

## Accepted Manuscript

Towards Particle System Based Stress Visualization

Mauno Rönkkö, Pekka Toivanen

PII: S1569-190X(11)00023-2  
DOI: [10.1016/j.simpat.2011.02.004](https://doi.org/10.1016/j.simpat.2011.02.004)  
Reference: SIMPAT 1062

To appear in: *Simulation Modeling Practices and Theory*

Received Date: 8 June 2009  
Revised Date: 16 November 2010  
Accepted Date: 14 February 2011



Please cite this article as: M. Rönkkö, P. Toivanen, Towards Particle System Based Stress Visualization, *Simulation Modeling Practices and Theory* (2011), doi: [10.1016/j.simpat.2011.02.004](https://doi.org/10.1016/j.simpat.2011.02.004)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# Towards Particle System Based Stress Visualization

Mauno Rönkkö<sup>a,\*</sup>, Pekka Toivanen<sup>a</sup>

<sup>a</sup> *School of Computing, University of Eastern Finland, P.O.Box 1627, FIN-70211 Kuopio, FINLAND*

---

## Abstract

A particle system, as understood in computer science, is a novel technique for modeling robots in their environment. Particle systems have traditionally been used for modeling complex dynamics of fluids and gases. In this paper, as the main contribution, we adapt our earlier work on particle systems, to compute a preliminary stress visualization for a bipedal robot walking on a soft sediment. The underlying problem of modeling rigid objects with particles is solved by introducing rack particles that enforce structural rigidness while maintaining deformability under stress. The presented approach opens many new possibilities, as it provides a computationally lightweight and unified, complementary framework for computing a stress of interacting, moving components with underspecified, non-trivial materialistic properties.

*Key words:* particle systems, robotics, stress analysis, emergent dynamics, complex systems

---

---

\* Corresponding author.

*Email addresses:* [mauno.ronkko@uef.fi](mailto:mauno.ronkko@uef.fi) (Mauno Rönkkö),  
[pekka.toivanen@uef.fi](mailto:pekka.toivanen@uef.fi) (Pekka Toivanen).

A stress analysis involving objects with materialistic properties presents numerous mathematical challenges [1–8]. The associated problems have traditionally been tackled by using numerical methods [2,9], including spring-mass models [8,10,11] and finite element methods [4–6,12–14]. These methods provide reliable results; however, their use is typically constrained to well specified dynamics. Because of this, the traditional methods do not fully support early stages of development, where many of the properties, including the interaction dynamics, are underspecified or even unknown.

In this paper, we investigate a complementary approach. As the main contribution, we propose here the use of a particle system to compute a preliminary stress visualization in early stages of the development process. To do this, we adapt our earlier work on particle systems [15–18], particularly on modeling and simulating non-linear dynamics of non-rigid bodies in a mechanical setting [19–24] to include also the dynamics of rigid but deformable bodies.

To illustrate the proposed approach, we model, simulate, and visualize the dynamics of a bipedal robot walking on a soft sediment. The research question explored here deals with the use of a walking robot in underwater archeology. Namely, does a bipedal robot maintain a better visibility than a vehicle with a propulsion system when the sediment floor contains scattered artifacts. Some earlier studies on underwater walking exists [25,26] but not in connection to robotics. Still, an excellent survey on various aspects related to design and control of underwater robotics is done by Yuh [27]. The research presented here is in its early stages, whereby we first wish to know how the shape of the robot affects its motion and structural stress during the walking. At this stage, we do not even consider the effect of the arms on the dynamics, which is a significant factor for small robots as pointed by Yuh [27]. Consequently, the effect of water currents on the robot is also omitted here.

### 1.1 *Related work*

Particle systems have been used successfully for decades in computer science [2,8,28–30] and in physics [30–32] to model complex dynamics of, for instance, fluids and gases as well as elastic and deformable bodies. Particle systems are computationally attractive, because the models are simple difference equations over matrices independently of the underlying complexity of the simulated objects. Consequently, the models are compact and computationally lightweight. Moreover, particle systems can be used to simulate non-linear dynamics of objects with underspecified properties, because the simulation is rich with

emergent dynamics [15,19,33] that compensates for the missing parameters. Emergent dynamics typically exhibits a level of details [19] that is hard or impossible to capture otherwise, and it may capture anything from simple friction [20] to complex hunting behavior [15]. Formally, particle systems capture so called basic emergence [33], which means behavior that is reducible to particle-to-particle interactions without any evolutionary processes involved. The presented work can be seen as a continuation to work done by Jansson et al. [2] and Terzopoulos et al. [8] with an emphasis on conservative incorporation of rigid body dynamics into the particle system framework. In particular, when comparing the presented work with that in [8], the forces here are computationally simpler, requiring less effort in the modeling phase.

Particle systems is a complementary approach to modeling with finite element methods [4,5]. When using a finite element method, the modeled space is partitioned into a predefined mesh and the focus is on the flow of matter and forces within the mesh, whereas with particle systems, the focus is on individual particles and their interaction within a continuous Euclidean space. Because of this, particle systems support modeling of a multi-scale heterogeneous setting better than finite element methods [5,6,13] which excel in modeling structures with homogenous properties. Conversely, when the number of particles increases, the use of finite element methods becomes attractive. As for the computational complexity, particle systems excel in having a computationally compact and lightweight model; according to Kipfer et al. [34], a particle system even with some  $1 \times 10^6$  particles can be computed in real-time by using a graphics processor. However, because particle systems tend to have a quadratic computational complexity with respect to the number of particles, the computation time for particle system with more than  $1 \times 10^9$  particles may well become infeasible. Because of this finite element models excel, for instance, in analyzing fluid flows [12]. In this paper, we consider a model with less than 40,000 particles.

Particle systems is also a complementary approach to use of spring-mass models [8,10,11]. When using a spring-mass model, the focus is in defining the springs and their coefficients between the mass points which represent rigid undeformable bodies. In contrast to this, when using particle systems, the interaction properties are assigned to the particles and particle-to-particle forces; in particular, the particle-to-particle forces may well be temporal and they do not need to be spring-like forces at all. Furthermore, with particle systems, the particles tend to be of uniform shape, whereby even the interacting bigger bodies are composed of smaller particles. In this respect, the introduction of rack particles, proposed in this paper to enforce the rigidness of interacting bodies, has obvious similarities with the idea of adding stiffness constraints to a spring-mass model, as put forward by Provot [10]. However, unlike the addition of stiffness constraints, the use of rack particles does not change the underlying computational process at all.

ACCEPTED MANUSCRIPT

In contrast to other numerical methods used in stress analysis, for instance, by Demirdžić et al. [9] and by Saadawi et al. [14], the particle systems tend to be computationally lighter and more mechanical, as they require no discretization or timing consideration for the modeled space. In particle systems, the particles move and interact in a continuous Euclidean space. In summary, particle systems are an attractive tool for modeling at early stages of the development, when the exact details of the shape parameters and interaction forces are still unknown, and use of less than a million particles suffices.

## 1.2 Overview

In Section 2, we extend and present the approach introduced by Rönkkö, Waldén, and Back [23], where the particle system is formalized using mathematical operators. Then, the model is modular, as it is composed of operators that can be reused in other models. This also means that the model can be extended by introducing new operators into the model. Furthermore, as each operator captures a single aspect of the model, the entire model is decomposable into smaller, intentional units. Also, as the entire model is composed of mathematical operators, the operators also induce the set of difference equations that are then used for computing the simulation. In particular, this paper presents a new application of the method introduced by Rönkkö, Waldén, and Back [23],

In Section 3, we introduce rack particles which are used to enforce structural rigidity to an otherwise non-rigid object composed of a large number of particles. The rack particles interact only with the particles of their host object, enforcing further rigidity by suppressing the wave propagation in the host object. Consequently, an object with rack particles remains deformable.

In Section 4, we present the particle system model for the bipedal walking robot, MagPod. The model consists also a mechanism for the movement.

In Section 5, we present and analyze the simulated motion of MagPod. The analysis confirms that use of rack particles allows modeling of rigid but deformable bodies with particles. A more detailed visual analysis of the simulated dynamics reveals many intricate emergent phenomena associated with bipedalism, for instance, MagPod's body tilts upright as it walks forward. The visual analysis reveals also another intricate emergent phenomenon associated with flat feet and foot biomechanics, namely, the outer sides of MagPod's feet show clear bending due to stress.

Finally, in Section 6, we conclude and discuss future work.

As in [23], we formalize particle systems using mathematical operators. We start by formalizing particles as 5-tuples. We then formalize operators as higher-order functions and operator sequences as forward compositions of operators. After that, we formalize the model of motion as an operator sequence, and present the induced set of difference equations that is used for computing the simulation. Lastly, we formalize the primitive operators for all the forces that are used in this paper to model the robot.

### 2.1 Particles and operators

In general, a particle may be of any shape [32]; however, we assume throughout this paper that a particle is a sphere with a varying radius, but with a constant and uniformly distributed mass. In an earlier article [23], we formalized a particle as a 4-tuple. In this paper, however, we need to use the original position of a particle as a reference, whereby we formalize a particle as a 5-tuple.

