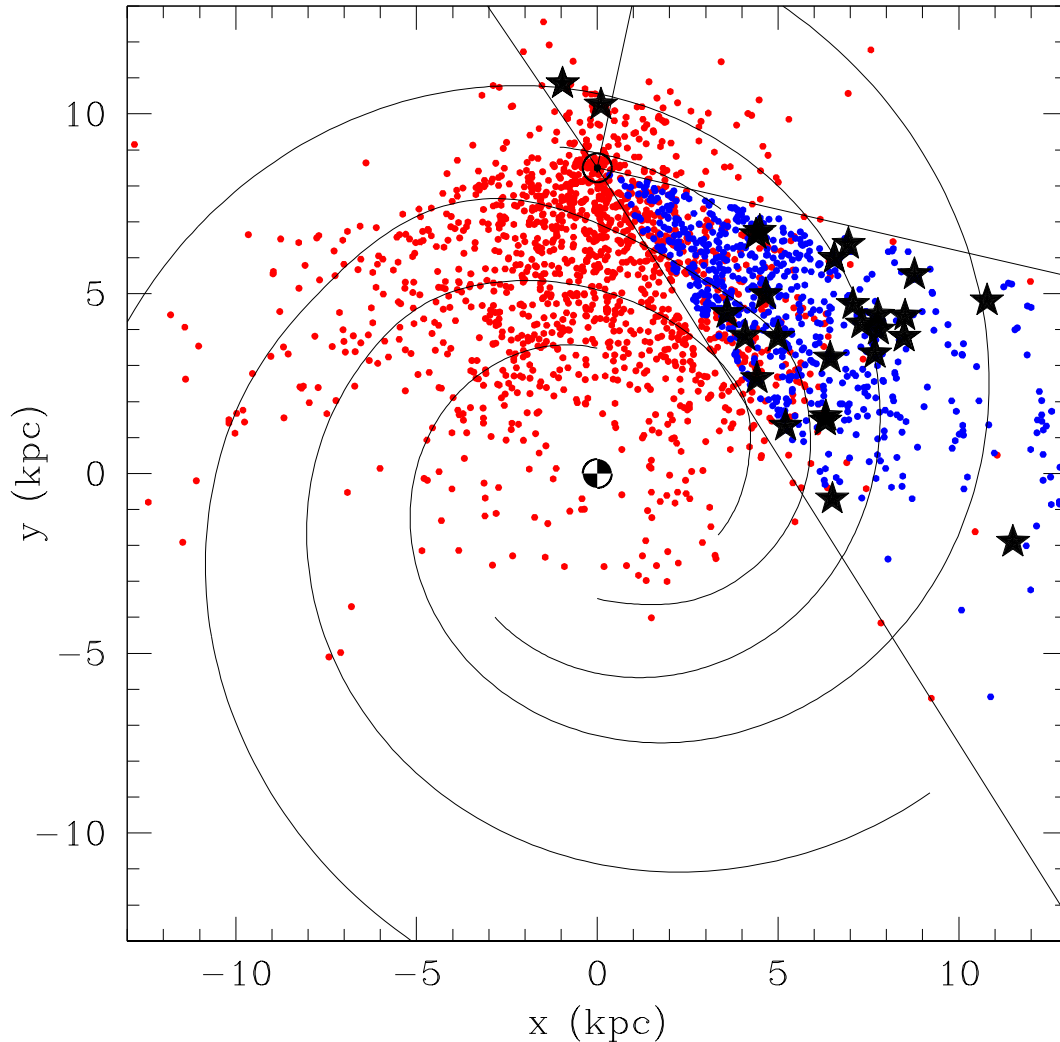


psrpop : Pulsar Population Synthesis Programs (Version 2.0)

Dunc Lorimer — West Virginia University — Duncan.Lorimer@mail.wvu.edu
Friday July 28, 2006



Summary: Pulsar population syntheses, which seek to correct the observationally biased observed sample of pulsars to infer properties about the true Galactic population, have been the subject of numerous papers over the years and have produced a rich variety of interesting and often contrasting results. Pulsar astronomy is currently enjoying an extremely productive phase with new surveys producing greatly increased sample sizes with which to work with. This document describes **psrpop**, a freely-available collection of programs written to simulate the Galactic population of pulsars. The software is constantly undergoing incremental evolution as new features are added. An example is given in the plot above which shows the spatial distribution of known and simulated pulsars on the Galactic $x - y$ plane. The Galactic centre is at the origin and the Sun is located at (0.0,8.5) kpc. The red points show previously known pulsars, while the blue are **psrpop**-simulated pulsars for an Arecibo L-band Feed Array (ALFA) survey. The filled stars show new pulsars discovered with ALFA. The figure was prepared by David Nice and further information about this survey can be found at <http://www.naic.edu/~palfa>.

