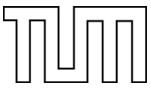

Technische Universität München

Department of Mathematics

Bachelor's Thesis

Evaluation of Time Integrations Schemes in DEM-Simulations

Tobias Hofmann

Supervisor: Frau Prof. Barbara Wohlmuth

Advisor: Frau Prof. Barbara Wohlmuth

Submission Date: 15. August 2011

I assure the single handed composition of this bachelor's thesis only supported by declared resources.

Garching,

Zusammenfassung

Die Software *pe*, kurz für *Physics Engine*, wurde am Institut für Informatik an der Friedrich-Alexander-Universität in Erlangen-Nürnberg entwickelt. Sie ermöglicht Starr-Körper-Simulationen und benutzt hierzu unter anderem *Diskrete Element Methoden*, kurz *DEM*. Mario Heene hat diese Methode erst kürzlich im Rahmen seiner Bachelor-Arbeit in die *pe* implementiert. In vorliegender Arbeit werden die von Heene ausgewählten Zeitintegrationsverfahren hinsichtlich Energieerhaltung beurteilt und mit symplektischen Verfahren verglichen.
Nach Einführung der theoretischen Grundlagen und Beschränkung der Simulation auf unter anderem kugelförmige Körper ohne Rotationsgeschwindigkeit und Ausrichtung wird der Restitutionskoeffizient als Maß der Elastizität eines Stoßes mit einem der drei in der *pe* implementierten Rückstoßkraft-Modelle berechnet. Mit ihm lassen sich Vorhersagen über die zu erwartende Gesamtenergie des simulierten Systems treffen. Schließlich werden die vier zu beurteilenden Zeitintegrationsverfahren definiert, wobei den beiden bereits implementierten Verfahren zwei symplektische von jeweils ähnlicher Konvergenzordnung entgegengesetzt werden.
Im zweiten Teil ergeben Matlab-Simulationen, dass sowohl das verwendete Taylor-Verfahren als auch das adaptive Runge-Kutta-Verfahren die Energie nicht wesentlich schlechter als die neuen, symplektischen Verfahren, der Velocity-Verlet und ein Gauß-Kollokations-Verfahren, erhalten.
Es wird vor einer Implementation in die *pe* eine umfassendere Evaluation und Optimierung hinsichtlich Rechengeschwindigkeit der beiden vorgestellten neuen Verfahren empfohlen.

Contents

1 Introduction

The framework *pe*, short for *Physics Engine*, was developed at the Department of Computer Science at the Friedrich-Alexander-University Erlangen-Nürnberg. It enables the user to calculate rigid body dynamic simulations of granular materials. One of its goals is the fast, but physical accurate simulation. There are different methods of finding and resolving contacts between the bodies. One of these methods, the *Discrete Element Method*, short *DEM* provides in combination with a suited time integration scheme the desired fast calculation and has recently been implemented by Mario Heene. His bachelor thesis [Hee11] is the foundation for this thesis.
Usually, numerical simulations of the Newton equations are mainly used to predict physical quantities like the position and the velocity of the simulated bodies as accurately as possible, whereas energy conservation is of secondary importance. The situation is different with granular materials. Using a time integration scheme with high convergence order yields a good approximation of the physical quantities, but needs much simulation time. If a symplectic scheme of a lower convergence order is applied, similar conclusions in terms of qualitative behaviour can be drawn in less time.
A granular material like sand often reveals unexpected behaviour: Despite its solid state it can behave like a liquid or even a gas under certain circumstances. In [Pös05] it is said that under vibration granular material tends to show convective behaviour". This phenomenon among other things supports the assumption that the qualitative physical behaviour of granular material can be reproduced better by simulation if the applied time integration algorithms account for energy conservation.
It is now of interest whether this implementation of DEM shows the desired physical behaviour like conservation of energy or impulse. In this thesis we compare three symplectic methods of different convergence order versus two explicit, non-symplectic time integration schemes which were used to implement the DEM in the first place.
In the chapter 2 we discuss the theoretical foundations to understand the Discrete Element Methods and in chapter 3 we compare the already implemented algorithms with the proposed new ones.

2 Theory

In this section, we present the required details and definitions of the simulation.

2.1 Particles

Instead of different rigid bodies like spheres, cuboids, cylinders and composite bodies we simulate only the motion of spheres for the simulation time T. Every particle i can be defined by the following quantities

- Position $q_i : [0, T] \Rightarrow \mathbb{R}^3$,
- Velocity $v_i : [0, T] \Rightarrow \mathbb{R}^3$,
- Mass $m_i \in \mathbb{R}^+$ and
- Radius $r_i \in \mathbb{R}^+$.

The *pe* also considers

- Orientation $\phi_i : [0, T] \Rightarrow \mathbb{R}^3$ and
- Rotational velocity $\omega : [0, T] \Rightarrow \mathbb{R}^3$

which are neglected for simplification.
$q \in \mathbb{R}^{3N}$ is the vector of all positions, $v \in \mathbb{R}^{3N}$ is the vector of all velocities, $r \in \mathbb{R}^N$ is the vector of all radices, $m \in \mathbb{R}^{3N \times 3N}$ is the mass matrix

$$\begin{pmatrix} M_1 & & & \\ & M_2 & & \\ & & \cdots & \\ & & & M_n \end{pmatrix}$$

with $M_i \in \mathbb{R}^{3\times 3} = m_i E_3$. m is given as a diagonal matrix instead of a vector in order to simplify algorithm notation.

2.2 Newton's equation of motion

We have to solve the equation of Newtonian mechanics which can be formulated in this context as

$$m\ddot{q} = F(q, v) \Leftrightarrow \begin{cases} m\dot{v} = F(q, v) \\ \dot{q} = v. \end{cases}$$

As we know from [Woh10], these equations yield a Hamilton system if the force F only depends on the position q of the particles with the Hamilton function

$$H(q, v) = \frac{1}{2} v^T v + m^{-1} U(q)$$

with $U(q) := \int F(q)dq$ an antiderivative of F. However, even the simplest implementation of the DEM requires a model for the force F which depends also on v and therefore results in unknown behaviour regarding energy conservation. This aspect is also investigated in this thesis.

2.3 Discrete Element Method

The Discrete Element Method was developed for the simulation of multi-body-dynamics like granular materials, gases, molecules or galaxies. It can be used to simulate different physical phenomenons like friction, recoil, gravity, adhesion or molecular forces. We restrict ourselves to the analysis of the recoil of two colliding particles, as it is the only phenomenon implemented in the *pe*.
To simulate the recoil, the definition of a so-called contact force is needed. If the simulated particles do not touch each other, this force is set to zero. For the simulation, an approach could be to calculate the point of collision of any two bodies, to run the time-integration scheme using large time until this point and then to set a force to recoil the bodies from each other. This leads to another, here not treated version of DEM, called *event-driven DEM*. Still the concept of event-driven simulations is implemented with the use of the adaptive Runge-Kutta-Fehlberg-method. We justify this claim in the section on time integration schemes.
The DEM in *pe* is called *time-driven*. In a time-driven DEM it is possible that after the time-integration scheme completes a step of computing and new values of q and v are available, some bodies overlap with others. This for instance happens if they are moving towards each other with large velocity. This event is called a *collision* and it triggers a force unequal zero in order to resolve the contact.
The key task which characterizes the DEM is the calculation of this contact force.

2.3.1 Contact force

For the components of the force holds

$$F_i(q, v) = \sum_{j \neq i} \tilde{F}_n(\xi_{i,j}, \dot{\xi}_{i,j}) \frac{q_i - q_j}{|q_i - q_j|} \in \mathbb{R}^3$$

where

$$\begin{aligned} \xi_{i,j} &:= r_i + r_j - |q_i - q_j| \\ \dot{\xi}_{i,j} &:= \frac{< v_i - v_j, q_i - q_j >}{|q_i - q_j|}. \end{aligned}$$

From these definitions we obtain important identities such as

$$\begin{aligned} \xi_{i,j} &= \xi_{j,i} \\ \dot{\xi}_{i,j} &= \dot{\xi}_{j,i} \\ F_j &= -F_i. \end{aligned}$$

Later these are of importance.
Furthermore we set

$$\tilde{F}_n(\xi_{i,j}, \dot{\xi}_{i,j}) = \begin{cases} F_n(\xi_{i,j}, \dot{\xi}_{i,j}) & \text{if} \quad \xi > 0 \quad \text{and} \quad F_n(\xi_{i,j}, \dot{\xi}_{i,j}) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The first condition assures that a collision has already taken place.
With the second condition we get that the acting force F_i is not repelling, because the vector $\frac{q_i - q_j}{|q_i - q_j|}$ always points away from the collision point between q_i and q_j.
The physical interpretation is that when the particles collide, they first compress each other at their contact point by a length ξ and then recoil from each other. The collision is modeled as partly elastic and partly inelastic.