**Particles.** We model particles using five variables: original position, current position, velocity, acceleration, and radius. From these variables, only the radius is a positive real-valued number,  $\mathbb{R}_+$ , whereas all the other variables are three-dimensional vectors,  $\mathbb{R}^3$ . As  $\mathbb{R}^3$  refers here to a subspace of Euclidean space, we shall denote the components of  $\mathbb{R}^3$  by  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$ , respectively.

Formally, a particle type  $\mathbf{P}$  is a 5-tuple:

$$\mathbf{P} \triangleq (\mathbb{R}^3, \mathbb{R}^3, \mathbb{R}^3, \mathbb{R}^3, \mathbb{R}_+) \quad (1)$$

In  $\mathbf{P}$ , the components represent in the given order the original position, the current position, the velocity, the acceleration, and the radius. We shall denote these components by  $\mathbf{o}$ ,  $\mathbf{p}$ ,  $\mathbf{v}$ ,  $\mathbf{a}$ , and  $\mathbf{r}$ , respectively. For instance, the velocity of a particle  $\mathbf{p}$  along the  $y$ -axis is referred to as  $\mathbf{p}_{vy}$ . A set of particles is expressed as a particle vector; thus, for instance,  $\mathbf{P}^{10}$  captures ten particles. Then, for instance, the velocity vector of the  $i$ 'th particle in a particle vector  $\mathbf{p}$  is referred to as  $\mathbf{p}_{iv}$ .

**Operators.** As in [23], we formalize an operator as a parameterized collection of maps. In general, a map is a function from one domain to another. In this paper, we discuss only maps that are functions from one particle vector to another. The particle vectors need not be of the same dimension. Such a

map from a vector with  $m$  particles to a vector with  $n$  particles is expressed as  $\mathbf{P}^m \rightarrow \mathbf{P}^n$ .

Let  $\Sigma$  denote a (possibly empty) parameter domain. Then, an operator  $\mathbf{o}$  is expressed as a higher-order function of the form  $\Sigma \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^n)$ . Thus, for some parameter values  $\sigma \in \Sigma$ , the expression  $\mathbf{o}(\sigma)$  returns a map of the form  $\mathbf{P}^m \rightarrow \mathbf{P}^n$ , and for a vector  $\mathbf{p}$  of the form  $\mathbf{P}^m$ , the expression  $\mathbf{o}(\sigma)(\mathbf{p})$  returns a vector of the form  $\mathbf{P}^n$ .

**Operator sequences.** Operators can be applied in sequence. As operators are higher-order functions, the sequential application of operators falls under the standard forward composition [35]. Consequently, a sequence of operators remains as an operator also in the formal sense.

Consider operators  $\mathbf{a} : \Sigma \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^n)$  and  $\mathbf{b} : \Gamma \rightarrow (\mathbf{P}^n \rightarrow \mathbf{P}^k)$ , parameters  $\sigma \in \Sigma$  and  $\gamma \in \Gamma$ , and a vector  $\mathbf{p}$  of the form  $\mathbf{P}^m$ . Then, the forward composition  $\mathbf{a}(\sigma) \bullet \mathbf{b}(\gamma)$  gives the map  $\mathbf{P}^m \rightarrow \mathbf{P}^k$  and is defined as [23]:

$$(\mathbf{a}(\sigma) \bullet \mathbf{b}(\gamma))(\mathbf{p}) \triangleq \mathbf{b}(\gamma)(\mathbf{a}(\sigma)(\mathbf{p})) \quad (2)$$

In particular, we define the convention:

$$\mathbf{p} \bullet \mathbf{a}(\sigma) \triangleq \mathbf{a}(\sigma)(\mathbf{p}) \quad (3)$$

Note that we use here a left associative forward composition, so that the application of the operators in a sequence can be read from left to right. This clarifies the equations.

**Matrix definitions.** In the sequel, we need to express simple matrices with specific element values. The matrices are used only locally and are not referenced later. To avoid unnecessary definition of temporary variables, we use the following convention to define a matrix with  $m$  rows and  $n$  columns, where each element has the value given by the expression  $expr(i, j)$ :

$$[expr(i, j)]_{i=1..m, j=1..n}$$

**Conditional expressions.** We also use conditional expressions within matrices. To clarify the conditional expressions in the formulas, we define the following function:

$$\langle P \rangle = \begin{cases} 1, & \text{if the predicate } P \text{ holds} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

## 2.2 Model of motion

The model of motion for the particle system is traditionally [32,23] given as an Euler approximation [36] of the Newtonian model of motion [37]. As in [23], we formalize it here as an operator sequence.

The model of motion is a sequence of two operators, `netForce` and `integrate`. The `netForce` captures all the forces in the model, and is, thus, model dependent. The `integrate` operator, however, integrates all the forces into the motion by computing the actual Euler approximation of the Newtonian dynamics, and can, therefore, be formalized independently of the model.

**Integrating.** The `integrate` operator retains the original position and the size of the particles, while integrating accumulated acceleration to the velocity, and the velocity to the position of the particles. The integration is computed as an Euler approximation of the trivial Newtonian model with forward error correction. The `integrate` operator also zeros the acceleration value, to allow a new accumulation of acceleration. As Euler approximation is a numerical approximation, the resulting accuracy depends on a coefficient that acts as a fixed time step. Let  $\mathbf{p}$  be a particle vector of the form  $\mathbf{P}^m$  and  $t$  be the fixed time step. Then, the integration is expressed an operator of the form `integrate` :  $(\mathbb{R}) \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^m)$ , and we define it as [23]:

$$\text{integrate}(t)(\mathbf{p}) \triangleq \mathbf{p} + [(\emptyset, t\mathbf{p}_{iv} + t^2\mathbf{p}_{ia}, t\mathbf{p}_{ia}, -\mathbf{p}_{ia}, \emptyset)]_{i=1..m} \quad (5)$$

Thus, when using forward error correction, the update velocity is obtained from the current velocity by adding the acceleration multiplied by the time step. Similarly, the updated position is obtained from the current position by adding the updated velocity multiplied by the time step. Forward error correction is used, because it improves the stability of simulated dynamics for a particle system [18]. For instance, the classical equation  $\mathbf{p}_p + t\mathbf{p}_v + \frac{1}{2}t^2\mathbf{p}_a$  gives a very unstable motion for the particles in the presence of collision and bidding forces defined later. Note that there are alternative integration schemes that could be used in the particle system. Note also that, when choosing the value for the time step, there is a trade off between the computation accuracy and the computation time; a smaller time step gives more accurate computation, but it also causes the computation to take more time.

**The motion.** Let  $\mathbf{p}$  be a particle vector of the form  $\mathbf{P}^m$ , and  $\text{netForce}$  be a model dependent operator of the form  $\text{netForce} : (\emptyset) \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^m)$ , capturing all the forces in the model. Also, let  $t$  be the fixed time step. Then, the model of motion is an operator of the form  $\text{motion} : (\mathbb{R}) \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^m)$ , and we define it as [23]:

$$\text{motion}(t) \triangleq \text{netForce}() \bullet \text{integrate}(t) \quad (6)$$

The actual simulation is obtained by iteratively computing the model of motion. There are three distinct computational phases: computation of all the affecting forces, computation of updated velocities, and computation of updated positions. Thus, the model of motion induces a set of difference equations that are used for computing the simulation:

$$\begin{aligned} \mathbf{p}'_a &= (\mathbf{p} \bullet \text{netForce}())_a \\ \mathbf{p}'_v &= \mathbf{p}_v + t\mathbf{p}'_a \\ \mathbf{p}'_p &= \mathbf{p}_p + t\mathbf{p}'_v \end{aligned} \quad (7)$$

### 2.3 Primitive operators for forces

Although the model of the walking robot comprises of many objects, all composed of particles, the operators forming the  $\text{netForce}$  operator are nevertheless based on a set of primitive operators that we formalize next. Note that some operators require coefficient matrices as parameters. The values for these matrices are obtained through experimentation. There is a correlation between the values in the coefficient matrices and the physical phenomena; however, due to particle system formulation, the actual values differ from those used in standard physics equations.

**Forcing.** A constant force on a particle is modeled implicitly as its effect, as acceleration. Thus, an operator modeling a constant force adds a constant acceleration to the particles. Let  $\mathbf{p}$  be a particle vector of the form  $\mathbf{P}^m$ , and  $\hat{a}$  be a matrix of the form  $\mathbb{R}^{m \times 3}$  capturing the acceleration. Then, a constant force is modeled as an operator of the form  $\text{force} : (\mathbb{R}^{m \times 3}) \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^m)$ , and we define it as [23]:

$$\text{force}(\hat{a})(\mathbf{p}) \triangleq \mathbf{p} + [(\emptyset, \emptyset, \emptyset, \hat{a}_i, \emptyset)]_{i=1..m} \quad (8)$$

**Damping.** The damping force captures loss of kinetic energy; thus, the actual damping force depends on the velocity of a particle. Let  $\mathbf{p}$  be a particle

vector of the form  $\mathbf{P}^m$ , and  $c$  be a real-valued number capturing the magnitude of the damping. The damping force is then modeled as an operator of the form  $\text{dampen} : (\mathbb{R}) \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^m)$ , and we define it as [23]:

$$\text{dampen}(c)(\mathbf{p}) \triangleq \text{force}([-c\mathbf{p}_{iv}]_{i=1..m})(\mathbf{p}) \quad (9)$$

Note that there could also be alternative damping forces defined in the model; however, here we only use linear damping forces.

