Hyantes

for version 1.3.0, 7 July 2009

L. S. Guelton (serge.guelton@enst-bretagne.fr)

This manual is for Hyantes (version 1.3.0, 7 July 2009), Hyantes is free software; you can redistribute it and/or modify it under the terms of the CeCILL-C License

Copyright © 2007 LIG

Table of Contents

1 Installation Instructions

1.1 Quick Installation

1.1.1 Using subversion repository

For this section, we suppose you have downloaded the library sources from the `subversion` repository, using the shell command

```
svn checkout svn://scm.gforge.inria.fr/svn/hyantes
```

and that you have entered into the source repository (most certainly the `hyantes` repository, we will use <hyantes_src_dir> in the following) using

```
$ cd <hyantes_src_dir>
```

Once you are in the source repository, you can lauch the shell command

```
/<hyantes_src_dir>$ autoreconf -vis
```

to (re)generate the build structure. `autoreconf` is a tool bundled with the `autotools` and it generates the `configure` script, as well as several auxiliary files. In the following section we will detail how to configure, build and install the library.

1.1.2 Building the hyantes library

It is better that hyantes library is compiled in a different directory from the source files. In the source repository (or another repository), you can create a `build` repository using the shell command

```
/<hyantes_src_dir>$ mkdir build ; cd build
```

In the `build` repository, you can now lauch the `configure` script:

```
/<hyantes_src_dir>/build$ ../configure
```

The shell command `../configure` should be enough to configure the library. In this case, the default installation path will be '`/usr/local`', it can be changed as detailed in Section 1.2.1 [Configure option], page 1. If any dependency check fails, you will be noticed. Then you just need install the missing softwares and run this shell command again.

The shell command `make` will build the software and `make install` will install the package.

```
/<hyantes_src_dir>/build$ make && make install
```

Now, you can enjoy hyantes library!

1.2 In depth Installation

1.2.1 `configure` option

The `configure` script always respect some typical options, plus some package-specific ones. Some of them are detailed here, please read `../configure --help` for full help.

- Those ones are package agnostic:

  '`--prefix=/my/install/path`'
  : sets the installation path

‘`"CFLAGS=-O3"`’
: sets the package `CFLAGS`

‘`"CPPFLAGS=-I /etc/include"`’
: sets additionnal include path

‘`"LDFLAGS=-L /etc/lib"`’
: sets additional library path

2 Overview

`Hyantes` is a library to compute neighbourhood population potential with scale control. It is developped by the Mescal team from the Laboratoire Informatique de Grenoble, as a part of Hypercarte project. The `Hypercarte` project aims to develop new methods for the cartographic representation of human distributions (population density, population increase, etc.) with various smoothing functions and opportunities for time-scale animations of maps. `Hyantes` provides one of the smoothing methods related to multiscalar neighbourhood density estimation. It is a `C` library that takes sets of geographic data as inputs and computes a smoothed representation of this data taking account of neighbourhood's influence.

3 Library Interface

All following '`C`' enumerations, structures and functions can be found in the header file '`hyantes.h`'. To use them, simply include it

```
#include <hyantes.h>
```

3.1 Enumerations

unamed enumeration *F_DISK F_AMORTIZED_DISK F_GAUSSIAN F_EXPONENTIAL F_PARETO* [Data Type]

`F_DISK` Disk smoothing method

`F_AMORTIZED_DISK`
smoothing method amortizing potential

`F_GAUSSIAN`
smoothing method using gaussian distribution

`F_EXPONENTIAL`
smoothing method using exponential distribution

`F_PARETO` smoothing method using pareto distribution

unamed enumeration *HS_PARSE_ONLY HS_THRESHOLD HS_LOAD_RAW HS_LOAD_PRECOMPUTED HS_SMOOTH_FUNC HS_MODULE_OPT* [Data Type]

`HS_PARSE_ONLY`
(deprecated) only require generation of precomputed quadtree, extra arg : "char *filename"

`HS_THRESHOLD`
(deprecated) set the threshold used for ignoring some area, extra arg: "double threshold"

`HS_LOAD_RAW`
(deprecated) tells the library to consider input file as a raw data file, no extra arg

`HS_LOAD_PRECOMPUTED`
(deprecated) tells the library to consider input file as a precomputed file, no extra arg

`HS_SMOOTH_FUNC`
tells the library to use given function and param to perform smoothing, extra arg: "char *funcname, double extra param, ...