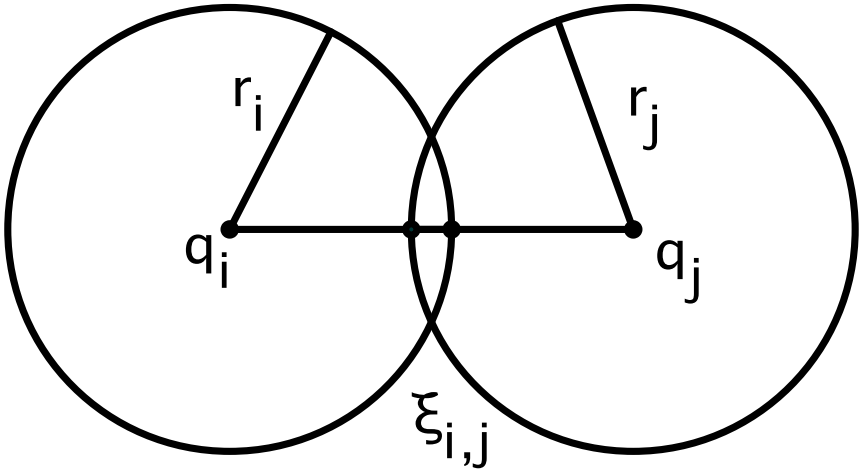

Figure 1: The overlap $\xi_{i,j}$ illustrated by two spheres

If the force only depends on ξ and is continuous, a Hamilton system can be achieved by integration of the force function. In general, this is not possible for a force function depending on $\dot{\xi}$.
Next we introduce the force models of the DEM as it is implemented in *pe*.

2.3.2 Force model parameters

For our simulation we provide three different force models with the parameters

- $\kappa, \gamma, \in \mathbb{R}^+$,
- $\hat{\kappa}, \hat{\gamma} \in \mathbb{R}^+$ or
- $\bar{\kappa}, \bar{\gamma}, \bar{\theta} \in \mathbb{R}^+$.

provided for our simulation which are material-specific and geometry-specific and are used in the calculation of the currently chosen repelling force. They represent the material, of which all of the particles are made. In the *pe* it is of course possible to simulate particles of different materials in the same simulation, this means that there are for example parameters like κ_i and γ_i for every body. This has been neglected here since it is not expected that it influences or complicates our calculation.

2.3.3 Linear force model

This force model is linear in ξ and $\dot{\xi}$ which enables us to solve the resulting system analytically. This is done in 2.6. The force on the particle i is defined as

$$F_n(\xi, \dot{\xi}) = \kappa\xi + \gamma\dot{\xi}. \tag{2}$$

The force is split in two parts with different interpretation. $\kappa\xi$ is interpreted as a conserving part in contrast to $\gamma\dot{\xi}$ which is interpreted as a dissipative part. We observe that the system becomes Hamiltonian for $\gamma = 0$ because

$$F_i(q) = \begin{cases} \kappa\xi \dfrac{q_j - q_i}{|q_i - q_j|} & \text{if} \quad \xi \geq 0 \\ 0 & \text{if} \quad \xi < 0. \end{cases}$$

The continuous force has the differentiable potential function

$$U_i(q) = \begin{cases} \dfrac{1}{2}\kappa|q_i - q_j|^2 - \kappa(r_i + r_j)|q_i - q_j| & \text{if} \quad \xi \geq 0 \\ 0 & \text{if} \quad \xi < 0. \end{cases}$$

which justifies the description *conservative* for the addend containing κ.
As the general linear force model is not a Hamiltonian system, we can not expect that symplectic integrators conserve energy or impulse. Nevertheless, with $\kappa \gg \gamma$ we expect the system to behave nearly symplecticly, so we choose symplectic integrators for our benchmarks.

2.3.4 Non-linear Kuwabara-Kono force model

In the second model the force is defined with the two parameters $\hat{\kappa}$ and $\hat{\gamma}$ as

$$F_n(\xi, \dot{\xi}) = \hat{\kappa}\xi^{\frac{3}{2}} + \hat{\gamma}\dot{\xi}\xi^{\frac{1}{2}}.$$

This model requires that $\hat{\gamma}$ is calculated dependent of the radii of the colliding spheres.

2.3.5 Extended non-linear force model

The third model is a generalization of the second one and is defined with the three parameters $\bar{\kappa}$, $\bar{\gamma}$ and $\bar{\theta}$ as

$$F_n(\xi, \dot{\xi}) = \bar{\kappa}\xi^{\frac{3}{2}} + \bar{\gamma}\dot{\xi}\xi^{\bar{\theta}}.$$

The non-linear models were mentioned for reasons of completeness and shall not be further analyzed.

2.4 Boundaries

It is handy to implement boundaries from which the particles recoil, because this gives us the opportunity to simulate many collisions of different angles and velocities in one scenario. Otherwise we would have to simulate many scenarios with different, wisely chosen initial values in order to test the algorithms in all realistic cases.

It seems reasonable to implement planes as boundaries. We define the contact point as the center of the intersection circle of the colliding sphere and the boundary plane. The solution provided by [Hee11] for the calculation of an appropriate contact force can then be interpreted as an assumption of a theoretical sphere at the contact point with the following properties:

- Position q at the contact point,
- Mass $M = \infty$ and
- Radius $R = 0$.

The contact force can then be calculated based on the original contact force model. Thus basically we can imagine a boundary plane as a collection of an infinite number of spheres where exactly one sphere is chosen for the collision with the simulated sphere. Moreover, from this definition we get the contact normal $n_k \in \mathbb{R}^3$ as the normal vector of the plane.

2.5 Energy

There are two different types of energy which we have to keep in mind. First the potential energy

$$E_{pot}(t) = g \sum_{i=1}^{n} m_{i,i} z_i(t)$$

resulting from the z-component of the bodies and second the kinetic energy

$$E_{kin}(t) = \frac{1}{2} \sum_{i=1}^{n} v_i(t)^T m_{i,i} v_i(t) = \frac{1}{2} v^T m v.$$

In simple mechanics, the sum of both energies is constant. This yields well-known Hamilton systems. In DEM-simulations this sum is not constant over time but decrease with every resolved collision. We show this in the following chapter. Our new goal is to keep the energy of the simulated system as near as possible to the real energy of the system in terms of DEM-Simulation.

2.6 Coefficient of Restitution

It is our interest to know how much energy is lost during a collision. This can be described with the so-called coefficient of restitution.
In this subsection, we calculate the coefficient of restitution which is used to calculate the amount of energy that is lost during a single DEM-collision. For the linear force model it can be derived analytically which is done in this section. For the non-linear force model only estimates based on experiments are available. The coefficient ϵ is defined as

$$\epsilon := \frac{v_{col}}{v_0} \tag{3}$$

where v_0 and v_{col} stand for the relative normal velocity of two bodies before and after a collision, respectively. Thus we expect that $\epsilon \in (0, 1)$ with $\epsilon \to 1$ for $\gamma \to 0$, as this means that the dissipative part disappears.

2.6.1 A differential equation

As a preparation we solve the differential equation given by

$$\begin{aligned} m\ddot{x} + \gamma\dot{x} + \kappa x &= 0 \\ x(0) &= 0 \\ \dot{x}(0) &= v_0 \end{aligned} \tag{4}$$

for $m, \gamma, \kappa, v_0 > 0$ and

$$4m\kappa - \gamma^2 \geq 0. \tag{5}$$

The characteristic polynomial of this differential equation

$$P(\lambda) = m\lambda^2 + \gamma\lambda + \kappa$$

has zeros at

$$\lambda_{1,2} = -\frac{\gamma}{2m} \pm \frac{\sqrt{\gamma^2 - 4m\kappa}}{2m} = -\frac{\gamma}{2m} \pm i\frac{\sqrt{4m\kappa - \gamma^2}}{2m}.$$

With these zeros, we obtain the two linearly independent solutions of the differential equation

$$\begin{aligned} x_1(t) &= \exp\left(-\frac{\gamma}{2m}t + i\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right) \\ x_2(t) &= \exp\left(-\frac{\gamma}{2m}t - i\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right) \end{aligned}$$

which span the solution space. Equation 5 tells us that the argument of the exponential function is already partitioned in the real and the imaginary part. With basis transformation we can characterize the solution space as a linear combination of the two solutions

$$\begin{aligned} x_1(t) &= \exp\left(-\frac{\gamma}{2m}t\right)\sin\left(\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right) \\ x_2(t) &= \exp\left(-\frac{\gamma}{2m}t\right)\cos\left(\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right) \end{aligned}$$

so that the solution can be written as

$$x(t) = A\exp\left(-\frac{\gamma}{2m}t\right)\sin\left(\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right) + B\exp\left(-\frac{\gamma}{2m}t\right)\cos\left(\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right)$$

with $A, B \in \mathbb{R}$. With the first initial value condition from Equation 4, we can see that $B = 0$. The second initial value condition gives us the equation

$$A\frac{\sqrt{4m\kappa - \gamma^2}}{2m} = v_0 \quad \Leftrightarrow \quad A = \frac{2mv_0}{\sqrt{4m\kappa - \gamma^2}}$$

for A. Thus the solution for the differential Equation 4 is

$$x(t) = \frac{2mv_0}{\sqrt{4m\kappa - \gamma^2}} \exp\left(-\frac{\gamma}{2m}t\right) \sin\left(\frac{\sqrt{4m\kappa - \gamma^2}}{2m}t\right).$$

