

Figure 23. from

galpy: A python Library for Galactic Dynamics

Bovy 2015 ApJS 216 29 doi:10.1088/0067-0049/216/2/29

<https://dx.doi.org/10.1088/0067-0049/216/2/29>

© 2015. The American Astronomical Society. All rights reserved.

```
1 from galpy.df import dehnendf
2 # Init. dehnendf w/ flat rot., hr=1/3, hs=1, and sr(1)=0.2
3 df= dehnendf(beta=0.,profileParams=(1./3.,1.0,0.2))
4 # Same, w/ correction factors to scale profiles
5 dfc= dehnendf(beta=0.,profileParams=(1./3.,1.0,0.2),
6         correct=True,niter=20)
7 # Log. diff. between scale and DF surf. dens.
8 numpy.log(df.surfacemass(0.5)/df.targetSurfacemass(0.5))
9 -0.056954077791649592
10 # Same for corrected DF
11 numpy.log(dfc.surfacemass(0.5)/dfc.targetSurfacemass(0.5))
12 -4.1440377205802041e-06
13 # Log. diff between scale and DF sr
14 numpy.log(df.sigmaR2(0.5)/df.targetSigma2(0.5))
15 -0.12786083001363127
16 # Same for corrected DF
17 numpy.log(dfc.sigmaR2(0.5)/dfc.targetSigma2(0.5))
18 -6.8065001252214986e-06
19 # Evaluate DF w/ R, vR, vT
20 df(numpy.array([0.9,0.1,0.8]))
21 array(0.1740247246180417)
22 # Evaluate corrected DF w/ Orbit instance
23 from galpy.orbit import Orbit
24 dfc(Orbit([0.9,0.1,0.8]))
25 array(0.16834863725552207)
26 # Calculate the mean velocities
27 df.meanvR(0.9), df.meanvT(0.9)
28 (0.0, 0.91144428051168291)
29 # Calculate the velocity dispersions
30 numpy.sqrt(dfc.sigmaR2(0.9)), numpy.sqrt(dfc.sigmaT2(0.9))
31 (0.22103383792719539, 0.17613725303902811)
32 # Calculate the skew of the velocity distribution
33 df.skewvR(0.9), df.skewvT(0.9)
34 (0.0, -0.47331638366025863)
35 # Calculate the kurtosis of the velocity distribution
36 df.kurtosisvR(0.9), df.kurtosisvT(0.9)
37 (-0.13561300880237059, 0.12612702099300721)
38 # Calculate a higher-order moment of the velocity DF
39 df.vmomentsurfacemass(1.,6.,2.)/df.surfacemass(1.)
40 0.00048953492205559054
41 # Calculate the Oort functions
42 dfc.oortA(1.), dfc.oortB(1.), dfc.oortC(1.), dfc.oortK(1.)
43 (0.40958989067012197, -0.49396172114486514, 0.0, 0.0)
44 # Sample Orbits from the DF, returns list of Orbits
45 os= dfc.sample(n=100,returnOrbit=True,nphi=1)
46 # check that these have the right mean radius = 2hr=2/3
47 rs= numpy.array([o.R() for o in os])
48 assert numpy.fabs(numpy.mean(rs)-2./3.) < 0.1
49 # Sample vR and vT at given R, check their mean
50 vrvt= dfc.sampleVRVT(0.7,n=500,target=True); vt= vrvt[:,1]
51 assert numpy.fabs(numpy.mean(vrvt[:,0])) < 0.05
52 assert numpy.fabs(numpy.mean(vt)-dfc.meanvT(0.7)) < 0.01
53 # Sample Orbits along a given line-of-sight
54 os= dfc.sampleLOS(45.,n=1000)
```