FERNET

Fluorescence **E**mission **R**ecipes and **N**um**E**rical routines **T**oolkit

1 About FERNET

FERNET is a toolkit developed to aid in the simulation and interpretation of confocal microscopy experiments. It uses Mcell engine to calculate 3D molecular trajectories over complicated scenarios and generates fluorescence traces and/or images.

Developed and mantained at the Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina by the following people

- Juan Francisco Angiolini (jfa@qb.fcen.uba.ar, website)
- Esteban Mocskos (emocskos@dc.uba.ar, website)

Latest version of this manual, source code, compiled executables and examples can be found at `http://dc.uba.ar/FERNET`.

2 Compiling FERNET

FERNET needs the following libraries to compile and run:

- argtable (`http://argtable.sourceforge.net/`)
- libconfig (`http://www.hyperrealm.com/libconfig/`)
- libtiff (`http://www.libtiff.org/`)

To compile, go to the *src* folder and run the following commands:

```
make
sudo make install
sudo make clean (optional, cleans temporal files)
```

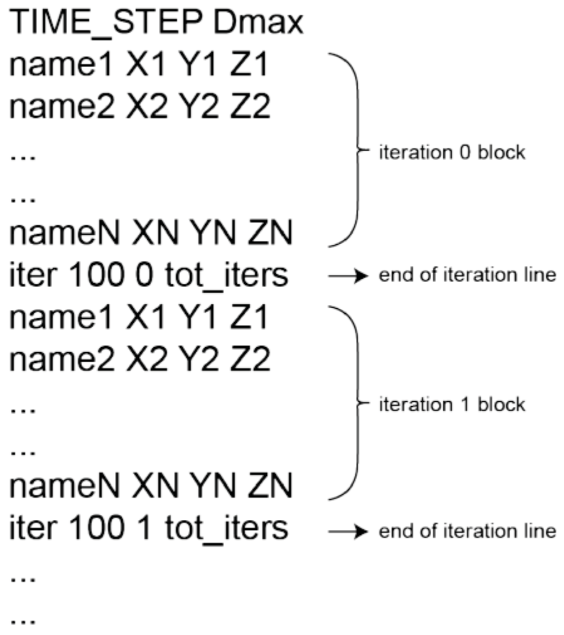
3 MCell and molecular trajectories generation

MCell and Blender usage is thoroughly detailed and explained with documentation and tutorials at

- `http://www.mcell.org/tutorials/`
- `http://www.mcell.org/documentation/`

Our custom version of MCell introduces two new visualization modes in order to work with FERNET: JOINED and PIPED. As any MCell visualization mode they need to be specified with the MOLECULE_VIZ_OUTPUT parameter, normally found in the *Scene.viz_output.mdl* file. These two new modes generate an output file containing the molecule name and position. Each iteration is separated by a special line. The first line of the file contains the TIME_STEP of MCell and the maximum value of the diffusion coefficient among the simulated molecules (Figure 1). FERNET uses these values to check if the selected sampling time is adequate for the simulated system (TIME_STEP needs to be 10 times smaller than $\tau_{Dmax} = \frac{\omega_{xy}^2}{4D_{max}}$.

This output file (joined_X.dat or PIPE_X.dat, where X is the seed number) is then given as the input file for FERNET (section 6). In JOINED mode, the file is physically created and once MCell finishes the simulation it can be viewed and edited before passing it to FERNET. However, this file tends to be very large (several GBs even for small simulations). If the user does not require analysing the simulated molecular trajectories and desires to speed up the simulations the PIPED mode can be used. Instead of creating a physical file with the trajectories, it creates a pipe file that connects the output of MCell with the input of FERNET. If MCell is started in PIPED mode, it will freeze until FERNET is started with the pipe file as input. Once MCell and FERNET are connected the later will output the fluorescence information.



```
TIME_STEP Dmax
name1 X1 Y1 Z1
name2 X2 Y2 Z2
...
...
nameN XN YN ZN
iter 100 0 tot_iters
name1 X1 Y1 Z1
name2 X2 Y2 Z2
...
...
nameN XN YN ZN
iter 100 1 tot_iters
...
...
```

Figure 1: Schematic representation of MCell output in JOINED/PIPED mode.

