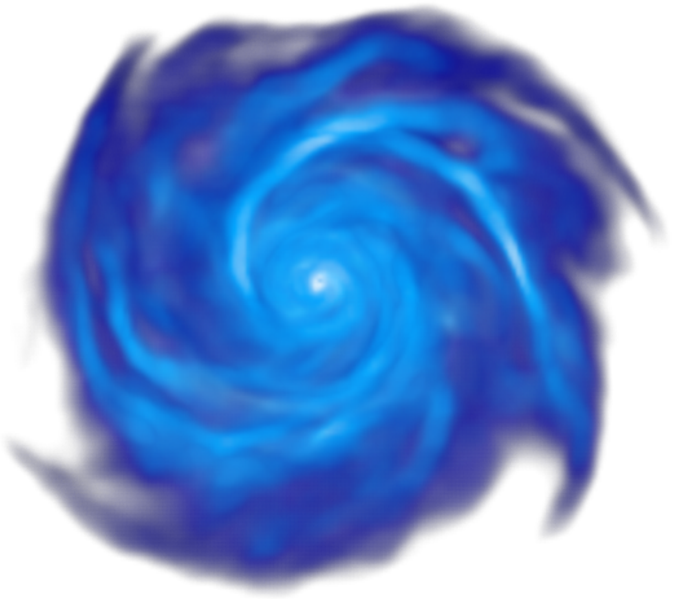
Laboratoire d'Astrophysique de Marseille

# DICE: Disk Initial Conditions Environment

Valentin Perret

October 18, 2012



## CONTENTS

# 1 About the code

## 1.1 Origins

First of all, it is important to note that this code was not written from scratch. It is a fork distribution, from a code named starscream, written by Jay Billings. I encourage every people interested in DICE to take a look to strarscream (`http://code.google.com/p/starscream/`). I first used starscream as part of my PhD project, involving numerical simulations of idealized galaxies. Since there was some limitations in the code and it was under a GPL license; I decided to modify it to fit my needs. It was important for me to follow the concept of public code followed by starscream, which is one of the few codes for generating initial conditions of idealized galaxies that is completely free, well documented, and embedded in a real cross-platform environment. DICE intends to follow this lead line, and to globally follow the spirit of starscream.

## 1.2 Purpose

DICE is designed to generate initial conditions file, allowing the user to explore the dynamics (and many other physics) of galaxies through secular evolution, or through encounter events. Generating idealized galactic initial conditions involves using a lot of different numerical tools and algorithms. Thus, reading and understanding the code should never be considered as a waste of time. DICE intends to be as much didactic as starscream was for me. With this code, you can generate initial conditions using only ascii files as input. The advantage of DICE is that you can generate with one executable a lot of different scenarios.

## 1.3 Required libraries

DICE uses the GSL & FFTW-3 libraries. You can find these libraries here:

- `http://www.fftw.org/download.html`
- `http://www.gnu.org/software/gsl/`

GSL is used to generate random numbers, and also to perform numerical integration. FFTW is used to compute Fourier Transforms through the fastest portable implementation available.
On Linux-based systems, it quite easy to perform a standard installation of these libraries using package tools such as **apt-get** or **yum**.
On OSX systems, **macport** (`http://www.macports.org`) is a good option to compile and install easily these libraries. You might be interested in using the threading capacities of the library FFTW. In this case, download the tarball, and perform the following commands:

```
tar -xzvf fftw-x.x.x.tar.gz
cd fftw-x.x.x
./configure --enable-threads
```

```
make
sudo make install
```

If your libraries are installed in a non-standard location, think about updating the DYLD_LIBRARY_PATH variable for shared libraries and LD_LIBRARY_PATH variable for static libraries.