**Colliding.** We model the generic collision forces on particles as their effect, as acceleration on the particles. We formalize the collision forces as an operator that adds acceleration due to collision forces on a particle vector that collides with another particle vector. The magnitude of the acceleration is controlled with a coefficient matrix.

For the collision operator, we first define an auxiliary function,  $\text{col}$ . It returns for a single particle the acceleration due to collision forces with a particle vector. The returned acceleration is the sum of all the accelerations due to the individual particle-to-particle collisions. The magnitude of the particle-to-particle collision forces are controlled with a coefficient matrix. Let  $\mathbf{p}$  be the colliding particle,  $\mathbf{u}$  be a particle vector of the form  $\mathbf{P}^n$ , and  $\hat{c}$  be a coefficient matrix of the form  $\mathbb{R}^n$ . Then, the function  $\text{col} : (\mathbf{P}, \mathbb{R}^n, \mathbf{P}^n) \rightarrow \mathbb{R}^3$  is defined as [23]:

$$\text{col}(\mathbf{p}, \hat{c}, \mathbf{u}) \triangleq \sum_{i=1}^n \hat{c}_i (\mathbf{u}_{ip} - \mathbf{p}_p) \min\{0, 1 - \frac{(\mathbf{u}_{ix} + \mathbf{p}_x)^2}{\|\mathbf{u}_{ip} - \mathbf{p}_p\|^2}\} \quad (10)$$

Here, for the pair of particles  $\mathbf{p}$  and  $\mathbf{u}_i$ , the  $\min\{\cdot\}$  expression evaluates to 0 if the two particles do not overlap, that is, their distance is greater than the sum of their radii. Consequently, the particles are considered to collide, only if they overlap. Then, the collision force for the particle  $\mathbf{p}$  is directed directly away from the particle  $\mathbf{u}_i$ , because the  $\min\{\cdot\}$  expression becomes negative. The magnitude of the collision force is controlled by  $\hat{c}_i$ .

The  $\text{col}$  function, as defined above, is a generalization of the collision function used before for computing the collision of particles in a particle system [15,19–22,16–18]. The characteristics of that collision function are discussed in more detail in a technical report [38].

We formalize now the generic collision operator for a particle vector. Let  $\mathbf{p}$  be the colliding particle vector of the form  $\mathbf{P}^m$ , and  $\mathbf{u}$  be a particle vector of the form  $\mathbf{P}^n$  capturing the particles that collide with  $\mathbf{p}$ . Also, let  $\hat{c}$  be a matrix of the form  $\mathbb{R}^{m \times n}$  capturing the collision coefficients for all the pairs of particles of  $\mathbf{p}$  and  $\mathbf{u}$ . Then, the collision operator is of the form

$$\text{collide}(\mathbf{u}, \hat{c})(\mathbf{p}) \triangleq \text{force}([\text{col}(\mathbf{p}_i, \hat{c}_i, \mathbf{u})]_{i=1..m})(\mathbf{p}) \quad (11)$$

**Binding.** Similarly to collision forces, we model the generic binding forces on particles as their effect, as acceleration on the particles. The purpose of the binding forces is to try to maintain the original distance between the bound particles. We formalize the binding forces as an operator that adds acceleration on a particle vector that is bound to another particle vector. The magnitude of the acceleration is controlled with a coefficient matrix.

For the binding operator, we first define an auxiliary function, **bin**. It returns for a single particle the acceleration due to binding forces with a particle vector. The returned acceleration is the sum of all the accelerations due to the individual particle-to-particle bindings. The magnitudes of the particle-to-particle binding forces are controlled with a coefficient matrix. Let  $\mathbf{p}$  be the particle to be bound,  $\mathbf{u}$  be a particle vector of the form  $\mathbf{P}^n$ , and  $\hat{c}$  be a coefficient matrix of the form  $\mathbb{R}^n$ . Then, the function  $\text{bin} : (\mathbf{P}, \mathbb{R}^n, \mathbf{P}^n) \rightarrow \mathbb{R}^3$  is defined as:

$$\text{bin}(\mathbf{p}, \hat{c}, \mathbf{u}) \triangleq \sum_{i=1}^n \hat{c}_i (\mathbf{u}_{ip} - \mathbf{p}_p) \left(1 - \frac{\|\mathbf{u}_{io} - \mathbf{p}_o\|^2}{\|\mathbf{u}_{ip} - \mathbf{p}_p\|^2}\right) \quad (12)$$

Here, for the pair of particles  $\mathbf{p}$  and  $\mathbf{u}_i$ , the rightmost term of the expression evaluates to 0 if the two particles are as far away from each other as originally. If the particles are closer to each other than originally, the rightmost expression becomes negative, causing the particles to repel each other. Similarly, if the particles are further away from each other than originally, the rightmost expression becomes positive, causing the particles to attract each other. Note that the magnitude of a repelling or attracting force depends on the distance of the particles polynomially. Furthermore, the magnitude is controlled by  $\hat{c}_i$ .

The **bin** function, as defined above, is a generalization of the binding function used before for computing the binding of particles in a particle system [15,19–22,16–18]. The characteristics of that binding function are discussed in more detail in a technical report [38].

We formalize now the generic binding operator for a particle vector as an operator called **bind**. Note we give the operator the same name as for the function, because it is clear by the parameters, which one of them is to be applied. Let  $\mathbf{p}$  be a particle vector of the form  $\mathbf{P}^m$  denoting the particles to be bound, and  $\mathbf{u}$  be a particle vector of the form  $\mathbf{P}^n$  capturing the particles that  $\mathbf{p}$  is bound to. Also, let  $\hat{c}$  be a matrix of the form  $\mathbb{R}^{m \times n}$  capturing the binding coefficients for all the pairs of particles of  $\mathbf{p}$  and  $\mathbf{u}$ . Then, the binding

$$\text{bind}(\mathbf{u}, \hat{c})(\mathbf{p}) \triangleq \text{force}([\text{bin}(\mathbf{p}_i, \hat{c}_i, \mathbf{u})]_{i=1..m})(\mathbf{p}) \quad (13)$$

**Attracting.** In addition to the binding forces, we also need linear attraction forces between the particles. Similarly to the binding forces, we model the attraction forces on particles as their effect, as acceleration on the particles. We formalize the attraction forces as an operator that adds acceleration on a particle vector that is attracted towards another particle vector. The magnitude of the acceleration is controlled with a coefficient matrix.

Let  $\mathbf{p}$  be a particle vector of the form  $\mathbf{P}^m$  denoting the particles to be attracted, and  $\mathbf{u}$  be a particle vector of the form  $\mathbf{P}^n$  capturing the particles that  $\mathbf{p}$  is attracted towards. Furthermore, let  $\hat{c}$  be a matrix of the form  $\mathbb{R}^{m \times n}$  capturing the attraction coefficients for all the pairs of particles of  $\mathbf{p}$  and  $\mathbf{u}$ . Then, the attractor operator is of the form  $\text{attract} : (\mathbf{P}^n, \mathbb{R}^{m \times n}) \rightarrow (\mathbf{P}^m \rightarrow \mathbf{P}^m)$ , and we define it as:

$$\text{attract}(\mathbf{u}, \hat{c})(\mathbf{p}) \triangleq \text{force}\left(\left[\sum_{j=1}^n \hat{c}_{ij} \frac{\mathbf{u}_{jp} - \mathbf{p}_{ip}}{\|\mathbf{u}_{jp} - \mathbf{p}_{ip}\|}\right]_{i=1..m}\right)(\mathbf{p}) \quad (14)$$

Note that, unlike for the binding forces, the magnitude of the attraction forces is completely controlled by  $\hat{c}$ . Also, the magnitude of an attraction force depends linearly on the distance of the attracted particles. Consequently, an attraction force is weaker for longer distances than a binding force, and because of this, an attraction force cannot break apart an object with binding forces.

### 3 Modeling rigid objects with particles

In this section, we propose the use of rack particles in modeling rigid objects by using particles and particle-to-particle binding forces only. We start by showing how local deformations propagate as waves throughout an object composed of particles. We then formalize rack particles and associated forces that are used to isolate local deformations by preventing wave propagation. Lastly, we illustrate how rack particles make an object rigid but deformable by showing how a two-dimensional plane with rack particles behaves under a local pulling force.

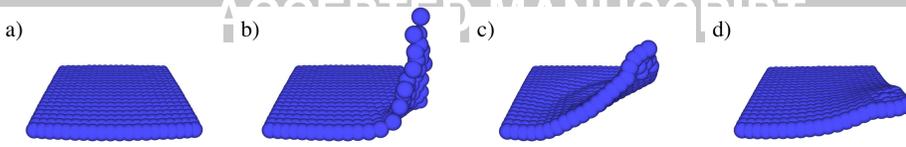


Fig. 1. An increase in the number of binding forces makes a plane stiffer but does not make it rigid, when pulled up from one corner. a) the initial setting, b) a plane with particles bound only to their neighboring particles, c) a plane with particles bound to half of the plane particles, d) a plane with particles bound all of the plane particles.

