Time- and Event-oriented Spreadsheet Modelling of ARGESIM Benchmark C12 'Collision Processes in Rows of Spheres'

Zsofia Kolozsvari[1], Lukas Lienbacher[1], Stefan Perebner[1], Henrique Teixeira Rego[1], Martin Bicher[2*], Andreas Körner[1], Horst Ecker[3], Felix Breitenecker[1]

[1]Mathematical Modelling and Simulation Group, Inst. of Analysis and Scientific Computing,
[2]Inst. of Information System Engineering, [3]Inst. of Mechanics and Mechatronics
TU Wien, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; **martin.bicher@tuwien.ac.at*

SNE 28(4), 2018, 205 - 214, DOI: 10.11128/sne.28.bne12.10452
Received: September 15, 2018; Revised: October 3, 2018;
Revised: November 20, 2018; Accepted: November 30, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This Benchmark Study with educational key aspects presents a spreadsheet-based approach to ARGESIM Benchmark C12 'Collision Processes in Rows of Spheres'. The process, the collision of spheres in a row, is seen as discrete process with a discrete time base, using two modelling approaches. A classical time-oriented approach describes the movement of the spheres by a discrete-time model, the collisions are approximated within the discretization points. The event-oriented approach makes use of the explicit movement formula and determines a sequence of exact collision times and collision events. Both approaches are implemented in the spreadsheet program Excel, which is quite suitable for simulation of discrete processes by means of recursive formulas. The implementation uses standard features of Excel, so that the provided sources can be used in arbitrary spreadsheet programs. Interestingly, with given medium accuracy and sufficient time resolution, time-oriented and event-oriented results coincide. The study also concentrates on educational aspects in giving a sketch on the physical background of elastic and inelastic collisions, in giving hints for proper implementation, and in providing background information on the selection of required experiments with the model (benchmark tasks).

Introduction

The ARGESIM Benchmark C12 *Collision Processes in Rows of Spheres*, defined in SNE in 1999 ([1]), is based on a continuous mechanical model with collision events, but mainly concentrates on discrete events within the movement of the spheres – on the collision of the spheres and tries to analyse the phenomena of collisions, from elastic to plastic, or inelastic, resp. The tasks of the benchmark – the experiments to be performed with the model – require determination of collision sequences depending on the collision type (between elastic and inelastic), boundary value problems for initial hits, analysis of the number of collisions and of final velocities of the spheres, and stochastic analysis for stochastically modelled collisions strength.

A time-oriented and an event-oriented discrete model approach is implemented in the spreadsheet program Excel, which is quite suitable for simulation of discrete processes by means of recursive formulas (in case of time constants of medium or large range). The implementation uses standard features of the spreadsheet tool Excel, so that the provided sources can be used in arbitrary spreadsheet programs.

Section 1 sketches the model and necessary equations, formulas, and algorithms for movement and for collisions. Section 2 presents two discrete model approaches – time-oriented and event-oriented – and the respective implementation in Excel. Section 3 presents results of time domain analysis for specific collision types in a comparative manner for the different implementation approaches. Section 4 shows the results for a structural collision analysis investigating the dependence of the number of collisions and of the final velocities of the spheres on the collisions coefficient. Section 5 sketches the experiment description and the results for a boundary value problem for the initial velocity, and the stochastic analysis of collision strength with results for final velocities.

1 Model Equations

The model consists of four spheres in a row, with positions x_i, velocities v_i, $(\dot{x}_i)$ and parameters mass m_i, diameter d and distance a (Figure 1).

The first sphere starts moving, hits the second, both move, the second hits the third, the first may re-hit the second, the third hits the fourth sphere, etc.

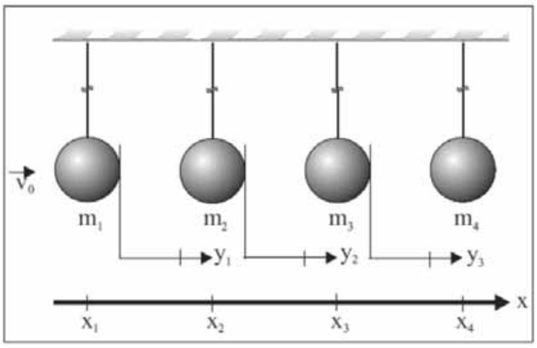

Figure 1: Spheres in a row – pendulum with infinite length.

With development of time, a sequence of collisions occurs – from minimal three collisions up to a theoretically infinite number of collisions n_C. Time instants of collisions are denoted by t_c^a or $t_{c,k}^a$ resp. Movements of the spheres after each collision follow a simple linear translational movement, i.e.

$$x_i(t) = v_{i,c} \cdot t + x_{i,c}, \qquad i = 1, \dots, 4 \leftrightarrow$$

$$\ddot{x}_i(t) = 0,\ \dot{x}_i(t_c^a) = v_{i,c},\ \ x_i(0) = x_{i,c} \qquad (1)$$

Here $x_{i,c}$ and $v_{i,c} = \dot{x}_{i,c}$ denote initial position and initial velocity after a collision: position is continued, and velocity is changed if the i - th sphere is involved, otherwise continued.

The effect of the collision is given by the momentum conservation law for the impact of two masses (Figure 2): the quality of impact is controlled by the collisions coefficient $e, 0 \leq e \leq 1$, or restitution coefficient resp., and may range from elastic collision (collisions coefficient $e_{el} = 1$), via the quasi-elastic collision (all spheres move with same velocity), until the inelastic or plastic collision (collisions coefficient $e_{pl} = 0$).

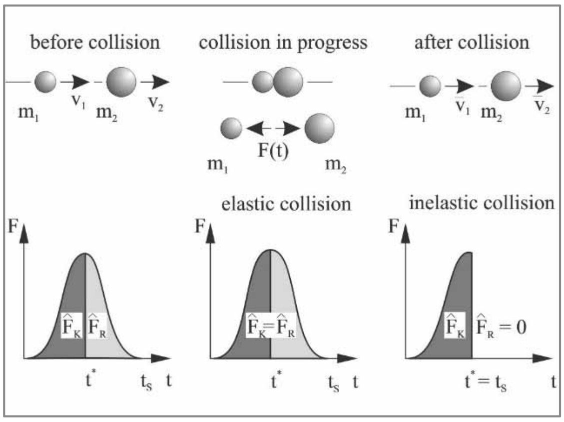

Figure 2: Central impact of two masses – momentum conservation law.