## 1.4 COMPILATION & INSTALLATION

DICE is available for download from the internet. You can download it and unzip it by typing at the prompt:

```
wget http://projets.oamp.fr/attachments/download/483/DICE-version.tar.gz
tar -xzvf DICE.tar.gz
cd DICE
```

You can also checkout the latest version in the SVN repository typing:

```
svn co http://svn.oamp.fr/repos/DICE
```

The DICE package comes with the CMake cross-platform build system. So technically, you don't have to worry so much about the compilation. Make sure you have cmake installed by typing:

```
cmake --version
```

If your version is older than cmake 2.6, you will have to update your system to a more recent version of cmake. To generate the makefile, type:

```
cd build
cmake ..
```

If no errors are detected through this step, compile the code and install the code like this:

```
make
make install
```

By default, DICE is installed in `$HOME/local/dice`, but you can specify a different installation directory using the flag `-DCMAKE_INSTALL_PREFIX=/install/path`
A cmake macro is implemented to locate standard installations of these libraries. Nevertheless, if you installed them in a different way, use the following keywords to help cmake locating these libraries:

- `-DGSL_PATH = /path/to/gsl`
- `-DFFTW3_PATH = /path/to/fftw3`

Once installed, think about adding to the `PATH` environment variable the location of the DICE binary. If you use bash and under standard installation process, add to your `.bashrc` (or `.profile` for OSX systems) :

```
export PATH=$HOME/local/dice/bin:$PATH
```

Optionally, if you want the code to run a bit faster, you can use the threading abilities of the FFTW library for the gravitational potential computation. In this case, and assuming that you have installed the treaded version of FFTW3, then replace the previous commands by:

```
cmake .. -DENABLE-THREADS=ON
make
make install
```

CMake will automatically look for the `fftw3_threads` library, and link it.

## 1.5 Quick start

DICE take as input the name of a file called the DICE configuration file. This file contains all the informations needed to start the program. It specifies:

- The location of ASCII files containing the physical parameters of the galaxy to generate. These files are called the galaxy parameters files. You can point up to 64 galaxy parameters files , just remember to put galaxies with same parameters one after the other: it lowers the computation time since DICE copy the result of the previous computation if it is strictly the same galaxy.
- The output format of the initial conditions. For now, DICE handle only two format. **Gadget1** and **RamsesGroup**. But you can use the **uns_projects** tools, calling the **unsio** API, to convert from **Gadget1** to **nemo** or **Gadget2**.
- The parameters to define a Keplerian trajectory between two and only two galaxies. If there more or less galaxies, or if DICE misses one or more of the three keywords needed, the galaxies will not be placed on such an orbit.

DICE is provided with an example, in order to quickly test the DICE initial conditions. Once DICE has been installed, you can run the example by typing in the prompt:

```
cd example
dice dice.config
```

Contents of "dice.config" file in example folder:

```
% List of galaxy parameters file
% Putting identical galaxies one beside the other allows DICE
% to re-use the previous computation,
% and thus reduces the global execution time
```

```
Galaxy ./galaxy1.params
Galaxy ./galaxy2.params


% Output format of the initial conditions
% Allowed format 'Gadget1','RamsesGroup'
ICformat Gadget1

% Keplerian trajectory parameters
% This keywords should be set
% if you want a Keplerian orbit between two and only two galaxies!!
% If there are more or less galaxies, trajectories will be set according
% to the xc,yc,zc,vel_xc,vel_yc,vel_zc keywords in the galaxy parameters files.
% It induces that the keywords xc,yc,zc,vel_xc,vel_yc,vel_zc are not considered
% when the Keplerian trajectory parameters are set.
% By default, galaxies are aligned along the X-axis

% Eccentricity of the Keplerian orbits
Eccentricity  1.0
% Initial distance between the galaxies [kpc]
Rinit 100.
% Pericentral distance of the Keplerian system [kpc]
Rperi 10.
% Polar angle of the normal vector of the orbital plane
OrbitPlanePhi 45.
% Azimuthal angle pf the normal vector of the orbital plane
OrbitPlaneTheta 45.
```

Each Galaxy keywords in the dice configuration file should point to an ASCII file containing the physical parameters of the galaxy to construct. Every keywords are needed, so check the content of this file carefully.