4 FERNET configuration file

This plain text file can be found inside the *config* folder and contains information and parameters of the simulated system. The FERNET configuration file is composed by the 7 blocks detailed below. The *cfgmaker* tool allows the creation of the FERNET configuration file with an user friendly graphical interface.

Common parameters block (shared for all the acquisition modes):

```
channel0: // Channel 0 configuration (mandatory)
{
    on  = 1;              // channel state (1 = on, 0 = off)
    molec   = ("A","B");  // list of molecule names from MCell that emit
                          in channel 0
    bright  = (100000,10000); // list with brightness in counts per second
                         per molecule for channel 0, length must be
                         equal to number of molecs in channel
};

channel1:   // Channel 1 configuration (for dual color experiments, optional)
{
    on  = 1;
    molec   = ("B");
    bright = (100000);
};

kappa  = 200e-9;  // time of single fluorescence event in seconds (already optimized
                  parameter, be careful if editing)
w_xy   = 0.2;     // PSF waist in XY plane
w_z    = 1.0;     // PSF waist in Z direction
```

In the example above, molecule A will emit in channel 0 with brightness of 100000 cpsm and molecule B will emit in channel 0 with brightness of 10000 cpsm and in channel 1 with 100000 cpsm. The same molecule can appear any number of times in the list with different brightness (for example, two different dyes of the same color in a molecule).

Point mode block:

```
shiftx  = 0.0;      // X center of PSF
shifty  = 0.0;      // Y center of PSF
shiftz  = 0.0;      // Z center of PSF
prefix  = "point";  // name for photon output file
```

The output file containing the fluorescence trace is named *prefix_c0.txt* for channel 0 and *prefix_c1.txt* for channel 1 (if needed).

Multi mode block:

```
prefix = "multi";   // prefix for output name
shiftz = 0.0;       // Z center of PSF
nPSFX  = 5;         // number of PSF in X axis
dx     = 0.2;       // separation between PSF in X axis
nPSFY  = 5;         // number of PSF in Y axis
dy     = 0.2;       // separation between PSF in Y axis
```

In multi mode, a grid of nPSFX×nPSFY PSFs is created, each center separated dx and dy microns on each axis. The output files are named *prefix_XXX_cY.txt*, where XXX is a number from 0 to (nPSFX×nPSFY)-1 and Y is the corresponding channel. An auxiliary file *index.txt* is created with the number of the PSF and its (X,Y) center.

Line mode block:

```
deadtime  = 0.001;       // dead line time in seconds
centerx   = 0.0;         // X center of PSF (um)
centery   = 0.0;         // Y center of PSF (um)
centerz   = 0.0;         // Z center of PSF (um)
n_columns = 15;          // number of columns
shift     = 0.2;         // shift between columns (um)
tiffname  = "linescan";  // output TIFF name prefix
```

In line mode, a line of (n_columns×shift) microns centred on (centerx,centery,centerz) is scanned along the x axis. The time that the scanning laser takes to return to the initial scan position can be simulated with the deadtime parameter. The output image containing the XT profile of fluorescence is named *tiffname_cY.tif*, where Y is the channel number.

Image mode block:

```
deadtime = 0.001;    // dead line time in seconds
centerz  = 0.0;      // Z center of image
pixel    = 0.05;     // pixel size in um
width    = 64;       // image width in pixels
height   = 64;       // image height in pixels
tiffname = "image";  // output TIFF name
```

Similar to line mode, except an area of width×height pixels centred on (0,0,centerz) is scanned.

Stack image block:

```
deadtime = 0.001;     // dead line time in seconds
pixel    = 0.05;      // pixel size in um
width    = 64;        // image width in pixels
height   = 64;        // image height in pixels
tiffname = "image";   // output TIFF name
top_z    = 1.0;       // top Z position of the stack in um
bot_z    = -1.0;      // bottom Z position of the stack in um
step     = 0.1;       // Z plane separation
```

Similar to image mode, except the scenario is scanned between top_z and bot_z positions at step intervals to obtain a 3D stack.

