

# modelWDLF v1.02

---

A Java tool for simulating White Dwarf  
Luminosity Functions

Nick Rowell  
August 22, 2013

## Contents

Introduction .....	2
Requirements.....	2
Documents .....	2
Usage.....	3
Initial window.....	3
Main windows.....	4
Star Formation Rate Input Form .....	5
WDLF Simulation Form .....	5
Input Physics Parameters.....	5
Survey Type & Number of WDs .....	6
Magnitude Bins .....	7
Future Developments .....	7
Change Log .....	7

## Introduction

This document describes the use of the synthetic white dwarf Luminosity function generator program 'modelWDLF'. This uses Monte Carlo methods to simulate a population of white dwarf stars then measure a range of statistics including the luminosity function.

## Requirements

modelWDLF is a Java application. It requires Java version 5 or higher; version 6 is needed for the correct appearance of the GUI.

It also uses the plotting program Gnuplot to produce graphs (see <http://www.gnuplot.info/>). Gnuplot version 4.2 or higher must be installed and be on the system path. In addition, the pngcairo terminal must be supported, but I think this is standard on most systems.

modelWDLF will work on both Linux and Windows (tested on XP), however in Windows the solar mass symbol (unicode 0x2299) won't work and will be replaced by '?' in some of the graphs. The software is untested on other operating systems.

## Documents

The project application files, resources etc are bundled in a Java archive (jar file):

File Name	Purpose
modelWDLF.jar	Main application files for modelling the white dwarf luminosity function
modelWDLF_UM.pdf	This user manual

The code itself isn't documented but is (reasonably) well commented. Much of it deals with the GUI interface. The main modelling routines are in `modelWDLF.jar`:

- `Modelling.MonteCarloWDLFSolver`
- `Modelling.ModellingState`
- `Modelling.WhiteDwarfs`

## Usage

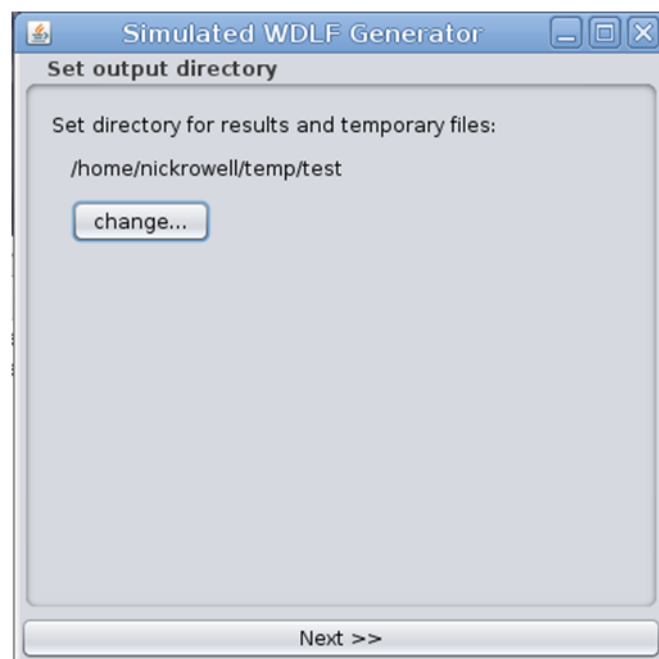
From the command prompt, type:

```
>> java -jar modelWDLF.jar
```

Note that you must be in the directory containing `modelWDLF.jar` and the `lib/` directory.