### 3.1 Deformation of a plane with binding forces

To illustrate how local deformations propagate as waves throughout the whole body of particles, we consider a square plane composed of 400 particles, with a side of 20 particles, as shown in image (a) of Figure 1. We shall vary the number of binding forces within the plane, to show how the plane behaves as a whole when pulled up from a corner. In the model, the motion of the plane is slightly dampened, to prevent cascading waves from occurring.

We shall denote the plane by a particle vector **plane** of the form  $\mathbf{P}^{400}$ . The plane is subjected to a force pulling it up from the corner nearest to the origin  $(0, 0, 0)$ . For the force, we define an acceleration matrix *pull* of the form  $\mathbb{R}^{400 \times 3}$ :

$$pull \triangleq \left[ \langle \|\mathbf{plane}_{i_o}\|^2 < 10 \rangle \cdot (0, 100, 0) \right]_{i=1..400} \quad (15)$$

For the internal binding forces we define an auxiliary function, *within*(*d*), that returns a matrix of the form  $\mathbb{R}^{400 \times 400}$  capturing the binding coefficients for all the pairs of particles of **plane**. The variable *d* determines a limit distance; all the particles *j*, whose original distance is within the limit distance *d* of a particle *i* will be bound to the particle *i* with a magnitude of 20 units. A particle *i*, however, is never bound to itself. Formally, *within*(*d*) is expressed as:

$$within(d) \triangleq \left[ 20 \cdot \langle \|\mathbf{plane}_{j_o} - \mathbf{plane}_{i_o}\| < d \rangle \langle i \neq j \rangle \right]_{i=1..400, j=1..400} \quad (16)$$

Now, the model of motion for the plane is:

$$\mathbf{plane} \bullet force(pull) \bullet bind(\mathbf{plane}, within(d)) \bullet dampen(1) \bullet integrate(0.01) \quad (17)$$

Figure 1 shows some selected images from the simulated dynamics of Equation 17. The image (a) of the figure shows the initial setting for the simulation. The image (b) shows an intermediate stage of a simulation with  $d = 2$ . The image (c) shows an intermediate stage of a simulation with  $d = 10$ . Lastly, the image

(d) shows an intermediate stage of a simulation with  $d = 100$ . As the image (b) of Figure 1 shows, an object with only a few binding forces may deform and stretch severely under a force affecting it locally. As the number of the binding forces is increased, the object becomes stiffer; however, the object is still very flexible and it may exhibit waves, as in the image (c) of Figure 1. Even the maximal number of binding forces does not prevent the local deformations from spreading throughout the body of an object, as shown in the image (d) of Figure 1. In the image (d), the plane does still bend visibly, although each particle of the plane is bound to all the particles of the plane. Because of this, the binding forces alone are not enough to capture the dynamics of a rigid object.

The increase in the number of the binding forces brings also another problem into the model; an object with a large number of binding forces becomes unstable in the presence of local forces. This is due to the used integration scheme formalized in Equation 5. If a force affects only one particle of the object, all the bound particles try to pull that single particle back to its “desired position”. As such a force is linear to the number of the bound particles, it may easily exceed the pulling force, leading to a series of escalating over corrections. For instance, if the magnitude of the binding forces appearing in the function *within()*, Equation 16, was increased from 20 to 100, the particles of the plane with *within*(100) would diverge violently after only a few computation rounds.

### 3.2 Formalization of the rack particles and forces

For modeling a rigid object with particles, we propose here the use of rack particles. Rack particles reside outside the particles of their host object, and they interact only with their host particles. In short, the host particles are bound to the rack particles and vice versa; however, there are no binding forces between any two rack particles or between any two host particles. Because of the missing binding forces between the host particles, the effect of the local forces remains local and there is no wave propagation. Still, because all the host particles are bound to all the rack particles, there is a global effect on the host object due to forces on the host particles. Consequently, the host object becomes rigid but maintains the ability to have local deformations.

In this paper, we use only six rack particles per object. The rack particles are located outside the bounding box of the object, so that there is one rack particle above and below the object, as well as on each side of the object. We place the rack particles so that they align at the center of the bounding box, but so that the distance from a rack particle to the corresponding face of the bounding box is five units. Note that the number and positioning of rack

particles affect the rigidity of the object, whereby a different kind of rack particle configuration might be preferred in other models.

We shall define first a function that constructs the rack particles for a vector of particles. Let  $\mathbf{p}$  be a particle vector of the form  $\mathbf{P}^m$ . Then, the two opposite corners of the bounding box for the particles  $\mathbf{p}$  are given by the functions  $\min$  and  $\max$ , of the form  $\mathbf{P}^m \rightarrow \mathbf{P}$ , which we define as:

$$\min(\mathbf{p}) \triangleq \left( \min_{i=1}^m \{\mathbf{p}_{iox} - \mathbf{p}_{ir}\}, \min_{i=1}^m \{\mathbf{p}_{ioy} - \mathbf{p}_{ir}\}, \min_{i=1}^m \{\mathbf{p}_{ioz} - \mathbf{p}_{ir}\} \right) \quad (18)$$

$$\max(\mathbf{p}) \triangleq \left( \max_{i=1}^m \{\mathbf{p}_{iox} + \mathbf{p}_{ir}\}, \max_{i=1}^m \{\mathbf{p}_{ioy} + \mathbf{p}_{ir}\}, \max_{i=1}^m \{\mathbf{p}_{ioz} + \mathbf{p}_{ir}\} \right) \quad (19)$$

Thus, the center of the bounding box is given by a function  $\text{ctr}$  of the form  $\mathbf{P}^m \rightarrow \mathbf{P}$  as:

$$\text{ctr}(\mathbf{p}) \triangleq \frac{1}{2} \cdot (\min(\mathbf{p}) + \max(\mathbf{p})) \quad (20)$$

Then, the positions of the six rack particles for a particle vector  $\mathbf{p}$  is given by a function  $\text{rck}$  of the form  $\mathbf{P}^m \rightarrow \mathbb{R}^{6 \times 3}$  as:

$$\text{rck}(\mathbf{p}) \triangleq \begin{bmatrix} (\min(\mathbf{p})_x, \text{ctr}(\mathbf{p})_y, \text{ctr}(\mathbf{p})_z) \\ (\text{ctr}(\mathbf{p})_x, \min(\mathbf{p})_y, \text{ctr}(\mathbf{p})_z) \\ (\text{ctr}(\mathbf{p})_x, \text{ctr}(\mathbf{p})_y, \min(\mathbf{p})_z) \\ (\max(\mathbf{p})_x, \text{ctr}(\mathbf{p})_y, \text{ctr}(\mathbf{p})_z) \\ (\text{ctr}(\mathbf{p})_x, \max(\mathbf{p})_y, \text{ctr}(\mathbf{p})_z) \\ (\text{ctr}(\mathbf{p})_x, \text{ctr}(\mathbf{p})_y, \max(\mathbf{p})_z) \end{bmatrix} \quad (21)$$

Now, the function  $\text{rack}$  of the form  $\mathbf{P}^m \rightarrow \mathbf{P}^6$  that constructs the rack particles from a given vector of particles is defined as:

$$\text{rack}(\mathbf{p}) \triangleq [(\text{rck}(\mathbf{p})_i, \text{rck}(\mathbf{p})_i, \emptyset, \emptyset, 1)]_{i=1..6} \quad (22)$$

There are two kind of forces affecting the rack particles: binding forces and damping forces. Throughout this paper we assume that the rack particles  $\text{rack}(\mathbf{p})$  are bound to all the host particles  $\mathbf{p}$  with a magnitude 0.5, and that there is only a minor damping of the rack particles. Thus, we define an operator  $\text{rackFroces}$  of the form  $\mathbf{P}^m \rightarrow (\mathbf{P}^6 \rightarrow \mathbf{P}^6)$  that captures the forces on the rack particles as:

$$\text{rackForces}(\mathbf{p}) \triangleq \text{bind}(\mathbf{p}, [0.5]_{i=1..6, j=1..m}) \bullet \text{dampen}(1) \quad (23)$$

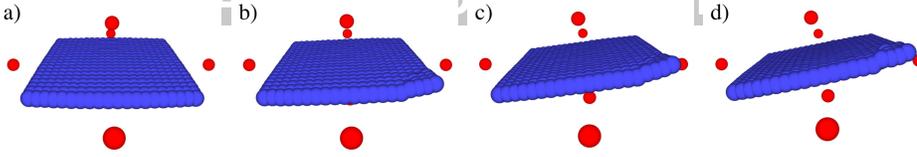


Fig. 2. A plane with rack particles behaves as a rigid object when it is pulled up from one corner: a) the initial setting, b) the state after 60 iterations, c) the state after 120 iterations, d) the state after 180 iterations.