The change of velocities in case of collision at t_c^a is based on the momentum conservation law depending on the collision (restitution) coefficient (Figure 2):

$$v_{i,new} = v_i(1-e)\frac{m_{i+1}}{m_i+m_{i+1}}\ (v_i - v_{i+1})$$
$$v_{i+1,new} = v_{i+1}(1+e)\frac{m_i}{m_i+m_{i+1}}\ (v_i - v_{i+1}) \qquad (2)$$

In order to avoid small faulty differences, relative quantities $y_j = x_{j+1} - x_j - d, j = 1, \dots, 3,\ \dot{y}_j = \dot{x}_{j+1} - \dot{x}_j = v_{j+1} - v_j$ have to be used as further variables.

The process starts with an active hit of the first sphere with an initial velocity v_0:

$$v_{1,c} = \ \dot{x}_{1,c} = v_0 \ \text{at}\ t_{c,0}^a = t_0 = 0, \text{with}\ x_{1,c} = 0$$

resulting in initial values for the other variables with

$$y_{1,c} = y_{2,c} = y_{3,c} = \ a;\ \dot{y}_{1,c} = -v_0,\ \dot{y}_{2,c} = \ \dot{y}_{3,c} = 0$$

so that next movement starting at collision time t_c^a is given by

$$\ddot{x}_1 = 0, \quad \dot{x}_1(t_c^a) = \ \dot{x}_{1,c}, \quad x_1(t_c^a) = \ x_{1,c} \qquad (3)$$
$$\ddot{y}_j = 0,\ \ \dot{y}_j(t_c^a) = \ \dot{y}_{j,c},\ \ y_j(t_c^a) = \ y_{j,c}, j = 1, \dots 3 \qquad (4)$$

The change of velocities at collision $C_{2\times3}$ (collision second sphere –third sphere) at collision time t_c^a is given by momentum law (2):

$$\dot{y}_1 = \dot{y}_1 + \frac{(1+e)*m_3}{m_3+m_2} * \dot{y}_2,$$
$$\dot{y}_3 = \dot{y}_3 + \frac{(1+e)*m_2}{m_3+m_2} * \dot{y}_2,\ \ \dot{y}_2 = -e * \dot{y}_2 \qquad (5)$$

These formulas for the collision $C_{u\times(u+1)}, 2 \leq u \leq n-1$ of two inner spheres is generally valid, also in case of n spheres $n > 4$. Similar formulas hold for collisions with first sphere $C_{1\times2}$ and last sphere $C_{3\times4}$ $C_{(n-1)\times n}$:

$$\dot{x}_1 = \dot{x}_1 + \frac{(1+e)*m_2}{m_1+m_2} * \dot{y}_1, \dot{y}_1 = -e * \dot{y}_1$$
$$\dot{y}_2 = \dot{y}_2 + \frac{(1+e)*m_1}{m_1+m_2} * \dot{y}_1$$
$$\dot{y}_2 = \dot{y}_2 + \frac{(1+e)*m_4}{m_3+m_4} * \dot{y}_3, \dot{y}_3 = -e * \dot{y}_3 \qquad (6)$$

The simple linear translational movement (1) allows to calculate the collision times in advance. After collision time $t_{c,k}^a$, two neighbouring spheres move with

$$x_i(t) = v_{i,c} \cdot t + x_{i,c},\ \ x_{i+1}(t) = v_{i+1,c} \cdot t + x_{i+1,c}$$

and collide if $v_{i+1,c} < v_{i,c}$ and $x_{i+1}(t) = x_i(t)$, given by

$$v_{i,c} \cdot t + x_{i,c} = v_{i+1,c} \cdot t + x_{i+1,c},$$

This results in the relative timespan $t_{c,i\times i+1}^r$ until next collision $C_{i\times(i+1)}$:

$$t = t^r_{c,i\times i+1} = \frac{x_{i+1,c}-x_{i,c}}{v_{i,c}-v_{i+1,c}} = \frac{y_{i,c}}{-\dot{y}_{i,c}} \qquad (7)$$

Depending on the velocity differences, one, two, three, or no collision can occur, where one must be the first. Consequently, the absolute time instant $t^a_{c,k+1}$ is given by

$$t^a_{c,k+1} = t^a_{c,k} + \min_{\exists i\times i+1}(t^r_{c,i\times i+1}) \qquad (8)$$

Obviously, for the dynamics of movement more complicated dynamics can be used, e.g. taking into account air resistance, so that instead of linear translational movement (1) nonlinear dynamics must be used:

$$\dot{x}_i(t) = v_i(\mathrm{t}),\cdot\ \dot{v}_i(\mathrm{t}) = f(x_i, v_i), \mathrm{i} = 1, \dots, 4$$

2 Implementation

Two types of approaches are implemented in MS Excel: a time-oriented approach and an event-oriented approach. The event-oriented approach computes the exact collision times, whereas the time-oriented approach detects the collisions within discrete time progress.

2.1 Time-oriented approach

The time-oriented approach defines a time grid with given step size Δt and calculates the dynamics in a recursive manner by a discretisation of model (1)

$$x_i(t_{k+1}) = v_{i,c}\cdot \Delta\mathrm{t} + x_i(t_k),\ \ \mathrm{i} = 1, \dots, 4, \qquad (9)$$

and analogously for the derivative variables and difference variables according to model (3-4).

In each recursive calculation (9) the occurrence of a collision is checked. If a collision has happened in the last recursion (between t_k and t_{k+1}) then it is handled by means of formulas (5) and (6) at time instant t_{k+1} – in general too late, but proper choice of the timestep can keep the error small.

The straightforward implementation in Excel makes use of the following variables, calculated in rows and partly updated recursively along the time column:

t current time t_k
x1 x2 x3 x4 $x_i(t_k)$ position of spheres
dx1 dx2 dx3 dx4 $\dot{x}_i(t_k)$ velocity of spheres
y1 y2 y3 $y_i(t_k)$ position differences
dy1 dy2 dy3 $\dot{y}_i(t_k)$ velocity differences
s12 s23 s34 collision indicator at (t_k) ('0' or '1')
delta_y1 delta_y2 delta_y3 $y_i(t_k) - y_i(t_{k-1})$ recursive difference of position differences.

Parameters are defined in fixed calls and named:

a, **d**, **v_0**, **m_1**, **m_2**, **m_3**, **m_4**, **e**, **t_0**, **delta_t**, **tend**.

Figure 3 shows the Excel worksheet with parameter definitions and results of the recursive updates, to be explained in more detail in the following.