2.6.2 Calculating the coefficient of restitution

In this paragraph, we derive the coefficient of restitution for a simplified example. In [Bie09] it is explained why this conclusion applies in general circumstances.
We consider two spheres of the same material with radii r_1 and r_2 which are located at $(0,0,0)^T$ and $(r_1 + r_2, 0, 0)^T$, respectively and have the velocities $(v_{1,0}, 0, 0)^T$ and $(v_{2,0}, 0, 0)^T$, respectively. We assume that

1. $v_{1,0} > v_{2,0}$,
2. there is no gravity or other body forces,
3. the linear force model is applied,
4. it holds $q_1(t) < q_2(t)$ for the whole collision time and
5. $4m_e\kappa - \gamma^2 > 0$.

Please note that assumption 1 is needed for a collision scenario and assumption 4 is justified by the claim of physical correctness: If $q_1(t)$ is larger than $q_2(t)$, both spheres would have been compressed to half their sizes.
With the initial conditions the Newtonian mechanics are reduced to

$$\begin{aligned} F_1(t) &= m_1\ddot{q_1}(t) \\ F_2(t) &= m_2\ddot{q_2}(t) \end{aligned} \tag{6}$$

with the initial conditions

$$\begin{aligned} q_1(0) &= 0 \\ q_2(0) &= r_1 + r_2 \\ \dot{q_1}(0) &= v_1 \\ \dot{q_2}(0) &= v_2. \end{aligned} \tag{7}$$

Application of assumption 4 removes the modulus in the definition of ξ:

$$\xi(t) = r_1 + r_2 + q_1(t) - q_2(t).$$

With the initial conditions and the assumption 1 we ensure that $\xi(0) = 0$ and $\xi(t) > 0$ for $0 < t < \varepsilon$ and assumption 2 tells us that there is no other force acting on the particles. Thus with assumption 3 it holds that

$$\begin{aligned} F_1(t) &= \left(\kappa\xi(t) + \gamma\dot{\xi}(t)\right) \frac{q_1(t) - q_2(t)}{|q_1(t) - q_2(t)|} &&= -\kappa\xi(t) - \gamma\dot{\xi}(t) \\ F_2(t) &= -F_1(t) &&= \kappa\xi(t) + \gamma\dot{\xi}(t). \end{aligned} \tag{8}$$

The fraction has been cancelled by use of assumption 4.
If we derive the definition of ξ two times, we get

$$\begin{aligned}\xi(t) &= r_1 + r_2 + q_1(t) - q_2(t)\\ \dot{\xi}(t) &= \dot{q}_1(t) - \dot{q}_2(t)\\ \ddot{\xi}(t) &= \ddot{q}_1(t) - \ddot{q}_2(t).\end{aligned}$$

Now we put Equation 6, 7 and 8 together and get

$$\begin{aligned}m_1\ddot{q}_1 &= -\gamma\dot{\xi} - \kappa\xi\\ m_2\ddot{q}_2 &= \gamma\dot{\xi} + \kappa\xi\\ \xi(0) &= 0\\ \dot{\xi}(0) &= v_{1,0} - v_{2,0} =: v_0.\end{aligned}$$

We can take q_1 and q_2 out of the system by multiplying the first equation with m_2, the second one with m_1 and subtracting:

$$\begin{aligned}m_1m_2\ddot{\xi} &= -(m_1+m_2)(\gamma\dot{\xi} + \kappa\xi)\\ \xi(0) &= 0\\ \dot{\xi}(0) &= v_1 - v_2 =: v_0.\end{aligned}$$

We introduce the *effective mass* $m_e := \frac{m_1m_2}{m_1+m_2}$ and get the system

$$\begin{aligned}m_e\ddot{\xi} + \gamma\dot{\xi} + \kappa\xi &= 0\\ \xi(0) &= 0\\ \dot{\xi}(0) &= v_0\end{aligned}$$

which is Equation 4 and therefore has the solution

$$\xi(t) = \frac{2m_ev_0}{\sqrt{4m_e\kappa - \gamma^2}} \exp\left(-\frac{\gamma}{2m_e}t\right) \sin\left(\frac{\sqrt{4m_e\kappa - \gamma^2}}{2m_e}t\right)$$

which is only valid as long as $\tilde{F}_n(t) = \kappa\xi + \gamma\dot{\xi}$. The conditions given by Equation 1 tell us that this is true as long as $\xi(t) \geq 0$ and $\kappa\xi + \gamma\dot{\xi} \geq 0$. We define

$$t_0 := \min_{t>0}\{\xi(t) = 0\}$$

and assume that

$$\forall t \in (0, t_0) : \kappa\xi(t) + \gamma\dot{\xi}(t) \geq 0. \tag{9}$$

Now we are able to calculate the *collision duration* t_0 with the equation

$$0 = \xi(t_0) = \sin\left(\frac{\sqrt{4m_e\kappa - \gamma^2}}{2m_e}t_0\right).$$

Equivalent hereto

$$\frac{\sqrt{4m_e\kappa - \gamma^2}}{2m_e} t_0 = \pi$$

as $\frac{\sqrt{4m_e\kappa-\gamma^2}}{2m_e} > 0$ holds with the common values for κ and γ. This gives us

$$t_0 = \pi \frac{2m_e}{\sqrt{4m_e\kappa - \gamma^2}} = \pi \left(\frac{\kappa}{m_e} - \left(\frac{\gamma}{2m_e} \right)^2 \right)^{-\frac{1}{2}}.$$

As interim result, we calculate

$$v_{col} = \dot{\xi}(t_0) = v_0 \exp\left(-\frac{\gamma}{2m_e} t_0 \right) = v_0 \exp\left(-\frac{\gamma}{2m_e} \pi \left(\frac{\kappa}{m_e} - \left(\frac{\gamma}{2m_e} \right)^2 \right)^{-\frac{1}{2}} \right). \tag{10}$$

We dete

$$v_{col} = \dot{\xi}(t_0) = v_0 \exp\left(-\frac{\gamma}{2m_e} t_0 \right) = v_0 \exp\left(-\frac{\gamma}{2m_e} \pi \left(\frac{\kappa}{m_e} - \left(\frac{\gamma}{2m_e} \right)^2 \right)^{-\frac{1}{2}} \right). \tag{11}$$

rmine the coefficient of restitution given by Equation 3 as

$$\epsilon := \frac{v_{col}}{v_0} = \exp\left(-\frac{\gamma}{2m_e} \pi \left(\frac{\kappa}{m_e} - \left(\frac{\gamma}{2m_e} \right)^2 \right)^{-\frac{1}{2}} \right). \tag{12}$$

This is consistent with equation 2.9 in [Pös05]. Assumption 5 implies that - for given parameters κ and γ - there is a model-dependent minimum effective mass of the simulated colliding bodies given by

$$m_{eff,min} = \frac{\gamma^2}{4\kappa}.$$

For two particles having the same mass m and therefore the effective mass $m_e = \frac{m}{2}$, the mass m cannot be smaller than $\frac{\gamma^2}{\kappa}$, because below this value the solution space of the differential equation has the hyperbolic functions in the base vectors expressions instead of the trigonometric functions. This leads to a solution where $\xi(t)$ does not have any zeros for $t \in (0, \infty)$. The repelling force is not only too small to separate the bodies, but their overlap ξ gets even arbitrary large for $t \to \infty$. The conclusion is that it is not possible to simulate DEM-collisions with the linear force model for masses smaller than $\frac{\gamma^2}{\kappa}$.
We still need to justify Equation 9. For this we calculate

$$
\begin{aligned}
F_n(t_0) &= \kappa\xi(t_0) + \gamma\dot{\xi}(t_0) \\
&= \kappa\frac{2m_e v_0}{\sqrt{4m_e\kappa - \gamma^2}} \exp\left(-\frac{\gamma}{2m_e}t\right) \sin\left(\frac{\sqrt{4m_e\kappa - \gamma^2}}{2m_e}t_0\right) \\
&+ \gamma\frac{\gamma v_0}{\sqrt{4m_e\kappa - \gamma^2}} \exp\left(-\frac{\gamma}{2m_e}t_0\right) \sin\left(\frac{\sqrt{4m_e\kappa - \gamma^2}}{2m_e}t_0\right) \\
&+ \gamma v_0 \exp\left(-\frac{\gamma}{2m_e}t_0\right) \cos\left(\frac{\sqrt{4m_e\kappa - \gamma^2}}{2m_e}t_0\right) \\
&= -\gamma v_0 \exp\left(-\frac{\gamma}{2m_e}t_0\right) \\
&< 0
\end{aligned}
$$

This means

$$\exists t_1 \in (0, t_0) : \tilde{F}_n(t_1) = 0$$

because

$$F_n(t_1) < 0$$

and Equation 1.
Therefore the calculation of the coefficient of restitution is only valid if attracting forces are allowed.