SPIM image block:

```
NA       = 1.6;       // numerical aperture of the microscope
lambda   = 500.0;     // emission wavelength in nm
waist    = 0.5;       // axial waist of the excitation PSF, overrides the value
                      of w_z in common block
pixel    = 0.1;       // pixel size in um
width    = 64;        // image width in pixels
height   = 64;        // image height in pixels
centerz  = 0.00;      // Z center of the image in um
frame_t  = 0.001;     // frame time in seconds
tiffname = "image";   // output TIFF name
```

CCD imaging mode. The fluorescence of an area of width×height of the simulated scenario pixels centred on (0,0,centerz) is binned at frame_t intervals.

5 Calculation of number of iterations

The TIME_STEP parameter in MCell configuration files is equal to the dwell time or pixel time in a confocal microscopy experiment. For point and multi mode, the length of the fluorescence trace will be equal to the number of iterations simulated in MCell (ITERATIONS parameter).

In line and raster-scan image modes, we introduced the deadtime parameter which is a time delay between consecutive lines acquisition to match the typical confocal line-scan frequencies (in the order of 100-1000 Hz). During deadtime MCell continues performing simulation steps however FERNET disregards these steps (Figure 2). We strongly recommend doing this calculation BEFORE performing MCell simulations in order to run the required number of simulation steps. The value of ITERATIONS in MCell required for line, raster and stack scan modes can be calculated as follows.

For line-scan method

$\text{ITERATIONS} = \text{number of desired lines} \times \left(\text{n\_colums} + \mathit{round}\left(\frac{\text{deadtime}}{\text{TIME\_STEP}}\right)\right)$

For raster-scan method

$\text{ITERATIONS} = \text{number of desired frames} \times \text{height} \times \left(\text{width} + \mathit{round}\left(\frac{\text{deadtime}}{\text{TIME\_STEP}}\right)\right)$

For stack method

For now FERNET takes a single 3D stack and disregards the exceeding simulated steps. The minimum and necessary MCell iterations for a single stack can be calculated as

$\text{slices} = \mathit{round}\left(\frac{\text{Top Z - Bot Z}}{\text{step}}\right)$

$\text{ITERATIONS} = \text{slices} \times \text{height} \times \left(\text{width} + \mathit{round}\left(\frac{\text{deadtime}}{\text{TIME\_STEP}}\right)\right)$

For SPIM method

SPIM imaging is performed with a CCD sensor, not in a raster mode. The number of MCell iterations needed can be calculated as

$\text{ITERATIONS} = \text{number of desired frames} \times \mathit{round}\left(\frac{\text{frame\_t}}{\text{TIME\_STEP}}\right)$