The implementation of the model and its formulas is straightforward and briefly sketched in Figure 3 for row no. 7, with recursive updates from row no. 6. Only the sometimes necessary distinction of cases with nested **IF** and **AND** constructs seems elaborate.

	A	B	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	U	V	W	X	Y	Z
1	ARGESIM BENCHMARK C12 'Collision of Speres' Time-oriented Approach																						
2																		Impact			Difference: Previous Pos. Diff.		
3	Parameters		Time	Position Variables				Velocity Variables				Position Difference Var.			Velocity Difference Var.			Indicator			minus Actual position Diff.		
4			t	x1	x2	x3	x4	dx1	dx2	dx3	dx4	y1	y2	y3	dy1	dy2	dy3	s12	s23	s34	delta_y1	delta_y2	delta_y3
5	a	1	0,00	0,00	2,00	4,00	6,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	-1,00	0,00	0,00						
6	d	1	0,05	0,05	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,95	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
7	v_0	1	0,10	0,10	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,90	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
8	m_1	1	0,15	0,15	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,85	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
9	m_2	1	0,20	0,20	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,80	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
10	m_3	1	0,25	0,25	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,75	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
11	m_4	1	0,30	0,30	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,70	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
12	e	0,2	0,35	0,35	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,65	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
13	t_0	0	0,40	0,40	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,60	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
14	delta_t	0,05	0,45	0,45	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,55	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
15	tend	15	0,50	0,50	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,50	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
16			0,55	0,55	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,45	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
17			0,60	0,60	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,40	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
18			0,65	0,65	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,35	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
19	Number	13	0,70	0,70	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,30	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
20	Collisions		0,75	0,75	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,25	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
21			0,80	0,80	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,20	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
22			0,85	0,85	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,15	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
23			0,90	0,90	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,10	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
24			0,95	0,95	2,00	4,00	6,00	1,00	0,00	0,00	0,00	0,05	1,00	1,00	-1,00	0,00	0,00	0	0	0	0,05	0,00	0,00
25			1,00	1,00	2,00	4,00	6,00	0,40	0,60	0,00	0,00	0,00	1,00	1,00	0,20	-0,60	0,00	1	0	0	0,05	0,00	0,00
26			1,05	1,02	2,03	4,00	6,00	0,40	0,60	0,00	0,00	0,01	0,97	1,00	0,20	-0,60	0,00	0	0	0	-0,01	0,03	0,00
27			:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:

Figure 3: Time-oriented approach, calculation spreadsheet.

Time Advance

t: **D7 = IF (D6<t_end; D6 + delta_t)**

If the previous time is smaller than the maximum time (15 seconds), the actual time is the previous time plus the time step.

Position update: The actual positions are the previous position plus the previous velocity times the time step:

x_1: **E7 = E6 + delta_t**, x_2, x_3, x_4 alike

Position difference is calculated by differences of positions minus sphere diameter:

y_1: **M7 = F7 - E7 - d**, y_2, y_3 alike

Collisions Detection:

The difference between previous position difference and actual position difference changes sign, if a collision has just occurred:

delta_y1: X7=M6-M7, **delta_y2** and **delta_y3** alike,

the type of collision is given by the impact indicator

s12: U7 = IF (M7<=0; IF (X7>0;1;0); 0)

s23 and **s34** alike, being '**1**' in case of detected collision $C_{1\times2}$, $C_{2\times3}$, or $C_{3\times4}$ resp.

Collision Handling:

New Velocity difference:

If the impact number **s12** is '**1**', the velocity difference **dy1** is calculated with formula (5) for $C_{1\times2}$

dy1: P7 = IF (U7=1; -e*P6;
IF (V7=1; P6+(1+e)*m_3/(m_2+m_3)*Q6; P6))

If the impact number **s23** is one, the velocity difference **dy3** is calculated with formula (5). If both impact numbers **s12** and **s23** are zero, the actual velocity difference **dy1** is the previous velocity difference **dy1** (**dy2** and **dy3** alike).

New Velocity: **dx1: I7 = IF(U7=1;**
I6+(1+e)*m_2/(m_1+m_2)*P6; I6)

If the impact number **s12** is one ($C_{1\times2}$), the actual velocity `dx1` is calculated with formula similar to (5), else the actual velocity is the previous velocity, **dx2**, **dx3** and **dx4** alike, and analogously for collisions $C_{2\times3}$.and $C_{3\times4}$.

The worksheet for the first task (Figure 3) shows the movement for the first 22 timesteps – with step size **delta_t** of 0.05; the first collision is detected and handled at t = **1**, indicated by a one for **s12** (highlighted in orange.

The columns for the collision indicators allow a simple calculation of the number of collisions: indicated in cell **A19** (highlighted in orange): n_C **= SUM (U : W).**

The advantage of the time-oriented approach is the flexibility in the model description – instead of the simple linear translational movement for the position update also a nonlinear movement described by ODEs can be used – instead of the update a Euler solver for the ODE must be used. The disadvantage of the approach is the fact, that all collisions time instants t_c^a are detected with a delay at the next grid time instant t_{k+1}. The classical interpolation strategy of simulators for event detection could help, but is elaborate: the collision time t_c^a could be interpolated between t_k and t_{k+1}.

Most of the up to now 14 published benchmark solutions are based on a simulator, which makes use of an ODE solver with integrated event detection for the collisions, e.g. [2-4].

2.2 Event-oriented approach

The event-oriented approach makes use of the special linear movement dynamics given by model (1), which allows to calculate the next collision time instant by means of the intersection formula (7). Each collision is a state event, which changes the velocities of the involved spheres due to collision formulas (5) and (6). From the viewpoint of events, the collision event is an algorithmic state event: intersection formulas and case-by-case analysis constitute the algorithm for the next collision time with the associated collision type.

The implementation in Excel follows the time-oriented approach, in order to provide a better comparability. Again, in rows the variables are calculated, but each row is now associated with the respective collision time. Variables and parameters are:

tca current absolute collision time t_c^a
tcr current relative collision time t_c^r
x1 x2 x3 x4 $x_i(t_k)$ position of spheres
dx1 dx2 dx3 dx4 $\dot{x}_i(t_k)$ velocity of spheres
y1 y2 y3 $y_i(t_k)$ position differences
dy1 dy2 dy3 $\dot{y}_i(t_k)$ velocity differences
tcr12 tcr23 tcr34 time until next possible collision time instant $t_{c,i\times i+1}^r$ ('`FALSE`' if no collision possible)
cind collision indicator for next collision type.