## Initial window

The first screen that appears looks like this:



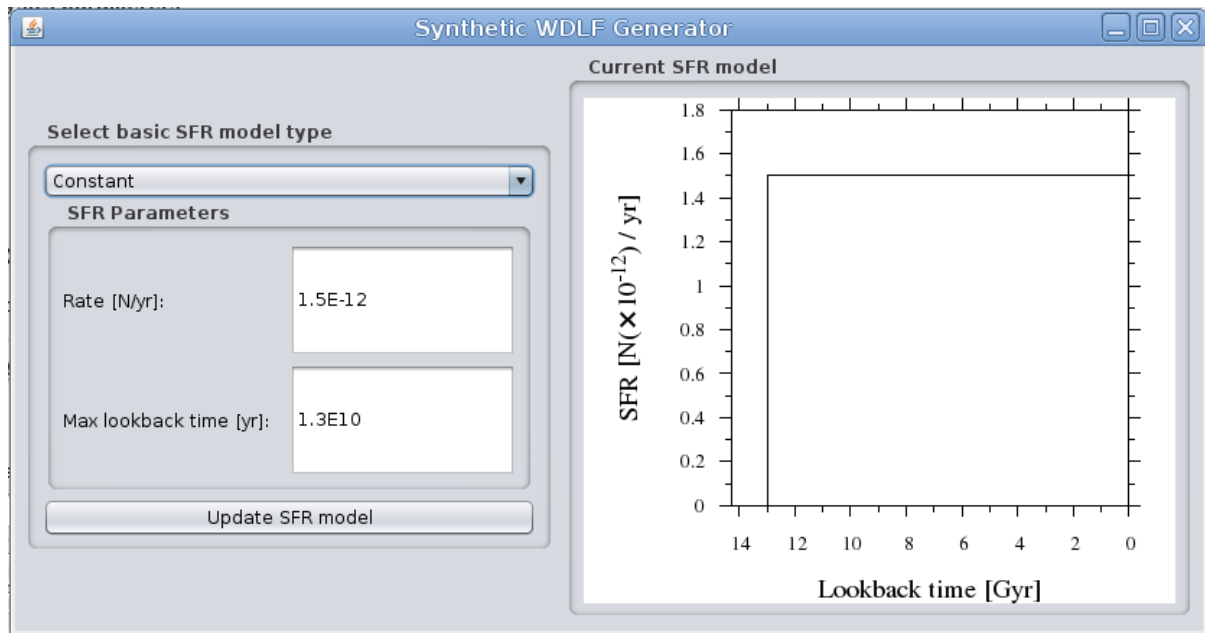
Choose a suitable output directory then hit 'Next >>'. The program will write temporary files and the simulation results to the directory, and will overwrite anything with the same name as any of the files written to disk. The user must have write permission for the directory selected.

On exit, all of the temporary files will be deleted, leaving just the results of the most recent simulation stored in a file called `wdlf_simulation.txt`. This file contains the luminosity function data, as well as the mean age and mass for white dwarfs in each magnitude bin (the age-luminosity and mass-luminosity relations). It also contains a header specifying what choices of input physics parameters and survey type were used in the simulation, and includes at the end the star formation rate - essentially everything that is used to determine the WDLF. This file is regenerated