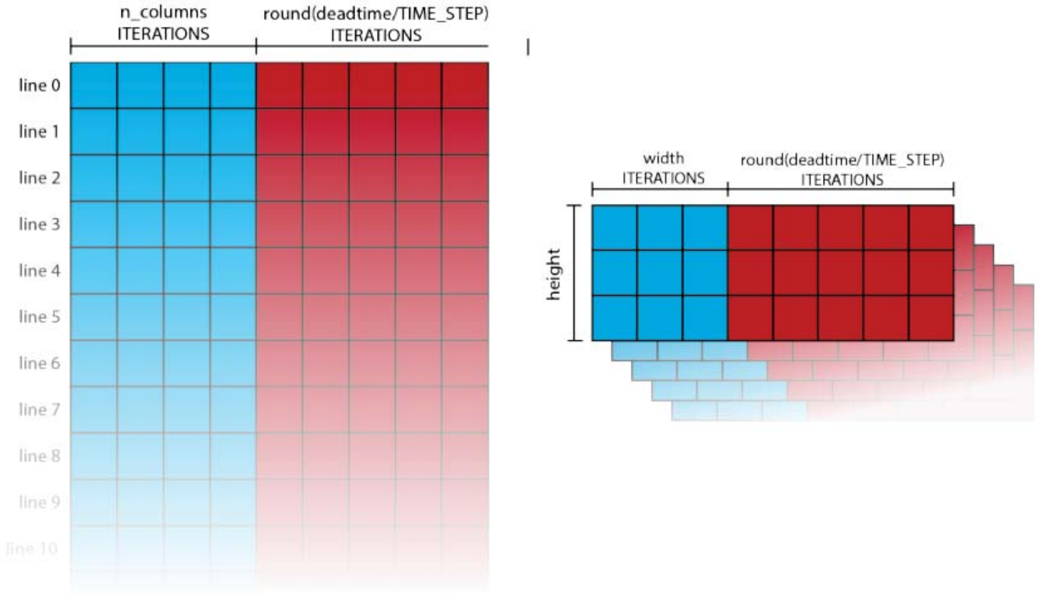


Figure 2: Schematic representation of FERNET line mode (left) and image mode (right).

6 FERNET usage

To run FERNET, simply invoke it from a terminal with the following arguments:

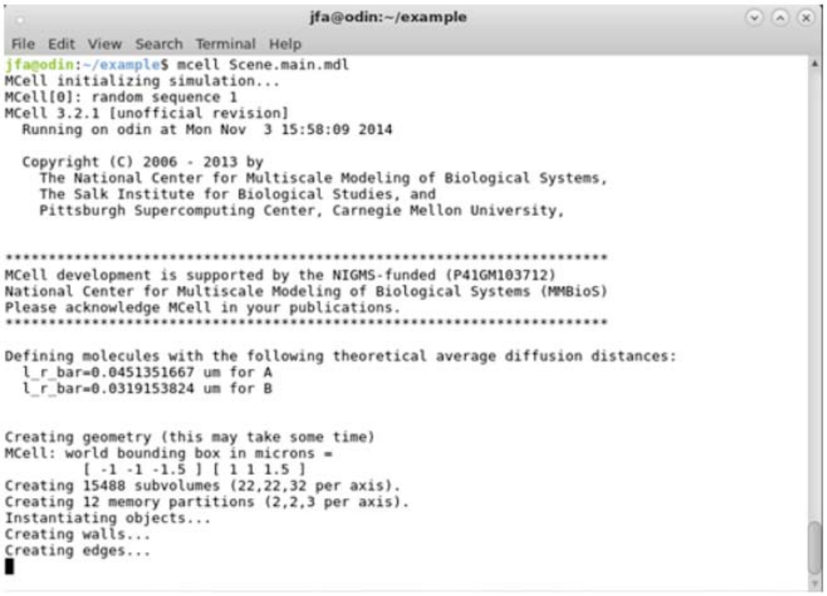
```
fernet INPUT_FILE -m MODE -c CONFIG
```

or

```
fernet INPUT_FILE --mode=MODE --config=CONFIG
```

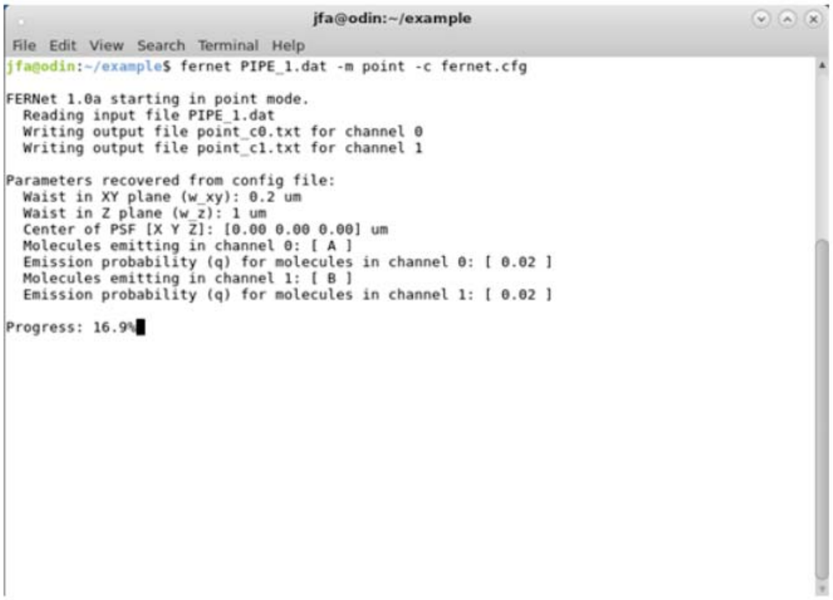
where INPUT_FILE is the file with the molecular trajectories generated with MCell (either in JOINED or PIPED mode), MODE is one of the possible modes (point, multi, line, image, stack, spim) and CONFIG is the FERNET configuration file. If no errors occurred, FERNET will display information related to the selected mode and a progress percentage.

