



C1 Lithium-Cluster Dynamics - Mathematica

Symbolical / Numerical approach

Simulator: Mathematica is an environment for numerical and symbolical calculations in the fields of scientific research, engineering analysis and modeling. It offers an integrated programming environment which allows performing symbolical and numerical tasks with well defined interfaces. Together with a wide range of visualisation commands Mathematica allows the treatment of mathematical / technical problems in a homogenous environment.

Model: The governing equation in this system is an initial value problem of a nonlinear ordinary differential equation. Mathematica offers the possibility for symbolical and numerical treatment. The Dsolve command,

```
Dsolve[equations, dep. vars, indep. vars]
```

tries to solve the given system symbolically. Of course, as expected the solution cannot be given explicitly. Therefore the differential equations have to be treated numerically. Mathematica offers three different numerical methods. For nonstiff equations, a Runge-Kutta-Fehlberg 45 and an Adams Method, for stiff equations the Gear Algorithm is provided. So first, let us analyse the model whether it is stiff or not. This is done symbolically to present Mathematica's features. Define the three equations as g1,g2 and g3:

```
g1=-dr r + kr m r; g2=...; g3=...
```

Then determine the Jacobian Matrix J:

```
J={{D[g1,r],D[g1,m],D[g1,f]},{...},{...}}
```

where the symbol D calculates the differential of the governed equation. The eigenvalues can be computed symbolically with the command

```
Eigenvalues(J)
```

Inserting all data (calculation of steady state see Task c) gives the eigenvalues lying in the range between [-1000, -0.1]. We conclude a stiff system.

Task a: Simulation of the System: Now we discuss the numerical treatment. Use of the NDSolve command

```
NDSolve[eqns + initconds, dep. Variables, indep. Range, Method->Gear]
```

chooses the Gear Method. The simulation duration was 0.1sec. on a 1.3 GHz processor. The methods for nonstiff systems failed to converge. It is worth to point out that NDSolve chooses automatically the options of algorithm and gives satisfactory results.

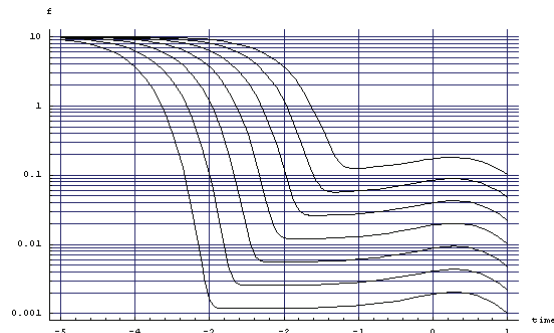
Task b: Parameter Variation: This task shows how to use the rich field of Mathematica Tables. For each parameter we determine the equations and store each function as a table entry

```
Sol=Table[NDSolve[
r[i]'[t]=...,
m[i]'[t]=...,
f[i]'[t]=10^((5+i)/3)f[i][t]...],{i,1,7}]
```

The LogPlot command

```
LogPlot[Evaluate[
Table[f[i][t],{i,1,7}]/.
Table[Sol[[i,1,3]],{i,1,7}]],
{t,-5,1},GridLines-Automatic]
```

can now easily be applied to produce following picture



The very flexible data structures in Mathematica offer easy to implement parameter variation tasks.

Task c: Calculation of steady states: We perform this task completely symbolic. The equations are already defined as g1,g2 and g3. With the replacement operator /. and the Solve command we can get steady state solution as

$$r \circledast \frac{k_f k_r p^3}{d_m d_r l f^3}, m \circledast \frac{k_f p^2}{d_m l f^2}, f \circledast \frac{p}{l f}$$

In the case of p==0 we get the zero solution as steady state. With help of the replacement operator /. we simply can calculate the steady state when p<>0:

```
Solmitp /. {kf->0.1,p->10000,...}
```

and obtain r=1000, m=10, f=10.

As presented above Mathematica offers the framework to easily perform above tasks. Where possible, symbolic calculations were performed. Only in Task a, (and for ODE solving) numerical computations were necessary.

Michael Wibmer
mwibmer@osiris.tuiwien.ac.at