Complexity in social dynamics : from the micro to the macro Laboratory 2

Franco Bagnoli

Namur 7-18/4/2008

Summary

1. Introduction to the laboratory.
2. FORTRAN 95 and gnuplot.
3. Cellular automata.
4. Probabilistic cellular automata (directed percolation). Trajectories and observables.
5. Phase transitions. Phase diagram.
6. The Domany-Kinzel model
7. Mean field approximation.

Introduction to the laboratory

- Language: We shall use FORTRAN 95 + gnuplot
- FORTRAN: ifort (intel), g95, gfortran (GNU)
- gnuplot (preferably versione 4.3)
- For making life easier, I suggest compiling (with a C compiler) a C wrapper for pipes (gnuplot.c, see attached material): gcc -c gnuplot.c
- For testing, try life.f90 (f95 life.f90 gnuplot.o)

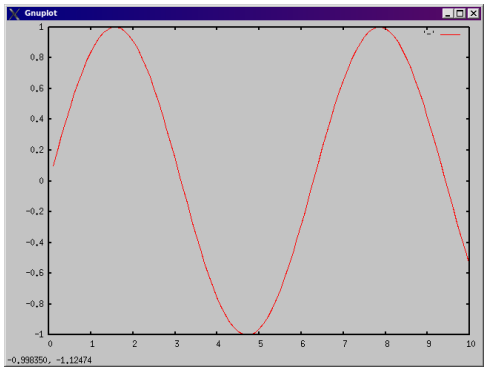
Exercise 1

- Production of data with fortran90: `firstExample.f90`.
- Compilation: `f95 firstExample.f90`. Produces `a.out`.
- Redirection to a file: `./a.out > firstExample.dat`
- Plot:

```
$ gnuplot <cr>
gnuplot> plot 'firstExample.dat' with lines <cr>
gnuplot>
```

Exercise 2

- Piping data from program: `secondExample.f90`
- gnuplot can read from the same stream both commands and data.
- Just use the special filename `'-'` (and write `end` at the end).
- Fortran does not have function for dealing with pipes: use `gnuplot.c`
- Compile it with `gcc -c gnuplot.c`

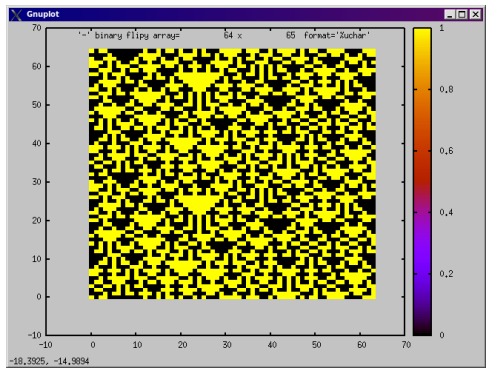


Cellular automata

- Program: `CA.f90`
- We use vector `y(0:N1)`+ for data
- first and last positions are for boundary: with periodi boundary conditions, `y(0)=y(N)` and `y(N+1)=y(1)`
- We use pointers for referring to "windows" on data
- `xl, xc, xr` corresponds to `y(0:N-1)`, `y(1:N1)`, `y(2:N+1)`
- In this way we can elaborate "in parallel" a whole configuration.

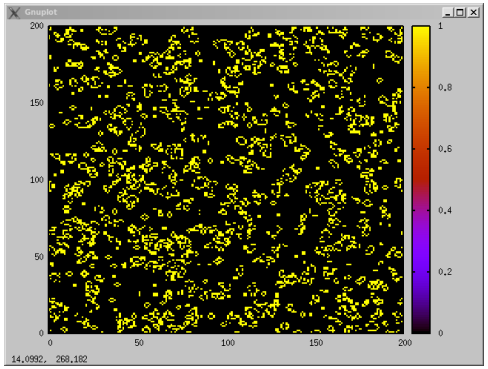
Visualization

- In the second example we sent data to gnuplot in ASCII.
- The program has to convert data from the internal representation to ASCII, and gnuplot has to parse them, and convert them back to the internal representation.
- Passing data in binary speeds up the visualization.
- In this case we have to tell gnuplot the size of data and its format.
- We use the function `gnuplotWrite` (just an interfate to `fwrite`) to send data in binary.



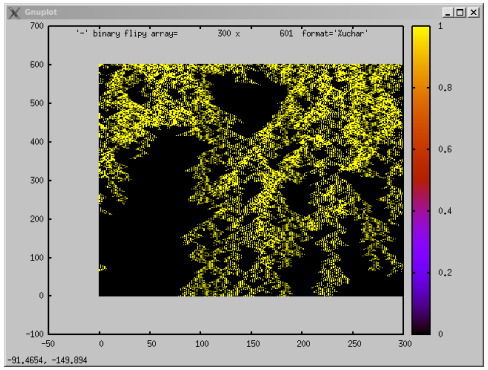
Two-dimensional CA

- Two-dimensional CA are similar: `life.f90`
- In this case we wisualize snapshots of the configuration.
- Try to investigate the variation of the asymptotic state and the transient as a function of the initial density.



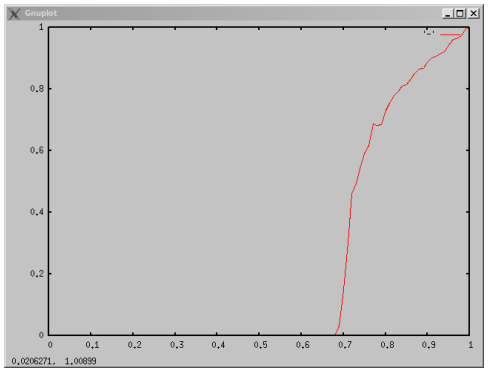
Probabilistic CA

- The simulation of probabilistic CA is similar: `DP.f90`
- In this case the input is the probability p of percolation.
- We have to extract random numbers and compare them with p.
- Varying p we observe different scenarios. Measure the asymptotic density d.



Phase portrait

- We want to plot the asymptotic density d as a function of p:`DPphase.f90`
- We have just to iterate the previous computations over difefrent values of p. We get a "phase portrait".
- Study what happens varying the system size and time length. What about transients before entering the absorbing state?



More parameters

- We can have more than one parameter. The Domany-Kinzel (DK) `DK.f90` model has two control parameters.
- Calling $p(1|A)$ the probability of getting 1 if the sum of neighbors is A, we have : $p(1|0) = 0$, $p(1|1) = p$ and $p(1|2) = q$.
- The directed percolation problem is a particular case of DK, when $p = q$ (bisectrix)
- The phase space is two-dimensional `DKphase.f90`