`HS_MODULE_OPT`
(deprecated) pass option to module

3.2 Data Structures

`hs_config_t` *g_file_serialize threshold g_is_raw_data fid fparam herrno status* [Data Type]

`g_file_serialize`
Type: FILE * deprecated

`threshold`
Type: double deprecated

`g_is_raw_data`
Type: int deprecated

`fid` **Type**: smoothing_fun_t smoothing function to use

`fparam` **Type**: double parameter used by smoothing function

`herrno` **Type**: int code of last error encountered

`status` **Type**: unsigned long status of the execution

`hs_potential_t` *lat lon pot* [Data Type]

`lat` **Type**: data_t latitude of the potential

`lon` **Type**: data_t longitude of the potential

`pot` **Type**: data_t value of the potential

`hs_coord_t` *mLat mLon MLat MLon* [Data Type]

`mLat` **Type**: data_t minimum latitude

`mLon` **Type**: data_t minimum longitude

`MLat` **Type**: data_t maximum latitude

`MLon` **Type**: data_t maximum longitude

3.3 User Functions

`void hs_display` (*size_t* `lonRange`, *size_t* `latRange`, *hs_potential_t pt*[*latRange*][*lonRange*]) [Library Function]
displays the matrix of processed potentials

lonRange the longitudinal resolution of the matrix

latRange the resolution of the matrix

pt the matrix of potential which is of size latRange by lonRange

`int hs_set_r` (*hs_config_t* *`config`, *hs_option_t* `opt`, ...) [Library Function]
sets the given option to the given parameters in the given configuration

config pointer to the configuration to use

opt option to set

Return value: 1 if setting went well, 0 otherwise

`int hs_set` (*hs_option_t* `opt`, ...) [Library Function]
sets the given option to the given parameters in the default configuration (deprecated, you should use your own configuration structure)

opt option to set

Return value: 1 if setting went well, 0 otherwise

`hs_potential_t * hs_smooth` (*int* `_resoLat`, *int* `_resoLon`, *hs_coord_t* `visu`, *FILE* * `pFileReference`) [Library Function]
performs the smoothing of target area inside visu, using potentials from pFileReference the smoothing is performed using smoothing method given by hs_set(HS_SMOOTH_FUNC, ...) the resolution of the output matrix will be resoLat x resoLon

_resoLat number of latitude points computed

_resoLon number of longitude points computed

visu visualization window

pFileReference
file containg the data in the format latitude longitude potential latitude longitude potential ... latitude longitude potential where latitude and longitude are given in degrees

Return value: an allocated array of size resoLat x resoLon containing a struct (lat, lon, pot)

`hs_potential_t * hs_smooth_r` (*int* `_resoLat`, *int* `_resoLon`, *hs_coord_t* `visu`, *FILE* * `pFileReference`, *hs_config_t* *`configuration`) [Library Function]
performs the smoothing of target area inside visu, using potentials from pFileReference and using given hs_config the smoothing is performed using smoothing method acording to the configuration given in the arguments the resolution of the output matrix will be resoLat x resoLon

_resoLat number of latitude points computed

_resoLon number of longitude points computed

visu visualization window

pFileReference
file containg the data in the format latitude longitude potential latitude longitude potential ... latitude longitude potential where latitude and longitude are given in degrees

configuration
configuration to use

Return value: an allocated array of size resoLat x resoLon containing structs (lat, lon, pot)

`hs_potential_t *  hs_smoothing` (*int* `_resoLat`, *int* `_resoLon`, *const char* *`function_name`, *double* `function_param`, *hs_coord_t* `visu`, *FILE* * `pFileReference`) [Library Function]

perform the smoothing of target area inside visu, using potentials from pFileReference the smoothing is performed using function_name smoothing method, with a radius of function_param the resolution of the output matrix will be resoLat x resoLon (obsolete function, use hs_smmoth_r instead)

