

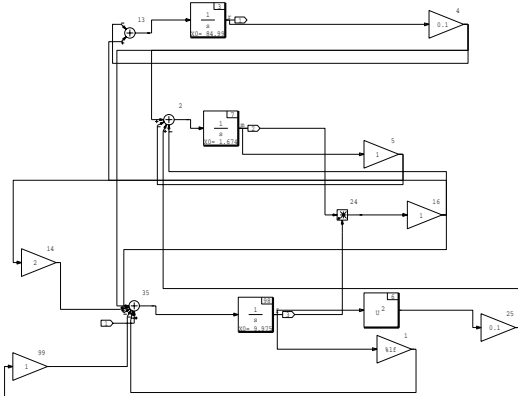


## A fully Numerical Solution to ARGESIM Comparison “C1 - Lithium-Cluster Dynamics” with MATRIXx

W. Weidinger, F. Breitenecker, TU Vienna  
wieweit@fsmath.at

**Simulator:** In 2003, a revival of MATRIXx took place. The Mathworks (MATLAB) had to resell MATRIXx, which is now serviced by National Instruments (NI). NI will distribute MATRIXx as stand-alone product and it will integrate it into NI's system LabView, especially the graphical simulator SystemBuild. MATRIXx is a so-called CNS – a Computer Numeric System, as MATLAB is (and as CONTRL\_C has been. The structure is very similar to MATLAB: it consists of an environment called XMath, and a graphical simulation tool called SystemBuild.

**Model:** SystemBuild allows straightforward modeling using classical basic blocks (integrators, summers, multipliers. The blocks are pasted from the model library and graphically wired (see next figure).

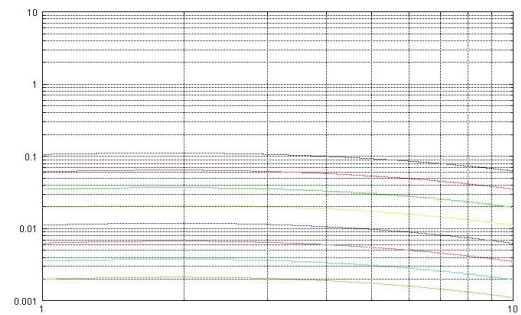


**Task a: Simulation of the stiff system.** Experimentation can be done directly in SystemBuild using menus, where a complete range of solvers is offered (three suited for problems). Best results were obtained with DASSL, based on the DASSL stiff system solver by Linda Petzold, and nearly similar results with ODASSL, a modification of DASSL for overdetermined DAEs. Interestingly, also the standard algorithm (Variable-step Kutta Merson) resulted in correct solutions; other algorithms clearly had to fail. Computation times are given in the following table.

Algorithm	Variable step Kutta Merson	DASSL	ODASSL
One Run	0.451	0.27	0.281

**Task b: Parameter variation of  $l_f$ .** Parameter variations must be made in the environment XMath, which can call SystemBuild for performing simulation runs (command **sim**). Programming in XMath is very similar to MATLAB, m-files (slightly changed) can be used in XMath!. The following program lines perform the logarithmic parameter variation and plot the results (next figure; unfortunately the logarithmic plot caused problems).

```
p = 0;
for n=9:-1:1
    t = [0:0.001:10]';
    u = ones(t);
    lf=10^(1.75+n/4);
    y = sim("c1b",t,u, {ialg=6});
    ymatrix = makematrix(y(3,1)');
    fc(n) = ymatrix;
    ti(n)=t;
endfor
plot(ti(1),fc(1),{log,
    title = "Lithium Cluster Dynamics });
for n=2:9
    plot(ti(n),fc(n),{log, keep});
endfor
```



**Task c: Calculation of steady states.** MATRIXx is a CNS, with matrix manipulation, interpolation, optimisation, and ODE model solving – where the ODE can be modelled graphically in SystemBuild. XMath offers a modified Newton algorithm (in principle for root finding), which can compute as command **trim** steady states (states, inputs and outputs). Using trim for  $p=0$  and  $p=10^4$  gave correct results shown below.

p	r	m	f
0	10e-06	0.001	10e-08
10000	10	10	1000

**C1 Classification: Numerical Approach**  
**Simulator: MATRIXx, Recent Rel. of 2003**