### 3.3 Simulating a rigid plane using the rack particles

We shall now illustrate the use of the rack particles in simulating a rigid but deformable plane. We use the same plane and the same force pulling the plane from one corner as earlier in Section 3.1. Thus, as earlier, we denote the plane by **plane**, which is a vector of the form  $\mathbf{P}^{400}$ . In addition, we denote the rack particles for the plane by **planerack**, which is a vector of the form  $\mathbf{P}^6$  defined initially as:

$$\mathbf{planerack} \triangleq \mathbf{rack}(\mathbf{plane}) \quad (24)$$

As we use rack particles, there are no internal binding forces for the plane; instead, the plane particles are bound to the rack particles. Each plane particle is bound to each rack particle with a magnitude 100. We also use the matrix *pull* for the pulling force, defined earlier in Equation 15. Now, the net forces on the plane are captured as an operator **planeForces** of the form  $\emptyset \rightarrow (\mathbf{P}^{400} \rightarrow \mathbf{P}^{400})$ , and it is defined as:

$$\begin{aligned} \mathbf{planeForces}() &\triangleq \mathbf{force}(\mathit{pull}) \\ &\bullet \mathbf{bind}(\mathbf{planerack}, [100]_{i=1..400, j=1..6}) \\ &\bullet \mathbf{dampen}(1) \end{aligned} \quad (25)$$

With these definitions, the particle system model for the plane and the rack particles is expressed as the composite:

$$\left[ \begin{array}{l} \mathbf{plane} \bullet \mathbf{planeForces}() \\ \mathbf{planerack} \bullet \mathbf{rackForces}(\mathbf{plane}) \end{array} \right] \bullet \mathbf{integrate}(0.01) \quad (26)$$

In the resulting vector, the top 400 particles describe a new state for the plane particles, and the bottom 6 particles describe a new state for the rack particles.

Figure 2 shows some selected images from the simulated dynamics of the model in Equation 26. As the image series of Figure 2 shows, the force pulling the plane from a corner affects only the corner particles causing a local deformation. However, the deformation remains local and does not spread throughout

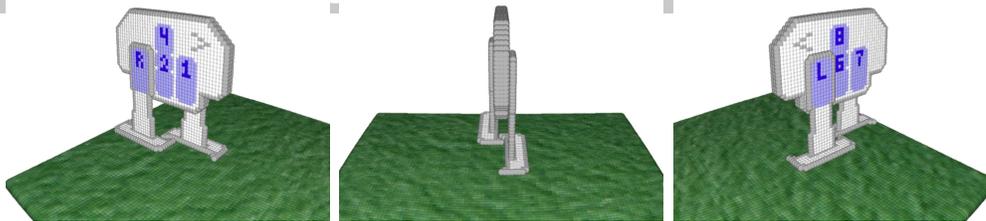


Fig. 3. MagPod shown the from the right, front, and left.

the whole plane as before in Figure 1. Note also that as the force keeps on pulling, the plane and the rack particles start to tilt. The motion is now uniform, and the plane does not bend as in Figure 1. The image series confirms that the rack particles do enforce the dynamics of a rigid object.

## 4 MagPod

MagPod, shown in Figure 3, is a conceptual model of a bipedal robot walking on a soft sediment. We are currently exploring the use of a walking robot in underwater archeology; a bipedal robot maintains a better visibility than a vehicle with a propulsion system when the sediment floor, containing scattered artifacts, is easily stirred.

MagPod has two legs and a body, and it walks by moving its legs synchronously. The motion of legs is realized by augmenting the body with attraction regions that act similarly to electromagnets. The attraction regions are switched on and off in a predetermined sequence and there is an alternation of phases, when a leg attracts the body and when the body attracts a leg. We assume that the body of MagPod is in balance with the buoyancy, and that its legs stick into the soft sediment, giving sufficient support to carry and pull the body. The research presented here is in its early stages, whereby we do not consider any motion of arms or effect of water currents on MagPod's body, yet.

We shall now proceed by formalizing the body and legs of MagPod, the properties of the sediment, and the model of motion.

### 4.1 Body

The body of MagPod is composed of 2907 particles and 6 rack particles, as shown in Figure 4. We shall denote the particles by a vector **body** which is of the form  $\mathbf{P}^{2907}$ . The particles of **body** have no velocity or acceleration

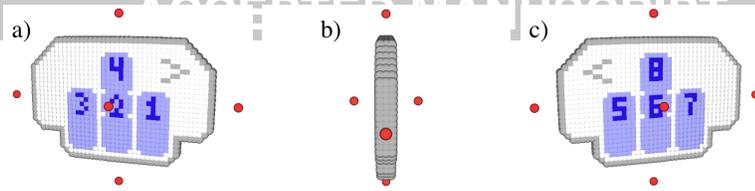


Fig. 4. MagPod’s body, where red particles are the rack particles and blue particles belong to indicated attraction regions: a) right side, b) front view, c) left side.

initially, and they have a radius of 1 unit. The corresponding rack particles, **bodyrack**, are initially given by the function `rack(body)`.

The motion principle is that the legs move synchronously from one attraction region to another in the following predefined sequence, where each tuple shows the number of the activated body region for the right and the left leg respectively:

$$(4, 6) \rightarrow (1, 7) \rightarrow (2, 8) \rightarrow (3, 5)$$

The activation sequence above is then repeated. To get the actual walking motion, the legs are used in turns to pull the body forward. When the body region 6 or 7 is active, the left leg is stuck to the sediment, pulling the body forward. Similarly, when the body region 2 or 3 is active, the right leg is stuck to the sediment, pulling the body forward. The body region 8 is used for pulling the left leg up from the sediment, and the body region 5 is used to push the left leg forward. Similarly, the body regions 4 and 1 are used for pulling the right leg up from the sediment and pushing it forward.

To control the use of the attraction regions in the body, we define an integer matrix, *breg*, which is of the form  $\mathbb{I}^{8 \times 2907}$ . Each row of the matrix gives indexes to the particles with respect to the corresponding region. For instance, *breg*<sub>3</sub> gives the indexes for the body particles when the body region 3 is active. As illustrated in Figure 5, only a particle with a non-zero index belongs to the attraction region. The actual index values are used for aligning attraction regions of the body and a leg, so that when there is an attraction force between the two, the force tries not only to attract, but also to align the leg in an upright position.

In addition to the alternating attraction forces between the attraction regions, the body is subject to a damping force, collision forces toward the legs, and binding forces toward the rack particles. There are no other forces affecting MagPod’s body, as it is assumed to be balanced with the buoyancy. As for the collision forces between the body and the legs, we balance them so that the body actively collides with a leg, only if the leg is pulling the body. If the body is pulling the leg, the leg actively collides with the body. In this way, we avoid simulating an anomaly, where the leg appears to push the body, although the body is supposed to be pulling the leg.

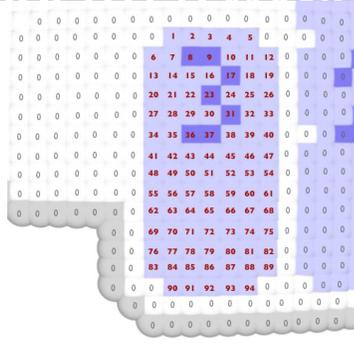


Fig. 5. A partial view of the matrix  $breg_3$  provides indexing for all the body particles with respect to the attraction region number 3.

We shall now express the body forces formally. Similarly to  $breg$ , let  $rreg$  and  $lreg$  denote the index matrices for the attraction regions of the right and the left leg. The matrices  $rreg$  and  $lreg$  are of the form  $\mathbb{I}^{382}$ . Also, let **rightleg** and **leftleg** be vectors of the form  $\mathbf{P}^{382}$  capturing the particles of the right and left leg. Furthermore, let  $r$  and  $l$  be integers denoting the active regions on the right and the left side of MagPod's body. Then, the net forces on MagPod's body are captured as an operator **bodyForces** of the form  $(\mathbb{I}, \mathbb{I}) \rightarrow (\mathbf{P}^{2907} \rightarrow \mathbf{P}^{2907})$ , defined as:

$$\begin{aligned}
 & \text{bodyForces}(r, l) \\
 & \triangleq \text{attract}(\mathbf{rightleg}, [200\langle r \in \{2, 3\} \wedge breg_{ri} = rreg_j \neq 0 \rangle]_{i=1..2907, j=1..382}) \\
 & \bullet \text{attract}(\mathbf{leftleg}, [200\langle l \in \{6, 7\} \wedge breg_{li} = lreg_j \neq 0 \rangle]_{i=1..2907, j=1..382}) \\
 & \bullet \text{collide}(\mathbf{rightleg}, [100\langle r \in \{2, 3\} \rangle]_{i=1..2907, j=1..382}) \\
 & \bullet \text{collide}(\mathbf{leftleg}, [100\langle l \in \{6, 7\} \rangle]_{i=1..2907, j=1..382}) \\
 & \bullet \text{bind}(\mathbf{bodyrack}, [100]_{i=1..2907, j=1..6}) \\
 & \bullet \text{dampen}(10)
 \end{aligned} \tag{27}$$

As the operator above clearly shows, the body collides with a leg only when it is attracted to that leg. The collision of a leg and the body, when the body is pulling the leg, is captured among the forces on the legs.