_resoLat number of latitude points computed

_resoLon number of longitude points computed

function_name
name of a smoothing method listed by hs_list_smoothing

parameter (in kilometers) of the smoothing method

visu visualization window

file containg the data in the format latitude longitude potential latitude longitude potential ... latitude longitude potential where latitude and longitude are given in degrees

Return value: an allocated array of size resoLat x resoLon containing structs (lat, lon, pot)

`unsigned long  hs_status` () [Library Function]

observer of the execution of the computation

Return value: number of computed input potential points from the beginning of the computation

`const char **  hs_list_smoothing` (*size_t* * `sz`) [Library Function]

list all available smoothing methods that can be configured using hs_config

pointer to the number of smoothing methods

Return value: array of string constant of size *sz. Memory is still owned by hyantes

4 hyantesite

4.1 Hyantesite Overview

`hyantesite` is a simple client example to `hyantes` library. It can be used as a stand-alone program to benefit from `hyantes` features, or as a sample code to build another `hyantes` client.

Basically, the program takes a file containing stocks as inputs, as follows

```
latitude lontitude stock
latitude lontitude stock
...
latitude lontitude stock
```

and, with given various parameters, it prints out a result file containing potential values, as follows

```
latitude lontitude potential
latitude lontitude potential
...
latitude lontitude potential
```

The computed potentials highly depend on the given parameters, see Section 4.2 [Command line invocation], page 8 for a detailed explanation of those parameters. During the execution, the program will display various information screen as well as a progression bar (updated each second).

4.2 Command line invocation

The following table provides a description of the arguments required by `hyantesite`.

‘`-d`’

‘`--dump-tree`’

Require the dump of processed input. No further computation is done. The result is printed out in file given by the ‘`--output`’ option. This file can be used for further call via the ‘`--precomputed-file`’ option.

‘`-p`’

‘`--precomputed`’

Indicate that input file was precomputed with ‘`--dump-tree`’ option. The file is directly loaded for computation instead of being parsed. It results in quicker load, but it is not portable !

‘`-t`’

‘`--threshold`’

(deprecated) Threshold used to skip small area. This is used to ignore area which do not contribute enough to current calculus.

‘`-w`’

‘`--window`’

Coordinate of the visualisation window, given in degrees as *minimum latitude*,*minimum longitude*,*maximum latitude*,*maximum longitude*

‘`-f`’
‘`--function`’
: Name of the smoothing function, chosen among `disk`, `amortized_disk`, `gaussian`, `pareto`, `exponential`

‘`-r`’
‘`--range`’
: Smoothing range in kilometers, used as parameter of *interaction function*

‘`-s`’
‘`--scale`’
: Resolution of output map, given in number of potentials per latitude (*lat resolution*) and longitude (*lon resolution*) as *lat resolution*x*lon resolution*

‘`-i`’
‘`--input`’
: Path to the input file, default is ‘`stdin`’. The file containing input data is in the format described in Section 3.3 [User Functions], page 5

‘`-o`’
‘`--output`’
: Path to the output file, default is ‘`stdout`’. The file containing out data is in the same format as *input file*

‘`-v`’
‘`--version`’
: Print out some version information and exit

‘`-h`’
‘`--help`’
: Print out some quick help and exit

4.3 Sample invocation

The following example is used to render a small dummy file read from ‘`stdin`’:

```
hyantesite ‘--window’=0,0,5,5\
 ‘--function’=disk\
 ‘--range’=10\
 ‘--scale’=10x10\
 ‘-o’ res.dat << EOF
0 0 10
0 5 100
5 0 50
EOF
```

Here is output:

```
0.000000 0.000000 0.031831
0.000000 0.555556 0.000000
0.000000 1.111111 0.000000
...
0.000000 3.888889 0.000000
0.000000 4.444444 0.000000
0.000000 5.000000 0.318310
```

```
0.555556 0.000000 0.000000
0.555556 0.555556 0.000000
0.555556 1.111111 0.000000
...
4.444444 3.888889 0.000000
4.444444 4.444444 0.000000
4.444444 5.000000 0.000000
5.000000 0.000000 0.159155
5.000000 0.555556 0.000000
5.000000 1.111111 0.000000
...
5.000000 5.000000 0.000000
```

As expected, the range was not enough to make the three stocks interact. The chosen scale is too small to see how each stock spead in a *disk* way. However, we can see that the potential at <*0 5*> is no longer *100* as in the input file. This is because the stock <*0 5 100*> spreads over *10* kilometers, wich means a normalized value of $100/(3.14 * (10^2))$.

