Previsualization in Robotics: An Atomic Approach

Mauno Rönkkö

Report A/2003/4

ISBN 951-781-268-X

UNIVERSITY OF KUOPIO

Department of Computer Science

P.O.Box 1627, FIN-70211 Kuopio, FINLAND

Previsualization in Robotics: An Atomic Approach

Mauno Rönkkö *

Abstract

In this paper, we discuss previsualization in robotics. Previsualization is a design technique used in film industry. In previsualization, a computer animates a highly underspecified scene; thus, helping the designers to detect missing components from the scene.

Previsualization can also be used in robotics for inspecting a highly underspecified model. However, for previsualization to work, it should reveal detailed dynamics not explicitly specified in the model. The details should appear as emergent dynamics. In robotics, in particular, the emergent dynamics should cover the interaction dynamics of the physical components. Such a requirement is non-trivial.

The contribution of this paper is an investigation of an atomic approach that supports previsualization in robotics. The approach is based on use of atoms. Atoms obey simple, compositional interaction laws. The laws produce emergent interaction dynamics for physical components composed of atoms. We shall also illustrate in this paper how atoms, despite their simplicity, can capture compactly non-trivial collision of non-rigid strings, and how the emerging dynamics shows nevertheless also intricate details.

keywords: robotics, previsualization, emergent dynamics

1 Introduction

Previsualization is a design technique used in film industry [1]. In previsualization, a computer animates an underspecified scene. The animation reveals quickly, if the scene is missing some key components, or if the observed dynamics is as desired.

In this paper, we generalize the idea of previsualization to *robotics* [2]. In robotics, previsualization is used as a design tool for improving a model until it is detailed enough for formal analysis and prototyping. Unlike *simulation* [3], previsualization does not deal with a detailed model; rather, the model is

*Department of Computer Science, University of Kuopio, P.O.Box 1627, FIN-70211 Kuopio, FINLAND, mauno.ronkko@uku.fi

highly underspecified and details appear as *emergent dynamics* [1]. Because of this, previsualization does not reveal exact dynamics; however, the emergent dynamics is detailed enough to be used for improving the model.

When using previsualization in robotics, we wish to avoid specifying explicitly the interactions between the physical components in the model; the goal of the previsualization is to determine those interactions. This means that not all the requirements are known ahead of the previsualization; thus, previsualization is used as a tool for clarifying the requirements. Consequently, the interaction dynamics of the physical components should be part of the emergent dynamics.

The interaction dynamics of the physical components does not, however, trivially appear in robotics as emergent dynamics. The fundamental problem lies in the inherent compositional diversity: in robotics, the interacting physical components themselves are composed of many different kinds of shapes and materials, as discussed in [2]. In addition, the dynamics is further composed of various kinds of motions, rotations, collisions, and other kinematics. These phenomena are well understood on their own, and explicit equations are found in a text-book on physics, such as [5].

Mere encoding of the text-book equations of various kinds of motion, however, is not enough, as they describe the dynamics in isolation. Figuring out the interaction equations, on the other hand, is not reasonable, since some of the requirements are not known ahead of the previsualization. Also, for the previsualization to be useful, modelling should be effortless [2]. Yet, the potential interaction dynamics of the physical components must be present in the previsualization, as the goal is to improve the model to better reflect the desired dynamics.

As a solution, we propose an atomic approach that does not reveal exact dynamics, but where the interaction dynamics of the physical components appear as emergent dynamics. The approach is based on use of *atoms* [3], indivisible particles of constant shape and mass. Atoms may move, resist movement, be bound to each other, and collide with each other. For these dynamics, there are simple equations that are also compositional. Moreover, these equations are enough to produce emergent dynamics capturing interactions between physical components that are composed of atoms. Such components may freely vary in shape and size. The atomic approach is, thus, a specific variant of a *particle system* [6] using a *particle-particle simulation method* [7, 8].

An advantage of the atomic approach is that it is easy to attach also software components, such as neural nets used for controlling [2], to the model. Then, in

[1] By emergent dynamics we mean observable dynamics in the model that has no dedicated counterpart in the model variables. Some discussion on emergent dynamics and emergent behaviour is found, for instance, in [4] and [2].

[2] Unconstraint interactions result in a quadratic number of equations capturing the interaction dynamics. If we also consider various materialistic parameters such as shape, elasticity, and friction, the number of equations becomes unmanageable even for a simple model in robotics.

[3] The concept of an atom was presented already by Greek philosophers Democritus and Leucippus around 400 BC [9]. Their notion of atoms, however, differs from the notion of atoms presented in this paper.

the model, the software components interact with the physical components by manipulating the forces on the atoms. In this way, we do not need to explicitly state the resulting physical dynamics in the model, because it appears as emergent dynamics. Consequently, we can easily experiment with different software components to see their effect on the *behavioral dynamics* [2, 10].

The main contribution of this paper is a formalization of the mathematics behind the atomic approach. Therefore, we shall use only simple examples to confirm that the laws governing the motion of atoms produce meaningful compositional dynamics. We shall also illustrate how atoms, despite their simplicity, can capture compactly non-trivial collision of non-rigid strings, and how the emerging dynamics shows nevertheless also intricate details. More versatile examples are a topic of future research.

Overview. We start by defining atoms and formalizing their motion in Section 2. In Section 3 we discuss the notion of an impulse on an atom; in particular, we define and formalize the interaction of atoms in terms of bonds and collisions. In Section 4, we address compositionality: we illustrate that our approach has desired compositional features, and discuss the importance of compositionality. Finally, in Section 5, follows the conclusion. In Appendix A, we describe an object-oriented implementation of our atomic approach.

2 An atom

An atom is a passive indivisible particle with four properties: *shape*, *mass*, *position*, and *velocity*. Each atom has the same, constant, shape of a *unit ball*, and the same *unit mass*. Only the position and the velocity may vary from atom to atom. However, the position and velocity are not free variables, as they are bound together by the model of motion. In that model, velocity is affected by external forces. An atom, in itself, has no internal forces.

As an atom exists in $\mathbb{R}^3$-space, we use three dimensional *vectors* [11] to capture the properties of an atom. Let $\vec{c}$ denote the position of an atom, i.e., the displacement of the center of the atom from the origin. Then, the *magnitude* [11] of $\vec{c}$, i.e., $|\vec{c}|$, gives the distance of the center of the atom to the origin. Hence, an atom occupies the set of points in $\mathbb{R}^3$ described by a vector $\vec{p}$, for which $|\vec{p} - \vec{c}| \leq 1$.

The unit mass is evenly distributed within an atom. Consequently, according to [5], the center of an atom is also its *center of mass*. We do not, however, consider atoms as objects of dense material; on the contrary, atoms are *sparse objects*. Therefore, several atoms may overlap in space without any interference, if so desired.

For atoms, we use as the starting point the trivial model of motion with acceleration [5]. Let $\vec{c}(t)$ denote the center of an atom at time t, $\vec{v}(t)$ denote its velocity at time t, and $\vec{a}(t)$ denote its acceleration at time t. Then, the the model of motion with acceleration is $\vec{c}'' = \vec{a}(t)$. As discussed in [12], this

second order differential equation can be represented with a velocity variable as a system of differential equations: $\vec{c}\,' = \vec{v},\ \vec{v}\,' = \vec{a}(t)$. From this form, it is easy to see that the model of motion preserves both *Newton's first* and *second law* [5].

As a computational model we use an approximation of the model of motion. The approximation is obtained, for instance, by applying Euler's method [12]. Let n and $n+1$ denote successive iteration rounds, and τ denote a reasonably small, fixed, time step. Then, Euler's method gives $\vec{c}(n+1) = \vec{c}(n) + \tau \cdot \vec{v}(n)$ and $\vec{v}(n+1) = \vec{v}(n) + \tau \cdot \vec{a}(n)$, approximating both the position and velocity of an atom, respectively. The duration is given by $\tau \cdot n$.