Parameters are defined in fixed calls and named:

a, **d**, **v_0**, **m_1**, **m_2**, **m_3**, **m_4**, **e**, **t_0**, **tend**.

Figure 4 shows the Excel worksheet with formula implementation. The implementation of the model and its formulas is straightforward and briefly sketched for row no. 9, with recursive updates from row no. 8:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	ARGESIM BENCHMARK C12 'Collision of Speres' Event-oriented Approach																								
2																									
3	Parameters			collision	collision	Positions				Velocities				Position Differences			Velocitiy Differences				relative time until next coll.				collision
4				time abs.	time rel																coll. 1x2	coll. 2x3	coll. 3x4		indicator
5	a	1		tca	tcr	x1	x2	x3	x4	dx1	dx2	dx3	dx4	y1	y2	y3	dy1	dy2	dy3		tcr12	tcr23	tcr34		cind
6	d	1		0	0	0	2	4	6	1	0	0	0	1	1	1	-1	0	0		1	FALSE	FALSE		C12
7	v_0	1		1	1	1	2	4	6	0,4	0,6	0	0	0	1	1	0,2	-0,6	0		FALSE	1,66667	FALSE		C23
8	m_1	1		2,66667	1,66667	1,667	3	4	6	0,4	0,24	0,36	0	0,333	0	1	-0,16	0,12	-0,36		2,08333	FALSE	2,77778		C12
9	m_2	1		4,75	2,08333	2,5	3,5	4,75	6	0,304	0,336	0,36	0	0	0,25	0,25	0,032	0,024	-0,36		FALSE	FALSE	0,69444		C34
10	m_3	1		5,44444	0,69444	2,711	3,733	5	6	0,304	0,336	0,144	0,216	0,022	0,267	0	0,032	-0,192	0,072		FALSE	1,38889	FALSE		C23
11	m_4	1		6,83333	1,38889	3,133	4,2	5,2	6,3	0,304	0,221	0,259	0,216	0,067	0	0,1	-0,083	0,038	-0,043		0,80128	FALSE	2,31481		C12
12	e	0,2		7,63462	0,80128	3,377	4,377	5,408	6,473	0,254	0,271	0,259	0,216	0	0,031	0,065	0,017	-0,012	-0,043		FALSE	2,67094	1,51353		C34
13	t_0	0		9,14815	1,51353	3,761	4,787	5,8	6,8	0,254	0,271	0,233	0,242	0,025	0,013	0	0,017	-0,037	0,009		FALSE	0,35613	FALSE		C23
14	tend	15		9,50427	0,35613	3,852	4,883	5,883	6,886	0,254	0,248	0,256	0,242	0,031	0	0,003	-0,006	0,007	-0,014		5,34188	FALSE	0,22258		C34
15				9,72685	0,22258	3,909	4,938	5,94	6,94	0,254	0,248	0,247	0,25	0,03	0,002	0	-0,006	-8E-04	0,003		5,1193	2,0668	FALSE		C23
16	Number			11,7937	2,0668	4,434	5,451	6,451	7,457	0,254	0,248	0,248	0,25	0,018	0	0,006	-0,006	2E-04	0,002		2,81836	FALSE	FALSE		C12
17	Collisions	13		14,612	2,81836	5,15	6,15	7,15	8,162	0,25	0,252	0,248	0,25	0	5E-04	0,012	0,001	-0,004	0,002		FALSE	0,12545	FALSE		C23
18				14,7375	0,12545	5,181	6,181	7,181	8,194	0,25	0,249	0,25	0,25	2E-04	0	0,012	-9E-04	7E-04	1E-04		0,17344	FALSE	FALSE		C12
19				14,9109	0,17344	5,225	6,225	7,225	8,237	0,25	0,25	0,25	0,25	0	1E-04	0,012	2E-04	2E-04	1E-04		FALSE	FALSE	FALSE		0
20				FALSE																					

Figure 4: Event-oriented approach, calculation spreadsheet

Collision time advance

Row update starts with the selection of the next relative collision time span $t^r_{c,i\times i+1}$ due to minimum selection in formula (7) by:

tcr: E9 = IF (D8>t_end; FALSE; MIN(U8:W8))

If the previous absolute collision time t^a_c is smaller than the final time, time span t^r_c for next collision is the minimum of precalculated relative collision times for all possible collisions.

Now the next (actual) absolute collision time $t^a_{c,k+1}$ can be calculated by adding the actual (next) relative collision time span t^r_c to the previous absolute collision time $t^a_{c,k}$ (formula (7)):

tca: D9 = IF (D8>tend; FALSE; D8+E9)

Position update: The next (actual) positions are the previous position plus the previous velocity times relative collision time: x_1: **F9 = F8 + E9*J8**, x_2, x_3, x_4 alike.

Position differences are calculated by differences of positions minus sphere diameter:

y_1: **N9 = G9 – F9 - d**, y_2, y_3 alike

Velocities, velocity differences

As a collision has happened, velocities and velocity differences must be updated due to formulas (5) and (6) for spheres involved (indicated by precalculated collision indicator **cind**) or must be simply continued:

dy_1: **Q9 = IF (Y8='C12'; -e*Q8;**
IF (Y8='C23';
Q8+(1+e)*m_1/(m_1+m_2)*R8; Q8))

dx_1: **J9=IF(Y8='C12';**
J8+(1+e)*m_2/(m_1+m_2)*Q8; J8)

Next collision determination

The intersection formula (7) allows to calculate the time spans until all next possible collisions

$\ldots t^r_{c,1\times 2}$: **U9 = IF(N9=0; FALSE;**
IF (Q9>=0; FALSE; N9/-Q9))

If the position difference y_i is not zero and the velocity difference dy_i is less than zero, the time span until next collision $t^r_{c,i\times i+1}$ is calculated by the position difference y_i divided with the negative velocity difference dy_i.

The collision indicator **cind** determines, which collision comes next:

Y9 = IF (U9=E10; 'C12'; IF (V9=E10; 'C23';
IF (W9=E10; 'C34'; '0')))

Indeed, there can be maximal two possible next collisions, but there can be also no next (further) collision, so that the sequence of collisions has ended (indicated by type **'0'**).

The number of collisions n_C can be easily determined by counting the number of rows with detected collision type or collision time

n_C: **B17 = COUNTIF (Y6:Y29;<>'0')** or

n_C: **B17 = COUNTIF (D6:D29; >=0)**

In general, the implementation is straightforward, only the case-by-case analysis with its nested **IF** constructs seems to be laborious. The necessary number of rows with recursive updates depends on the number of collisions – which is not known in advance.