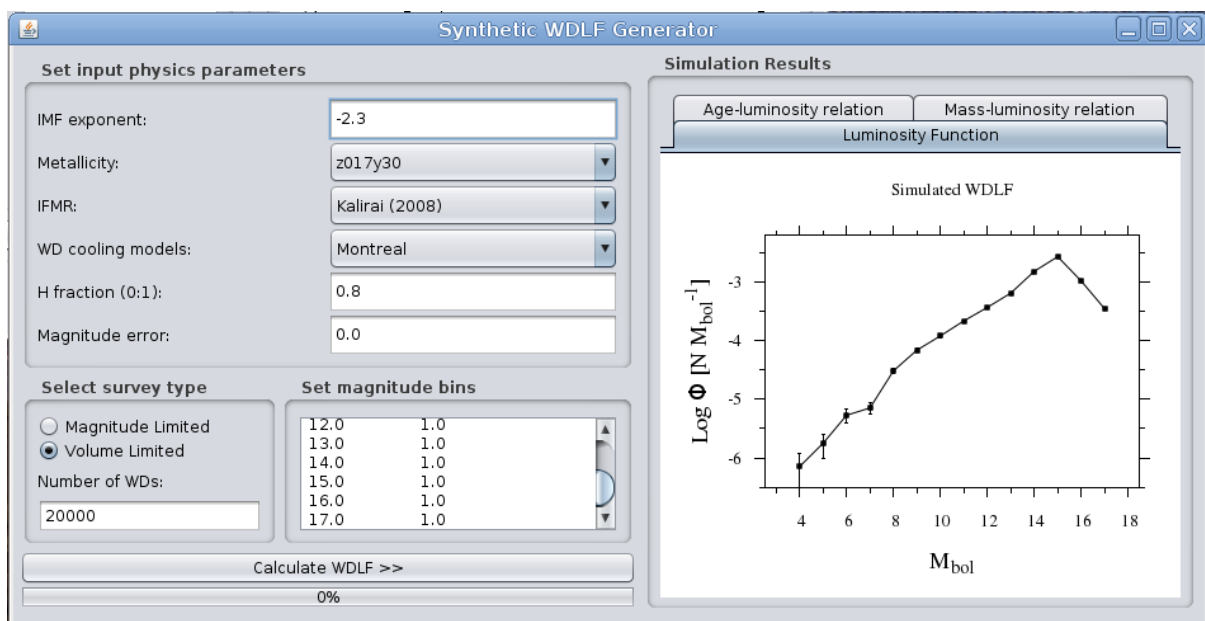
every time the 'Calculate WDLF >>' button is pressed and will clobber any existing file: in order to simulate many WDLFs, copy the file out of the directory between runs. The WDLF data in the file always corresponds to what is currently shown in the GUI.

## Main windows

The next two windows present the main modelling interface. The first is used to set the star formation rate function, and looks like this:



The second is used to set the various input physics and survey parameters, perform the WDLF simulation and display the results. It looks like this:



Hit the 'Calculate WDLF >>' button to read the current parameter selections and generate a new WDLF. This is plotted on the panel on the right, along with the age-luminosity relation and mass-luminosity relation for the white dwarf population.

## Star Formation Rate Input Form

A set of five configurable star formation rate functions are provided via the drop down menu. Each has a different set of parameters that are assigned through the GUI when the corresponding function is selected. Hit 'Update SFR Model' to read the current parameters and plot the resulting SFR on the right. The function plotted in the graph is always the one used to calculate the WDLF.

Note that the fractal SFR model is randomly generated, and will change each time the button is pressed even if none of the parameters change.

Star Formation Rate model	Description
Constant	Simple constant SFR
Exponential Decay	Decaying SFR from initial peak
Single Burst	Single burst at the onset of star formation
Fractal	Randomly generated using fractional Brownian motion. Magnitude sets the number & width of time bins, Hurst parameter sets the roughness at all scales – play around with the parameters and you'll get the idea
Freeform	Allows text entry of general SFR models

## WDLF Simulation Form

This window contains a range of input forms for the different simulation parameters.

### Input Physics Parameters

#### *Initial Mass Function*

A simple power law with a configurable slope parameter  $\alpha$ , where

$$\frac{dN}{dM} = AM^{\alpha}$$

and A is calculated so that IMF is normalised over the range  $0.6 \rightarrow 7.0 M_{\odot}$ .

#### *Metallicity/Main Sequence Lifetimes*

Main sequence lifetimes at different metallicities are taken from the evolutionary models of the Padova group and are described in the papers **Bertelli G., Girardi L., Marigo P., Nasi E., 2008, A&A, 484, 815** and **Bertelli G., Nasi E., Girardi L., Marigo P., 2009, A&A, 508, 355**. The names in the list refer to the metallicity of the model, e.g. "z017y30" has metal content  $Z = 0.017$  and helium content  $Y = 0.30$ . The metallicity isn't interpolated, so each simulated WDLF is for a single metallicity.

When modelling the WDLF, we are only interested in the length of the pre-WD phase at a given stellar mass. For low mass stars, I've included the time spent on the horizontal branch.