## 4.2 Legs

MagPod has two legs: the right leg and the left leg. Each leg is composed of 382 particles and 6 rack particles, as shown in Figure 6. We shall denote the right leg particles by a vector **rightleg** and the left leg particles by a vector **leftleg**. Both of these vectors are of the form  $\mathbf{P}^{382}$ . The leg particles have no velocity or acceleration initially, and they have a radius of 1 unit. The corresponding rack particles, **rightrack** and **lefttrack**, are initially given by

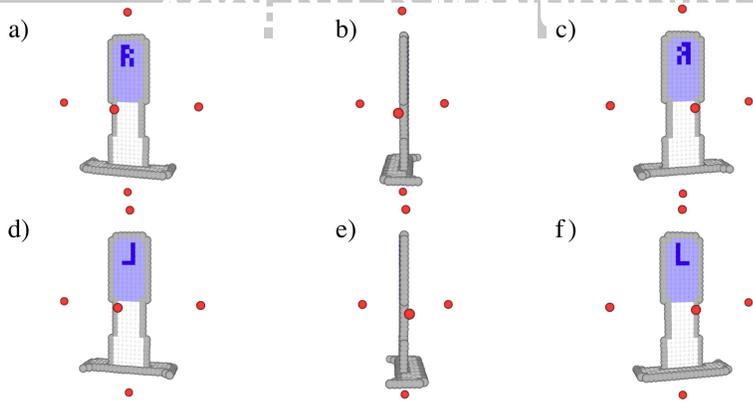


Fig. 6. MagPod’s legs with red rack particles and blue particles indicating the attraction region: a) right side of right leg, b) front view of right leg, c) left side of right leg, d) right side of left leg, e) front view of left leg, f) left side of left leg.

the functions  $\text{rack}(\text{rightleg})$  and  $\text{rack}(\text{leftleg})$ .

Similarly to the attraction regions of the body, we define matrices  $rreg$  and  $lreg$  to control the use of the attraction regions of the legs. Both  $rreg$  and  $lreg$  are of the form  $\mathbb{I}^{382}$ , and they give indexes to the leg particles with respect to the leg regions, the same way as the matrix  $breg$ , as explained earlier in Section 4.1.

The legs are subject to attraction and collision forces as well as attraction forces toward the body. In addition, the legs are subject to collision forces toward the sediment, binding forces toward the rack particles, damping forces, and a push-down force. A leg is attracted toward the body only if it is not pulling the body. A leg collides always with a body; however, if the leg is pulling the body, the collision magnitude is much smaller. This models the fact that the leg gets support from the sediment and does not move easily, even if the body is pushing it. Moreover, when any leg particle collides with the sediment, the motion of the whole leg is strongly dampened. This models the fact that the leg gets stuck in the sediment. The push-down force,  $(0, -50, 0)$ , is used to model the fact that the robot actively pushes down the legs to reach for the sediment.

We shall now express the forces on a leg formally. Let  $mag$  and  $breg$  denote the index matrices of the attraction regions for the leg and the body as before. Also, let  $b$  be an integer denoting the active attraction region of the body. Furthermore, let  $\mathbf{p}$  be a vector of the form  $\mathbf{P}^{382}$  capturing the leg particles,  $\mathbf{body}$  be a vector of the form  $\mathbf{P}^{2907}$  capturing the body particles,  $\mathbf{sediment}$  be a vector of the form  $\mathbf{P}^{30000}$  capturing the sediment particles, and  $\mathbf{r}$  be a vector of the form  $\mathbf{P}^6$  capturing the rack particles for the leg. Then, the net forces on a leg are captured as an operator  $\text{legForces}$  of the form  $(\mathbf{P}^6, \mathbb{I}^{8 \times 2907}, \mathbb{I}) \rightarrow$

$\text{legForces}(\mathbf{r}, \text{mag}, b)(\mathbf{p})$

$\triangleq \mathbf{p}$

- $\text{attract}(\mathbf{body}, [300 \langle b \in \{1, 4, 5, 8\} \wedge \text{mag}_i = b \text{reg}_{bj} \neq 0 \rangle]_{i=1..382, j=1..2907})$
  - $\text{collide}(\mathbf{body}, [100 \langle b \in \{1, 4, 5, 8\} \rangle]_{i=1..382, j=1..2907})$
  - $\text{collide}(\mathbf{body}, [5 \langle b \in \{2, 3, 6, 7\} \rangle]_{i=1..382, j=1..2907})$
  - $\text{collide}(\mathbf{sediment}, [50]_{i=1..382, j=1..30000})$
  - $\text{bind}(\mathbf{r}, [100]_{i=1..382, j=1..6})$
  - $\text{force}([(0, -50, 0)]_{i=1..m})$
  - $\text{dampen}(10 + 85 \langle \exists i, j : \|\text{col}(\mathbf{p}_i, 1, \mathbf{sediment}_j)\| > 0 \wedge b \in \{2, 3, 6, 7\} \rangle)$
- (28)

Note that the definition above starts with a forward composition “ $\mathbf{p} \bullet \dots$ ”. The reason for stating  $\mathbf{p}$  explicitly in the definition is that we need to express a condition for the damping force that depends on  $\mathbf{p}$ . The stated condition evaluates to a positive number, if any of the particles of  $\mathbf{p}$  collide with any of the particles of **sediment**.

The leg force, as defined above, is generic to both legs. We shall now specialize it to each leg. Let  $r$  be an integer denoting the active attraction region on the right side of MagPod’s body. Then, the net forces on MagPod’s right leg are captured as an operator  $\text{rightForces}$  of the form  $(\mathbb{I}) \rightarrow (\mathbf{P}^{382} \rightarrow \mathbf{P}^{382})$ , defined as:

$$\text{rightForces}(r) \triangleq \text{legForces}(\mathbf{rightrack}, r \text{reg}, r) \quad (29)$$

Let  $l$  be an integer denoting the active attraction region on the left side of MagPod’s body. Then, the net forces on MagPod’s left leg are captured as an operator  $\text{leftForces}$  of the form  $(\mathbb{I}) \rightarrow (\mathbf{P}^{382} \rightarrow \mathbf{P}^{382})$ , defined as:

$$\text{leftForces}(l) \triangleq \text{legForces}(\mathbf{leftrack}, l \text{reg}, l) \quad (30)$$

### 4.3 Sediment

We model the sediment as a flat plane composed of 300 by 100 particles, as shown in Figure 7. The radius of a sediment particle is 2 units. We shall denote the sediment particles by a vector **sediment** which is of the form  $\mathbf{P}^{30000}$ . As the sediment is soft material, there are no rack particles involved with it.

The sediment particles are subject to two kinds of forces: collision forces toward MagPod’s legs and damping forces. The magnitude of the collision forces



Fig. 7. The sediment is composed of 300 by 100 particles, each having a radius of 2 units. The particles form a flat plane, although the added colors give an impression of an uneven terrain.

is small, as the sediment gives in slowly under the weight of a leg. Furthermore, the motion of sediment particles is strongly dampened, to prevent the sediment particles from moving without an external force.

We shall now express the forces on the sediment particles formally. Let **rightleg** and **leftleg** be vectors of the form  $\mathbf{P}^{382}$  capturing the leg particles. Then, the net forces on the sediment particles are captured as an operator **sedimentForces** of the form  $\emptyset \rightarrow (\mathbf{P}^{30000} \rightarrow \mathbf{P}^{30000})$ , defined as:

$$\begin{aligned} \text{sedimentForces}() &\triangleq \text{collide}(\mathbf{rightleg}, [1]_{i=1..30000, j=1..382}) \\ &\bullet \text{collide}(\mathbf{leftleg}, [1]_{i=1..30000, j=1..382}) \\ &\bullet \text{dampen}(50) \end{aligned} \quad (31)$$

#### 4.4 Model of motion

For the overall model of motion we need to combine all the net forces of the components, the body, the legs, and the sediment. We define the overall net force as an operator working on 33,689 particles, as that is the total number of particles in all the components.

Let **body**, **rightleg**, **leftleg**, **bodyrack**, **rightrack**, **leftrack**, and **sediment** denote the particle vectors of the corresponding components. Also, let  $r$  and  $l$  be integers denoting the active attraction regions on the right and the left side of MagPod's body. Then, the overall net forces are captured as an operator

$$\begin{bmatrix} \mathbf{body} \\ \mathbf{rightleg} \\ \mathbf{leftleg} \\ \mathbf{bodyrack} \\ \mathbf{rightrack} \\ \mathbf{leftrack} \\ \mathbf{sediment} \end{bmatrix} \bullet \mathbf{netForces}(r, l) \triangleq \begin{bmatrix} \mathbf{body} \bullet \mathbf{bodyForces}(r, l) \\ \mathbf{rightleg} \bullet \mathbf{rightForces}(r) \\ \mathbf{leftleg} \bullet \mathbf{leftForces}(l) \\ \mathbf{bodyrack} \bullet \mathbf{rackForces}(\mathbf{body}) \\ \mathbf{rightrack} \bullet \mathbf{rackForces}(\mathbf{rightleg}) \\ \mathbf{leftrack} \bullet \mathbf{rackForces}(\mathbf{leftleg}) \\ \mathbf{sediment} \bullet \mathbf{sedimentForces}(l) \end{bmatrix} \quad (32)$$

Now, the overall model of motion with respect to the active attraction regions  $r$  and  $l$  is captured as an operator of the form  $\mathbf{motion} : (\mathbb{I}, \mathbb{I}) \rightarrow (\mathbf{P}^{33689} \rightarrow \mathbf{P}^{33689})$ , and we define it as:

$$\mathbf{motion}(r, l) \triangleq \mathbf{netForce}(r, l) \bullet \mathbf{integrate}(0.01) \quad (33)$$

## 5 Simulation analyzed

To simulate the modeled motion, we compute iteratively  $\mathbf{motion}(r, l)$ , where the values of  $(r, l)$  follow the predefined sequence described earlier:

$$(4, 6) \rightarrow (1, 7) \rightarrow (2, 8) \rightarrow (3, 5)$$

For each tuple in the above sequence, the operator  $\mathbf{motion}(r, l)$  is iterated 1200 times. Thus, the whole sequence, modeling a short walk where MagPod takes one step with both of its legs, requires the iteration of  $\mathbf{motion}(r, l)$  a total of 4800 times. The computation was implemented on top of an open-source particle system package called Atoms [39]. A non-optimized computation of 1200 iteration rounds run with a **1.83** GHz Intel CoreDuo processor in a single thread took 142 seconds. Such a computation, when run with a modern graphics processor, could be completed within a second for instance by using the approach reported by Kipfer et al. [34].