2.6.3 The resulting energy differences

Now let us consider some equations of the original differential equation system:

$$
\begin{aligned}
m_1\ddot{q}_1 &= -\gamma\dot{\xi} - \kappa\xi \\
m_2\ddot{q}_2 &= \gamma\dot{\xi} + \kappa\xi \\
\ddot{\xi} &= \ddot{q}_1 - \ddot{q}_2,
\end{aligned}
$$

The sum of the first two equations gives

$$m_1\ddot{q}_1 + m_2\ddot{q}_2 = 0.$$

Elimination of $\ddot{q}_1$ leads us to

$$(m_1 + m_2)\ddot{q}_2 = -m_1\ddot{\xi}$$

and therefore

$$
\begin{aligned}
\ddot{q}_1 &= \frac{m_2}{m_1 + m_2}\ddot{\xi} \\
\ddot{q}_2 &= -\frac{m_1}{m_1 + m_2}\ddot{\xi}.
\end{aligned}
$$

After integration and using the initial conditions we obtain the following expression

$$\dot{q}_1(t) = \frac{m_2}{m_1+m_2}\dot{\xi}(t) - \frac{m_2 v_0}{m_1+m_2} + v_{1,0}$$
$$\dot{q}_2(t) = -\frac{m_1}{m_1+m_2}\dot{\xi}(t) + \frac{m_1 v_0}{m_1+m_2} + v_{2,0}$$

Thus the (normal) velocities after the collision are:

$$\dot{q}_1(t_0) = \frac{m_2}{m_1+m_2}\epsilon v_0 - \frac{m_2 v_0}{m_1+m_2} + v_{1,0} = \frac{m_2 v_0}{m_1+m_2}(\epsilon - 1) + v_{1,0}$$
$$\dot{q}_2(t_0) = -\frac{m_1}{m_1+m_2}\epsilon v_0 + \frac{m_1 v_0}{m_1+m_2} + v_{2,0} = -\frac{m_1 v_0}{m_1+m_2}(\epsilon - 1) + v_{2,0}$$

The energy before the collision is

$$E_0 = \frac{1}{2}m_1\dot{q}_1^2(0) + \frac{1}{2}m_2\dot{q}_2^2(0) = \frac{1}{2}(m_1 v_{1,0}^2 + m_2 v_{2,0}^2),$$

whereas the energy after the collision is

$$\begin{aligned}
E_1 &= \frac{1}{2}m_1\dot{q}_1^2(t_0) + \frac{1}{2}m_2\dot{q}_2^2(t_0) = \\
&= \frac{1}{2}\left(m_1\left(\frac{m_2 v_0}{m_1+m_2}(\epsilon-1) + v_{1,0}\right)^2 + m_2\left(\frac{m_1 v_0}{m_1+m_2}(\epsilon-1) - v_{2,0}\right)^2\right) \\
&= \frac{1}{2}\left(\frac{m_1 m_2^2 v_0^2 + m_2 m_1^2 v_0^2}{(m_1+m_2)^2})(\epsilon-1)^2 + 2\frac{m_1 m_2 v_0(\epsilon-1)}{m_1+m_2}\left(v_{1,0} - v_{2,0}\right)\right) + \frac{1}{2}(m_1 v_{1,0}^2 + m_2 v_{2,0}^2) \\
&= \frac{1}{2}m_e v_0^2\left((\epsilon-1)^2 + 2(\epsilon-1)\right) + \frac{1}{2}(m_1 v_{1,0}^2 + m_2 v_{2,0}^2) \\
&= \frac{1}{2}m_e v_0^2\left(\epsilon^2 - 1\right) + \frac{1}{2}(m_1 v_{1,0}^2 + m_2 v_{2,0}^2).
\end{aligned}$$

This results in the following energy difference of collisions:

$$\begin{aligned}
\Delta E &= E_1 - E_0 \\
&= \frac{1}{2}m_e v_0^2\left(\epsilon^2 - 1\right) + \frac{1}{2}(m_1 v_{1,0}^2 + m_2 v_{2,0}^2) - \frac{1}{2}(m_1 v_{1,0}^2 + m_2 v_{2,0}^2) \\
&= \frac{1}{2}m_e v_0^2\left(\epsilon^2 - 1\right)
\end{aligned} \tag{13}$$

Thus the expected energy loss can be easily calculated from the effective mass of the colliding bodies.

2.6.4 General cases

In this example we restrict ourselves to a very special case where there is no tangential motion. In [Bie09] it was found that this result can be generalized to all collisions, so that Equation 11 holds for all collisions if normal velocities are taken into account.
At the end of this section, we take a look at the applicability of the coefficient of restitution to general cases with $v_t \neq 0$. ξ, the collision time and the coefficient of restitution depend on the motion along the tangential plane. However, as we don't consider tangential forces,

there is no correcting factor for the loss of energy by tangential motion, but we take only the normal velocities before the collision into account. This means, in a collision between particles x_i and x_j, the ΔE given by Equation 12 is added to E_0.

2.7 Time integration schemes

The two already implemented methods are complemented by two new symplectic methods which were chosen to achieve the desired properties.

2.7.1 Taylor expansion *Taylor*

The Taylor expansion has been the first method utilized by [Hee11] for the DEM implementation. It was taken out of [KE08] where it was described as

$$\begin{aligned} q(t+\Delta t) &= q(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \frac{1}{6}b(t)\Delta t^3 \\ v(t+\Delta t) &= v(t) + a(t)\Delta t + \frac{1}{2}b(t)\Delta t^2, \end{aligned} \tag{14}$$

where

$$\begin{aligned} a(t) &= \frac{d}{dt}v(t) = m^{-1}F(t) \\ b(t) &= \frac{d}{dt}a(t) = m^{-1}\frac{d}{dq}F(t)\cdot\frac{d}{dt}q(t). \end{aligned}$$

If some does not want to take additional time for calculating this derivative of the force function or the force function is not differentiable, it is possible to estimate $b(t)$ which is done in [Hee11] by a finite backward difference, namely

$$b(t) = \frac{d}{dt}a(t) \approx \frac{a(t) - a(t-\Delta t)}{\Delta t} =: \hat{b}(t).$$

This results in the algorithm

$$\begin{aligned} \hat{q}(t+\Delta t) &= q(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \frac{1}{6}(a(t) - a(t-\Delta t))\Delta t^2 \\ \hat{v}(t+\Delta t) &= v(t) + a(t)\Delta t + \frac{1}{2}(a(t) - a(t-\Delta t))\Delta t. \end{aligned} \tag{15}$$

This raises the question whether the new method has the same convergence order as the original one. We show that the approximation does not affect the convergence order of the algorithm if $b \in C^1$.
The local consistency fault for the original method is given in [KE08] as

$$\begin{aligned} q(t+\Delta t) &= \tilde{q}(t+\Delta t) + O(\Delta t^4) \\ v(t+\Delta t) &= \tilde{v}(t+\Delta t) + O(\Delta t^3), \end{aligned} \tag{16}$$

where $\tilde{q}$ and $\tilde{v}$ are the correct solutions and $q(t) = \tilde{q}(t)$ and $v(t) = \tilde{v}(t)$ for a fix t. Next we derive the difference between the backward difference quotient $\hat{b}(t)$ and the derivative $b(t)$. Using Taylor series gives

$$\begin{aligned}\hat{b}(t) &= \frac{a(t) - a(t-\Delta t)}{\Delta t} \\ &= \frac{a(t) - a(t) + \Delta t \cdot b(t) - \frac{1}{2}\Delta t^2 \frac{d}{dt} b(t) + O(\Delta t^3)}{\Delta t} \\ &= b(t) - \frac{1}{2}\Delta t \frac{d}{dt} b(t) + O(\Delta t^2).\end{aligned} \tag{17}$$

