A fully Numerical Approach to ARGESIM Comparison “C1 Lithium-Cluster Dynamics” with Dymola

St. Pawlik, F. Breitenecker, Vienna University of Technology; pawlikb@fsmat.at

Simulator. Dymola, Dynamic Modeling Laboratory, is a simulation environment built for modelling various kinds of physical objects. It uses an object oriented approach for modelling of large, complex and heterogeneous physical systems. A graphical editor and a specific language (Modelica) allow the user to construct models composed of mechanical, electrical and hydraulic subsystems out of predefined models.

Model. Because of the simplicity of the present model there is no need for graphic modelling, therefore a pure textual approach at Modelica level was taken:

```
model C1
… declaration of parameters and variables …
  equation
  der(r)=-dr*r + kr*m*f;
  der(m)=dr*r - dm*m + kf*f^2 - kr*m*f;
  der(f)=dr*r+2*dm*m-kr*m*f-2*kf*f^2-lf*f+p;
end C1;
```

Task a - Comparing ODE Solvers. The System is a highly stiff one. Among the more than 10 ODE solvers five solvers worked efficiently. Two are based on the LSODE Solvers, two on the DASSL code, and the fifth is a generalized RK – method.

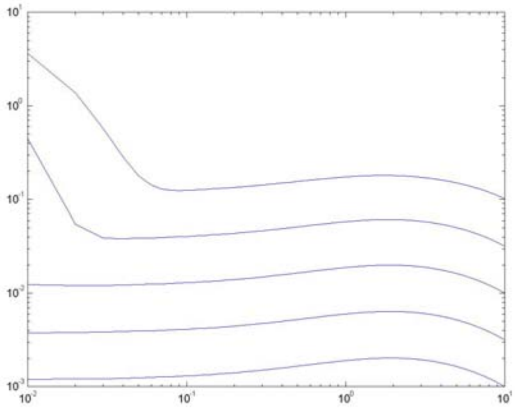
For this task, a relative accuracy of 10^{-6} was set, of the five algorithms used the DASSL-algorithm (DAE solver of Petzold of order 1-5) was the fastest, further results given in the next table. A classical RK4 - algorithm takes about 1.5 s, but one has to choose the stepsize very carefully.

LSODE2	0,36s
LSODARr	0,36s
GRK4T	0,578s
DASSL	0,343s
ODASSL	0,344s

Task b - Parameter Variation: A variation of the parameter l_f from 100 to 10000 in logarithmic steps is to be documented for the function f in a double-logarithmic plot.

Dymola offers now a script language, where such parameter variations could be done, but up to not very comfortable. In this solution the parameter variation was done ‘manually’, performing different runs with precalculated values of l_f . As Dymola’s environment does not support directly logarithmic plots, the resulting time courses of f were exported to MATLAB (supported directly in Dymola).

The following figure shows the MATLAB plot, with 1000 evenly distributed output points for each graph, with $l_f = 10^{(3+i)/2}$, i = 1, …, 5.



Task c: Calculation of steady states: A direct approach is, to simulate in the time domain until the derivatives become zero or almost zero. The model has to be extended by the following lines:

```
when der(r) <= 1e-10 and der(m) <= 1e-10 and
der(f) <= 1e-10 then
       a = r;    b = m;    c = f;
```

Additionally the simulation horizon has to be set large enough. The results are surprisingly exact: a = b = c = 0 without bombardment p(t) and a = 1000, b = 10, c = 10 for a constant bombardment of p = 10000.

Another (relatively new) feature of Dymola is the possibility, that simulation runs can start in steady states. This is implemented in an initial statement by means of an algebraic equation for the derivatives:

```
initial equation der(r) = 0;   der(m) = 0;
der(f) = 0;
```

Dymola solves this equation iteratively with respect to the states, giving accurate results as given above.

C1 Classification: Fully Numerical Approach

Simulator: Dymola 5.1