We may further refine the approximation equations by noting that $m \cdot \tau \cdot \vec{a}(n)$ is according to [5] the impulse $\vec{j}(n)$ over the fixed time step τ on a particle with a mass m. Since an atom has a unit mass, we may substitute in the approximation $\tau \cdot \vec{a}(n)$ with $\vec{j}(n)$. Thus, we have:

$$\vec{c}(n+1) = \vec{c}(n) + \tau \cdot \vec{v}(n), \qquad \vec{v}(n+1) = \vec{v}(n) + \vec{j}(n) \tag{1}$$

In the sequel, we shall use Equation 1 as the (computational) model of motion. An advantage of Equation 1 is that it is easily implemented using any programming language. A more important feature of Equation 1 is that it is compositional; in particular, it allows us to consider the position, velocity, and the impulse of an atom independently of each other within one iteration round.

For clarity, we shall assume in the sequel that a model always has a fixed number of atoms. Then, we may refer to a specific atom in the model using subscripts. Thus, we shall denote the center of the i'th atom in the model by $\vec{c}_i$, the velocity by $\vec{v}_i$, and the impulse by $\vec{j}_i$.

3 An impulse

An impulse is, by definition [5], the sum of forces affecting the velocity. In our model, there are only four kind of forces: *damping forces*, *bond forces*, *collision forces*, and other *case-specific external forces*. An example of a case-specific external force is, for instance, gravity.

3.1 A damping force

A damping force resists the motion of an atom. Its direction is opposite to the direction of the velocity, and the magnitude is always less or equal to the magnitude of the velocity.

Assuming $m \in [0, 1]$, a damping force $\vec{fd}(i, m)$ with a magnitude of $m \cdot 100\%$ on the i'th atom in the model is:

$$\vec{fd}(i, m) = -m \cdot \vec{v}_i \tag{2}$$

3.2 A bond force

A bond force maintains a predefined distance between two specific atoms. Generally speaking, a bond force is modelled as a function over the desired [4] and the current distance between the two atoms. We model a bond force with a non-linear function. It supports both short bond distances and stability under moderate velocities. A linear function would work properly only with long bond distances, or with low velocities.

Assuming $r \in [0, \infty)$, a bond force $\vec{fb}(i, r, k)$ on the i'th atom in the model with a desired distance r to the k'th atom in the model is:

$$\vec{fb}(i, r, k) = (\vec{c}_k - \vec{c}_i) \cdot (1 - r^2 |\vec{c}_k - \vec{c}_i|^{-2}) \tag{3}$$

Since the bond force on the i'th atom is opposite to the bond force on the k'th atom, i.e., $\vec{fb}(i, r, k) = -\vec{fb}(k, r, i)$, a pair of bond forces, $\vec{fb}(i, r, k)$ and $\vec{fb}(k, r, i)$, preserves *Newton's third law* [5] [5].

Equation 3 above can also be presented in a more readable form. Let $\widehat{\vec{c}_k - \vec{c}_i}$ denote a *unit vector* [11] along $\vec{c}_k - \vec{c}_i$. Then, the equation can be written as $\vec{b}(i, r, k) = (\widehat{\vec{c}_k - \vec{c}_i}) \cdot (|\vec{c}_k - \vec{c}_i|^2 - r^2)|\vec{c}_k - \vec{c}_i|^{-1}$. This equation clearly shows that the direction of a bond force is determined by the *alignment* of the atoms, and the magnitude of a bond force is proportional to the deviation from the desired distance. Note that an altered equation $(\widehat{\vec{c}_k - \vec{c}_i}) \cdot (|\vec{c}_k - \vec{c}_i|^2 - r^2)(|\vec{c}_k - \vec{c}_i| + r)^{-1}$, captures a linear bond force, as it simplifies to $\widehat{\vec{c}_k - \vec{c}_i} \cdot (|\vec{c}_k - \vec{c}_i| - r)$.

An advantage of Equation 3 over a linear bond force is that it prevents atoms from slipping through each other even with short desired distances and moderate velocities. This shows in a comparison between a non-linear coefficient $(|\vec{c}_k - \vec{c}_i|^2 - r^2)|\vec{c}_k - \vec{c}_i|^{-1}$ and a linear coefficient $|\vec{c}_k - \vec{c}_i| - r$ depicted in Figure 1. The figure clearly shows that a non-linear coefficient has a bigger magnitude, especially when the distance of the atoms is shorter than the desired distance.

Another advantage of Equation 3 over a linear bond force is that it involves no square roots. This is important, since square roots slow down computation considerably.

Clearly, Equation 3 results in an accelerating movement. Consequently, an additional force, such as a damping force, may be needed for the motion to *converge*.

Note that an atom may well be bound to many atoms at the same time in the model. Then, we may consider each bond independently, and the sum of the bond forces is the total bond force on the atom. Due to the model of motion, Equation 1, the bond forces can be evaluated in any order without affecting the result.

[4] We consider only the distance between the centers of the atoms, as the center of an atom is also its center of mass.

[5] The significance of Newton's third law is that it preserves *symmetry*. Then, from a pair of atoms, we may freely choose one atom, and use it as the center of events. Due to symmetry, the result is independent of our choice.

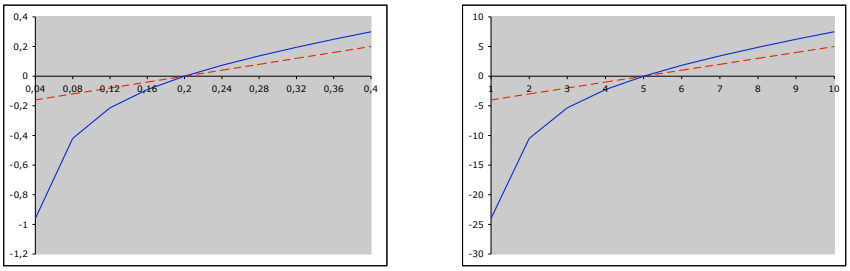

Figure 1: A comparison between a linear and a non-linear coefficient with a desired distance 0.2 on left and 5 on right; The red dashed line shows the values of $|\vec{c}_k - \vec{c}_i| - r$, and the blue solid line shows the values of $(|\vec{c}_k - \vec{c}_i|^2 - r^2)|\vec{c}_k - \vec{c}_i|^{-1}$, while $|\vec{c}_k - \vec{c}_i|$ varies from 0.04 to 0.4 on left, and from 1 to 10 on right

3.3 A collision force

A collision force is a semi-bond force. It affects only overlapping atoms by pushing them apart. Atoms are considered to be apart, when their distance exceeds the sum of their radii, 2.

Formally, a collision force $\vec{fc}(i,k)$ on the i'th atom in the model with respect to the k'th atom in the model is:

$$\vec{fc}(i,k) = min\{0,\ \vec{fb}(i,2,k)\} \tag{4}$$

As for a bond force, the collision force on the i'th atom is opposite to the collision force on the k'th atom, i.e., $\vec{fc}(i,k) = -\vec{fc}(k,i)$. Hence, a pair of collision forces, $\vec{fc}(i,k)$ and $\vec{fc}(k,i)$, preserves also Newton's third law.

Note that an atom has no materialistic variables, such as elasticity. Consequently, Equation 4 above defines exhaustively a collision between two atoms. More specifically, Equation 4 captures an *elastic* collision, as we shall illustrate later. This means, in turn, that an inelastic collision can never occur between two atoms, unless there is an additional force affecting the atoms.

Similarly to the bonds, we may consider each collision independently, and the sum of the collision forces is the total collision force on the atom. Due to the model of motion, Equation 1, the collision forces can also be evaluated in any order without affecting the result.

3.4 Other case-specific external forces

We do not attempt to give a general formalization for the other case-specific external forces. The reason for this is simply the overwhelming variety of such forces. Examples of case-specific external forces are, among others, gravity, initial impulses that set the model into motion, impulses originating from various power sources like motors, accumulators, and electromagnetic fields, and im-

pulses used mainly for visualization purposes, i.e., forcing the atoms to move into desired directions to see the emergent dynamics of that motion.

4 Compositionality

The model of motion, Equation 1, supports compositionality on many levels. As already mentioned, the model of motion allows us to consider the impulse and the changes in the velocity separately, independently, and before considering the changes in the positions of the atoms. Similarly, the impulse can be considered in parts, because the summed forces can be considered separately and independently of each other. The resulting impulse is, thus, independent of the order in which the forces are evaluated.