For simplicity, a maximal number of rows is implemented, and if collisions stop, collisions type is set to zero, and collision time is set to **FALSE**. (see Figure 4, row 19 and row 20).

3 Task a – Time Domain Analysis

The first task concentrates on time domain analysis for given values of the collision coefficient: first a classical simulation showing the behaviour with numerous hits, and second, the simulation-based analysis of specific collisions cases –- elastic collisions, quasi-plastic collision, and inelastic collision.

3.1 Task a1 – Basic time analysis

The first task requires a graphical representation of the position differences $y_i(t)$ for fixed collision coefficient $e = 0.2$, and initial velocity of first sphere $\dot{x}_1(0) = v_0 = 1$, to be observed on time interval [0,15].

Figure 5 and Figure 6 show the positions and the position differences for the time-oriented approach.

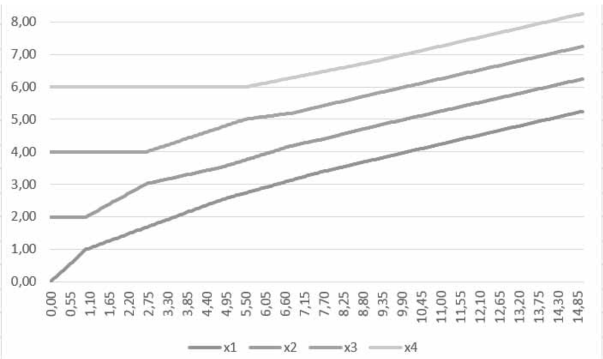

Figure 4: Positions over time - time oriented approach.

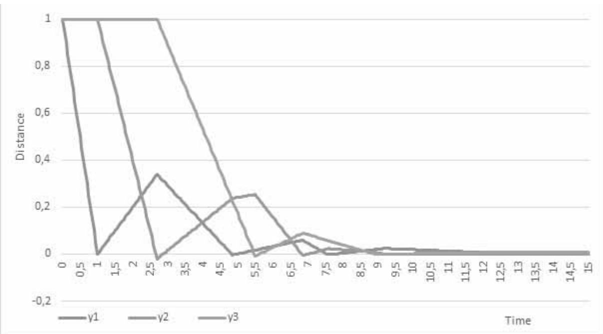

Figure 5: Position differences over time - time oriented approach.

Figure 6 displays the position differences for the event-oriented approach – the time scale grid represents the event times, and interpolation to a synchronous grid shows only small differences to results of the time-oriented approach. A more precise comparison is given in Table 1, which lists the collision times and collision types for both approaches (13 collisions in both approaches). As expected, the collision times in case of the time-oriented approach are delayed: the relative delay is less than the time step, but absolutely the delay is aggregated.

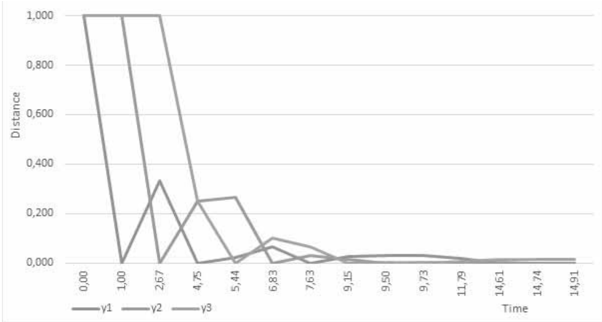

Figure 6: Position differences over collision - event-oriented approach.

Time-oriented Approach		Event-oriented approach	
Collision time	Collision Type	Collision time	Collision Type
1	1-2	1	1-2
2.7	2-3	2.67	2-3
4.85	1-2	4.75	1-2
5.5	3-4	5.44	3-4
6.85	2-3	6.83	2-3
7.6	1-2	7.63	1-2
8.95	3-4	9.15	3-4
9.2	2-3	9.5	2-3
9.25	3-4	9.73	3-4
9.3	2-3	11.79	2-3
13.05	1-2	14.61	1-2
13.1	2-3	14.47	2-3
13.2	1-2	14.91	1-2

Table 1: Comparison of collision times – time-oriented vs. event-oriented approach.

A realistic collision takes time. This fact is not taken into account in the event-oriented solution. However, in the time-oriented solution this impact time is implicitly taken into account by the delay error. Because of this, the solution of the time-oriented approach seems to be more realistic.

3.2 Task a2 – elastic vs. quasi-plastic case

This task requires the simulation for special cases of the collision coefficient e – the elastic case, and the quasi-plastic case.

Elastic case. In the elastic case with $e_{el} = 1$ each collision gives the full moment to the sphere hit, so that the hitting sphere stops, and the sphere hit continues with full velocity. Simple analytical considerations derive the results for the final velocities: the first three spheres are stopped and do not move, so that $x_i(tend) = 0, i = 1,2,3$, and the 4[th] sphere moves on with the initial velocity ($x_4(tend) = v_0 = 1$). Simulation results (Table 2) coincide with the expected values, also for the time-oriented approach, because the collision time instants of the event-oriented approach are a subset of the discrete time grid for the chosen resolution $\Delta t = 0.05$.

Final Velocities elastic case	Time-oriented Approach	Event-oriented approach
$x_i(tend), i = 1,2,3$	0	0
$x_4(tend)$	1	1

Table 2: Final velocities for elastic collisions.

Quasi-plastic case. While elastic collisions and inelastic collision represent generic cases, the quasi-plastic case lies in-between: all spheres move after some time and after some collisions with the same velocity. These movements with same velocity are result of a proper chosen collision coefficient e_{qp}, which forwards energy at sufficient many collisions in that amount, that i) all spheres move with same velocity, and so that ii) no further collisions occur. The proper sequence of collisions results in a dynamic equilibrium.

Simple guess and variation of the collision coefficient could give sufficient accurate approximation of the value for e_{qp}. Simulation trials indicate, that an appropriate collision coefficient must be small –- below 0.25 – and the number of collisions rises above 10.