Installation procedure

Go to <http://psrpop.sourceforge.net> and follow the instructions for downloading and installing the code. Note that `psrpop` has currently only been used on 32-bit linux machines which support Fortran 77. Ports to other operating systems should be possible, but might require some fiddling. Please contact Duncan.Lorimer@mail.wvu.edu with any questions concerning installation and with any updates on new ports that you have been able to achieve. These will then be incorporated in future versions of the code.

The programs

The currently available programs in `psrpop` are: `populate` which creates a static pulsar population for subsequent analysis; `survey` which searches a population generated by `populate` using one or more pulsar survey models; `visualize` which looks at output from `survey` and compare with catalogued pulsars or another model; `pkspdrs` which generates a known pulsar population from various Parkes surveys for input to `visualize`; In the following subsections, we detail the functions of each program.

`populate`

This program generates static populations of pulsars, i.e. there is no spatial or temporal evolution involved, the population represents a snapshot according to various prescribed distribution functions. The program was developed for the analysis described in detail by Lorimer et al. (2006, astro-ph/0607640). A summary of the command-line help is given below.

```
usage: populate {options} {survey file(s)}

-o file - output file (def=psrpop.model)
-n ngen - number of pulsars to generate/detect
-d dmod - distance model (lmt=0;bwhv=1;cwfsr=2;tc93=3;ne2001=4=default)
-e derr - distance error (fraction; def=0.0)
-z zkpc - use exponential scale height (def=model)
-s seed - seed for random number generator (def=based on time of day)
-f file - fix sky positions according to file
-p dist - period distribution (def=from file)
-w duty - width from duty cycle in % (def=from file)
-r dist - radial distribution (def=from file)
-si m s - Mean and sigma of spectral indices (def= -1.60000002 0.349999994)
-rf mhz - Reference frequency for luminosities (def= 1374. MHz)
-rrats - generate RRATs with no spin period info
-quiet - suppress output text to file populate.out
-spiral - generates spiral arms a la fk06
-scatt0 - set all scattering times to zero
```

We can use `populate` in two main ways. In the first example, let us create a population of 5000 pulsars with a Gaussian radial distribution with a standard deviation of 5 kpc, a z scale height of 500 pc and a power-law luminosity distribution in the range 0.1–1000 mJy kpc² with a slope of $d\log N/d\log L = -1$, a log-normal period distribution with a mean of 1000 ms (i.e. the log is 3) and standard deviation of 100

ms (i.e. the log is 2) and a pulse width distribution equal to 6% of the pulse period with a Gaussian scatter as defined by Lorimer et al. (2006). The command-line syntax for this would be as follows:

```
populate -n 5000 -r 5 -z 0.5 -l 0.1 1000 -1 -p lnorm 3 2 -w 6
```

You should see the following messages on the screen

```
Using log-normal period distribution
Pmean = 1000. ms Psigma = 100. ms
Using fixed duty cycle of 6.%
Using Gaussian radial distribution with scale length: 5. kpc
Using exponential z-scale height of 0.5 kpc
Spectral index mean and standard deviation -1.60000002 0.349999994
Luminosity function min, max and slope: 0.100000001 1000. -1.
Reference frequency 1374. MHz.
Initial seed for random numbers: 4307
About to generate 5000 pulsars...
NE2001 distance model
```

and the program will terminate by generating a binary output file `psrpop.model`. This default can be changed with the `-o` option. If we add the `-rrats` option to the above, we generate Rotating Radio Transients (RRATs) instead of pulsars. The way subsequent code currently recognizes this is that the RRATs have no period attached to them in the population file.

