         plt.xlabel('R')
         plt.ylabel('T9')

         plt.legend()
```

```
Out[66]: <matplotlib.legend.Legend at 0x7fce80609438>
```

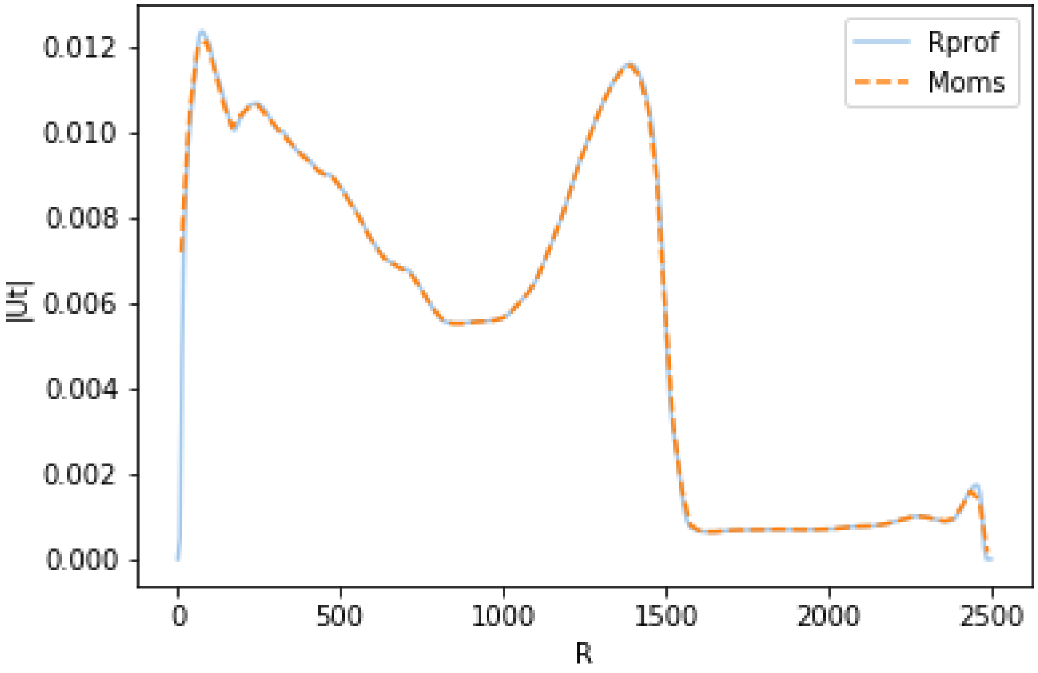
```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



```
In [67]: # plot
         ifig += 1; plt.close(ifig);  plt.figure(ifig)

         plt.plot(R_rprof,Ut_rprof,label='Rprof',ls=cb(0)[0],color=cb(0)[2])
         plt.plot(radial_axis,ut_avg,label='Moms',ls=cb(1)[0],color=cb(1)[2])

         plt.xlabel('R')
         plt.ylabel('|Ut|')

         plt.legend()
```

```
Out[67]: <matplotlib.legend.Legend at 0x7fce805ab4e0>
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



1.4.1 Planar Slice Image

```
In [68]: x,y,z,r = moms[runid].get_grid()
```

```
         # they are flattened arrays, rearrange
         resolution = moms[runid].momsdata.resolution
         r_matrix = np.reshape(r,(resolution,resolution,resolution))

         # extent x,y
         extent=[min(x),max(x),min(y),max(y)]

         # slice number
         slice_num = int(resolution/2)
```

T9

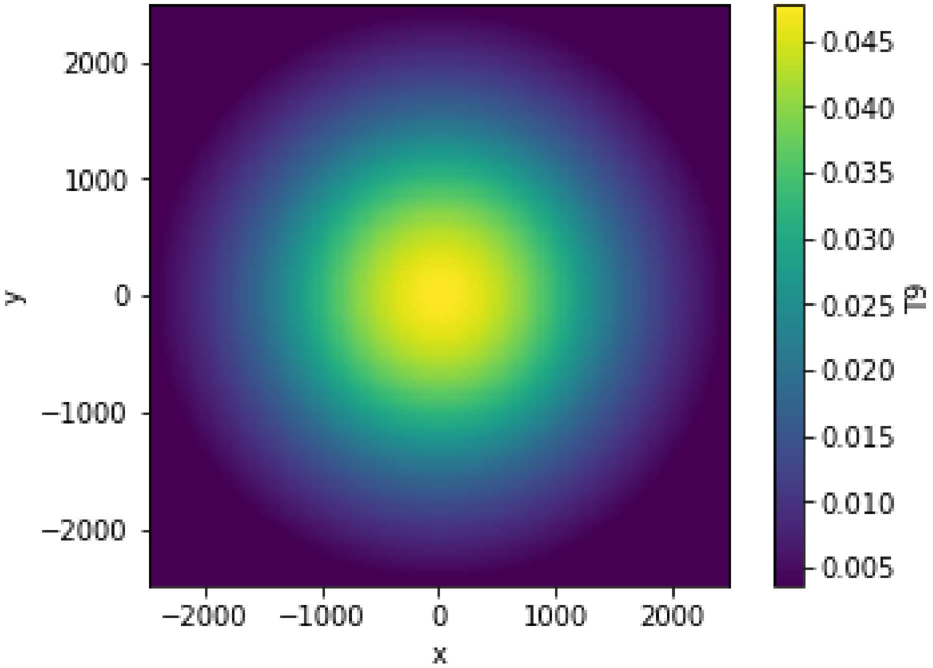
```
In [69]: T_matrix = np.reshape(T,(resolution,resolution,resolution))
```