If we run the example again, with a higher *scale* (more units to compute), that is

```
hyantesite '--window'=0,0,5,5\
 '--function'=disk\
 '--range'=10\
 '--scale'=1000x1000\
 '-o' res.dat << EOF
0 0 10
0 5 100
5 0 50
EOF
```

We will get a more accurate visualisation of the output. Indeed '`res.dat`' now looks like this

```
0.000000 0.000000 0.031831
0.000000 0.005005 0.031831
0.000000 0.010010 0.031831
0.000000 0.015015 0.031831
0.000000 0.020020 0.031831
0.000000 0.025025 0.031831
0.000000 0.030030 0.031831
0.000000 0.035035 0.031831
0.000000 0.040040 0.031831
0.000000 0.045045 0.031831
0.000000 0.050050 0.031831
0.000000 0.055055 0.031831
0.000000 0.060060 0.031831
0.000000 0.065065 0.031831
0.000000 0.070070 0.031831
0.000000 0.075075 0.031831
0.000000 0.080080 0.031831
0.000000 0.085085 0.031831
```

```
0.000000 0.090090 0.000000
0.000000 0.095095 0.000000
...
```

As this is not very easy to visualize, let us use the well known `gnuplot` software

```
gnuplot << EOF
set terminal png
set out 'res.png'
splot 'res.dat'
EOF
```

It will produce a `png` image from '`res.dat`'. It is often a good idea to run `gnuplot` interactively,

```
gnuplot
> splot 'res.dat'
```

5 Examples

These two quick examples may help to understand how the value of a stock spreads over its neighbourhood.

5.1 Manage input and output

The input format is

```
latitude longitude stock
```

But sometimes you have a diffrently formatted input file. Here are some tips:

select fields

```
awk '{ printf("$1 $3 $4\n"); }' foo.bar
```

It will only select first, third and forth fields from *foo.bar*. This command can also be used to remove multiple spaces ...

swap entries

```
awk '{ printf("$2 $1 $3\n"); }' foo.bar
```

It will swap latitude and longitude from *foo.bar*.

remove null value

```
grep -v " 0.0000" foo.bar
```

It will remove all null value from output file *foo.bar*.

preview result

```
gnuplot
> splot 'foo.bar'
```

It will display a 3d view of output file *foo.bar*.

another preview

```
gnuplot
> set pm3d map
> set palette gray
> splot "europe_np.out"
```

It will display a 2D view of output file *europe_np.out*.

5.2 Single Stock

In this example, we will consider a simple single stock, that is

```
cat > single.txt << EOF
5 5 100
EOF
```

