



## C1 Lithium-Cluster Dynamics AnyLogic 4.1

### Fully Numerical Approach

AnyLogic ([www.xjtek.com](http://www.xjtek.com)) is a general-purpose simulator for discrete but also for continuous and hybrid applications. AnyLogic supports modelling with UML – RT and modelling technology is based on Java so that building simulation models using AnyLogic should be easy for experienced programmers.

**Model:** As you know from java based software, the model is represented by an arbitrary number of Java classes. According to the task of the comparison the complete model is implemented in one Active Object Classes (Root), which is slightly adapted to the subtasks a) – c).

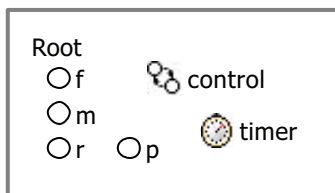


Fig. 1: model layout of root class in AnyLogic

Equations defined in Class Root look like:

```

import
timer.restart(simTime);
Equations
d(f)/dt = d_r*r + 2*d_m*m - k_r*m*f - 2*k_f*f*f - l_f*f + p
d(m)/dt = d_r*r - d_m*m + k_f*f*f - k_r*m*f
d(r)/dt = -d_r*r + k_r*m*f
Additional class code
double startTime=0;
double simRealTime=0;
public void executionControl() {
    for ( int i = 1; i <= nRep; i++) {
        Engine.execute();
    }
}
  
```

This class holds control of model repetition and the definition of the ODEs from the model. Control of the model replication is implemented in the static timer and in the control – state chart.

The set of numerical methods included with the simulation engine can handle both stiff and non-stiff systems and you can use explicit algorithms as well as implicit algorithms to deal with the ODEs. The choice of the numerical method is no part of the model but a setting in the AnyLogic engine-options.

**Results task a: simulation in the time domain comparing integration algorithms.** AnyLogic provides various types of output as charts and real-time values of object variables with the pitfall of increasing demand of system resources and therefore slower execution of simulation. To standardise simulation with different numerical methods, the only generated output is the real simulation time written in an output file (see Figure 2).

Considering the smallest mean simulation-time and sufficient accuracy, it seems to be appropriate to use the default setting (automatic) of the provided ODE – solving method.

	min	max	mean
Automatic	110	230	119,58
Radau (impl.)	110	400	131,34
LSODA (impl.)	260	771	287
RK853 (expl.)	210	230	210,92

Fig. 2: time in [ms] per simulation after 50 replications

**Task b: Parameter variation of  $l_f$ .** The parameter  $l_f$  measures the loss of F – centers at the surface. The value of  $l_f$  will be varied between  $1.0E2$  to  $1.0E4$  in logarithmic step size (additional Java code).

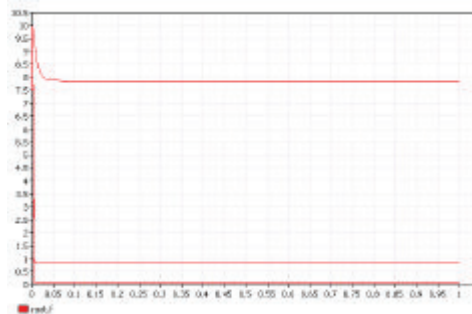


Fig. 3:  $f(t, l_f)$  with constant rate of  $p(t) = 800$

$f(t, l_f)$  tends always to equilibrium depending on the values of the parameter  $p$  and the escape factor  $l_f$ . The plotting function of AnyLogic is unfortunately only capable of linear scaling, so it is hard to visualize the behaviour in the first second.

**Task c: Calculation of steady states.** AnyLogic does not provide any function for computing equilibrium points of a system – so the simulation runs as long as the values for  $df$ ,  $dm$  and  $dr$  converges towards 0 respectively their values become lower than a threshold of  $1.0E-7$ .

The equilibrium points of  $f$ ,  $m$  and  $r$  with bombardment  $p(t) = 0$  converges towards 0. Constant bombardment  $p(t) = 10000$  changes steady states for M – centers and F – centers to  $\sim 10$  and for R – centers to  $\sim 1000$ .

$p(t)$	F - center	M - center	R - center
0	0	0	0
10000	10	10	1000

Fig. 4: Results for steady states

Michael Gyimesi  
mgyimesi@osiris.tuwien.ac.at