### 5.1 Emergent properties

The image series in Figure 8 shows the simulated motion of MagPod for the first 9600 iteration rounds. The simulation shows several interesting emergent properties. Two of the properties, however, are of special interest: MagPod's

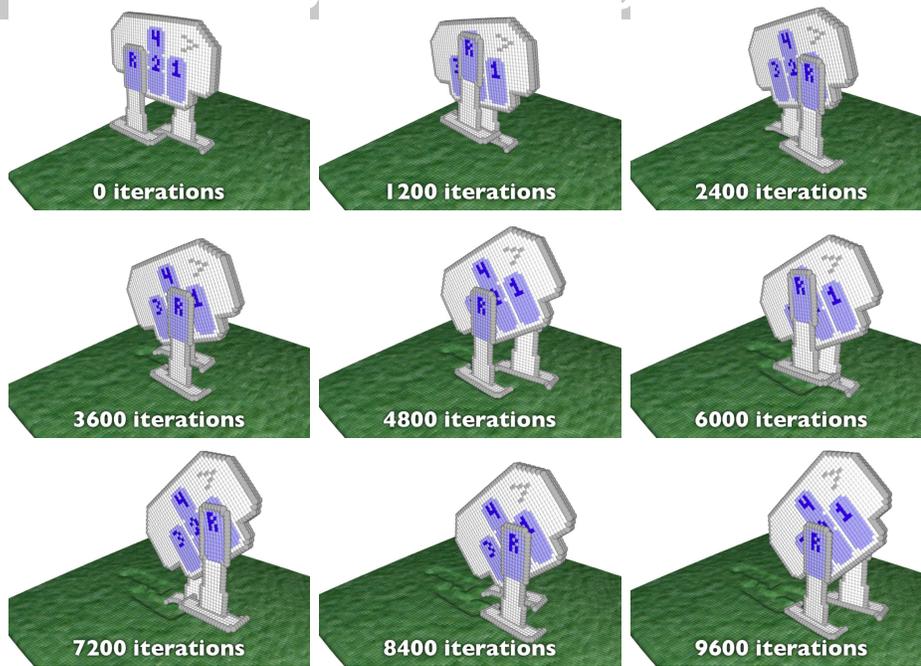


Fig. 8. Image series showing MagPod's walking motion on the sediment; the number of computed iterations is shown at the bottom of each image.

body tilts upright as it walks, and MagPod leaves a trail of footprints in the sediment.

As Figure 8 shows, MagPod's body tilts upright as it walks forward. This is a very intricate property typically associated with bipedalism, especially with kangaroos, dinosaurs, and some birds. As we have omitted water currents in the model and as the model is deterministic, this phenomenon is due to the style of walking. It is explained by the fact that the legs pull the back of the body forwards while the front of the body has mass and, therefore, resists the forward motion. If we were to include water currents in the model, this property could become either exaggerated or dampened depending on the shape and design of MagPod's body.

The effect of the walking motion to the whole body of MagPod is visible in the image series of Figure 9. It shows clearly how MagPod swings and sways when walking. This phenomenon is also emergent and typically associated with bipedalism. In fact, from the front view, MagPod's walk resembles that of a human; even the feet turn outwards when MagPod is walking.

The second emergent property of interest is the distinct trail of footprints that MagPod leaves behind in the sediment. The trail is shown from a low angle in Figure 10. As the figure shows, the distance between the footprints in the trail is somewhat equal; however, the shape of the imprint is neither



Fig. 9. Image series showing MagPod swings and sways while walking; the number of computed iterations is shown at the bottom of each image.

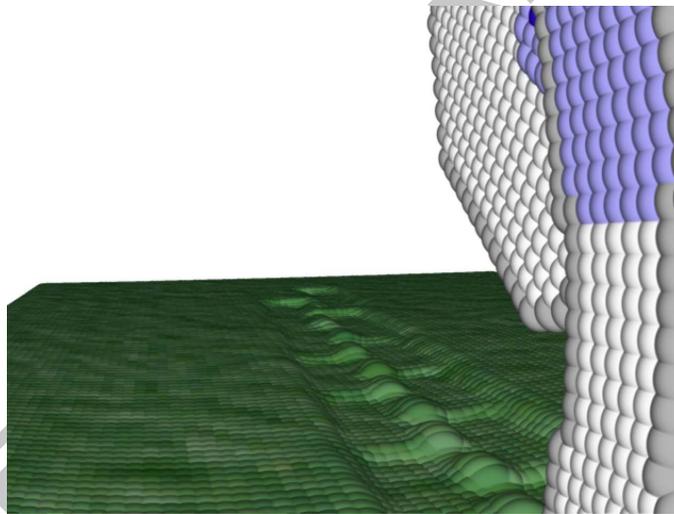


Fig. 10. MagPod leaves a distinct trail of footprints in the sediment. Note that the leg of MagPod appears to be bend because of perspective distortion.

flat nor uniform. This indicates the presence of diverse forces. A closer look at the step motion of a leg, as illustrated in Figure 11, also indicates that the presence of diverse forces that could cause stress on MagPod's structure. Note in particular, how the area of MagPod's leg around the attraction region "R" bends.



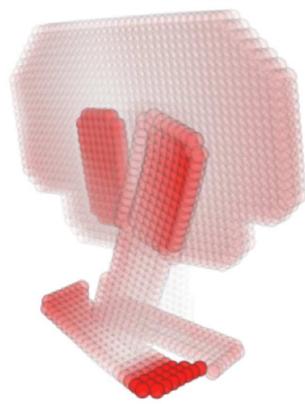
Fig. 11. Image series showing MagPod’s step motion on the sediment; the number of computed iterations is shown at the bottom of each image.

## 5.2 Preliminary stress visualization

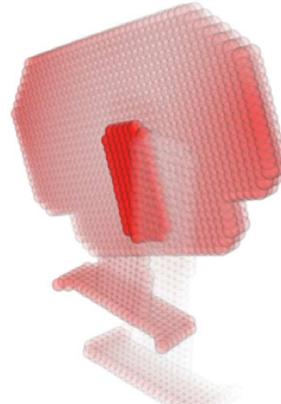
For the preliminary stress visualization, we use the fact that forces cause local deformations in MagPod’s parts, although the parts themselves behave as rigid objects. We interpret the magnitude and the duration of local deformations as stress. To visualize stress, we use false coloring. We paint a particle the redder and more opaque the further away it has drifted away from its “desired position”, that is, from its initial relative position. We paint a particle that has maintained its initial relative position grey and translucent. In this way, we obtain images that highlight the stress points.

Figure 12 shows four selected images from the simulation drawn using the false coloring. It should be noted that as stress is not addressed in the model, it is an emergent property of the model. In Figure 12, the image after 1,320 computed iterations displays clearly shearing stress due to torsion around and on the attraction regions. It also shows how the front particles of the right foot bend due to a collision with the sediment. The image after 2,600 computed iterations shows how a pulse force propagates through the whole body when the left leg is lifted from the sediment. The lifting of a leg causes the center of mass to shift, leading to the swinging and swaying that was captured in Figure 9. Out of all the images, the image after 4,720 computed iterations is the most interesting, as it shows bending of the outer side on the right foot. This phenomenon is mostly associated with foot biomechanics, especially with flat feet. Lastly, the image after 17,480 computed iterations shows clearly, how one of MagPod’s right foot toes breaks up in the sediment due to MagPod’s weight.

In summary, the simulated motion of MagPod confirms that a particle system



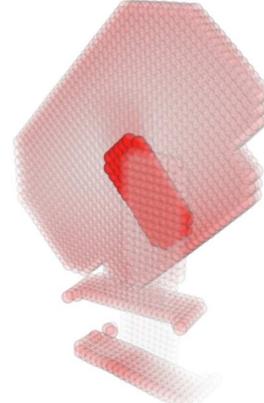
1320 iterations



2600 iterations



4720 iterations



17480 iterations

Fig. 12. A stress visualization on MagPod shows shearing stress due to torsion, generic shear stress, bending, and a broken toe. The number of computed iterations is shown at the bottom of each image.

can be used to simulate the interaction of both non-rigid and rigid objects. In particular, the motion analysis above confirms that use of rack particles is sufficient for modeling rigid object with particles and particle-to-particle forces. The advantage of using a particle based approach is that the computed motion dynamics can then be used for performing also a preliminary stress visualization, as discussed above. All in all, the discussion above indicates that particle systems are a valuable tool for simulating complex motion dynamics and for performing preliminary analysis on the simulated dynamics. Thus, a particle system based approach saves time, because it can be used to model systems with a modest modeling effort, to identify interesting dynamics and structures that can then be analyzed further with more rigorous approaches, as discussed earlier in related work section.