The second main use of this program is not to fix the number of pulsars that is generated, but rather keep generating until a specified number of *detectable* pulsars are found. The detectable pulsars are those which fall above the thresholds of an arbitrary number of surveys. For example, let's change the above configuration to keep generating until 1000 pulsars have been found by models of the Parkes multibeam survey of the Galactic plane (PMSURV) and high Galactic latitudes (PHSURV). Let's also change the default distance model to use the Lyne et al. (1985) electron density distribution (`-d 0`). The command-line syntax would then be

```
populate -n 1000 -d 0 -r 5 -z 0.5 -l 0.1 1000 -1 -p lnorm 3 2 -w 6 PMSURV PHSURV
```

You should find that a total of about 215,000 pulsars get generated. The exact number will depend on the starting seed of your random number generator. By default, `populate` will choose a seed based on the time of day. You can change this to a different number using the `-s` option.

In addition to the `populate.model` (or name of your choice) output file, a secondary file `populate.cmd` gets generated. This file contains a record of the command line switches you have been using each time you've run `populate` in your current directory. When experimenting with different input populations, this file can often be quite an invaluable means of keeping track of your work!

The second example above brings up the survey parameter files. Here we used two of them: PMSURV and PHSURV. If you look in your `psrpop-2.0/surveys` directory you will see a whole slew of other parameter files including these two. Each file gives a description of the survey parameters and is used by `populate` and, later on, `survey` to model the detection thresholds and sky coverage of each survey. Note that the programs will first look in the `psrpop-2.0/surveys` directory for each file you supply on the command line. If they are not found there, the current working directory is searched for a file of that

name. You can create any number of these files with parameters that are tailored for each survey. You should, however, use one of the examples in `psrpop-2.0/surveys` as a starting point since the order of the parameters is fixed! As an example, let's look at `psrpop-2.0/surveys/PMSURV`

```
#####
# model of the Parkes multibeam survey - main parameters
#####
 1.2 ! survey degradation factor
 0.6 ! antenna gain (K/Jy)
2100 ! integration time (s)
 0.25 ! sampling time (ms)
 25. ! system temperature (K)
1374 ! centre frequency (MHz)
 288 ! bandwidth (MHz)
 3.0 ! channel bandwidth (MHz)
 2 ! number of polarizations
14. ! full-width half maximum (arcmin)
 0. ! minimum RA (deg)
360. ! maximum RA (deg)
-90. ! minimum DEC (deg)
 90. ! maximum DEC (deg)
-150 ! minimum Galactic longitude (deg)
 50. ! maximum Galactic longitude (deg)
 0. ! minimum abs(Galactic latitude) (deg)
 6. ! maximum abs(Galactic latitude) (deg)
 1.0 ! fractional survey coverage (0-1)
 9.0 ! signal-to-noise ratio
```

Where, hopefully all the parameters and units are self-explanatory¹. They basically relate to the classic Dewey et al. formula for calculating the signal-to-noise ratio (S/N) that a pulsar would have in a given survey, where

$$S/N = S_{1400} G \frac{\sqrt{n_p \Delta \nu \tau}}{\beta T} \sqrt{\frac{P - W}{W}}. \quad (1)$$

Here S_{1400} is the pulsar mean flux density at 1400 MHz (mJy), G is the effective telescope gain (K/Jy), n_p is the number of polarizations summed, $\Delta \nu$ is the observing bandwidth (MHz), τ is the integration time (s), β accounts for S/N losses, T is the system temperature (K), P is the pulse period (s) and W is the observed pulse width (s). Further details can be found in Lorimer et al. (2006).

¹One parameter that is perhaps not obvious is the fractional survey coverage. This is usually equal to unity but values between 0 and 1 simulate the fraction of data processed.

survey

This program takes as input a binary population file generated by `populate` and filters the pulsars through models of one or more sets of survey parameters. The syntax is simply

```
usage: survey name_of_popfile survey1 survey2...
```