### *Initial-Final Mass Relation*

The initial-final mass relation (IFMR) determines the mass of the WD formed by a main sequence star of a given mass. Several different models are available, taken from the literature:

IFMR Name	Reference
Kalirai (2009)	Kalirai, Davis, Richer, Bergeron, Catelan, Hansen & Rich ApJ 705:408-425 (2009)
Kalirai (2008)	Kalirai, Hansen, Kelson, Reitzel, Rich & Richer ApJ 676:594-609 (2008)
Ferrario (2005) linear	Ferrario, Wickramasinghe, Liebert & Williams MNRAS 361:1131-1135 (2005)
Catalan (2008)	Catalan, Isern, Garcia-Berro & Ribas MNRAS 387:1693-1706 (2008)

### *White Dwarf Cooling Models*

We use WD cooling models in order to interpolate the bolometric magnitude at a given cooling time, or the cooling time at a given bolometric magnitude. Both Hydrogen and Helium atmosphere white dwarfs are considered.

There are three sets of models available. The first is from the Montreal group (see [www.astro.umontreal.ca/~bergeron/CoolingModels](http://www.astro.umontreal.ca/~bergeron/CoolingModels)) and are described in **Bergeron P., et al 2011, ApJ, 737, 28** and references therein. The second and third sets are taken from the BaSTI database (see <http://www.oa-teramo.inaf.it/BASTI>) and are described in **Salaris M., et al 2010, ApJ, 716, 1241**. These sets differ according to the treatment of cooling at low temperatures: the models labelled “BaSTI, inc. phase separation” include the effects of sedimentation and phase separation of carbon and oxygen in the WD core when crystallisation sets in, which slows the cooling at low temperatures. The models labelled “BaSTI, exc. phase separation” exclude these effects.

### *Fraction of H and He atmosphere white dwarfs*

Each simulated WD is randomly assigned either a hydrogen or helium atmosphere. The atmosphere type has an effect on the cooling rate. The parameter ‘H Fraction’ sets the probability that a simulated WD will be assigned a hydrogen atmosphere. Values around 0.8 are appropriate.

### *Magnitude Error*

This is used to simulate the effect of observational errors on the photometric parallax: basically, each simulated WD has a random error added to the bolometric magnitude before it is added to the luminosity function. The error is drawn from a Gaussian distribution with a standard deviation set by this parameter. This has the effect of smoothing out any sharp features in the LF. I generally use a value of 0.25 for comparison to SuperCOSMOS WDLFs, and 0.1 for the SDSS photometric WDLFs.

### *Survey Type & Number of WDs*

Volume and magnitude limited surveys have quite different statistical noise profiles – the magnitude limited WDLF has larger errors at faint magnitudes for the same number of survey stars. This option allows both types of survey to be simulated. Note that magnitude limited samples take much longer to compute.

The number of simulated WDs affects the accuracy of the WDLF. It can be set low to reproduce the results of different observational WDLFs, or high to produce low noise WDLFs for fitting to data. There is in principle no upper limit on the number of WDs that can be used, though for magnitude limited surveys a large number of stars might take an unfeasibly long time to compute.

## Magnitude Bins

Enter the magnitude range and resolution of the WDLF as a table of magnitude bin centre and width pairs. The bins can be non-contiguous but can't overlap. Note that the WDLF is always expressed in density per-magnitude, not per-bin.

## Future Developments

A few ideas for further work are

- Another solver that uses Romberg integration to solve the WDLF equation.
- Include the present day WD mass function in the simulation output.
- Additional basic types of SFR function

Other suggestions welcome.

## Change Log

Date	Version	Comments
29/03/2013	1.00	First release
10/06/2013	1.01	Removed the upper limit on the number of simulated WDs, which was set to around 2E6 (depending on the RAM available) Corrected a bug that meant WD models were stuck on Montreal and the combo box in the GUI didn't have any effect
22/08/2013	1.02	Merged several source packages into one (modelWDLF.jar), and included source code in jar. Removed command to exit when run on Mac OS. Removed the Ferrario et al (2005) polynomial initial-final mass relation.

- Nick Rowell, 22/08/13

nickrowell@computing.dundee.ac.uk