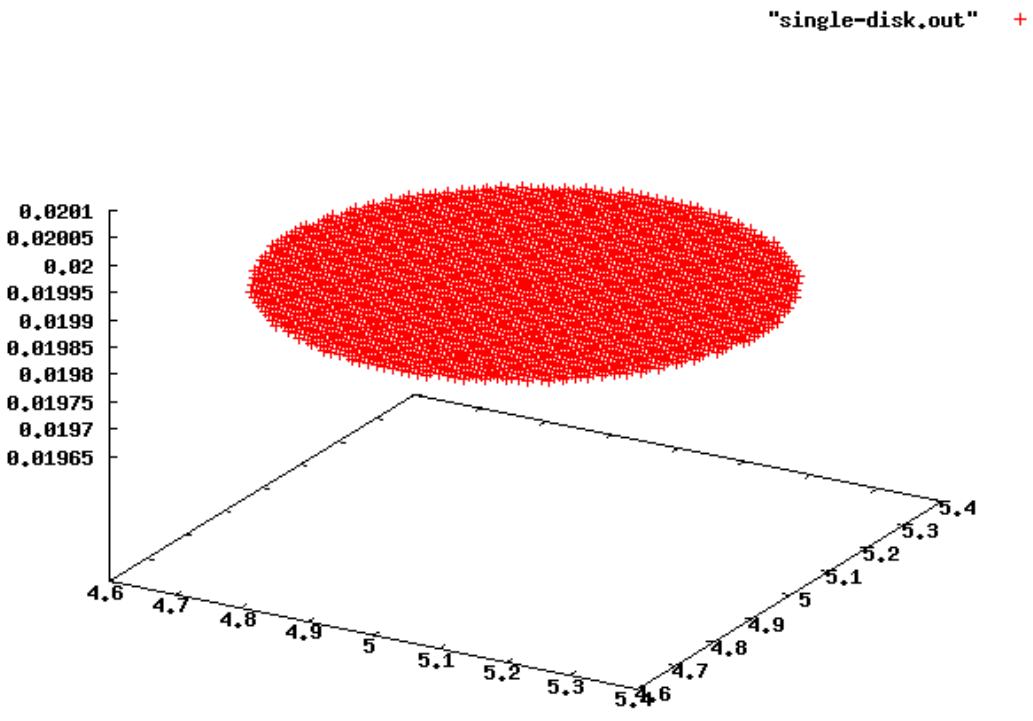
'`single.txt`' now contains a single entry at latitude and longitude 5 and of value 100.

We can render it using the shell command

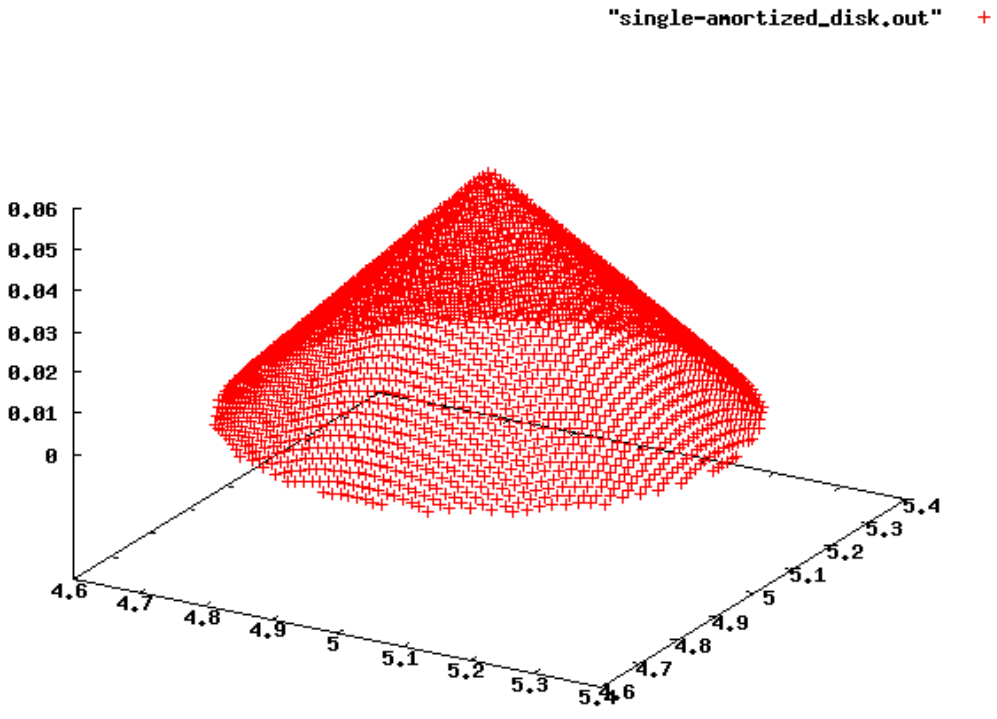
```
hyantesite '-i' single.txt\
 '-w' 0,0,10,10\
 '-f' disk\
 '-r' 40\
```

```
‘-s’ 800x800\
‘-o’ single-disk.out
```