Subtracting 14 and 13 we get the local consistency fault:

$$\begin{aligned}\hat{q}(t+\Delta t) - q(t+\Delta t) = \frac{1}{6}(\hat{b}(t) - b(t))\Delta t^3 \Leftrightarrow \\ \hat{q}(t+\Delta t) = q(t+\Delta t) + \frac{1}{6}(\hat{b}(t) - b(t))\Delta t^3 \\ \hat{v}(t+\Delta t) - v(t+\Delta t) = \frac{1}{2}(\hat{b}(t) - b(t))\Delta t^2 \Leftrightarrow \\ \hat{v}(t+\Delta t) = v(t+\Delta t) + \frac{1}{2}(\hat{b}(t) - b(t))\Delta t^2.\end{aligned}$$

Together with 16 and 15 we get:

$$\begin{aligned}\hat{q}(t+\Delta t) &= \tilde{q}(t+\Delta t) + O(\Delta t^4) + \frac{1}{6}(-\frac{1}{2}\Delta t \frac{d}{dt} b(t) + O(\Delta t^2))\Delta t^3 \\ &= \tilde{q}(t+\Delta t) - \frac{1}{12}\frac{d}{dt} b(t)\Delta t^4 + O(\Delta t^4) \\ &= \tilde{q}(t+\Delta t) + O(\Delta t^4) \\ \hat{v}(t) &= v(t+\Delta t) + O(\Delta t^3) + \frac{1}{2}(-\frac{1}{2}\Delta t \frac{d}{dt} b(t) + O(\Delta t^2))\Delta t^2 \\ &= \tilde{v}(t+\Delta t) - \frac{1}{4}\frac{d}{dt} b(t)\Delta t^3 + O(\Delta t^3) \\ &= \tilde{v}(t+\Delta t) + O(\Delta t^3).\end{aligned}$$

This means that the same consistency fault and the same convergence order is achieved. If we consider the linear force model, we recognize that the force function and the acceleration a are not differentiable. b can not defined in the linear force model, thus the Taylor expansion is not expected to assure a truncation error of $O(\Delta t^4)$ for positions and $O(\Delta t^3)$ for velocities if the linear force model is applied.

2.7.2 Velocity Verlet *VelVer*

We are using the velocity Verlet as proposed for DEM in [Bie09] and [KE08]. There it is defined as

$$\begin{aligned}v(t + \frac{\Delta t}{2}) &= v(t - \frac{\Delta t}{2}) + m^{-1}\left(F(q(t), v(t - \frac{\Delta t}{2}))\right)\Delta t \\ q(t+\Delta t) &= x(t) + v(t + \frac{\Delta t}{2})\Delta t\end{aligned}$$

The method has the disadvantage that the calculation of positions and velocities is performed with a shift of $\Delta t/2$ between both of them. However, using very small time steps

the velocity Verlet still provides a good estimate of the position and velocity at the same point of time.
The convergence order of this method is two and needs one force evaluation per simulation step, is not adaptive or iterative, is symplectic in Hamiltonian systems and was therefore chosen as a comparable scheme for the Taylor expansion.

2.7.3 Adaptive Runge-Kutta-Fehlberg-method *Rk45*

The adaptive Runge-Kutta method is a method of fourth or fifth order which uses the Butcher tableau listed in Figure 2 and has three additional parameters, the minimum step size h_{min}, the maximum step size h_{max} and the tolerance ϵ.

$$
A = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
\frac{3}{32} & \frac{9}{32} & 0 & 0 & 0 & 0 \\
\frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & 0 & 0 & 0 \\
\frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & 0 & 0 \\
-\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{40} & 0
\end{pmatrix}
$$

$$
b_4 = \begin{pmatrix} \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & \end{pmatrix}
$$

$$
b_5 = \begin{pmatrix} \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55} \end{pmatrix}
$$

Figure 2: Butcher tableau for the adaptive Runge-Kutta-Fehlberg-method

The method uses two different Runge-Kutta methods from order four and five, respectively, which compute two different approximations for the next iterative.

$$
(q_4, v_4) = \Psi_4(q, v, h)
$$
$$
(q_5, v_5) = \Psi_5(q, v, h)
$$

The first Runge-Kutta method is given by the sub-matrix $(A_4)_{ij} = A_{ij}$ for $i, j \in \{1, \ldots, 5\}$ and b_4. The second Runge-Kutta method is given by the total matrix $A = A_5$ and b_5.
We define the error q_e as

$$
q_e := ||q_4 - q_5||_\infty
$$

This value is used to estimate the consistency error made during this step. If the error is too large, the step gets rejected and is recalculated with a smaller step size.
The authors have different opinions about whether q_5 and v_5 or q_4 and v_4 should be taken as next iterative in case of a successful step. In [Hee11], the solution of lower order q_4 and v_4 were chosen.
The maximum time step h_{max} is arbitrarily set as 10^{-3} to avoid having to calculating the step again with smaller step size due to rejection based on large collision forces. The minimum time step h_{min} is provided by the user. Another fair approach would be to try to

adjust the average step size to the user-provided time step, but this would require further evaluation during the simulation. The time step for the next iteration is then chosen as

$$h = \begin{cases} h_{min} & \text{if} \quad h_{opt} < h_{min} \\ h_{max} & \text{if} \quad h_{opt} > h_{max} \\ h_{opt} & \text{otherwise,} \end{cases}$$

where $h_{opt} = h_{opt}(h, q_e)$ is an estimated optimum time step.
There seems to exist different definitions on how to choose a proper step size and how to define the acceptance criterion. We explain why the current implementation is not useful and propose a better solution based on the original source in [YCL09].
The current implementation as described in [Hee11] calculates the optimum step size as

$$h_{opt} = \frac{\epsilon h}{q_e}$$

and then accepts the step if

$$h_{opt} \geq \frac{3}{4} * h.$$

This leads us to the following simplification: The step is accepted if

$$q_e \leq \frac{4}{3}\epsilon,$$

which is equivalent to an absolute error criterion.

Adaptive Runge-Kutta-Fehlberg-method with another step size strategy *Rk45N*

In [YCL09] h_{opt} is defined as

$$h_{opt} = \left(\frac{\epsilon h}{2q_e}\right)^{\frac{1}{4}} h$$

which leads to the condition

$$\frac{q_e}{h} \leq \frac{4^4}{3^4}\frac{\epsilon}{2}$$

for the acceptance of the step. This is equivalent to a relative error criterion which implies that the error q_e is allowed to be larger if the step size h is also larger in contrast to the current implementation which results in the condition that the error has to be below a certain bound. In the section about step size strategy in [DR08] it is told that this strategy fits the higher convergence order of this method better.
Additionally, we took the estimate q_5 as the result of this method.
In [Hee11] ϵ was set as 1 or 0,5. Regarding the condition for the acceptance, it seems obvious that the tolerance ϵ has to be chosen with respect to the expected size of q.
If the acceptance criterion is not fulfilled for a calculation with minimum step size, the simulation gets aborted and is classified as failed.

2.7.4 Gauß collocation methods *GauCol* and *GauCo2*

In order to compare the adaptive Runge-Kutta of fourth order with another symplectic scheme of higher order, a symplectic Gauß collocation method of fourth order out of [Hai06] was chosen. Its Butcher tableau is given in Figure 3.

$$A = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \end{pmatrix}$$
$$b = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Figure 3: Butcher tableau for the symplectic Gauß collocation method

The solution algorithm can be formulated with the definition of Runge-Kutta methods out of [Hai06] as

- set $k_1 = 0$ and $k_2 = 0$ as initial value (1) or take the results of the previous iteration step (2),
- calculate k_1^* and k_2^* explicitly according to the defining equation of k_i,
- repeat the previous step as long as $max(||(k_2^* - k_2)||_\infty, ||(k_1^* - k_1)||_\infty) > TOL$ and
- calculate $q(t + \Delta t)$ and $v(t + \Delta t)$ according to the defining equation of y_1.

Whether both k_i are set to zero as initial value or the results of previous iteration are taken, leads to the Gauß collocation method *GauCol* and the Gauß collocation method *GauCo2*, respectively.
In [Hai06] there is also a chapter about different algorithms for solving implicit Runge-Kutta schemes, but it was disregarded due to the limited extent of this thesis.

2.8 Summary of the algorithm properties

The properties of the algorithms are summarized in Figure 4.
The adaptive Runge-Kutta-Fehlberg method calls more function evaluations per time step because it is possible that the current calculated step is not accepted and a recalculation with new function evaluations is required.
In general the Gauß collocation method calls more function calls per time step because the sequence of the Runge-Kutta coefficients requires more than one iteration step to converge against its limit.

3 Comparison by numerical simulation

In this chapter we explain our basic scenario for the simulation and draw conclusions from the graphical outcomes.

