Math 3P40 Assignment 1: Spatial growth

Preparatory part (nothing to submit)

Create on your X drive (or USB drive) a folder named `math3p40`, and then inside it a subfolder `assign1`. Open web browser, go to ISAAK web page for Math 3P40 and download (using right click and "Save link as") to your folder `assign1` the following five files:

- rand.c
- randomnumbers.cpp
- monoferapopulation.cpp
- monoferainspace.cpp
- plotpop.gnu

Now open Cygwin xterm and change directory to your folder:

```
cd /cygdrive/x/math3p40/assign1
```

Open the first of the supplied programs with notepad editor by typing in the terminal window:

```
notepad randomnumbers.cpp
```

This will be our first C++ program, demonstrating how to use the pseudo-random number generator supplied in the file rand.c (you do not need to understand how the generator works, but if you are curious, you can open rand.c and inspect the source.)
We will now compile the program with GNU C++ compiler,

```
g++ randomnumbers.cpp
```

This produces executable `a.exe`, which you can run by typing `./a.exe`. Please compare the output of the program with the source, and make sure your understand how the output is produced.

Once the above is working, open `monoferainspace.cpp`, compile it and run. This is an implementation of the "monofera in restricted space" model discussed in class. Again, inspect the code and make sure your understand how it works.

Finally, open `monoferainspace.cpp`, compile it and run. This program prints percentage of sites occupied by monofera as a function of time step. It outputs time t and $U(t)/M$ in two columns (notation as in class notes), assuming that $\Delta t = 1$.

Plotting with gnuplot

We will now plot the graph of $U(t)/M$ versus t. Issue the following commands:

```
g++ monoferapopulation.cpp
./a.exe > pop.dat
```

This will redirect the output of `a.exe` to the file `pop.dat`. This is a very convenient way of saving output to a file in Unix/Linux environment. Now type in the terminal `gnuplot`, this starts Gnuplot plotting program. Type

```
plot ''pop.dat''
```

You should see the desired graph. Close the window by pressing `q`.

Now we will fit the logistic curve to the data obtained in pop.dat. Type the following commands (while still in gnuplot):

```
U(x)=1.0/(exp(-lambda*(x-ti))+1)
lambda= 0.15
ti=30
fit U(x) "pop.dat" via ti,lambda
plot  U(x), "pop.dat"
```

This defines the function U to be fitted, gives starting values of parameters, fits and plots the result (both data and fitted curve). Note that after issuing `fit`, gnuplot prints the resulting values of parameters as well as some other information.

Now close the plot window (press `q`) and quit gnuplot by pressing `Ctrl D`. You returned to the Cygwin shell. Now open with notepad the file `plotpop.gnu`. Note that it contains all commands issued before. You can run gnuplot in batch mode by typing in the shell

`gnuplot plotpop.gnu`

This saves a lot o typing, especially if you are modifying the graph many times.

Finally, we will learn how to produce publication-quality graph with gnuplot. Create file plotpdf.gnu with the following content:

```
set terminal pdf color enhanced
set out "figure.pdf"
set xlabel "t"
set ylabel "U(t)/M"
set nokey
U(x)=1.0/(exp(-lambda*(x-ti))+1)
lambda= 0.15
ti=30
fit U(x) "pop.dat" via ti,lambda
plot   U(x), "pop.dat" pt 7 ps 0.3
```

The above file is also available on ISAAK as `plotpdf.gnu`. When you type in Cygwin shell `gnuplot plotpdf.gnu`, it will produce `figure.pdf` file, which can be viewed in Acrobat Reader or other PDF viewer, printed or included in another document. In the above `pt` denotes point type and `ps` denotes point size. You can experiment with other point types and sizes if you wish.

Math 3P40 Assignment 1: Spatial growth

Due date: January 26, 2017 at 12pm in the assignment box across J434.

Problem 1

In the preparatory part you plotted $U(t)/M$ vs. t for monofera model in space. The resulting curve was not very smooth. Open the file `monoferapopulation.cpp` and uncomment line 22 from the top. This will produce initial lattice occupied with randomly placed monofera where 1% of all sites are occupied. Additionally, increase M to 1000. Compile and run the program, save the output, then plot and fit the logistic curve to the data, as before. Is the curve much better now? Why you think it is the case? Try to increase M even further, to 10000. Is there any change? Attach printout of your best curve and write down the resulting values of t_i and λ.

Problem 2

The "normalized" population of monofera $U(t) = \frac{P(t)}{M}$ is given by

$$U(t) = \frac{1}{e^{-rM(t-t_i)} + 1},$$

where $r = \lambda/M$, $t_i = \frac{\ln m}{rM}$, and $m = \frac{M}{P_0} - 1$. Recall that t_i is the inflection point. In the program, we can change parameters λ, M, and P_0 (initial population). Change parameter(s) in such a way that the inflection point occurs twice later than in problem 1. Run the program, fit the curve and attach printout with parameter values clearly spelled out.

Problem 3

Modify the `monoferapopulation.cpp` program from Problem 1 so that it prints out, instead of time step t, the "universal" time τ, as defined in class. Verify that the resulting data fit to the "universal" curve.

$$U(\tau) = \frac{1}{1 + 9^{-\tau}}$$

Repeat this for parameters obtained in Problem 2. Have you obtained the same universal curve in both cases? Attach evidence of your claim.