C12 Spheres' Collision – Maple

Numerical approach / hybrid simulation

Simulator: Maple V has been developed as a symbolic formula-manipulation program and became one of the most used programs in this area. In the last years the adding of numerical algorithms formed this program to a software suitable to perform simulation of rather complex processes.

Model: In SNE 28 Almeder used a symbolic approach where the differential equations have been explicitly solved. With an explicit formula for the position and velocity of each sphere he implemented a state event finder. In this work a numerical approach is used and the Maple state event finder is tested.

This comparison solution requires a numerical treatment of differential equation being able to capture state events. Maple V has implemented this feature in two standard numerical solvers for differential equations; these are a Runge Kutta Fehlberg algorithm of 4th order for nonstiff system (RKF45) and a Rosenbrock method for stiff systems.

Maple is no typical simulation software for continuous processes, therefore the implementation of the solution algorithm requires a different attempt compared to "classical" simulation languages like ACSL, Dymola, etc. In these programs one typically implements an algorithms-section where the events and differential equations are defined. The numerical treatment of the events is hided from the user. The programmer defines the event and the actions on that event. The handling is then the task of the simulation software.

Maple does not offer such high-level abstractions. Although the detection of an event is part of the numerical algorithms stated above, the actions after this event have to be implemented by the user. The whole implementation is necessary, not only the definition. In the case of this comparison the treatment of the spheres' collision can be treated by Maple's easy to use interface for event detection. One simply defines the stop condition **stopcond** and adds it as an option to the **dsolve** command:

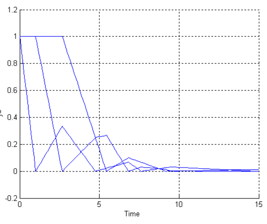
```
dsys:={diff(y1(t),t)=y1p(t),
       diff(y1p(t),t)=0 ,….};
ic:={y1(0)=y10, y1p(0)=y1p0,….};
stopond := y1(t);
s:=dsolve(dsys union ic, numeric,
    stop_cond=stopcond);
```

If an event is detected it returns all necessary information to set actions on this event. **dsolve** returns a list of the all variables. After event detection three different cases can occur in our case.

We assume that within the accuracy limits only one event, that is one sphere's collision at the same time, can occur:

```
if (abs(y1) < tol) then
    … set new initial condition
elif (abs(y2) < tol) then
    … set new initial condition
elif (abs(y3) < tol) then
    … set new initial condition
end if;
s:=dsolve(dsys union ic, numeric,
    stop_cond=stopcond);
```

Task a. Simulation in Time Domain / Final Values of Velocities. The following figure shows the distance-time function for the parameter value e=0.2. The RKF45 algorithm has been used (the equations are not stiff). With the standard option the event finder was not able to detect all the collisions. An increase of the relative and absolute error tolerances up to 10^{-12} was necessary. For e=1 the final velocities are $v_1=v_2=v_3=0$ and $v_4=1$.



Task b. Variation of Restitution Coefficient. The quasiplastic case is reached for e=0.154304556. The figures below show the number of collisions (left) and the final velocities (right) as a function of e, which is varied from 1 to the quasi-plastic case.

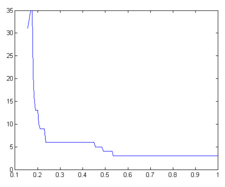


Figure 2

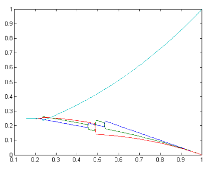


Figure 3

Task c. Boundary Value Problem / Statistical Deviation of Restitution Coefficient. With help of the standard Maple functions **random** and **normald** the value e=0.587401 results in an end velocity $v_4 = v_0/2$.

For a normally distributed e the end velocity of the fourth ball is normally distributed with m = 0.4243, s = 0.0421. The confidence intervals are [0.4216<mu< 0.4269] and [0.0403<s< 0.0440].

Michael Wibmer, TU Vienna
mwibmer@osiris.tuwien.ac.at