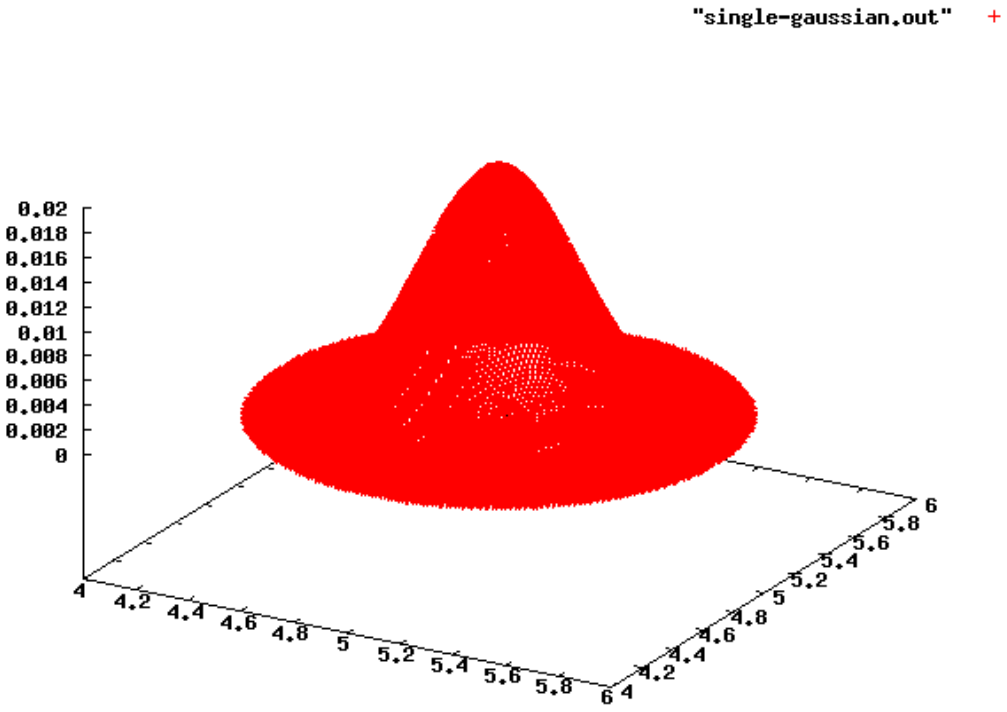
The visualization window is (0,0) (10,10), centered on our stock. Here we used a disk interaction of range 40 kilometers:



We could have used an amortized disk to get a pine cone:



A Gaussian will spread more:



5.3 Two Stocks

In this example, we will consider a simple couple of stocks, that is

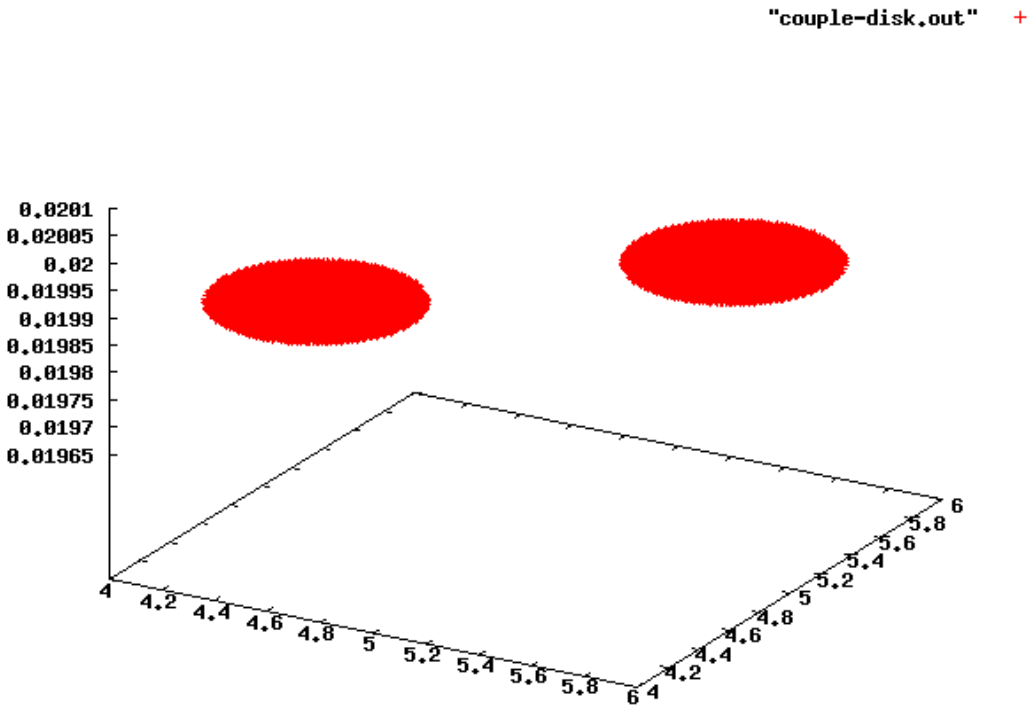
```
cat > couple.txt << EOF
4.5 4.5 100
5.5 5.5 100
EOF
```