We shall next illustrate with many examples how central compositionality is in our approach. We start by considering compositionality of a bond force and a damping force. We then consider multiple bond forces. After that we consider aligned and non-aligned collisions with motion. Then, we consider compositionality of bond forces and collision forces. Finally, we conclude with a non-trivial example involving a non-aligned collision of two non-rigid strings in motion.

In all the examples we assume a modest time step, $\tau = 0.01$, for the model of motion. More details on the implementation is found in Appendix A.

4.1 Composing a bond force and a damping force

We start by illustrating the compositionality of a bond force and a damping force. We first consider a bond between two atoms in isolation, and then with a damping force. After that, we summarize our findings.

A bond force in isolation. Due to the model of motion, a bond force does not immediately restore the desired distance between the atoms. Rather, it causes the atoms to move towards the desired distance. This slackness not only gives time for the other forces to interact simultaneously, but also supports stability by preventing the atoms from zooming back and forth uncontrollably.

Consider two atoms whose distance is initially 4, and a bond force that tries to restore a desired distance of 2. Formally, the model is:

$$\begin{array}{llll} \vec{c}_1(0) = (0,-2,0), & \vec{v}_1(0) = (0,0,0), & \vec{j}_1 = \vec{fb}(1,2,2) \\ \vec{c}_2(0) = (0,2,0), & \vec{v}_2(0) = (0,0,0), & \vec{j}_2 = \vec{fb}(2,2,1) \end{array}$$

The bond force first pulls the atoms towards each other, but as soon as the atoms pass the desired distance, the bond force starts pushing the atoms apart. Since a bond force cannot immediately stop the atoms, they have a stable, periodic motion around the desired distance. Figure 2 illustrates this motion of atoms. Clearly, the bond for is not sufficient in this case for convergence.

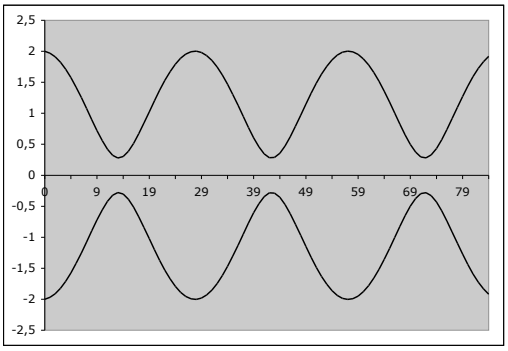

Figure 2: A bond force in isolation causes two atoms to have a periodic motion around the desired distance; the two lines above depict the projections of the centers of the atoms onto the y-axis with respect to the number of iterations

Simultaneous bond and damping force. Consider the case above, but with an additional damping force of 8% on both atoms. Formally, such a model is:

$$\vec{c}_1(0) = (0, -2, 0), \quad \vec{v}_1(0) = (0, 0, 0), \quad \vec{j}_1 = \vec{fb}(1, 2, 2) + \vec{fd}(1, 0.08)$$
$$\vec{c}_2(0) = (0, 2, 0), \quad \vec{v}_2(0) = (0, 0, 0), \quad \vec{j}_2 = \vec{fb}(2, 2, 1) + \vec{fd}(2, 0.08)$$

The added damping force slows down the acceleration of both atoms enough to cause a quick convergence that restores the desired distance between the atoms. The motion is now very different from the motion in the case above, as shown in Figure 3.

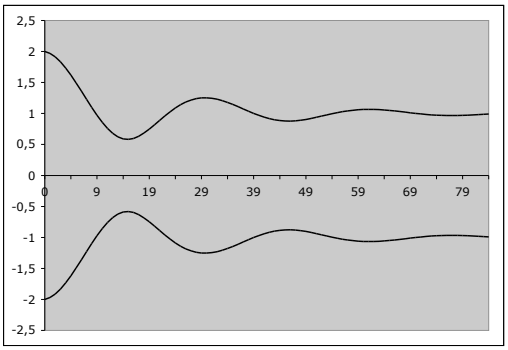

Figure 3: A bond force together with a damping force cause the atoms to have a convergent motion that restores the desired distance between the atoms; the two lines above depict the projections of the centers of the atoms onto the y-axis with respect to the number of iterations

Summary. The two cases above illustrate the role of compositionality in our atomic approach. Note that, we did not explicitly define the trajectories of the atoms. The trajectories emerged as a result of applying the model of motion together with the defined forces on the atoms. In the latter example, in particular, the emergent trajectories were the result of composed forces on the atoms.

4.2 Multiple bond forces

Next we consider compositionality of bonds with themselves. As it was mentioned earlier, we may freely combine bonds with atoms; yet, it makes no difference in which order the bonds are evaluated due to the model of motion. In this example, we show how the number of bonds per atom affects the dynamics of a string of atoms.

We consider a string of 20 atoms aligned along the x-axis so that, initially, the distance between any two neighboring atoms is 1. Thus, the atoms are partially overlapping. The bonds will be defined based on the initial distance between the atoms. In addition, the two atoms residing at the opposite ends of the string are given orthogonal initial velocities of magnitude 10, one along the y-axis and the other along the z-axis. Figure 4 depicts this setting.

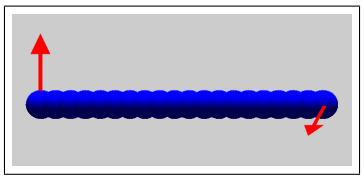

Figure 4: A string of 20 atoms in its initial position; the initial velocities of the two atoms residing at the opposite end of the string are shown by arrows

We shall study three cases involving the string in Figure 4: with 2 bonds per atom, with 8 bonds per atom, and with 19 bonds per atom. At the end, we summarize our findings.

Two bonds per atom. In the first case, we bind each atom to their neighboring atoms. Thus, the model is formally:

$$\vec{c}_i(0) = (i - 10, 0, 0),\ i \in [1..20]$$
$$\vec{v}_1(0) = (0, 10, 0)$$
$$\vec{v}_i(0) = (0, 0, 0),\ i \in [2..19]$$
$$\vec{v}_{20}(0) = (0, 0, 10)$$
$$\vec{j}_1 = \vec{fb}(1, 1, 2)$$
$$\vec{j}_i = \vec{fb}(i, 1, i - 1) + \vec{fb}(i, 1, i + 1),\ i \in [2..19]$$
$$\vec{j}_{20} = \vec{fb}(20, 1, 19)$$

Figure 5 shows a series of images depicting the motion of the string. As the series of images reveals, the initial velocities gradually distribute throughout

the entire string of atoms. It should be emphasized that this distribution of the initial velocities is emergent dynamics; it is not explicitly specified in the model.

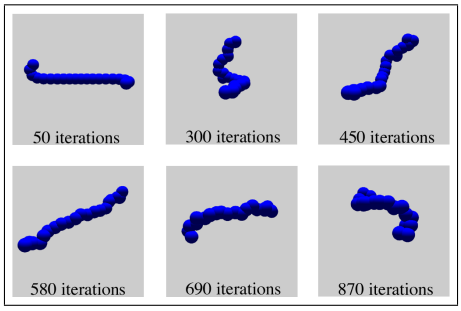

Figure 5: Motion of the string with two bonds per atom; the number of computed iteration rounds is shown in each image

Eight bonds per atom. In the second case, we bind each atom to eight neighboring atoms, to four nearest atoms on its left and to four nearest atoms on its right, whenever possible. Thus, the model is formally:

$$\vec{c}_i(0) = (i - 10, 0, 0),\ i \in [1..20]$$
$$\vec{v}_1(0) = (0, 10, 0)$$
$$\vec{v}_i(0) = (0, 0, 0),\ i \in [2..19]$$
$$\vec{v}_{20}(0) = (0, 0, 10)$$
$$\vec{j}_i = \sum_{k=max\{1,i-4\}}^{min\{i+4,20\}} \vec{fb}(i, |k - i|, k),\ i \in [1..20],\ k \neq i$$

Figure 6 shows a series of images depicting the motion of the string. This time, the motion becomes more rigid as the number of bonds is increased. In particular, the string shows now an emerging rotational motion around the centermost atom.