Algorithm	Function evaluation calls per step (Minimum)	Convergence order
Taylor expansion	1	2 and 3
Velocity Verlet	1	2
Adaptive Runge-Kutta-Fehlberg-method	6	4 or 5
Gauß collocation method	2	4

Figure 4: Properties of the different algorithms in comparison

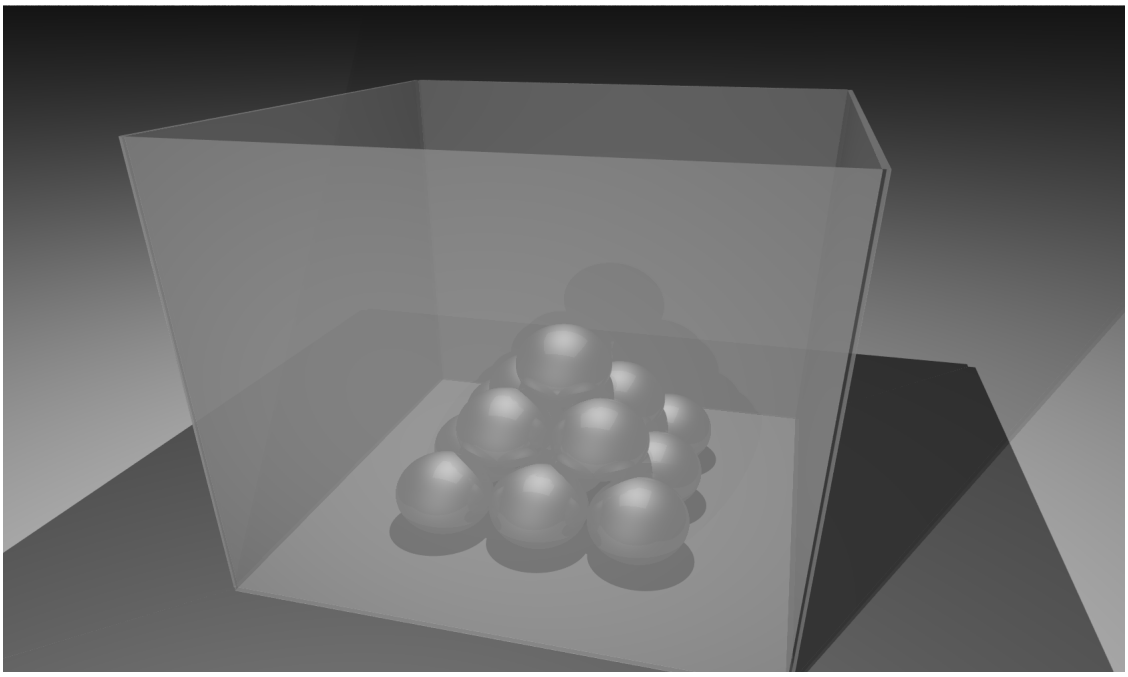

Figure 5: Initial state of the scenario

3.1 Simulation scenario and measured parameters

We chose a scenario with thirteen aluminium spheres of ten centimeters in diameter built as a pyramid inside a cube with a side length of one meter and used the linear force model. The spheres were piled up nearly touching each others with velocity zero. Even if they were put exactly on each other, they would not stay fixed due to the lack of static friction in our simplified model.
In Figure 5 we can see the initial state of this scenario which was then simulated with different time steps $h \in \{10^{-3}, 10^{-4}, 10^{-5}\}$.
We calculated the total energy of the system 25 times per second. This is equal to a timeframe of 0.04 seconds per evaluation.
Additionally in each Figure we trace the expected energy according to the Equation 12. At the point in time of a registered collision, the plot of the estimated total energy immediately drops by the calculated value for ΔE. As all tested algorithms increase the energy the velocities calculated by our simulation are larger than the true velocities. Therefore the energy difference ΔE which is subtracted from the initial energy is too

large because it is increasing with the collision velocity $v_0 = v_1 - v_2$. Altogether the expected energy traced by our algorithms is accurate as long as the calculated velocities are accurate.
Further simulations could show in what way this expected energy based on the theoretical energy loss in the DEM collisions can be enhanced.
It was also measured

- how much time the simulation requires,
- how many function calls are requested and
- how many steps the method takes.

Even if not much optimizing was done for the implementation this results should provide a decent insight in the underlying character of the methods and what is to be expected if they are applied for the solution of DEM.
The value ϵ for the adaptive Runge-Kutta-Fehlberg-methods and the tolerance TOL in the Gauß collocation methods have both been chosen to 10^{-4}, as this reflects a sufficient accuracy of the visualized solution.

3.2 Comparison of the Taylor expansion with the Velocity Verlet

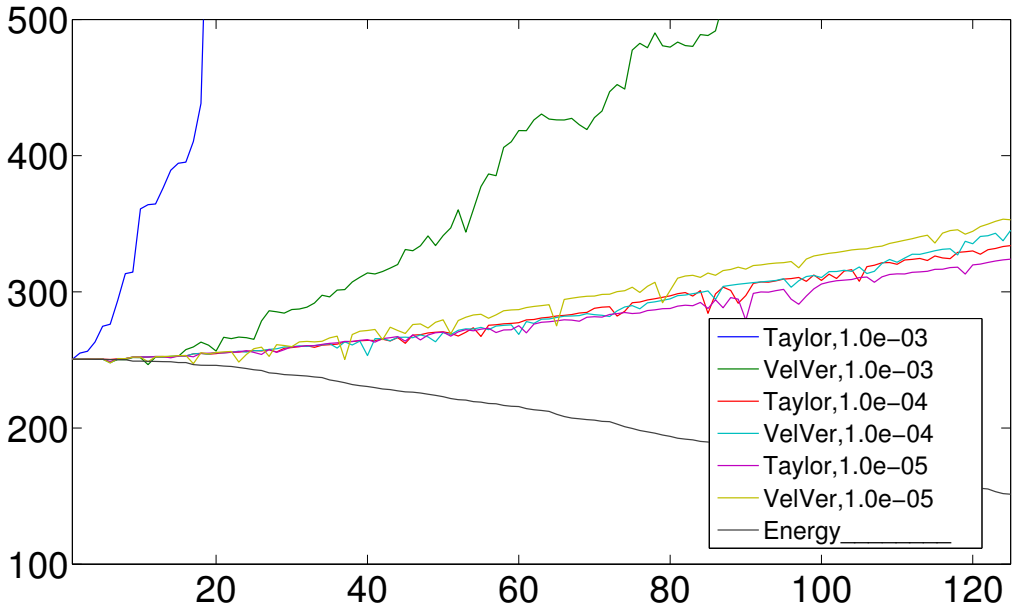

Figure 6: Energy plot of Taylor expansion (*Taylor*) versus velocity Verlet (*VelVer*)

We simulated the scenario for five seconds or 125 data points shown in Figure 6. The starting energy of nearly 250 was clearly increasing by both algorithms which is a known issue for explicit time integration schemes. With the greatest time step of 10^{-3}, the Taylor

expansion delivered poor results whereas the velocity Verlet at least did only triple the energy. If a time step of 10^{-4} or 10^{-5} was applied made little difference. Both algorithms computed good results.
To interrelate the multiplication of the energy with the physical consequences, two pictures in Figure 7 were generated from the data given by the velocity Verlet with time step 10^{-3} and 10^{-5} which multiplied the energy after five seconds roughly by 3 and 1.5, respectively. Even though no velocities can be illustrated in this diagram, one can guess from the elevation of the different bodies that the simulation on the left side is not reflecting the true physics.

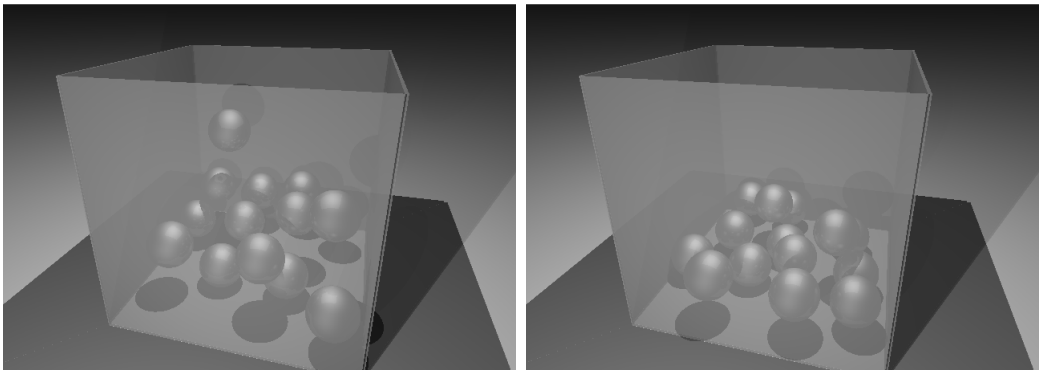

Figure 7: Pictures based on simulation with the velocity Verlet with $h = 10^{-3}$ and $h = 10^{-5}$ after five seconds

In terms of calculation time, the experiment met our expectations. As both algorithms have neither any inner iteration nor adaptive step sizes their computation time depends linear on the evaluation time for the contact force and the number of requested time steps. The existing differences can be explained by the normal fluctuation in computer calculation and not optimized algorithm implementation.