From algorithmic viewpoint, the search for an appropriate collision coefficient e_{qp} is an optimisation problem, or boundary problem, resp. The boundary value, or the goal function, resp., is expressed in terms of same velocities for all spheres, or equivalently with zero velocity differences for all spheres. An appropriate goal function is for instance:

$$F(e) = \sum_{i=1}^{3} |y_i(t_{end})| \to 0 \qquad (10)$$

Excel provides as standard feature the *Goal Seeking Function* in the *What If Analysis* – suitable for approximating the collision coefficient value e_{qp} for the quasi-plastic case. Figure 7 shows the spreadsheet for this approximation. The cells **B20-B23** contain the final velocities, cells **B25-B27** the final velocity differences, and cell **B28** defines the goal function (10):

B28 = =ABS(B25)+ABS(B26)+ABS(B27)

The *Goal Seeking Function* needs as input the cell of the parameter to be iterated – the value of the collision coefficient e in cell **B12**, the goal function evaluation – formula (10) in B**28**, and the goal value – zero in this case. The *Goal Seeking Function* results in values given in Figure 7 (event-oriented approach), with tuning parameters of a maximum of 1000 iteration steps and a deviation of 0.0001.

	A	B	C	D	E	F
1	ARGESIM BENCHMARK C12 'Collision of Speres' Event-oriented Approacl					
2						
3	Parameters			collision	collision	Positions
4				time abs.	time rel	
5	a	1		tca	tcr	x1
6	d	1		0	0	0
7	v_0	1		1	1	1
8	m_1	1		2,6896588	1,6896588	1,6896588
9	m_2	1		4,5524703	1,8628116	2,4499924
10	m_3	1		5,5446056	0,9921353	2,7571235
11	m_4	1		6,6222959	1,0776903	3,0907395
12	e	0,1836709		7,3578167	0,7355208	3,3184316
13	t_0	0		7,9460849	0,5882682	3,4705597
14	tend	15		8,338899	0,392814	3,5721426
15				8,5554333	0,2165344	3,6281391
16	Number Collisions			9,4994705	0,9440372	3,8722702
17		17		9,6974983	0,1980278	3,9234808
18				9,7075366	0,0100383	3,9260031
19	Final Velocities			9,7198529	0,0123163	3,9290978
20	dx1	0,2500068		11,378927	1,659074	4,344147
21	dx2	0,2499916		11,688162	0,3092356	4,4215081
22	dx3	0,250012		12,186975	0,4988126	4,5462957
23	dx4	0,2499896		14,133557	1,9465822	5,0332705
24	Goal Function			FALSE		
25	dy1	1,518E-05				
26	dy2	-2,035E-05				
27	dy3	2,232E-05				
28	sum_dy	5,784E-05				

Figure 7: Spreadsheet for goal seeking of restitution coefficient for quasi-plastic case - event-oriented approach.

Approach	Time-oriented	Event-oriented
Collision Coefficient	$e_{qp} = 0.191$	$e_{qp} = 0.184$
Final Velocities	$x_i(tend) = 0.25$	$x_i(tend) = 0.25$
Collisions	17	17

Table 3: Collision coefficient for quasi-plastic case - results of Goal Seeking tool.

Table 3 confronts the results for the two modelling approaches: the event-oriented approach results in a smaller coefficient $e_{qp} = 0.184$ than in the time-oriented approach with $e_{qp} = 0.191$ – caused by the delayed collision times of the time-oriented approach. But both approaches manage 17 collisions until the quasi-plastic equilibrium.

It is evident, that the inelastic case (plastic case) with $e_{pl} = 0$ results in spheres moving with same velocity. If the first sphere hits the second, both stick together and move on with half velocity – but double mass !, etc., so that after three collisions all spheres stick together and move with same velocity. The implementation does not regard this special case with 'increasing' mass. On the other hand, the inelastic case happens, if the collision coefficient is decreased until zero – with theoretically infinite number of collisions – see results Task b1.

4 Task b – Dependence on Collision Coefficient

This task analyses the dependence of the number of collisions and of the final velocities on the collision coefficient. Numerical problems for small values of the collision coefficient can be expected. Variation of the collision coefficient e in cell **B12** requires recalculation of the spreadsheet – here Excel macros are of help, which vary the collision coefficient, perform a spreadsheet update calculation, and write results into columns.

4.1 Task b1 – Number of collisions

It is known, that the elastic case $e_{el} = 1$ produces three consecutive collisions ($n_C = 3$) – but with decreasing collision coefficient $e < 1$, multiple collisions occur, so that the number of collisions is increasing ($n_C > 3$).

A simple Excel macro helps to determine the number of collisions $n_C(e)$ dependent on the collision coefficient and stores results (Figure 8):

i) increase of e by increment 0.001 in cell **B12,** and store value in consecutive cell of x–axis columns **AAn** for varying e (start with $e = 0.15$),

ii) update spreadsheet,

iii) store resulting number of collisions n_C from cell **B17** in consecutive cell of y–axis column **AFn** for storing function $n_C(e)$

Figure 8 shows the results for $n_C(e)$ numerically in columns **AAn** and **AFn**, and Figure 9 plots the corresponding graph $n_C(e)$ – using the event-oriented approach. For a collision coefficient $e > 0.54$, only three collisions are occuring ($n_C = 3$). With decreasing collision coefficient $e < 0.54$ the number of collisions $n_C(e)$ is rising, for $e \rightarrow 0$ the number theoretically increases monotonically to infinity: $n_C \rightarrow \infty$. However, there is a limit in reducing the collision coefficient due to general numerical computation limitations – for values $e < 0.15$ computations fail.

The event-oriented approach is able to calculate up to $n_C = 48$ collisions for a value of $e = 0.17$, for values below $e < 0.17$ the results are not reliable (number of collisions $n_C(e)$ seems to decrase) or simply fail.

The time-oriented approach – plot in Figure 10 – is faced with an additional erroneous behaviour. Because of the delay in collision detection, collisions times not only are delayed, but collisions themselves may vanish, if they are to near to the previous ones. As consequence, the time-oriented approach is able to calculate only up to $n_C = 20$ collisions for a value of $e = 0.18$; for values below $e < 0.18$ the computations fail.