‘`couple.txt`’ now contains two entries of value 100 at latitude and longitude 5.5 and 4.5

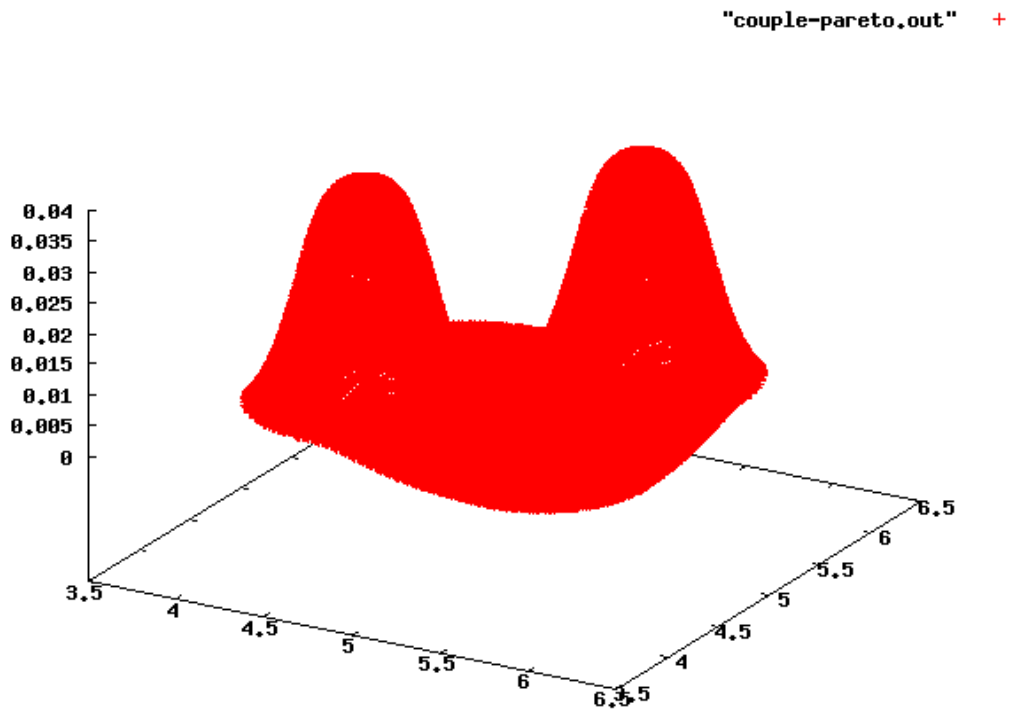
Once again we can render it using the shell command

```
hyantesite ‘-i’ couple.txt\
 ‘-w’ 0,0,10,10\
 ‘-f’ disk\
 ‘-r’ 40\
 ‘-s’ 800x800\
 ‘-o’ couple-disk.out
```