Algorithm	Step size	No. time steps	Simulation time (seconds)
Taylor expansion	10^{-3}	5000	11
Taylor expansion	10^{-4}	50000	48
Taylor expansion	10^{-5}	500000	483
Velocity Verlet	10^{-3}	5000	6
Velocity Verlet	10^{-4}	50000	48
Velocity Verlet	10^{-5}	500000	477

Figure 8: Simulation data of the Taylor expansion *Taylor* and the velocity Verlet *VelVer*

3.3 Comparison of both step size algorithms in the adaptive Runge-Kutta-Fehlberg-method

The scenario was run for five seconds. The decrease of the minimum time step did not have strong influences on the algorithm. The first method chose time steps $h \geq 10^{-3}$ thus they get the same results for $h_{min} \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. The method with new step size algorithm did not converge for the greatest time step, but calculated the same results for $h_{min} \in \{10^{-4}, 10^{-5}\}$. This is reflected in Figure 9.

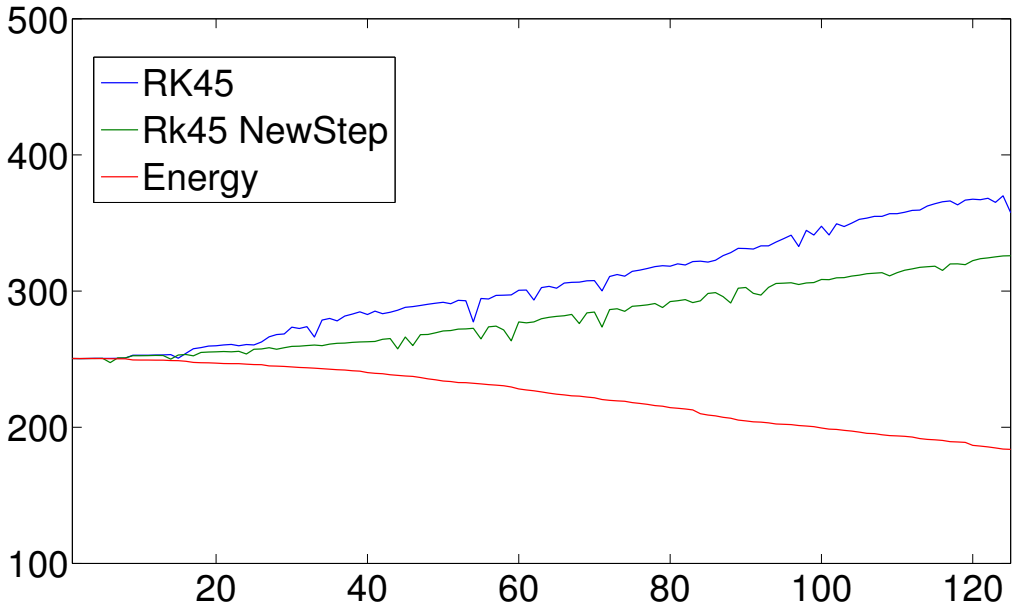

Figure 9: Energy plot of the adaptive Runge-Kutta-Fehlberg-methods *Rk45* and *Rk45N*

The new algorithm requires more calculation time and more function calls, but gets better energy conservation based on the higher convergence order. The simulation time is linearly dependent of the number of function calls.

Algorithm	Minimal step size	No. function calls	Simulation time	Average time step
Rk45	10^{-3}	30000	22	$1,0 \cdot 10^{-3}$
Rk45	10^{-4}	30000	20	$1,0 \cdot 10^{-3}$
Rk45	10^{-5}	30000	20	$1,0 \cdot 10^{-3}$
Rk45N	10^{-3}	109	0.1	-
Rk45N	10^{-4}	44773	29	$7,4 \cdot 10^{-4}$
Rk45N	10^{-5}	44773	29	$7,4 \cdot 10^{-4}$

Figure 10: Simulation data of the adaptive Runge-Kutta-Fehlberg-methods *Rk45* and *Rk45N*

3.4 Evaluation of the initial value setting in the Gauß collocation method

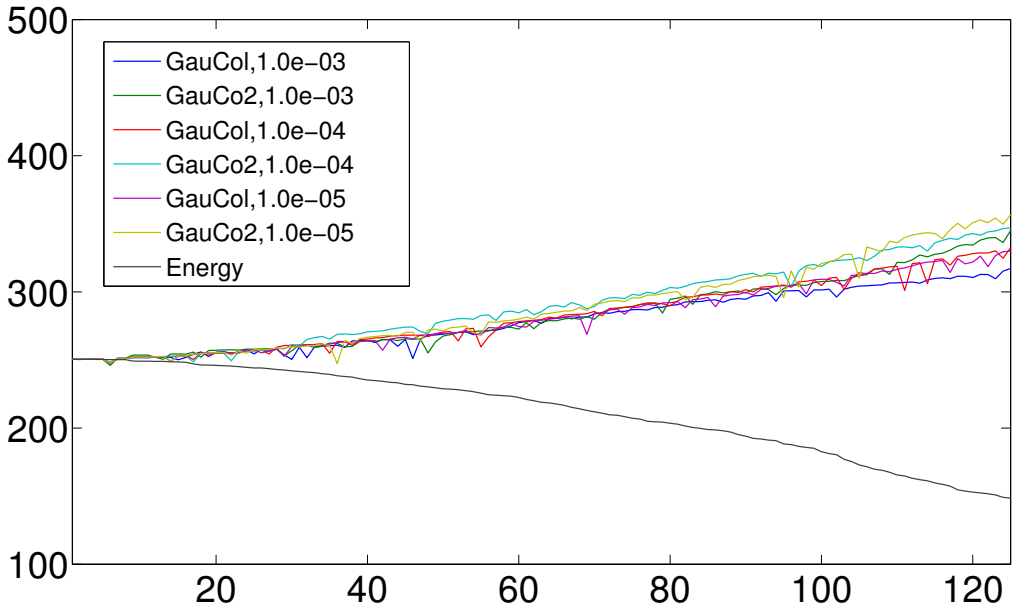

Figure 11: Energy plot of the Gauß collocation methods *GauCol* and *GauCo2*

We simulated the scenario for five seconds. Both methods for the setting of the initial values for the k_i were tested. In Figure 11 the results are presented.
As the Gauß collocation method is an implicit method one expects that it decreases the energy of a dissipative or Hamiltonian system over time instead of increasing it which has been observed. Decrease of time step size does not increase the accuracy.
In Figure 12 the details about the simulation are shown. If the step size is decreased the number of required iterations per step is also decreased. Also the number of required iterations in the second method where the k_i are set to their last values is smaller than with the first method where the k_i are set to zero in every iteration. In [Hai06] other algorithms for finding a good estimate for the next k_i are provided like a linear extrapolation of the last two iterations.

$$\text{GauCol} = \begin{cases} \dfrac{439.303}{78.733} \approx 5,6 < 10 \dfrac{3.092.647}{439.303} \approx 7,0 < 10 \end{cases}$$

$$\text{GauCo2} = \begin{cases} \dfrac{347.971}{76.929} \approx 4,5 < 10 \dfrac{2.245.553}{347.971} \approx 6,5 < 10 \end{cases}$$

If we divide the number of required function calls with step size 10^{-4} and 10^{-5} by the number of required function calls with step size 10^{-3} and 10^{-4}, respectively, we see that this ratio is below ten which is the ratio we would get with an explicit scheme. The smaller step size causes ten times more integration steps but in the implicit scheme convergence is achieved faster. If a step size of 10^{-5} is applied instead of 10^{-3}, only two or three iterations per step are required instead of nearly eight iterations per step.

Algorithm	Step size	No. function calls	Simulation time	Average no. of iterations per time step
GauCol	10^{-3}	78.733	49	7,9
GauCol	10^{-4}	439.303	272	4,4
GauCol	10^{-5}	3.092.647	2.062	3,1
GauCo2	10^{-3}	76.929	48	7,7
GauCo2	10^{-4}	347.971	226	3,5
GauCo2	10^{-5}	2.245.553	1.559	2,2

Figure 12: Simulation data of the Gauß collocation methods *GauCol* and *GauCo2*

3.5 Comparison of the Runge-Kutta-Fehlberg method with the Gauß collocation method

We simulated the scenario for five second. In Figure 13 we see that both methods perform equally in terms of energy conservation due to their order. However, as known from before, at the moment the Gauß collocation method performs poorly in terms of simulation time and the number of requested function evaluations.

