**!!! To work correctly, the program requires the installation of libraries gsl and gnu. !!!**

1. **Behypercycle**

* **behypercycle.cpp –** code written in C ++. It calculates the evolution of the system and records the result in the corresponding files.

The input to the program is 3 files:

**Matrix\_A.txt** – interaction matrix;

**u0.txt** – initial state vector for solving an ODE;

**constr.txt** – restriction on changes in the elements of the interaction matrix when solving the LPP.

Read from keyboard:

The size of the interaction matrix;

The number of iterations of evolution;

The final point in time for the ODE;

Grid for solving ODE;

The step of solving an ODE (at which iterations of evolution we solve an ODE).

* **behypercycle.m –** code written in Matlab. Visualization of data obtained using the **behypercycle.cpp** file**.**

1. **New Elements**

* **new\_elem.cpp –** code written in C ++. It calculates the evolution of the system in the case when new types can be added to the system and writes the result into the corresponding files.

The input to the program is 3 files:

**Matrix\_A.txt** – interaction matrix;

**u0.txt** – initial state vector for solving an ODE;

**constr.txt** – restriction on changes in the elements of the interaction matrix when solving the LPP.

Read from keyboard:

The size of the interaction matrix;

The number of iterations of evolution;

The parameter 0 ≤ λ1 ≤ 1 for generating the first Poisson process, determining whether it is necessary to make changes. The smaller λ1, the less likely it is that changes will be made;

The parameter 0 ≤ λ2 ≤ 1 for generating the second Poisson process, which defines whether to add a new or replace an existing element, in case changes need to be made. The smaller λ2, the less likely it is that you need to add a new element, respectively, the greater the likelihood that you need to replace an existing element;

The final point in time for the ODE;

Grid for solving ODE;

The step of solving an ODE (at which iterations of evolution we solve an ODE);

Maximum number of changes.

* **New\_elem.m –** code written in Matlab. Visualization of data obtained using the file **new\_elem.cpp**.