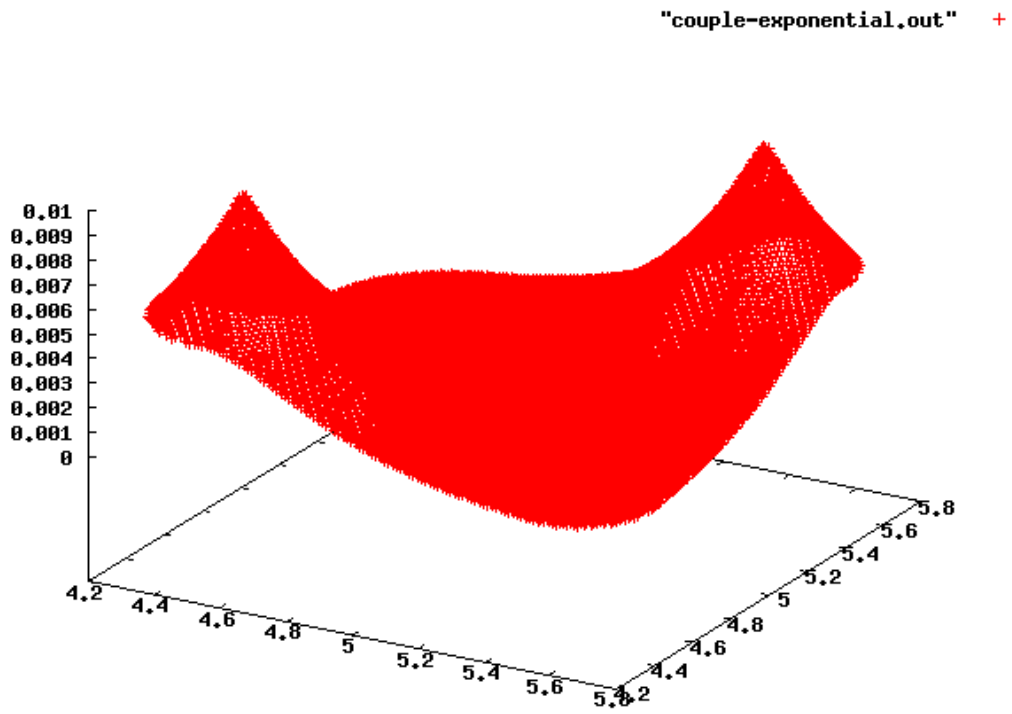
The visualization window is (0,0) (10,10), centered on our stocks. Here we used a disk interaction of range 40 kilometers. There will be no interaction: the range is two small !



We could have used a pareto interaction to observe a kind of interaction:



An exponential would work too, but in a different way:



Note that both disks are cut because of the visualisation window !

6 Contacts

6.1 LIG

`Hyantes` is developed in a team from the

Laboratoire LIG
110 av. de la Chimie - Domaine Universitaire
BP 53 - 38041 Grenoble - France cedex 9
Tel: 04 76 51 43 61

6.2 MESCAL

The team itself is known as Mescal.

Laboratoire Informatique et Distribution (ID)-IMAG
ENSIMAG - antenne de MontbonnotZIRST 51, avenue Jean Kuntzmann
38330 Montbonnot Saint Martin / France

6.3 Authors

`Hyantes` was originally created by Serge Guelton `serge.guelton @ enst-bretagne.fr` on an idea from Jean-Marc Vincent `jean-marc.vincent @ imag.fr` . Here is a list of historical maintainers.

- Jean-Marc Vincent `jean-marc.vincent @ imag.fr`
- Serge Guelton `serge.guelton @ imag.fr`
- Said Oulahal `said.oulahal @ imag.fr`
- Sbastien Martinez `sebastien.martinez @ telecom-bretagne.eu`

6.4 Thanks

`Hyantes` has originally been written by Serge Guelton. Many people further contributed to `Hyantes` by reporting problems, suggesting various improvements or submitting actual code. Here is a list of these people. Help us keep it complete and exempt of errors.

- Liyun Guelton `liyun.he @ gmail.com` - *for being my wife (and a nice doc reviewer)*
- Christine Plumejeaud `christine.plumejeaud @ imag.fr` - *loyal user*
- Vincent Danjean `vincent.danjean @ imag.fr` - *packaging guru*
- Said Oulahal `said.oulahal @ imag.fr` - *for all the reviews and the meaningful advices*
- Jean-Louis Roch `jean-louis.roch @ imag.fr` - *for the adaptive algorithm*
- Ronan Keryell `ronan.keryell @ telecom-bretagne.eu` - *for the c99 enlightenment*
- Sebastien Martinez `sebastien.martinez @ telecom-bretagne.eu` - *for algorithmic enhancement, validation stuff and warcraft3 games*