The example folder contains all the necessary files to run a simulation with two types of molecules inside a box, molecule A emitting in channel 0 and molecule B emitting in channel 1. The first step is to start MCell with the *Scene.main.mdl* file. Since the output was set to PIPED mode, MCell will freeze until FERNET is started (Figure 3). Once FERNET is called (in point mode for this example), MCell and FERNET will each display its current state (Figures 4 and5). Once the simulation is finished, the fluorescence trace for each channel is obtained (Figure 6).



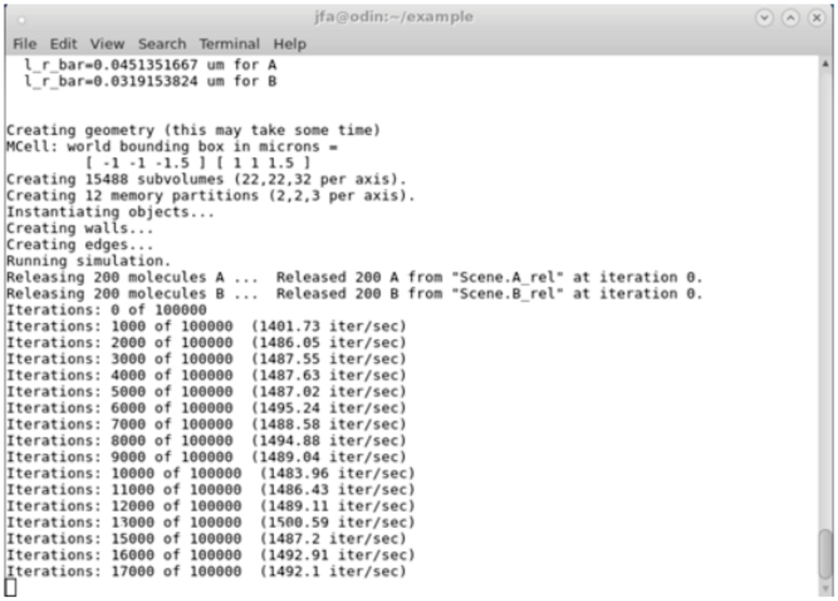
```
jfa@odin:~/example$ mcell Scene.main.mdl
MCell initializing simulation...
MCell[0]: random sequence 1
MCell 3.2.1 [unofficial revision]
  Running on odin at Mon Nov  3 15:58:09 2014

  Copyright (C) 2006 - 2013 by
    The National Center for Multiscale Modeling of Biological Systems,
    The Salk Institute for Biological Studies, and
    Pittsburgh Supercomputing Center, Carnegie Mellon University,


**********************************************************************
MCell development is supported by the NIGMS-funded (P41GM103712)
National Center for Multiscale Modeling of Biological Systems (MMBioS)
Please acknowledge MCell in your publications.
**********************************************************************

Defining molecules with the following theoretical average diffusion distances:
  l_r_bar=0.0451351667 um for A
  l_r_bar=0.0319153824 um for B


Creating geometry (this may take some time)
MCell: world bounding box in microns =
         [ -1 -1 -1.5 ] [ 1 1 1.5 ]
Creating 15488 subvolumes (22,22,32 per axis).
Creating 12 memory partitions (2,2,3 per axis).
Instantiating objects...
Creating walls...
Creating edges...
█
```

Figure 3: Starting MCell in PIPED mode.



```
jfa@odin:~/example$ fernet PIPE_1.dat -m point -c fernet.cfg

FERNet 1.0a starting in point mode.
  Reading input file PIPE_1.dat
  Writing output file point_c0.txt for channel 0
  Writing output file point_c1.txt for channel 1

Parameters recovered from config file:
  Waist in XY plane (w_xy): 0.2 um
  Waist in Z plane (w_z): 1 um
  Center of PSF [X Y Z]: [0.00 0.00 0.00] um
  Molecules emitting in channel 0: [ A ]
  Emission probability (q) for molecules in channel 0: [ 0.02 ]
  Molecules emitting in channel 1: [ B ]
  Emission probability (q) for molecules in channel 1: [ 0.02 ]

Progress: 16.9%
```