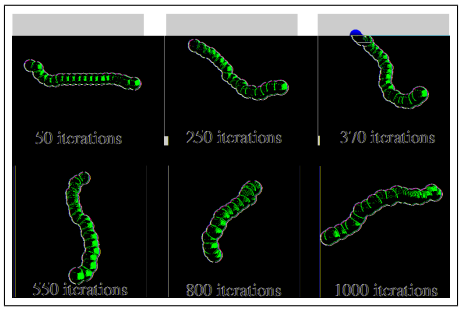

Figure 6: Motion of the string with bonds to eight nearest atoms; the number of computed iteration rounds is shown in each image

Nineteen bonds per atom. In the last case, we bind each atom to each other atom in the string. Thus, the model is formally:

$$\vec{c}_i(0) = (i - 10, 0, 0),\ i \in [1..20]$$
$$\vec{v}_1(0) = (0, 10, 0)$$
$$\vec{v}_i(0) = (0, 0, 0),\ i \in [2..19]$$
$$\vec{v}_{20}(0) = (0, 0, 10)$$
$$\vec{j}_i = \sum_{k=1}^{20} \vec{fb}(i, |k - i|, k),\ i \in [1..20],\ k \neq i$$

Figure 7 shows a series of images depicting the motion of the string. As the series of images in the figure shows, the string has now almost the dynamics of a rigid body. There are still some minute waves passing back and forth the string. These waves are the result of the initial velocity being distributed throughout the string by the bond forces. Also, with nineteen bonds per atom, the string shows a clear rotational motion around the centermost atoms.

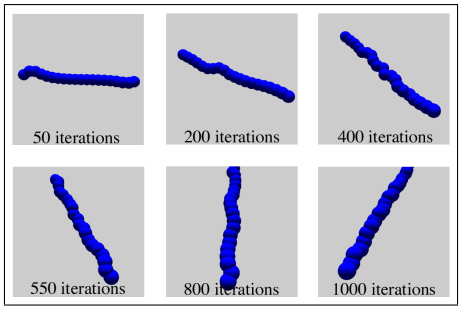

Figure 7: Motion of the string with each atom is bound to each other atom; the number of computed iteration rounds is shown in each image

Summary. As these three cases show, bonds not only bind atoms together, they also define the rigidness of the body. This shows clearly as the emergent rotational dynamics of the string; the more bonds there are, the more clearly the string of atoms rotates around the centermost atoms.

The three cases also reveal another interesting property: the string of atoms holds together even without a damping force. The reason for this is that the desired distances of the bond forces are in accordance with the initial distances between the atoms. Then, the motion in the string depends entirely on the initial velocities given to the atoms at the opposite ends of the string.

The periodic motion caused by the bond forces, as shown earlier in Figure 2, is still present, but it is mostly masked out by the motion originating from the initial velocities. The periodic motion shows as minute waves passing through the string. However, it should be noted that these waves are also present in any ordinary object, too. It shows, for instance, as sound waves carrying through the object when being hit. An extreme variant of minute waves is, of course, *resonance* [5].

The computational model of the string above is very light. For instance, with two bonds per atom, the computation of 100 000 iteration rounds takes about 0.8 seconds with a 500MHz PowerPC G3-processor; thus, the computation time required for the 870 iteration rounds shown in Figure 5 takes about 0.007 seconds.

4.3 Head-to-head collisions with motion

As it was mentioned earlier, the collision force formulated in Equation 4 captures an elastic collision between the atoms. To confirm this, we consider next examples where a collision is caused by some initial motion.

For simplicity, we start by considering only two atoms having a perfectly aligned, straight-line collision. We study two such cases: in the first case one of the atoms is at rest, and in the other case both of the atoms move. At the end, we summarize our findings.

Head-to-head collision with an atom at rest. We start with the simplest setting, where we give an initial velocity to only one of the two atoms while the other atom is at rest. The initial velocity is directed exactly towards the center of the other atom with a magnitude of 1. Thus, the formal model is:

$$\begin{array}{lll} \vec{c}_1(0) = (0,-2,0), & \vec{v}_1(0) = (0,1,0), & \vec{j}_1 = \vec{fc}(1,2) \\ \vec{c}_2(0) = (0,2,0), & \vec{v}_2(0) = (0,0,0), & \vec{j}_2 = \vec{fc}(2,1) \end{array}$$

Figure 8 shows the motion of the atoms before, during, and after the collision. As shown in the figure, the momentum of the moving atom is totally transferred to the atom at rest. This dynamics follows exactly the dynamics of an elastic straight-line collision with one of the bodies at rest [5].

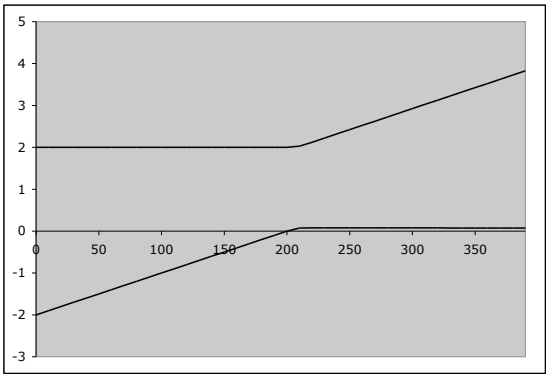

Figure 8: A collision of two atoms with one of the atoms at rest; the two lines above depict the projections of the centers of the atoms onto the y-axis with respect to the number of iterations

Head-to-head collision of moving atoms. Next we illustrate how the collision force not only conserves kinetic energy and momentum but also preserves symmetry of motion. We consider two atoms moving against each other in a straight line with opposite velocities of magnitude 1 and 0.5. Then, the formal model is:

$$\begin{array}{lll} \vec{c}_1(0) = (0, -2, 0), & \vec{v}_1(0) = (0, 1, 0), & \vec{j}_1 = \vec{fc}(1, 2) \\ \vec{c}_2(0) = (0, 2, 0), & \vec{v}_2(0) = (0, -0.5, 0), & \vec{j}_2 = \vec{fc}(2, 1) \end{array}$$

Figure 9 shows the motion of the atoms before, during, and after the collision. The dynamics observed in Figure 9 follows the conservation laws of kinetic energy and momentum [5]. The symmetry of motion is also preserved, since momentums are exchanged during the collision. This is nevertheless in accordance with text-book equations on collisions [5].

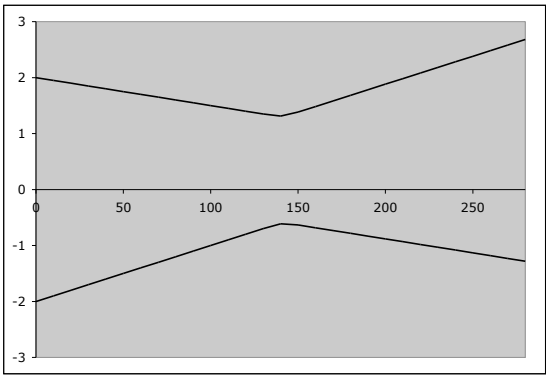

Figure 9: A head-to-head collision of two atoms with opposite velocities; the two lines above depict the projections of the centers of the atoms onto the y-axis with respect to the number of iterations

Summary. These two examples confirm that the collision force conserves both kinetic energy and momentum in a collision, where the atoms move along a straight line connecting the center of the atoms.

There is, however, a subtle difference between the text-book equations for collisions, such as in [5], and the collision following Equation 4. Namely, the text-book equations capture an instantaneous collision. A collision following Equation 4, on the other hand, is not instantaneous; it takes some iteration rounds for the collision force to push the atoms apart.

It should be also noted that the dynamics shown in Figure 9 is emergent dynamics. We did not explicitly specify how the kinetic energy should be distributed during a collision. The motion seen in Figure 9 is merely the result of applying Equation 4 to atoms with some initial velocities.

4.4 Non-aligned collisions with motion

Encouraged by the previous results, we shall next study a more general setting, where the centers of the atoms are no longer aligned along each other's trajectories, i.e., non-aligned collisions.

We shall study three cases involving two atoms. In the first case, one of the atoms is at rest. In the second case, atoms have velocities that cause crossing trajectories. In the third case, the atoms move against each other with non-aligned trajectories. At the end, we summarize our findings.