In this paper, we investigated a complementary approach to stress visualization in early stages of the development process, the use of a particle system. In order to model also rigid but deformable objects with particles, we proposed here use of rack particles. Rack particles interact only with the particles of their host object. The purpose of the rack particles was to isolate local deformations in the host object by preventing wave propagation.

We illustrated the use of rack particles in preliminary stress visualization by formalizing, modeling, and simulating a bipedal robot walking on a soft sediment. The simulated dynamics exhibited non-trivial interaction dynamics that emerged from the model. Two of the emergent properties were of particular interest; firstly, MagPod's body tilted upright as it walked forward, and secondly, the outer sides of MagPod's feet showed clear bending when MagPod was walking. Both of these intricate properties are typically associated with bipedalism, especially with kangaroos, dinosaurs, and some birds; thus, confirming to some degree that the simulated dynamics is that of a walking robot.

Technically, the preliminary stress visualization was extracted from the simulation simply by using false coloring. We painted a particle the redder and more opaque the further away it had drifted from its initial relative position. In this way, we obtained images that highlight the stress points. The preliminary analysis indicated sections of the body and legs that showed shear stress, torsion, and local bending.

Overall, the analysis confirmed that particle system simulations are a complementary tool for performing a preliminary stress point analysis of a body, to identify the parts of the structures that should be subjected to more rigorous analysis by mathematical methods. In particular, the simulation results confirmed a particle system can be used to simulate the interaction of both non-rigid and rigid objects by using rack particles. Thus, particle systems provide a uniform platform for modeling and analyzing the dynamics of interacting bodies with varying materialistic properties using particles only.

We are encouraged by the obtained results; they raise many interesting topics for future research. Clearly, an important topic for future research to study modelling of more complex scenarios in robotics using particle systems. Also incorporation of bigger volumes into the models is a central topic for future research; in particular, an attractive topic is the inclusion of water into the model of MagPod, to study water currents and their interaction to MagPod's shape and design. As such models tend to comprise of millions of particles, there is also a need to develop the computational tools. We believe that, for such tools, the use of graphics processor for the computation [34] is a necessity.

We wish to thank the anonymous reviewers for reading and insightful comments.

## References

- [1] C. Foias, D. D. Holm, E. S. Titi, The navierstokes-alpha model of fluid turbulence, *Physica D: Nonlinear Phenomena* 152-153 (2001) 505–519.
- [2] J. Jansson, J. S. M. Vergeest, A discrete mechanics model for deformable bodies, *Computer-Aided Design* 34 (2002) 913–928.
- [3] H. Jin, J. Glimm, Verification and validation for turbulent mixing, *Nonlinear Analysis: Theory, Methods & Applications* 69 (2008) 874–879.
- [4] V. S. Kumar, B. R. Kumar, P. Das, Web-spline based mesh-free finite element analysis for the approximation of the stationary navier-stokes problem, *Nonlinear Analysis: Theory, Methods & Applications* 68 (2008) 3266–3282.
- [5] A. Ibrahimbegović, D. Markovič, Strong coupling methods in multi-phase and multi-scale modeling of inelastic behavior of heterogeneous structures, *Computer Methods in Applied Mechanics and Engineering* 192 (2003) 3089–3107.
- [6] F. T. Fisher, L. C. Brinson, Viscoelastic interphases in polymer-matrix composites: theoretical models and finite-element analysis, *Composites Science and Technology* 61 (2001) 731–748.
- [7] C. Yang, D. Li, Analysis of electrokinetic effects on the liquid flow in rectangular microchannels, *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 143 (1998) 339–353.
- [8] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, Elastically deformable models, *ACM SIGGRAPH Computer Graphics* 21 (1987) 205–214.
- [9] I. Demirdžić, S. Muzaferija, Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, *Computer Methods in Applied Mechanics and Engineering* 125 (1995) 235–255.
- [10] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behavior, in: *Proceedings Graphics Interface, 1995*, pp. 147–154.
- [11] X. Tu, D. Terzopoulos, Artificial fishes: Physics, locomotion, perception, behaviour, in: *Proceedings of 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, 1994, pp. 43–50.

- [12] K.-J. Bathe, H. Zhang, S. Ji, Finite element analysis of fluid flows fully coupled with structural interactions, *Computers and Structures* 72 (1999) 1–16.
- [13] J. Kim, J.-C. Yoon, B.-S. Kang, Finite element analysis and modeling of structure with bolted joints, *Applied Mathematical Modelling* 31 (2007) 895–911.
- [14] H. Saadawi, G. Wainer, Modeling physical systems using finite element cell-devs, *Simulation Modelling Practice and Theory* 15 (2007) 1268–1291.
- [15] M. Rönkkö, An artificial ecosystem: Emergent dynamics and life-like properties, *Artificial Life* 13 (2) (2007) 159–187.
- [16] J. Holopainen, M. Rönkkö, Plausible motion simulation: Inchworm vs. roller, in: *Proceedings of the Second International Conference on Computer Graphics Theory and Applications*, INSTICC Press, 2007, pp. 71–74.
- [17] M. Rönkkö, G. Wong, Modeling the *c. elegans* nematode and its environment using a particle system, *Journal of Theoretical Biology*. In Press.
- [18] M. Rönkkö, Previsualization in robotics: An atomic approach, in: *Proceedings of the First International conference on Informatics in Control, Automation and Robotics*, INSTICC Press, 2004, pp. 110–117.
- [19] M. Rönkkö, J. Choudhury, Particle system: Motion analysis, in: *Proceedings of the 16th IASTED International Conference on Applied Simulation and Modeling*, Acta Press, 2007, pp. 50–55.
- [20] J. Holopainen, M. Rönkkö, Embedding control into a particle system: A case study, in: *Proceedings of the Ninth IASTED International Conference on Intelligent Systems and Control*, Acta Press, Anaheim, 2006, pp. 92–97.
- [21] J. Holopainen, M. Rönkkö, Roller: the effect of the environment to the emergence of behavior, in: *Proceedings of the 2007 International Conference on Intelligent Agent Technology*, IEEE, 2007, pp. 35–38.
- [22] M. Rönkkö, Hybrid systems: Modeling and analysis using emergent dynamics, *Nonlinear analysis: hybrid systems* 1 (2007) 560–576.
- [23] M. Rönkkö, M. Waldén, R.-J. Back, Beyond particle systems: Operator networks, *Simulation Modelling Practice and Theory*. (16) (2008) 834–847.
- [24] J. Holopainen, M. Rönkkö, P. Toivanen, Roller: Adding target-oriented behavior to a wheeled mobile agent, in: *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering*, IEEE, 2009, pp. 92–96.
- [25] M. M. Martinez, Issues for aquatic pedestrian locomotion, *American Zoologist* 36 (6) (1996) 619–627.
- [26] M. M. Martinez, R. J. Full, M. A. Koehl, Underwater punting by an intertidal crab: a novel gait revealed by the kinematics of pedestrian locomotion in air versus water, *Journal of Experimental Biology* 201 (18) (1998) 2609–2623.

- ACCEPTED MANUSCRIPT
- [27] J. Yuh, Design and control of autonomous underwater robots: A survey, *Autonomous Robots* 8 (2004) 7–24.
- [28] W. T. Reeves, Particle systems - a technique for modeling a class of fuzzy objects, *Computer Graphics* 17 (3) (1983) 359–376.
- [29] B. Eberhardt, A. Weber, W. Strasser, A fast, flexible, particle-system model for cloth draping, *Computer Graphics and Applications* 16 (5) (1996) 52–59.
- [30] N. Foster, R. Fedkiw, Practical animation of liquids, in: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press, New York, 2001, pp. 23–30.
- [31] S. T. Thornton, J. B. Marion, *Classical Dynamics of Particles and Systems*, 5th Edition, Brooks/Cole – Thomson Learning, Belmont, 2004.
- [32] D. Tonnesen, Dynamically coupled particle systems for geometric modeling, reconstruction, and animation, Ph.D. thesis, University of Toronto, Canada (1998).
- [33] A. Kubík, Toward a formalization of emergence, *Artificial Life* 9 (2003) 41–65.
- [34] P. Kipfer, M. Segal, R. Westermann, Overflow: a gpu-based particle engine, in: *HWWS '04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, ACM, 2004, pp. 115–122.
- [35] R. J. R. Back, J. von Wright, *Refinement Calculus: A Systematic Introduction*, Graduate Texts in Computer Science, Springer-Verlag, New York, 1998.
- [36] D. G. Zill, M. R. Cullen, *Differential Equations with Boundary-Value Problems*, 4th Edition, Brooks/Cole Publishing Company: Pacific Grove, 1997.
- [37] H. D. Young, R. A. Freedman, *University Physics with Modern Physics*, 12th Edition, Addison-Wesley, San Francisco, 2007.
- [38] M. Rönkkö, Previsualization in robotics: An atomic approach, Tech. Rep. A/2003/4, Department of Computer Science, University of Kuopio (2003).
- [39] M. Rönkkö, Atoms: a particle system framework, Department of Computer Science, University of Kuopio, <http://www.cs.uku.fi/~mronkko/atoms.html> (2009).