

Figure 5. 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.

C code:

```
1 # LogarithmicHaloPotential.c
2 #include <math.h>
3 #include <galpy_potentials.h>
4 //3 arguments: amp, c2, and q
5 double LogarithmicHaloPotentialEval(double R,double Z, double phi,
6                                     double t,
7                                     struct potentialArg * potentialArgs){
8     double * args= potentialArgs->args;
9     double amp= *args;
10    double q= *(args+1);
11    double c2= *(args+2);
12    double zq= Z/q;
13    return 0.5 * amp * log(R*R+zq*zq+c2);
14 }
15 double LogarithmicHaloPotentialRforce(double R,double Z, double phi,
16                                       double t,
17                                       struct potentialArg * potentialArgs){
18     double * args= potentialArgs->args;
19     double amp= *args++;
20     double q= *args++;
21     double c2= *args--;
22     double zq= Z/q;
23     return - amp * R/(R*R+zq*zq+c2);
24 }
25 double LogarithmicHaloPotentialzforce(double R,double z,double phi,
26                                       double t,
27                                       struct potentialArg * potentialArgs){
28     double * args= potentialArgs->args;
29     double amp= *args++;
30     double q= *args++;
31     double c2= *args--;
32     double zq= z/q;
33     return -amp * z/q/q/(R*R+zq*zq+c2);
34 }
35 ...
36 # In galpy/potential_src/potential_c_ext/galpy_potentials.h
37 ...
38 //LogarithmicHaloPotential
39 double LogarithmicHaloPotentialEval(double ,double , double, double,
40                                     struct potentialArg *);
41 double LogarithmicHaloPotentialRforce(double ,double , double, double,
42                                     struct potentialArg *);
43 double LogarithmicHaloPotentialzforce(double,double,double,double,
44                                     struct potentialArg *);
45 ...
46 # In galpy/orbit_src/orbit_c_ext/integrateFullOrbit.c's parse_leapFuncArgs_Full
47 ...
48 case 0: //LogarithmicHaloPotential, 3 arguments, unique case integer should match the integer in line 64 below
49     potentialArgs->Rforce= &LogarithmicHaloPotentialRforce;
50     potentialArgs->zforce= &LogarithmicHaloPotentialzforce;
51     potentialArgs->phiforce= &ZeroForce;
52     potentialArgs->nargs= 3;
53     break;
54 ...
```

python code:

```
61 # In galpy/orbit_src/integrateFullOrbit.py's _parse_pot
62 ...
63 if isinstance(p,potential.LogarithmicHaloPotential):
64     pot_type.append(0) # See line 48 above
65     pot_args.extend([p._amp,p._q,p._core2])
66 ...
67 # In galpy/potential_src/LogarithmicHaloPotential.py's __init__
68 ...
69 self.hasC= True
```