Non-aligned collision with an atom at rest. We start by considering a non-aligned collision of two atoms, where one of the atoms is at rest. Formally, the model is:

$$\vec{c}_1(0) = (-2, 0.7, 0), \quad \vec{v}_1(0) = (10, 0, 0), \quad \vec{j}_1 = \vec{fc}(1, 2)$$
$$\vec{c}_2(0) = (1, -0.7, 0), \quad \vec{v}_2(0) = (0, 0, 0), \quad \vec{j}_2 = \vec{fc}(2, 1)$$

The trajectories of these atoms are plotted in Figure 10. The figure shows how one of the atoms stays at rest until the other atom collides with it. The trajectories of both of the atoms change considerably in the collision, as their centers are not aligned along each other's initial trajectories.

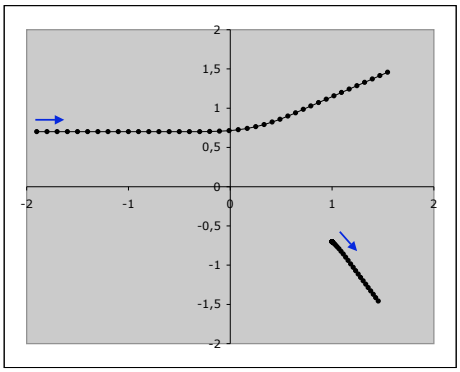

Figure 10: A non-aligned collision of two atoms; the trajectories are plotted after each iteration round, and the two arrows indicate both the beginning and the direction of the trajectories

Non-aligned collision to a moving atom from behind. Next we consider a collision of atoms with crossing trajectories. The trajectories are such that an atom collides to another atom from behind with respect to the initial trajectories. Formally, the model is:

$$\vec{c}_1(0) = (-2, 0.7, 0), \quad \vec{v}_1(0) = (10, 0, 0), \quad \vec{j}_1 = \vec{fc}(1, 2)$$
$$\vec{c}_2(0) = (0, -1, 0), \quad \vec{v}_2(0) = (5, 2, 0), \quad \vec{j}_2 = \vec{fc}(2, 1)$$

The trajectories of these atoms are plotted in Figure 11. The figure shows how the trajectories during the collision; in particular, the figure shows clearly

how it takes some iteration rounds for the collision force to push the colliding atoms apart.

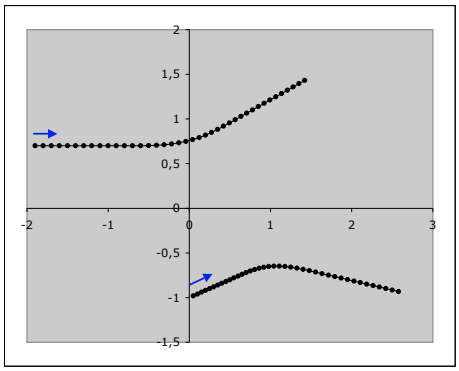

Figure 11: A non-aligned collision of two atoms with initially crossing trajectories; the trajectories are plotted after each iteration round, and the two arrows indicate both the beginning and the direction of the trajectories

Non-aligned collision of two atoms moving against each other. In this third case, we consider a non-aligned collision of two atoms moving against each other. This time, the formal model is:

$$\vec{c}_1(0) = (-2, 0.7, 0), \quad \vec{v}_1(0) = (10, 0, 0), \quad \vec{j}_1 = \vec{fc}(1, 2)$$
$$\vec{c}_2(0) = (2, -0.7, 0), \quad \vec{v}_2(0) = (-5, 0, 0), \quad \vec{j}_2 = \vec{fc}(2, 1)$$

The trajectories of these atoms are plotted in Figure 12. The figure shows clearly, how the principal direction of motion is kept by both of the atoms; the collision causes some adjustment to the initial trajectories.

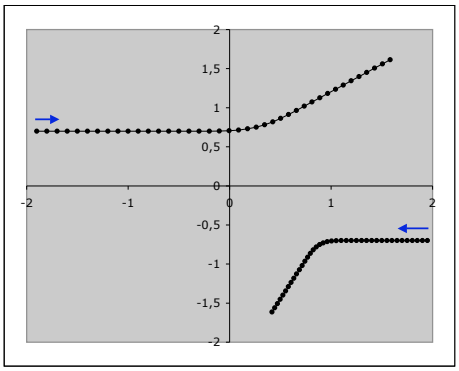

Figure 12: A non-aligned collision of two atoms moving against each other; the trajectories are plotted after each iteration round, and the two arrows indicate both the beginning and the direction of the trajectories

Summary. The resulting final velocities in each case are in accordance with the text-book equations for a collision in a plane [5]. In the first case, the velocities for the atoms after the last iteration round are $\vec{v}_1 = (7.5361, 4.3098, 0)$ and $\vec{v}_2 = (2.4639, -4.3098, 0)$. These velocities conserve both kinetic energy and momentum; in particular, these velocities conserve *componentwise* momentum along the axes, as required for a collision in a plane [5]. In the second case, the velocities for the atoms after the last iteration round are $\vec{v}_1 = (7.1472, 3.669, 0)$ and $\vec{v}_2 = (7.8527, -1.6691, 0)$. Also, in the last case, the velocities after the last iteration round are $\vec{v}_1 = (7.7064, 5.4098, 0)$ and $\vec{v}_2 = (-2.7063, -5.4098, 0)$.

As these three cases illustrate, the collision force defined in Equation 4 conserves both kinetic energy and momentum in non-aligned cases, too. The reason for this is that the collision force, in itself, preserves Newton's third law. It guarantees that momentum taken from one atom is given to the other atom in a proper manner.

4.5 Head-to-head collision of strings of atoms

Next we consider the compositionality of bond forces and collision forces. We study four cases, where the lengths of the colliding strings of atoms vary. The strings of atoms are aligned along the x-axis as shown in Figure 13.

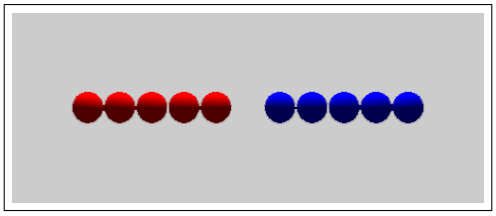

Figure 13: Initial position of two strings of atoms

We shall consider four cases: a collision of two equally long strings of atoms without bonds, a collision of the same strings of atoms with bonds, a collision of a shorter string of atoms to a longer string of atoms, and a collision of a longer string of atoms to a shorter string of atoms. In all but the first case, both strings are considered rigid, i.e., there is a bond per each pair of atoms in the string. At the end, we summarize our findings.

Collision of strings without bonds. We consider first a collision of two strings of atoms without bonds. Initially, the strings are positioned along x-axis according to Figure 13. The blue string of five atoms is set into a motion that causes a head-to-head collision with the red string of five atoms. All the atoms in the red string are at rest.

We shall denote the atoms in the red (left) string by indices 1..5, and the atoms in the blue (right) string by indices 6..10. Then, the formal model is:

$\vec{c}_i(0) = (-2i, 0, 0),\ i \in [1..5]$
$\vec{c}_i(0) = (2i - 10, 0, 0),\ i \in [6..10]$
$\vec{v}_i(0) = (0, 0, 0),\ i \in [1..5]$
$\vec{v}_i(0) = (-1, 0, 0),\ i \in [6..10]$
$\vec{j}_1 = \vec{fc}(1, 2)$
$\vec{j}_i = \vec{fc}(i, i-1) + \vec{fc}(i, i+1),\ i \in [2..9]$
$\vec{j}_{10} = \vec{fc}(10, 9)$

Figure 14 depicts the resulting motion of the atoms in the strings. In this case, there are multiple collisions as each atom collides to their neighboring atoms. As the figure shows, the collision forces keep all the atoms apart. Moreover, as indicated by the diverging lines, all the atoms drift slowly apart from each other due to the lack of binding and damping forces. This motion of string preserves, nevertheless, both kinetic energy and momentum.