	A	B	C	Z	AA	AB	AC	AD	AE	AF
1	ARGESIM BENCHMARK C12 'Collision of Speres' Event-oriented Approach									
3	Parameters				Varation	Final Velocities				number collisions
5	a	1			e	dx1	dx2	dx3	dx4	nC
6	d	1			0,15	0,25	0,25	0,25	0,25	44
7	v_0	1			0,16	0,25	0,25	0,25	0,25	48
8	m_1	1			0,17	0,25	0,25	0,25	0,25	48
9	m_2	1			0,18	0,25	0,25	0,25	0,25	21
10	m_3	1			0,19	0,25	0,25	0,25	0,25	13
11	m_4	1			0,20	0,25	0,25	0,25	0,25	13
12	e	1			0,21	0,25	0,25	0,25	0,25	9
13	t_0	0			0,22	0,25	0,25	0,25	0,25	9
14	tend	15			0,23	0,25	0,25	0,25	0,25	9
15					0,24	0,24	0,26	0,26	0,24	6
16	Number				0,25	0,24	0,26	0,26	0,24	6
17	Collision	3			0,26	0,24	0,26	0,26	0,25	6
18					0,27	0,24	0,25	0,25	0,26	6
:					:	:	:	:	:	:
83					0,92	0,04	0,04	0,04	0,88	3
84					0,93	0,03	0,03	0,03	0,90	3
85					0,94	0,03	0,03	0,03	0,91	3
86					0,95	0,02	0,02	0,02	0,93	3
87					0,96	0,02	0,02	0,02	0,94	3
88					0,97	0,01	0,01	0,01	0,96	3
89					0,98	0,01	0,01	0,01	0,97	3
90					0,99	0,00	0,00	0,00	0,99	3
91					1,00	0,00	0,00	0,00	1,00	3

Figure 8: Spreadsheet with calculation of collisions number final velocities dxi(e) for varying collision coefficient e – event-oriented approach.

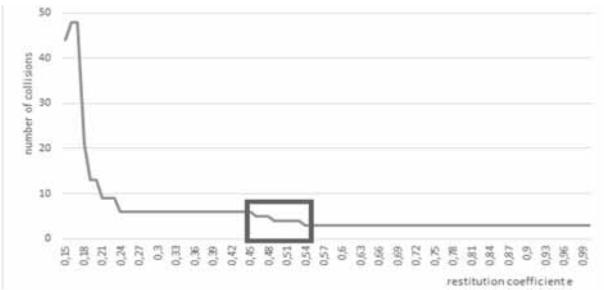

Figure 9: Number of collisions as function of collision coefficient – time-oriented approach.

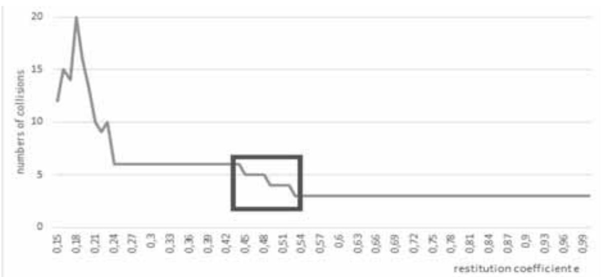

Figure 10: Number of collisions as function of collision coefficient – time-oriented approach.

Delay error and vanishing collisions result in another erroneous phenomenon: the function $n_C(e)$ is not monotonous – in Figure 10, for $e \sim 0.22$ the number of collisions seem to decrease (not only below the 'numerical' border $e < 0.18$). A smaller time step does not really help, because it only shifts the errors in relation to the time step decrease.

Both approaches coincide with the increase of the collision number from $n_C(e) = 3$ to $n_C(e) = 6$ within the range of $e = 0.54$ down to $e = 0.45$, simulating correctly the first re-hits of the spheres (indicated by green rectangles in Figure 9 and Figure 10).

For $e \to 0$, the number of collisions is going to infinity, so that from this viewpoint the plastic case, or the inelastic case resp., is associated with an infinite number of collisions: the spheres do not stick together and move as ‘increased’ mass, the spheres get infinitely near to each other. As consequence, a proper model for the plastic case must follow the strategy of three collisions with increasing mass of ‘multiple’ spheres.

The number of collisions indeed is increasing drastically. A benchmark solution with a high-accuracy computation tool ([5]) results in $n_C(e) = 11216$ for a collision coefficient of $e = 0.1715729$. Interestingly, the number of collisions does not really matter for very small values of the collision coefficient – in any case, the result is the quasi-plastic case; for instance, another high-accuracy benchmark solution ([6]) reaches a minimal collision coefficient of $e = 0.0811$ (with limiting the collision number in the decreasing loop).

4.2 Task b2 – Final velocities

Task 1b already investigated the final velocities of the spheres for the elastic case and for the quasi-plastic case. For further analysis, the final velocities are now investigated as function of the collision coefficient e. The development of $\dot{x}_i(e)$ is expected to be within the range of the elastic case – $\dot{x}_i(e = 1) = 0,\ \dot{x}_4(e = 1) = 1$ – and the quasi-plastic case – $\dot{x}_i(e = 0.184) = 0.25$.

As before, a simple Excel macro determines the final velocities $\dot{x}_i^{tend}(e)$ dependent on the collision coefficient and stores results. Figure 8 shows implementation and the numerical results, Figure 11 plots the functions $\dot{x}_i^{tend}(e)$ generated by the event-oriented approach.

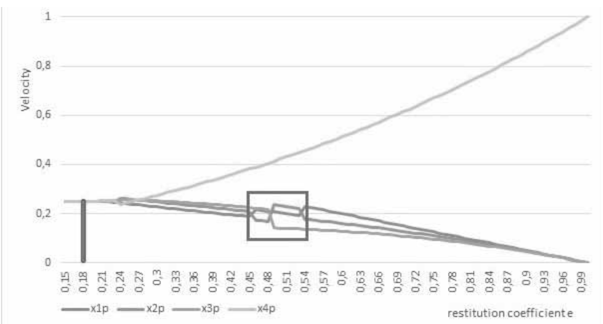

Figure 11: Final velocity of each sphere as function of collision coefficient - event--oriented approach.

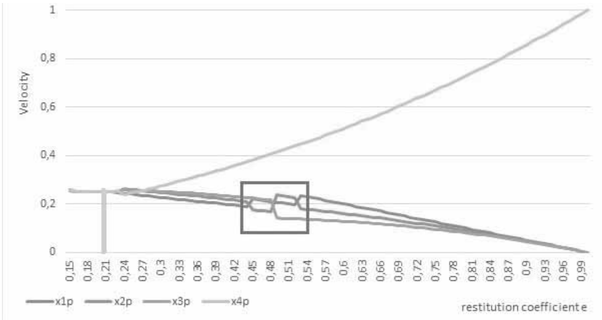

Figure 12: Final velocity of each sphere as function of collision coefficient - time-oriented approach.

The results for the time-oriented approach in Figure 12 show almost no difference to the results of the event-oriented approach in Figure 11. The final velocities of the first three spheres $\dot{x}_i^{tend}(e)$ tend towards zero with an increasing collision coefficient e, and the final velocity of the fourth sphere $\dot{x}_4^{tend}(e)$ rises until one with increasing collision coefficient.