```
In [70]: ifig += 1; plt.close(ifig);  plt.figure(ifig)

         plt.imshow(T_matrix[:][:][slice_num],extent=extent)
         plt.ylabel('y')
         plt.xlabel('x')
         cbar = plt.colorbar()

         # label colorbar
         cbar.ax.set_ylabel('T9')
```

```
DEBUG:matplotlib.colorbar:locator: <matplotlib.colorbar._ColorbarAutoLocator object at 0x7fce8
DEBUG:matplotlib.colorbar:Using auto colorbar locator on colorbar
DEBUG:matplotlib.colorbar:locator: <matplotlib.colorbar._ColorbarAutoLocator object at 0x7fce8
DEBUG:matplotlib.colorbar:Setting pcolormesh
```

```
Out[70]: Text(0, 0.5, 'T9')
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



|Ut|

```
In [71]: ut_matrix = np.reshape(thismoms.get(4,thisdump),(resolution,resolution,resolution))
```

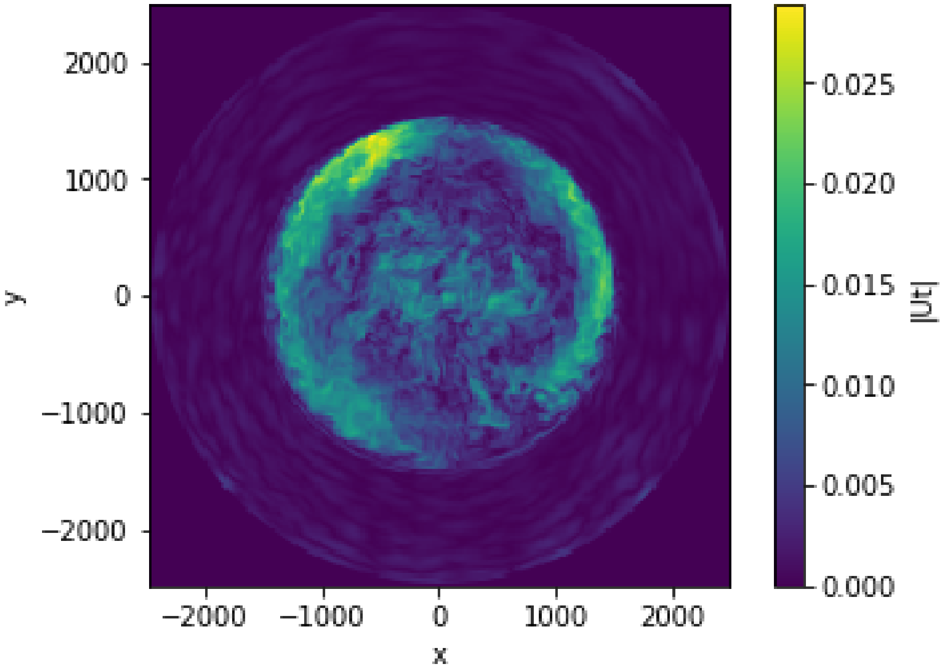
```
In [72]: ifig += 1; plt.close(ifig); plt.figure(ifig)
         plt.imshow(ut_matrix[:][:][slice_num],extent=extent)
         plt.ylabel('y')
         plt.xlabel('x')
         cbar = plt.colorbar()

         # label colorbar
         cbar.ax.set_ylabel('|Ut|')
```

```
DEBUG:matplotlib.colorbar:locator: <matplotlib.colorbar._ColorbarAutoLocator object at 0x7fce8
DEBUG:matplotlib.colorbar:Using auto colorbar locator on colorbar
DEBUG:matplotlib.colorbar:locator: <matplotlib.colorbar._ColorbarAutoLocator object at 0x7fce8
DEBUG:matplotlib.colorbar:Setting pcolormesh
```

```
Out[72]: Text(0, 0.5, '|Ut|')
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```



1.5 FV Colourmap of a Plane (x=y=0)

```
In [73]: # x=y=0, a particular z slice. convert to an 8-bit number
         fv = np.reshape(thismoms.get(9,thisdump),(resolution,resolution,resolution))
         fv_bit = 251 + 13.35455532 * np.log(fv[:][:][96])
```

```
In [74]: FV_cmap_str = '''
         Anot: 0 0.0
         Anot: 18 0.1058824
         Anot: 56 0.2745098
         Anot: 75 0.7843137
         Anot: 123 1.0
         Anot: 158 1.0
         Anot: 184 0.5490196
         Anot: 203 0.454902
         Anot: 255 0.1254902
         Cnot: 0 0.0 0.0 0.0
```

```
Cnot: 48 0.0 0.0 0.2509804
Cnot: 56 0.0 0.2352941 0.627451
Cnot: 65 0.0 0.7843137 1.0
Cnot: 75 1.0 1.0 1.0
Cnot: 100 1.0 1.0 0.0
Cnot: 186 1.0 0.0 0.0
Cnot: 244 0.5019608 0.0 0.0
Cnot: 255 0.5019608 0.0 0.0
'''
cmap = ppm.colourmap_from_str(FV_cmap_str, segment=(5, 251))

# normalize to our 255 bit range
norm = mpl.colors.Normalize(vmin=5, vmax=251)
```

Square Image

```
In [75]: my_dpi = 300
         ifig+=1; plt.close(ifig); plt.figure(ifig,figsize=(536/my_dpi, 536/my_dpi), dpi=my_dp
         x,y,z,r = thismoms.get_grid()
         plt.pcolor(np.unique(x),np.unique(y),fv_bit,cmap=cmap,norm=norm)

         plt.axis('off')
```

```
Out[75]: (-2486.97900390625, 2486.979248046875, -2486.97900390625, 2486.979248046875)
```

```
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
DEBUG:matplotlib.axes._base:update_title_pos
```