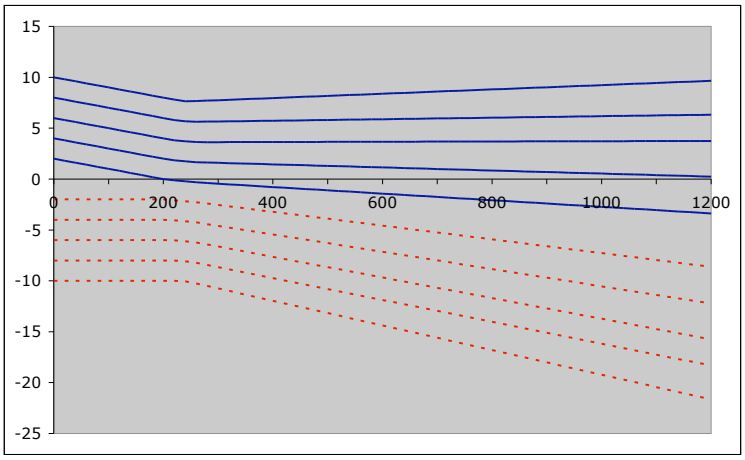

Figure 14: Collision of two strings of atoms without bonds; the blue lines indicate the positions of the atoms in the blue string on the x-axis, and the red lines indicate the positions of the atoms in the red string on the x-axis with respect to the number of iterations

Head-to-head collision of rigid strings. Next, we consider a collision of two rigid strings of atoms, i.e., there is a bond per each pair of atoms in a string. Each bond is in accordance with the initial distance between the atoms, and there are no bonds between atoms in different strings. Again, the blue string of atoms, indices 1..5, is set into a motion that causes a head-to-head collision with the red string of atoms, indices 6..10. Thus, the formal model is:

$$\vec{c}_i(0) = (-2i, 0, 0),\ i \in [1..5]$$
$$\vec{c}_i(0) = (2i - 10, 0, 0),\ i \in [6..10]$$
$$\vec{v}_i(0) = (0, 0, 0),\ i \in [1..5]$$
$$\vec{v}_i(0) = (-1, 0, 0),\ i \in [6..10]$$
$$\vec{j}_i = \sum_{k=1}^{5} \vec{fb}(i, 2|k-i|, k),\ i \in [1..4],\ k \neq i$$
$$\vec{j}_5 = \vec{fc}(5, 6) + \sum_{k=1}^{4} \vec{fb}(5, 2|k-5|, k)$$
$$\vec{j}_6 = \vec{fc}(6, 5) + \sum_{k=7}^{10} \vec{fb}(6, 2|k-6|, k)$$
$$\vec{j}_i = \sum_{k=6}^{10} \vec{fb}(i, 2|k-i|, k),\ i \in [7..10],\ k \neq i$$

Figure 15 depicts the resulting motion of the strings. As the figure shows, the two rigid strings behave just like a pair of colliding atoms, shown in Figure 8. Thus, the dynamics of the rigid strings correspond to dynamics of individual atoms with a mass of five atoms. Such dynamics is in accordance with the traditional text-book equations [5].

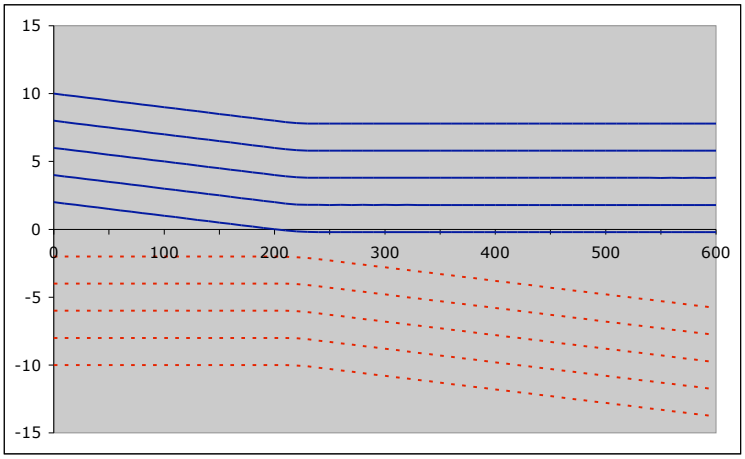

Figure 15: Collision of two rigid strings of atoms; the blue lines indicate the positions of the atoms in the blue string on the x-axis, and the red lines indicate the positions of the atoms in the red string on the x-axis with respect to the number of iterations

Head-to-head collision of a shorter rigid string to a longer one. Next, we consider the collision of two strings with different lengths. The red string has now 10 atoms, indices 1..10, while the blue string has 5 atoms, indices 11..15. Again, the blue string of atoms is set into a motion that causes a head-to-head collision with the red string of atoms. Thus, the formal model is:

$\vec{c}_i(0) = (-2i, 0, 0),\ i \in [1..10]$
$\vec{c}_i(0) = (2i - 20, 0, 0),\ i \in [11..15]$
$\vec{v}_i(0) = (0, 0, 0),\ i \in [1..10]$
$\vec{v}_i(0) = (-1, 0, 0),\ i \in [11..15]$
$\vec{j}_i = \sum_{k=1}^{10} \vec{fb}(i, 2|k-i|, k),\ i \in [1..9],\ k \neq i$
$\vec{j}_{10} = \vec{fc}(10, 11) + \sum_{k=1}^{9} \vec{fb}(10, 2|k-10|, k)$
$\vec{j}_{11} = \vec{fc}(11, 10) + \sum_{k=12}^{15} \vec{fb}(11, 2|k-11|, k)$
$\vec{j}_i = \sum_{k=11}^{15} \vec{fb}(i, 2|k-i|, k),\ i \in [12..15],\ k \neq i$

Figure 16 depicts the motion of the two strings. As the figure shows, the shorter blue string bounces back from the longer red string. The red string also starts moving after the collision to the initial direction of the blue string. The velocity of the blue string after the collision is approximately [6] $(0.33, 0, 0)$ while the velocity of the red string is approximately $(-0.66, 0, 0)$. These velocities are in accordance with velocities computed using text-book equations on a collision of particles with different masses [7] [5].

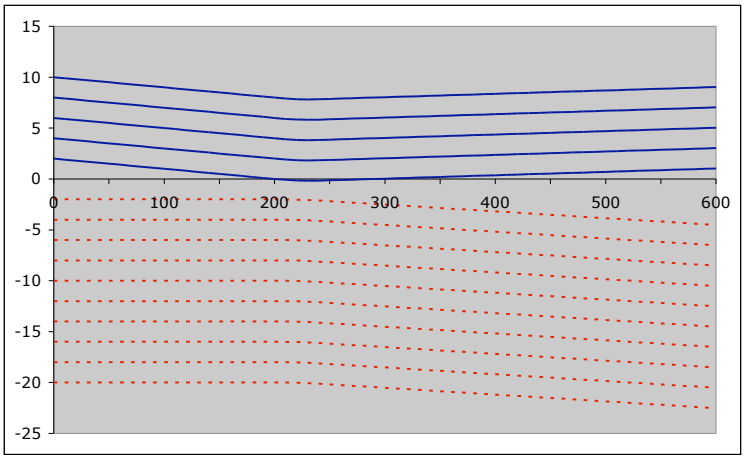

Figure 16: Collision of a shorter rigid string of atoms to a longer rigid string of atoms; the blue lines indicate the positions of the atoms in the shorter string on the x-axis, and the red lines indicate the positions of the atoms in the longer string on the x-axis with respect to the number of iterations

[6] As discussed earlier, the strings do not have a precise velocity as there are minor waves going back and forth the strings. These waves are caused by bond forces as they distribute collision forces in the strings, cf. Section 4.2.

[7] Although the sum of the velocities $(0.33, 0, 0)$ and $(-0.66, 0, 0)$ is not the initial velocity $(1, 0, 0)$, the collision conserves both kinetic energy and momentum, because there are initially 5 moving atoms, and after the collision there are altogether 15 moving atoms.