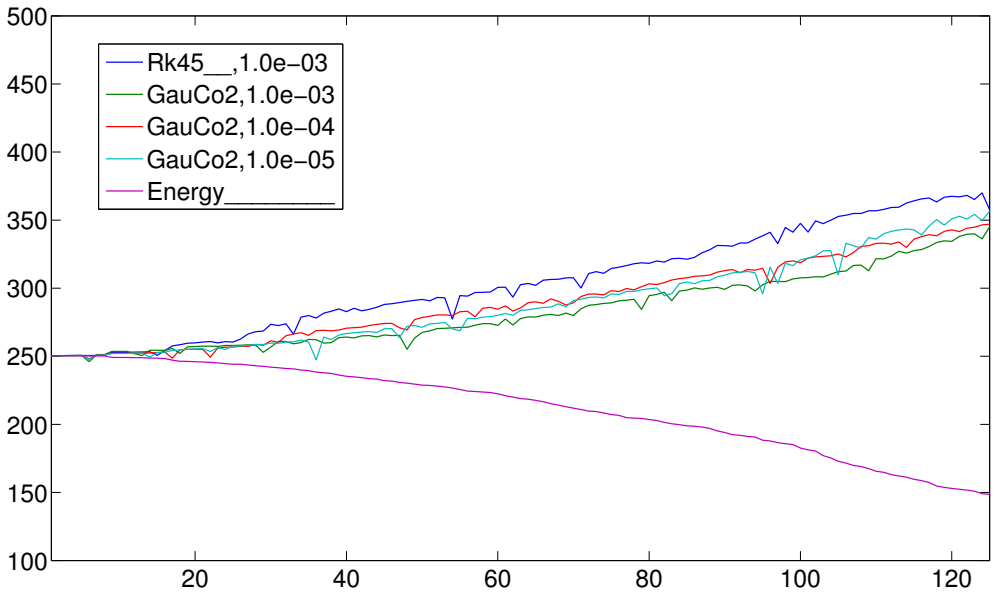

Figure 13: Energy plot of the Gauß collocation methods *Rk45N* and *GauCo2*

3.6 Comparison of algorithms of different convergence order

In order to evaluate the effect of algorithms of different convergence orders on the total energy we compared all introduced algorithms. We used the Taylor expansion *Taylor*, the velocity Verlet *VelVer*, the adaptive Runge-Kutta-Fehlberg-method *Rk45N* with the

new step size algorithm and the Gauß collocation method with the introduced estimate of k_i-estimates *GauCo2* with a simulation time of five seconds and the time steps $h \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. The results can be seen in Figure 14. With a time step of 10^{-3} only the adaptive Runge-Kutta-Fehlberg method converges. With smaller time steps all algorithms provide good estimates of the energy.

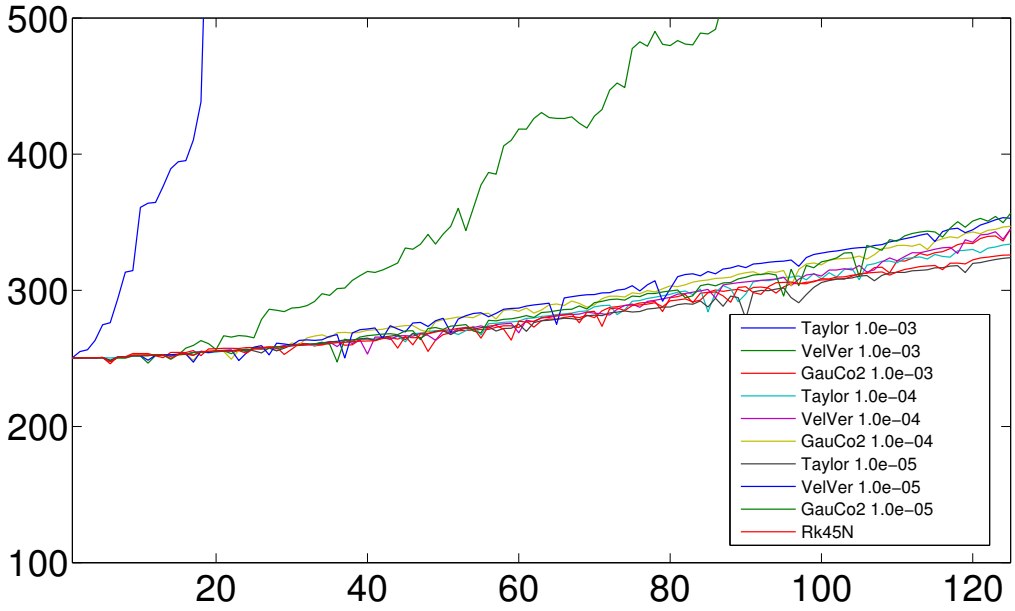

Figure 14: Energy plot of different algorithms

3.7 Comparison of different force models

Finally we compare the three different introduced force models. We use the Taylor expansion and the time steps 10^{-4} and 10^{-5}. The plot is shown in Figure 15. The results are that

- the energy conservation is improved by decrease of the time step and
- the extended nonlinear force model diverges faster than the Kuwabara-Kono model which diverges faster than the linear force model

Both simulations using nonlinear force models required smaller time steps than the simulation using the linear force model. This results in larger computation time if similar accuracy is desired.

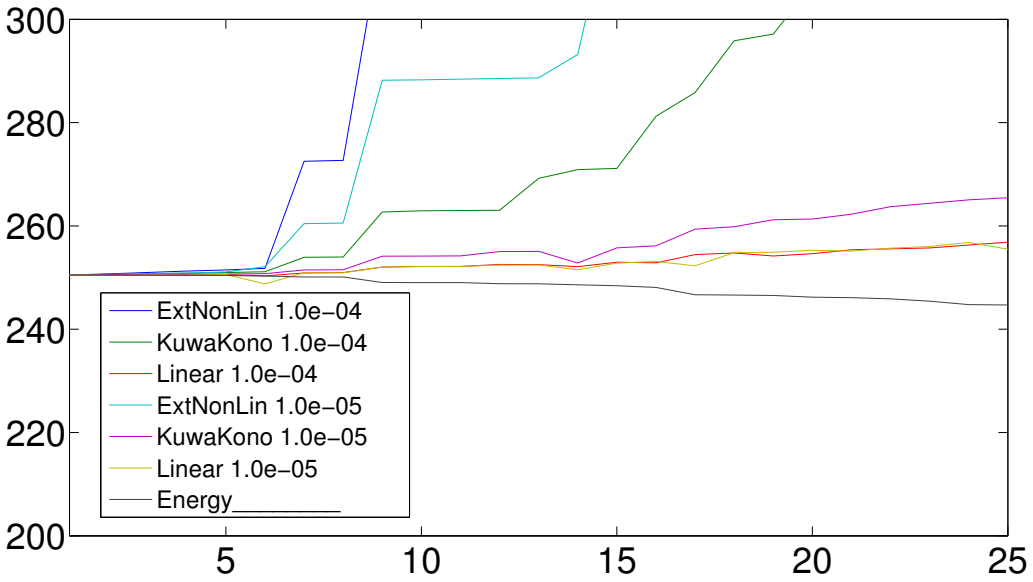

Figure 15: Energy plot of the Taylor expansion *Taylor* with three different force models

4 Summary and conclusion

As said in the beginning, the used algorithms have to provide fast as well as accurate calculation. It has been shown that there are no differences in orders of magnitude in the accuracy of the used algorithms. The proposed methods can guarantee additional energy conservation properties, but the high order of the used algorithms already insures much of this property.
There has been no explanation why even the implicit algorithms increase the total energy in the dissipative system.
As in most N-body-simulations the dominant time consuming step is the calculation of the acting forces, the Gauß collocation method has to be redesigned in terms of less function evaluations or examined for the applicability of larger time steps.
Further investigations of the *pe* can confirm the results by profiling more typical scenarios. Then the proposed change in the adaptive Runge-Kutta-Fehlberg-method can be implemented.

References

[Bie09] Claas Bierwisch. *Numerical simulations of granular flow and filling.* Shaker, Aachen, 2009.

[DR08] Wolfgang Dahmen and Arnold Reusken. *Numerik für Ingenieure und Naturwissenschaftler (Springer-Lehrbuch) (German Edition).* Springer, 2008.

[Hai06] Ernst Hairer. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer, Berlin, 2006.

[Hee11] Mario Heene. *Extension of the pe Physics Engine by Discrete Element Methods*. 2011.

[KE08] H. Kruggel-Emden. *Selection of an appropriate time integration scheme for the discrete element method (DEM). ScienceDirect*, 2008.

[Pös05] Thorsten Pöschel. *Computational Granular Dynamics: Models and algorithms.* Springer-Verlag, Berlin, 2005.

[Woh10] Barbara Wohlmuth. *Hauptseminar Strukturerhaltende Zeitintegratoren*. 2010.

[YCL09] T. Yee, C. Christara, and J. Lait. *A Study of a DEM-Based Granular Dynamics Solver (DRAFT 4).* 2009.