Content of galaxy1.params file in example folder:

```
% Virial velocity of the galaxy [km/s]
v200  100.
% Redshit of the galaxy
redshift 0.0
% Mass fraction of the stellar disk
m_d  0.05
% Bulge-disk scale factor
f_b  0.10
% Disk angular momentum fraction
j_d  0.05
% Mass fraction of the stellar bulge
```

```
m_b  0.001
% Halo spin parameter
lambda  0.05
% Halo concentration parameter [kpc]
c_halo  10.0
% Stellar disk scale length [kpc]
r_disk  2.0
% Gaseous disk scale length [kpc]
r_gas  3.0
% Number of element for thepotential grid
% Setting ngrid as 2^n with n positive integer
% allows FFTW algorithm to run much more faster
ngrid  128
% Size of the box to compute the gravitational potential [kpc]
boxsize  52.0
% Mass fraction of gas compare to stellar disk
gas_fraction  0.15
% Initial temperature of the gas [K]
t_init  1e4
% Stellar disk thickness parameter
disk_temp  0.2
% Gaseous disk thickness parameter
gas_temp  0.05
% Minimal value for the Toomre parameter
Q_lim 1.5
% Number of halo particles
nhalo 50000
% Number of stellar disk particles
ndisk   50000
% Number of gaseous disk particles
ngas   50000
% Number of stellar bulge particles
nbulge   5000
% Halo radial density cut [kpc]
halo_cut  25.0
% Stellar disk radial density cut [kpc]
disk_rcut  9.0
% Stellar disk azimutal density cut [kpc]
disk_zcut  1.2
% Gaseous disk radial density cut [kpc]
gas_rcut  9.0
% Gaseous disk azimutal density cut [kpc]
gas_zcut  0.45
% Stellar bulge density cut [kpc]
```

```
bulge_cut  2.0
% Halo model
% [1 = Hernquist profile, 2 = Plummer profile, 3 = Jaffe profile,
% 4 = Uniform profile, 5 = NFW profile]
halo_model 1
% Disk model
% [1 = Exponential-r Sech2-z disk, 2 = Myamoto-Nagai profile,
% 3=Exponential-r Exponential-z disk]
disk_model 1
% Gas model
% [1 = Exponential-r Sech2-z disk, 2 = Myamoto-Nagai profile,
% 3=Exponential-r Exponential-z disk]
gas_model 3
% Bulge model
% [1 = Hernquist profile, 2 = Plummer profile, 3 = Jaffe profile,
% 4 = Uniform profile]
bulge_model 1
% Position of the galactic center on x-axis [kpc]
xc 0.
% Position of the galactic center on y-axis [kpc]
yc 0.
% Position of the galactic center on z-axis [kpc]
zc 0.
% X-axis velocity component of the galaxy [km/s]
vel_xc 0.
% Y-axis velocity component of the galaxy [km/s]
vel_yc 0.
% Z-axis velocity component of the galaxy [km/s]
vel_zc 0.
% Spin angle of the disk in the XY plane [degree]
spin 0.
% Inclination of the disk compare to XY plane [degree]
incl  0.
```

## 2 About the Physics

## 3 Publication Policy

DICE is a free software distributed under the GPL license, and thus you can do almost what you want to do with it. But if you have found this software useful for your research, I would appreciate an acknowledgment to use of the *'???'* by Perret et al. (2012).

## 4 Acknowledgment

I would like to thank all the people that helped me working on this project. Here is some people involved at some points in the code:

- Benoit Epinat (LAM)
- Philippe Amram (LAM)
- Juan Carlos Bastos Pineda (IAU)
- Frederic Bournaud (CEA)
- Jean Charles Lambert (LAM)