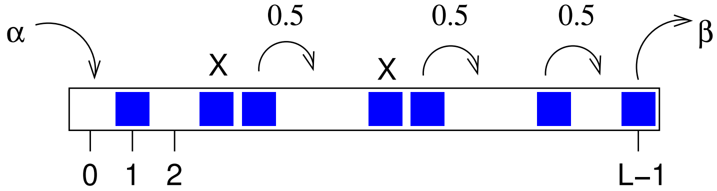
Prof. Dr. A.K. Hartmann
Exercise sheet 2, 12.12.2023

WiSe 2023/24, 5.04.4521

Exercises for Computational Physics

1 Totally Asymmetric Exclusion Process (TASEP)

The TASEP is a simple model for transport in non-equilibrium.
A system consists of a one-dimensional chain of L sites $(0, \ldots, L-1)$, which are either empty (0) or occupied by a particle (1).



The dynamics of the system is described by the following rules. They apply for one "sweep", where the lattice is iterated starting from site $L-1$ until site 0.

1. If the last site is occupied by a particle, the particle leaves the system with probability β.
2. A particle inside the system hops to the next site (only in positive direction, thus "totally asymmetric"), if the next site is empty, with probability 0.5. If the next site is occupied, the particle does not move.
3. If site 0, i.e., the left-most one, is not occupied, it will be occupied with probability α.

Important quantities to measure are the density ρ (fraction of occupied sites) and the current density j (average number of hopping particles per sweep and per number L of sites)

- Download from StudIP the partially completed code `tasep_fragment.c`. Your task will be to complete the code.
- Get to know and understand the data structures, the main program, and the other functions which are already present.

  Hint: All system data is stored in the variable `sys`, which is of self-defined type `tasep_sys_t` .

  You can compile with `cc -o tasep tasep_fragment.c -lm -Wall` .

- Complete the function `tasep_sweep()`, such that the dynamic as explained above is implemented. Test your code with the debugger `gdb` by following some iterations of the loop for the dynamics step by step (while looking at variables). Note that you can set arguments in `gdb` with `set args`, set breakpoints with `break`, start with `run` and, after reaching a breakpoint, perform single-line steps with `step` and investigate variables with `print`.

- Design and implement a function `tasep_density(...)`, which returns the density (fraction of occupied sites). You should pass the system-wide variables `sys` by a pointer. Modify the output which is printed in the main function such that the density is also printed.

- Extend the main program such that not only the current density ρ and current density j are printed, but also the "running" averages $\overline{\rho}, \overline{j}$ as take over the so-far simulation time t.

- Perform simulations ($L = 100$) and "suitable" (test) number of sweeps for three sets of parameters ($\alpha = 0.2, \beta = 0.1$), ($\alpha = 0.1, \beta = 0.2$) and ($\alpha = 0.7, \beta = 0.7$). Measure and plot $\rho(t)$, $j(t)$ or the running averages (most simple with `gunplot` etc). The number of sweeps is suitable if the system is "equilibrated", i.e., the running averages do not change much. Then you will observe the stationary values $\lim_{t\to\infty} \overline{\rho}(t), \lim_{t\to\infty} \overline{j}(t)$?

- Additional tasks:

  How does the equilibration time t_{eq} depend on the system size L?

  Perform a series of simulations for several combinations of the parameters α, β to obtain a phase diagram. Identify different phases according to the values of $\overline{\rho}, \overline{j}$ being "small"/"large" (relatively).