Head-to-head collision of a longer rigid string to a shorter one. In the last case, the red string has only 5 atoms, indices 1..5, while the blue string has 10 atoms, indices 6..15. Again, the blue string of atoms is set into a motion that causes a head-to-head collision with the red string of atoms, and the red string of atoms is at rest. Thus, the formal model is:

$$\vec{c}_i(0) = (-2i, 0, 0),\ i \in [1..5]$$
$$\vec{c}_i(0) = (2i - 10, 0, 0),\ i \in [6..15]$$
$$\vec{v}_i(0) = (0, 0, 0),\ i \in [1..5]$$
$$\vec{v}_i(0) = (-1, 0, 0),\ i \in [6..15]$$
$$\vec{j}_i = \sum_{k=1}^{5} \vec{fb}(i, 2|k - i|, k),\ i \in [1..4],\ k \neq i$$
$$\vec{j}_5 = \vec{fc}(5, 6) + \sum_{k=1}^{4} \vec{fb}(5, 2|k - 5|, k)$$
$$\vec{j}_6 = \vec{fc}(6, 5) + \sum_{k=7}^{15} \vec{fb}(6, 2|k - 6|, k)$$
$$\vec{j}_i = \sum_{k=6}^{15} \vec{fb}(i, 2|k - i|, k),\ i \in [7..15],\ k \neq i$$

Figure 16 depicts the motion of the two strings. As the figure shows, the shorter red string bounces away from the longer blue string. The blue string maintains some of its initial velocity, as its velocity after the collision is approximately $(-0.33, 0, 0)$ while the velocity of the red string is approximately $(-1.33, 0, 0)$. These velocities are again in accordance with velocities computed using text-book equations on a collision of particles with different masses [5].

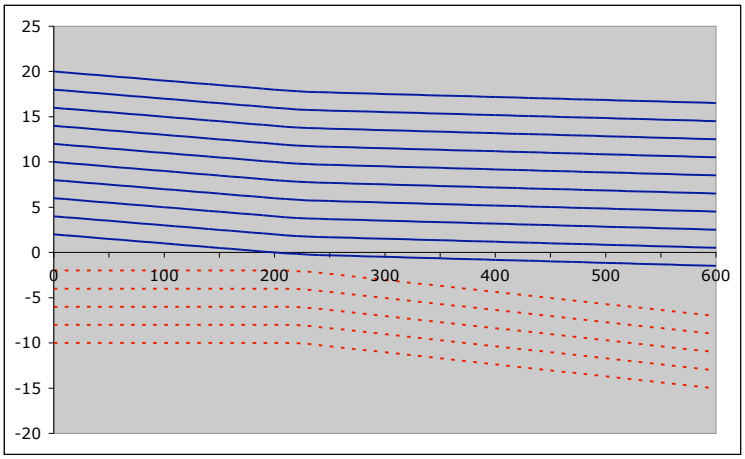

Figure 17: Collision of a longer rigid string of atoms to a shorter rigid string of atoms; the blue lines indicate the positions of the first and the last atom in the longer string on the x-axis, and the red lines indicate the positions of the first and the last atom in the shorter string on the x-axis with respect to the number of iterations

Summary. All the cases above show the expected dynamics when strings of atoms, with various lengths and bonds, have a perfectly aligned head-to-head collision. They confirm that composing bond forces together with collision forces produces results that are in accordance with dynamics of rigid, indivisible, particles. In particular, these cases confirm the earlier claim in Section 4.2 that bonds define the rigidness of a string of atoms.

4.6 Non-aligned collision of two non-rigid strings

In this very last example, we go beyond text-book equations for collisions, and study a non-aligned collision of two non-rigid strings. For such strings, there are no easily computable functions that express the exact motion before, during, and after the collision. In our atomic approach, however, it is easy to define the model of strings with proper forces. Moreover, the computational complexity depends only on the number of forces in the model.

In this example, we consider two strings with 70 atoms in each string. Thus, the atoms of one string have the indices 1..70, and the atoms of the other string have the indices 71..140.

The atoms in a string partially overlap, giving an impression of dense material for the collision. The strings are aligned diagonally with respect to the y and z-axes. Also, the alignment of the strings with respect to each other is diagonal and non-symmetric. Consequently, only some of the atoms in the two strings participate in the collision.

The atoms in the strings are bound only to their neighboring atoms. In addition, the initially non-overlapping atoms are prevented from overlapping by including proper collision forces.

Formally, the model for this last example is:

$$
\begin{aligned}
&\vec{c}_i(0) = (-6,\ 0.25i - 10.75,\ 0.25i - 10.75),\ i \in [1..70] \\
&\vec{c}_i(0) = (6,\ 0.25i - 28.25,\ 28.25 - 0.25i),\ i \in [71..140] \\
&\vec{v}_i(0) = (3, 0, 0),\ i \in [1..70] \\
&\vec{v}_i(0) = (-3, 0, 0),\ i \in [71..140] \\
&\vec{j}_i = \textstyle\sum_{k=1}^{70} \vec{fc}(i,k) + \sum_{m=71}^{140} \vec{fc}(i,m) \\
&\quad + \textstyle\sum_{l=max\{1,i-1\}}^{min\{70,i+1\}} \vec{fb}(i, \sqrt{0.125}, l), \quad i \in [1..70],\ |k-i| \geq 8,\ l \neq i \\
&\vec{j}_i = \textstyle\sum_{k=1}^{70} \vec{fc}(i,k) + \sum_{m=71}^{140} \vec{fc}(i,m) \\
&\quad + \textstyle\sum_{l=max\{71,i-1\}}^{min\{140,i+1\}} \vec{fb}(i, \sqrt{0.125}, l), \quad i \in [71..140],\ |m-i| \geq 8,\ l \neq i
\end{aligned}
$$

Figure 18 shows a series of images depicting the motion of the strings. As the series of images reveal, there is not only one collision, but two collisions, as the strings collide again some time after the first collision. This really shows the importance of emergent dynamics, as the second collision was not anticipated when the model was formulated. Note also the intricate entanglement of the strings during the collision as revealed by the series of images in the figure.

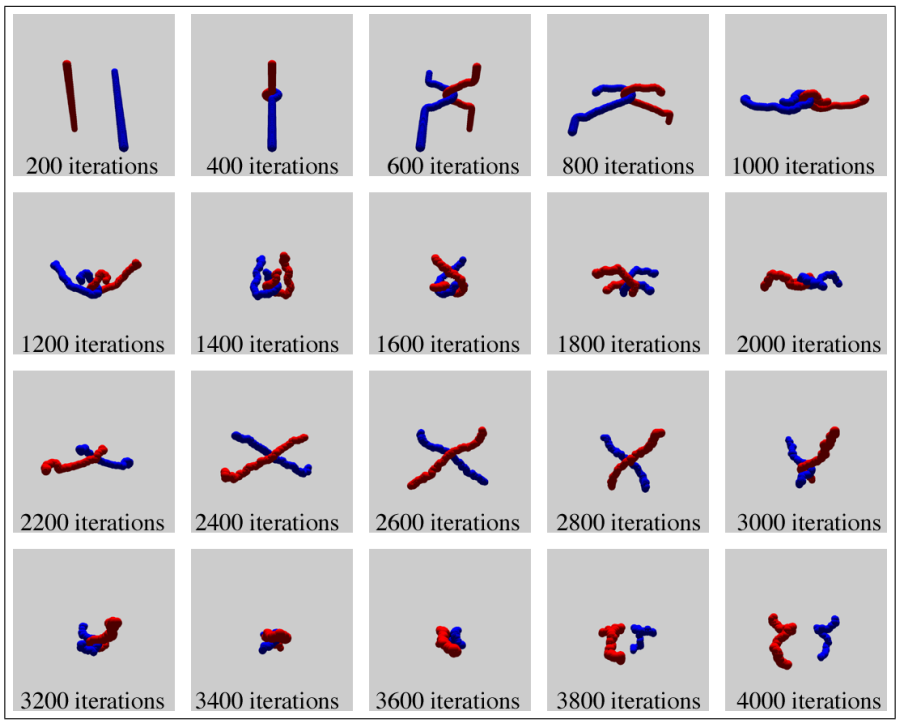

Figure 18: Motion of two non-rigid strings having a non-aligned collision; the number of computed iteration rounds is shown in each image

For this model, the 4000 iteration rounds is quickly computed even with a modest computer. For instance, a 500MHz PowerPC G3-processor computes it in about 3.2 seconds. This time is really spent by floating point operations, since there are 280 bond forces and and 17864 collision forces to be computed per each iteration round. Thus, within 4000 iterations rounds, the computer has to compute a total of 1 120 000 bond forces and 35 728 000 collision forces.

This last example shows well how in our atomic approach detailed dynamics emerges from an underspecified formal model. Moreover, despite the computational load of the model, the time needed by the computations remains acceptable.