Figure 4: Starting FERNET with MCell output.



```
  l_r_bar=0.0451351667 um for A
  l_r_bar=0.0319153824 um for B


Creating geometry (this may take some time)
MCell: world bounding box in microns =
         [ -1 -1 -1.5 ] [ 1 1 1.5 ]
Creating 15488 subvolumes (22,22,32 per axis).
Creating 12 memory partitions (2,2,3 per axis).
Instantiating objects...
Creating walls...
Creating edges...
Running simulation.
Releasing 200 molecules A ...  Released 200 A from "Scene.A_rel" at iteration 0.
Releasing 200 molecules B ...  Released 200 B from "Scene.B_rel" at iteration 0.
Iterations: 0 of 100000
Iterations: 1000 of 100000  (1401.73 iter/sec)
Iterations: 2000 of 100000  (1486.05 iter/sec)
Iterations: 3000 of 100000  (1487.55 iter/sec)
Iterations: 4000 of 100000  (1487.63 iter/sec)
Iterations: 5000 of 100000  (1487.02 iter/sec)
Iterations: 6000 of 100000  (1495.24 iter/sec)
Iterations: 7000 of 100000  (1488.58 iter/sec)
Iterations: 8000 of 100000  (1494.88 iter/sec)
Iterations: 9000 of 100000  (1489.04 iter/sec)
Iterations: 10000 of 100000  (1483.96 iter/sec)
Iterations: 11000 of 100000  (1486.43 iter/sec)
Iterations: 12000 of 100000  (1489.11 iter/sec)
Iterations: 13000 of 100000  (1500.59 iter/sec)
Iterations: 15000 of 100000  (1487.2 iter/sec)
Iterations: 16000 of 100000  (1492.91 iter/sec)
Iterations: 17000 of 100000  (1492.1 iter/sec)
```

Figure 5: MCell progress.

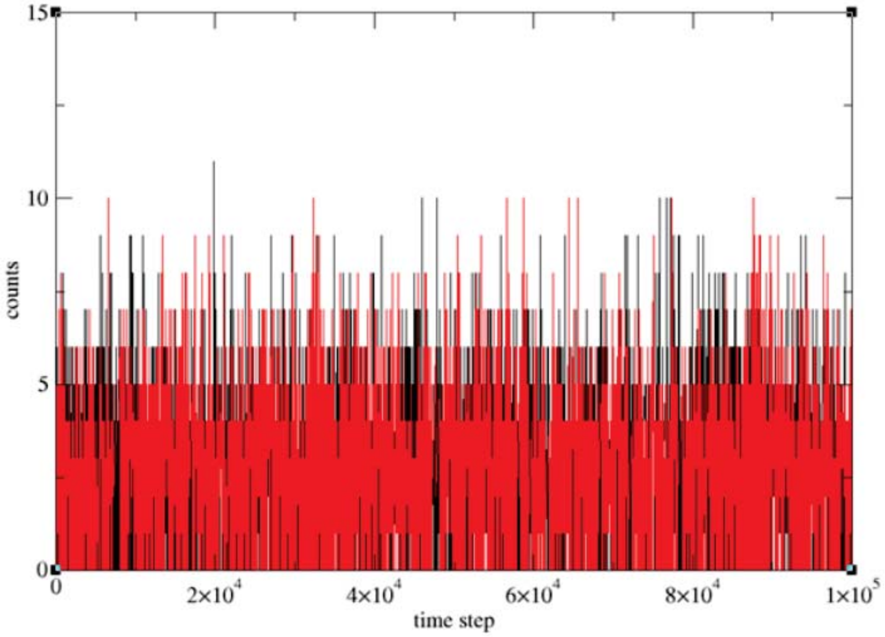


Figure 6: Fluorescence trace for each channel.