As an example, let's search the population of 5000 pulsars that we generated in the previous section

```
populate -n 5000 -r 5 -z 0.5 -l 0.1 1000 -1 -p lnorm 3 2 -w 6
```

using PMSURV and a model for the square kilometre array (SKA). The syntax would be

```
survey psrpop.model PMSURV SKA
```

and you should get the following message on the screen:

```
Fast sampled radio survey simulator
Starting seed for random numbers: 2507
Read in 5000 pulsars from psrpop.model
Generated via: populate -n 5000 -r 5 -z 0.5 -l 0.1 1000 -1 -p lnorm 3 2 -w 6
5000 pulsars in model Galaxy
4168 pulsars found
  Survey      Dis      Det      Out  Smear  Weak
  PMSURV      26       0    4313   658    3
    SKA     4142      26       0    529   303
```

The final lines in the screen output give the number of pulsars discovered by each survey (Dis), the number of previously known ones detected (Det), the number outside the survey region (Out), the number that were in the survey region but not detectable due to dispersion and/or scatter broadening (Smear) and the number in the survey region that were below the detection threshold (Weak). So, in this example, the SKA detects $(4142 + 26)/5000 \simeq 83\%$ of all pulsars in the model galaxy. Note that all 26 pulsars found in the model PMSURV are detected by the SKA. This information is also written to a file `survey.info` which, for this example, looks like this

```
5000 4168
  PMSURV      26       0    4313   658    3
    SKA     4142      26       0    529   303
```

In addition to this summary, all the properties of the detected pulsars are listed in the ASCII file `survey.det`. This file can then be read in by other programs to plot the resulting distributions and carry out statistical analyses. Although the file format is not shown here, a simple inspection of a `survey.det` file should provide enough information to read it into your favourite plotting/analysis program.

visualize

To analyse files produced by `survey`, and compare the model populations with the pulsar catalogue, the program `visualize` is available in the package. This graphical tool is written in Tcl/Tk and requires an additional library which is currently not packaged in the current release. When time permits, I will include links to the necessary libraries. Please contact me if you are interested in using `visualize`.

pksprrs

This program generates ASCII files in the same format as those produced by `survey` for known pulsars in the ATNF catalogue. The command-line syntax is as follows:

```
usage: pksprrs distance_model survey1 survey2 ...
```

It assumes that you have the `psrcat` program installed on your machine. Currently this program has limited options — it only selects isolated pulsars for four different surveys using the Parkes multibeam system: PMSURV, PHSURV, SWINMB1 and SWINMB2. The latter two symbols denote the Edwards et al. survey of intermediate latitudes and the Jacoby et al. survey of higher latitudes. Both of these Parkes multibeam surveys were carried out by the Swinburne pulsar group. As an example, let's generate a catalogue of known pulsars from the PMSURV and SWINMB1 using the Lyne et al. electron density model. The syntax would be `pksprrs 0 PMSURV SWINMB1` and you would see the following output:

```
LMT distance model
1095 pulsars from PSRCAT -> pksmb.real
```

which signifies that the ASCII data has been written to the file `pksmb.real`.

Version history and plans for future work

By placing `psrpop` in the public domain, I hope to encourage others to use the package and suggest/implement improvements. The file `version.history` summarizes the work done so far:

```
1.0 - 2005/04/04 - first version developed from psrdist code
1.1 - 2005/05/16 - added RRATs option to populate and survey
1.2 - 2005/06/03 - fixed scattering bug and added psfile option to visualize
1.3 - 2006/01/10 - added "-isotropic" option to populate
1.4 - 2006/01/11 - added npol and snmin to surveys
1.5 - 2006/01/12 - added spiral arm option to populate and units to visualize
1.6 - 2006/02/06 - added detailed pointing table for PMSURV
1.7 - 2006/03/17 - tidied up a number of features for pmVI paper
1.8 - 2006/04/13 - will read a survey.DMs file to get trial DMs and tsamps
1.9 - 2006/06/16 - the actual version used for pmVI paper analysis
2.0 - 2006/07/28 - tidied up release following acceptance of pm paper VI
```

The current major planned addition to `psrpop` is a program to dynamically evolve pulsar populations, i.e. follow their kinematic and spin-down evolution.

Acknowledgements

I thank Matthew Bailes, Paul Harrison and Rachel Dewey for getting me hooked on pulsar statistics back in the early 1990s and supplying some of the routines used in `psrpop`. I also wish to thank Vicky Kalogera and Chunglee Kim for recent collaborations.