5 Conclusion

In this paper, we discussed previsualization in *robotics* [2]. For this purpose, we proposed an atomic approach based on use of atoms. Atoms are indivisible objects of constant shape and mass. They obey simple, compositional, interaction laws. The laws produce emergent interaction dynamics, as components composed of atoms interact with each other.

We formalized in this paper atoms, their model of motion, and the basic

forces affecting the atoms: damping force, binding force, and a collision force. We illustrated the effect of the forces with many examples that combined the forces in different ways. The results were in accordance with the results obtained using corresponding text-book equations.

In the last example, we illustrated the advantage of our atomic approach, as we formalized and visualized a non-trivial setting, where two non-rigid strings have a non-aligned collision. The model required a lot of computation; however, the time spent by the computation was nevertheless acceptable.

The last example also showed, how detailed dynamics emerges from an abstract and underspecified model. Without emergent dynamics, we would have to define a detailed model, and it would show only the defined dynamics. Discoveries, like the second collision between the two strings in the last example, do not show without emergent dynamics.

Although our atomic approach gives reasonable results, the results are not exact. One reason for this is that we use an approximation equation as the model of motion. Another reason is that the computed model lacks some details. However, this is not a limitation, since a better approximation equation for the model of motion, or a more detailed model would yield more accurate results. The drawback of these improvements is that the computation becomes slower; however, loss of performance can be compensated using better computation techniques [7, 13]. Still, the increase in the details of a model results nevertheless in loss of generality.

Although the results in this paper confirm that our atomic approach has potential for previsualization in robotics, there are still many interesting open issues. Clearly, the atomic approach must also be tested with more versatile examples than those given in this paper.

Perhaps, the most important open issue is modularity. Currently, the more atoms there are in a model, the more impulse equations we are forced to write. In the worst case, we are forced to rewrite the same impulse equations for any two components that are similar. From the modelling point of the view, this is unacceptable. With modularity, we would avoid this problem. We could then also compose bigger components out of smaller components. Clearly, modularity reflects also somehow in the emergent dynamics; thus, making the study of modularity in our approach a very appealing topic for future research.

Acknowledgments

I wish to thank Pekka Kilpeläinen and Tapio Grönfors for their reading and comments on the earlier versions of this report.

References

[1] B. Ferster. *Idea Editing: Previsualization for Feature Films.* `http://www.stagetools.com/previs.htm`, October 9, 2003.

[2] R. C. Arkin. *Behavior-Based Robotics.* MIT Press, Cambridge 1998.

[3] P. Mosterman. An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages. In F. W. Vaandrager, and J. H. van Schuppen. editors, *Proceedings of Hybrid Systems: Computation and Control*, LNCS 1569, pp. 165–177, Springer-Verlag, Berlin, 1999.

[4] *Emergent Systems: A Discussion.* Center for Science in Society, Bryn Mawr College.
`http://serendip.brynmawr.edu/local/scisoc/emergence`, October 14, 2003.

[5] H. D. Young, and R. A. Freedman; contributing authors, T. R. Sandin, A. Lewis Ford. *University Physics with Modern Physics.* Tenth edition, Addison-Wesley, San Francisco, 2000.

[6] W. T. Reeves. Particle Systems - A Technique for Modeling a Class of Fuzzy Objects. *Computer Graphics*, vol. 17, no. 3, pp. 359-376, 1983.

[7] *N-Body / Particle Simulation Methods.* Amara Graps.
`http://www.amara.com/papers/nbody.html`, October 9, 2003.

[8] *Physical Based modeling: Principles and Practice.* Andrew Witkin, Online Siggraph '97 Course notes.
`http://www-2.cs.cmu.edu/~baraff/sigcourse`, October 9, 2003.

[9] *The Internet Encyclopedia of Philosophy.*
`http://www.utm.edu/research/iep/`, October 9, 2003.

[10] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, vol. 21, no. 4, pp. 25-34, 1987.

[11] R. A. Adams. *Calculus: A Complete Course.* Fourth edition, Addison-Wesley, Don Mills, 1999.

[12] D. G. Zill, and M. R. Cullen. *Differential Equations with Boundary-value problems.* Fourth edition, Brooks/Cole Publishing Company, Pacific Grove, 1997.

[13] D. O. Kutz, and R. R. Eckert. Enhancing the Control and Performance of Particle Systems Through the Use of Local Environments. Manuscript, State University of New York at Binghamton,
`http://www.cs.binghamton.edu/~reckert`, October 14, 2003.

[14] B. Stroustrup. *The C++ Programming Language, Third Edition.* Addison Wesley Longman, Reading, 1997.

[15] *Home page of OpenGL.*
`http://www.opengl.org/`, October 9, 2003.

A An Object-Oriented Implementation

The examples shown in this paper were implemented using *C++* [14] and *OpenGL* [15]. The code common to all examples was captured using four classes and a singleton class capturing the visualization routines. The four classes encode atoms, forces, and vector computations along with an interface to OpenGL. Clearly, the same could have been achieved using less classes; however, the shown class hierarchy was the result of some experimentation aiming at a compact and efficient class description for an atom. Figure 19 depicts the classes, routines, and their associations.

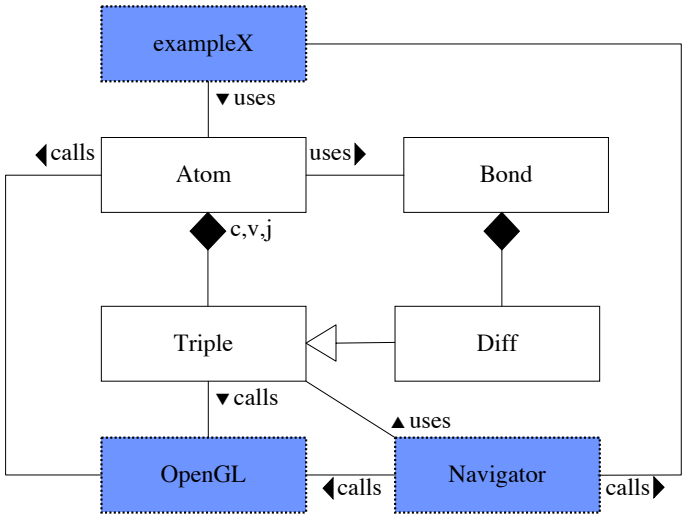

Figure 19: Classes used to implement the examples in this paper; ExampleX is a sigleton class of routines defining the atoms and forces of each example; Navigator is a singleton class containing visualization routines; OpenGL represents the library of OpenGL routines

From the classes shown in the figure above:

- *Triple* captures vector computations and provides an interface to OpenGL
- *Diff* is a support class for computing the difference of two coordinates given as triples
- *Bond* is a support class for computing the bond coefficient with respect to two coordinates given as triples, and a desired distance
- *Atom* captures an atom with its properties and forces

There were mainly two reasons for having the four classes above: compactness and efficiency. In short, some details concerning the bonds are expressed in

respective classes; consequently, copy constructors can be used to enhance the performance of computations involving objects. As a result, the class capturing an atom, shown in Figure 20, is also compact.

```
#ifndef ATOM_H
#define ATOM_H
#include "bond.h"

class Atom
{ private:
  Triple c,v,j;

  public:
  static const float tick=0.01;
  static const float radius=1.0;
  static const float radii=2.0*radius;

  Atom() : c(),v(),j() {}
  Atom(const Triple &_c,const Triple &_v) : c(_c),v(_v),j(){}
  Atom(const Atom &_a) : c(_a.c),v(_a.v),j(_a.j) {}

  void impulse(float _c, const Triple &_d) {j.add(_c,_d);}
  void move() {v.add(1,j); j=Triple(); c.add(tick,v);}

  void damp(float _c) {impulse(-_c,v);}
  void bind(const Bond &_b, Atom &_a) {impulse(_b.c,_b.d); _a.impulse(-_b.c,_b.d);}
  void bind(float _d, Atom &_a) {Bond b=Bond(c,_a.c,_d); bind(b,_a);}
  void collide(Atom &_a) {Bond b=Bond(c,_a.c,radii); if (b.c<0) bind(b,_a);}
};

#endif
```

Figure 20: C++ class code for an atom