

Documentation - Computational Approximation of the Gravitational Potential Due to Caustic Ring Dark Matter on NEMO

Introduction:

One of the great unsolved mysteries of modern physics is the story behind dark matter. Thus far, dark matter has only been observed through its large-scale gravitational interactions with light matter. At first this seems to pose a huge problem for determining dark matter's identity since two leading candidates, axions and WIMPs, though fundamentally different particles, behave very similarly at large scales. However, as pointed out by Pierre Sikivie's theory, it may be possible to break this behavioral symmetry though dark matter infall into galaxies thanks to the rotation of their velocity fields. In the case of axions, dark matter falls in with a net overall rotation and is predicted to form caustic ring density patterns, while the case of WIMP dark matter is predicted to produce tent-like density structures. The provided caustics.c potential program makes the assumption that dark matter is composed of axions, and estimates the caustic rings' gravitational influence for NEMO simulations. For a far more in-depth explanation of Sikivie's caustic ring theory, the following papers may be very useful:

Caustic rings of dark matter: <http://arxiv.org/abs/astro-ph/9705038>

Axion BEC Dark Matter: <http://arxiv.org/abs/1111.3976>

An argument that the dark matter is axions : <http://arxiv.org/abs/1210.0040>

Installation:

The following instructions have been found to work for Linux Mint 14.

First, dependencies must be taken care of. Below are the names of the dependencies as written on the NEMO site on the left (<http://bima.astro.umd.edu/nemo/#installation>), and the libraries I used (apt-get'd) on the right:

- pgplot or pplot pgplot5 and libpplot11
- hdf4 libhdf4-0
- cfitsio libcfitsio3
- pvm libpvm3
- png libpng12-0
- gsl libgsl0-dev
- fftw fftw2
- rdl (readline) libreadline5 (not 6!)

In addition, we need Miriad installed. But it's not a nice apt-get library. We've got to manually build this stuff. Do the following somewhere good:

- curl ftp://ftp.astro.umd.edu/progs/bima/miriad.tar.gz | tar xzf -
- wget -O - ftp://ftp.astro.umd.edu/progs/bima/miriad.tar.gz | tar xzf -
- cd miriad/
- ./autogen
- ./configure
- ./make
- Watch lots and lots of text scrolls. It may take a few minutes.

- `./make install`

Now let's make sure that NEMO can find Miriad. While in the Miriad directory, do this:

- `MIR=$PWD; export MIR`

And add something like that (replace the `$PWD` with the proper directory name, which you can get with `echo $MIR`) to your `.bashrc` so your computer will remember where to find Miriad.

Also check the value of `$VENDOR`. I didn't even have it set, which made NEMO fail when it wanted to try some vendor-specific stuff that didn't apply to me. I just set `VENDOR="DUMMY"`. Make sure to export it and shove it into your `.bashrc` like you did with `MIR`.

Now we're finally ready to tackle NEMO:

- `wget ftp://ftp.astro.umd.edu/pub/nemo/nemo_cvs.tar.gz`
- `tar xzf nemo_cvs.tar.gz`
- `curl ftp://ftp.astro.umd.edu/pub/nemo/nemo_cvs.tar.gz | tar xzf -`
- `cd nemo_cvs`
- `cvs login ; cvs update -d` (I just hit enter on the login. Login probably does nothing.)

But wait! There's a typo in one of the files that may not be solved by the time you read this. Check to make sure unless this is really dang old documentation:

- `gedit ./src/orbit/misc/orbintv.c`
- Scroll all the way to the bottom, and replace the `f[3]` with an `f[6]`
- Exit `gedit`

And now we can build everything (hopefully). The first time we build stuff, some won't get built correctly. Do it again, and the issue is resolved. It seems to be an issue with linking stuff. Below is the command you run twice:

- `src/scripts/test_a_new_nemo_cvs reuse=1 nemo=.`

If all went well, you should see a bunch of “OK”s after the second one completes. If there are any “Problems”s, then you have a problem. Otherwise, you're ready to move on to the dark matter caustics module. The module needs to know where to throw its files, so let's set up environmental variables to make the job simple. Like with that `MIR` variable earlier, we need to go into the root directory of a program, this time the top of the NEMO folder (which should be called `nemo_cvs...` unless this documentation is really old). Once in there, do this:

- `NEMO=$PWD; export NEMO`
- `NEMOINC=$NEMO/inc; export NEMOINC`
- `NEMOOBJ=$NEMO/obj; export NEMOOBJ`
- `NEMOLIB=$NEMO/lib; export NEMOLIB`
- `NEMOHOST=`$NEMO/src/scripts/hosttype`; export NEMOHOST`
- `POTPATH=$NEMO/obj/potential; export POTPATH`

These are also used by NEMO and should be included in your `.bashrc` like `MIR`. Next, go into the

nemo_caustics folder, and use the add_caustics_call script to add the caustics.c potential to your own potential (make sure your potential is in \$NEMO/src/orbit/potential/data)

- source add_caustics_call yourpotential

This will create a modified version of your potential, yourpotential_caustics.c. The script just adds a call to one of the apply_caustics_* functions in the yourpotential_caustics.c file, recompiles it, and links it to caustics.c. The script should work for all potentials included with NEMO, but may not work for all potentials you may want to use it for. If it does not work, you must manually combine the potentials. Some advice on how to do so is included in comments at the top of the script.

At this point it would be a good idea to make sure the installation was successful. A makefile, Testfile, should be provided in nemo_caustics. Run the following commands in nemo_caustics, which will run NEMO with an example potential which combines Miyamoto, Plummer, and caustic potentials, and compare the results with known “correct” values:

- source add_caustics_call mpc
- make -f Testfile all

If all tests pass, then you should be all set. Otherwise, have fun figuring out any issues that pop up!

How it works:

This program makes use of two estimates for the gravitational influence of a caustic ring. The first, “close” estimate was derived in the limit of approaching the ring. Originally derived and written in C++ by Heywood Tam, this approximation has been adapted to the mpc potential. The full derivation of the “close” estimate can be found here: <http://arxiv.org/abs/1205.1260>. A second, “far” estimate was later derived by Pierre Sikivie, as discussed in Dumas et al., in preparation 2014. This is calculated independently for all caustic rings, and summed up for the full caustics potential.