For decreasing collision coefficient all velocities tend towards the value $\dot{x}_i^{tend}(e_{qp}) = 0.25$ and reach the value at the slightly different collision coefficients for the quasi-plastic collision case (red bar in Figure 11 for event-oriented approach, yellow bar in Figure 12 for the time-oriented approach). Interestingly, while the calculations for the number of collisions for collision coefficients below 0.15 fail in both approaches (see Figure 9 and Figure 10), calculations for the final velocities do not fail as given in Figures 11 and Figure 12: although the number of collisions may vary chaotically, the associated changes of velocities are below any significant values.

Both approaches coincide in the changes of the velocities within the range of $e = 0.54$ down to $e = 0.45$, in accordance with the increase of the number of collisions from $n_C(e) = 3$ to $n_C(e) = 6$ (indicated in figures 9-12 by the green rectangle).

5 Task c – Boundary Value – Stochastic Parameters

In most benchmarks, the last task is a challenging one: complex experiments, sophisticated experiment control, model extensions, etc. Also Benchmark C12 requires in the third task elaborate experiments with the model: a boundary value problem, and statistical parameter analysis. But spreadsheet tools offer features for such tasks as standard features. Here the various data analysis features play an important role.

5.1 Task c1 – Velocity boundary value

This task requests the calculation of the collision coefficient $e_{v_0/2}$ which results in half initial velocity for the final velocity of the fourth sphere: $\dot{x}_4^{tend}\left(e_{v_0/2}\right) = \frac{v_0}{2} = 0.5$. Excel provides as standard feature the *Goal Seeking Function* in the *What If Analysis* – suitable for approximating the collision coefficient value $e_{v_0/2}$. Analogously to task b2 a goal function has to be defined, in this case simply

$$F(e) = \left|\dot{x}_4^{tend} - \frac{v_0}{2}\right| \to 0 \qquad \textbf{B28 = =ABS (B25 - v\_0)}$$

The *Goal Seeking Function* needs as input the cell of the parameter to be iterated – e in cell **B12**, the goal function evaluation – the above formula in **B28**, and goal value zero. The *Goal Seeking Function* gives as result the value $e_{v_0/2} = 0.587$ for the event-oriented approach, with four collisions (first re-hit of $C_{1\times 2}$ as fourth collisions). With accuracy limited to three digits, the time-oriented approach gives the same result.

5.2 Task c2 – Statistical analysis

This task extends to some extent task b2, calculation of the dependence of the final velocity on the collision coefficient. With given normally distributed stochastic values for the collision parameter $e_{\mu,\sigma}$, the challenge is to calculate the distribution function, the mean value, the standard deviation and the confidence interval (with confidence probability of 95%) for the final velocity $\dot{x}_4^{tend}\left(e_{\mu,\sigma}\right), \mu = 0.5, \sigma = 0.05$ for a sufficient large sample size.

Excel provides many statistical functions, and again Excel Macros perform the statistical parameter variation and the storage of the final velocities in the spreadsheet (similar as in Figure 7). The stored sample for the final velocities is then statistically analysed by standard features resulting in the following values:

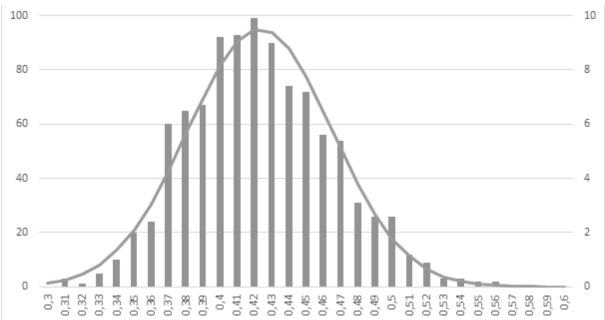

Figure 13: Histogram of the final velocities of the fourth sphere for normal distributed collision coefficient – event-oriented approach.

Mean value: $\ldots\mu(\dot{x}_4^{tend}) = 0.42313\ldots.$

Standard deviation: $\sigma(\dot{x}_4^{tend}) = 0.041907805$

Confidence interval: [0.420532527≤ $\mu \leq$ 0.425727473]

Excel graphic features offer bar charts, used for Figure 13, the histogram for the final velocities of the fourth sphere. Additionally, the formula for the density function allows to plot the distribution approximation (orange curve in Figure 13). With accuracy limited to three digits, the time-oriented approach gives the same result.

6 Conclusion

A spreadsheet tool is definitely not a simulator – modelling features for ODEs, processes, events, etc. are missing. But spreadsheet programs are an excellent experiment environment with statistical analysis, optimisation, what-if analysis, date handling, etc. Of course, macros and external programming could be used, but to some extent the standard features allow to implement discrete dynamic processes – recursive formulas for the development of states along with the flow of time with sufficient time resolution.

It is also possible to implement event-driven dynamics, by calculating the time advance due to state-dependent conditions. This implementation technique is used for the event-oriented approach in this benchmark study. A general disadvantage is the lack of accuracy in the Excel standard configuration – possible but laborious to increase. On the other hand, a spreadsheet tool is a very suitable tool for education, so that this C12 benchmark study is mainly intended for educational use.

References

[1] Hohmann R, Gotzel C, Pöge C. Comparison 12: Collision Processes in Rows of Spheres, Definition. *SNE* vol. 9(3) (*SNE 27*), p. 36-37, 1999.

[2] Hohmann R, Gotzel C, Pöge C. Comparison 12 – ACSL: Hybrid Approach. *SNE* vol. 9(3) (SNE 27), p. 38, 1999.

[3] Filippov A, Kornev A. C12 Spheres' Collision – AnyLogic: Numerical simulation / Time-oriented Model. *SNE* vol. 11(2/3) (*SNE 32/33*), p. 50, 2001.

[4] Lingl M. Comparison 12 – MATLAB/SIMULINK: Continuous Approach – State Events. *SNE* vol. 9(3) (*SNE 27*), p. 39, 1999.

[5] Gotzel C, Hohmann R, Pöge C. Schwerdt J. C12 Spheres'Collision – SLX /LEDA: Analytical Simulation / Event-oriented Model. *SNE* vol. 11(1) (*SNE 31*), p. 31, 2001.

[6] Willensdorfer M, Breitenecker F. Sphere's Collision - FORTRAN Algorithmic Simulation / Time-oriented Model. *SNE* vol. 11(1), (*SNE 31*), p.36, 2001.