Constraint-based Simulation of Passive Suction Cups

Antonin Bernardin, Eulalie Coevoet, Paul Kry, Sheldon Andrews, Christian Duriez, Maud Marchal

Constraint-based Simulation of Passive Suction Cups

ANTONIN BERNARDIN, Univ. Rennes, INSA, IRISA, Inria, France
EULALIE COEVOET, Univ. Rennes, INSA, IRISA, Inria, France
PAUL KRY, McGill University, Canada
SHELDON ANDREWS, École de technologie supérieure, Canada
CHRISTIAN DURIEZ, Inria, Univ. Lille, CNRS, Centrale Lille, France
MAUD MARCHAL, Univ. Rennes, INSA, IRISA, Inria, France

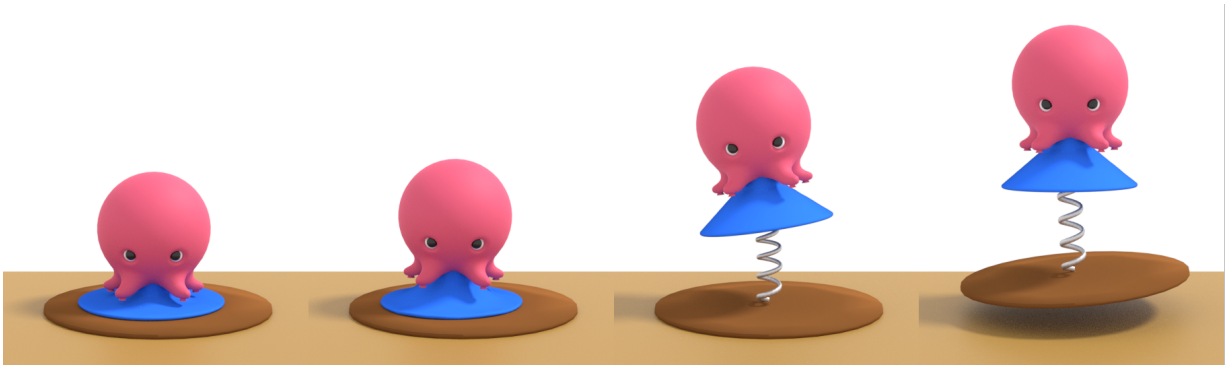

Fig. 1. The Monster pop-up toy sticks to its base until the spring forces release it due to air leakage, making the whole structure to jump.

In this paper, we propose a physics-based model of suction phenomenon to achieve simulation of deformable objects like suction cups. Our model uses a constraint-based formulation to simulate the variations of pressure inside suction cups. The respective internal pressures are represented as pressure constraints which are coupled with anti-interpenetration and friction constraints. Furthermore, our method is able to detect multiple air cavities using information from collision detection. We solve the pressure constraints based on the ideal gas law while considering several cavity states. We test our model with a number of scenarios reflecting a variety of uses, for instance, a spring loaded jumping toy, a manipulator performing a pick and place task, and an octopus tentacle grasping a soda can. We also evaluate the ability of our model to reproduce the physics of suction cups of varying shapes, lifting objects of different masses, and sliding on a slippery surface. The results show promise for various applications such as the simulation in soft robotics and computer animation.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; *Collision detection.*

Additional Key Words and Phrases: constraints, deformation, suction, friction

Authors' addresses: Antonin Bernardin, antonin.bernardin@irisa.fr, Univ. Rennes, INSA, IRISA, Inria, , Rennes, France; Eulalie Coevoet, eulalie.coevoet@gmail.com, Univ. Rennes, INSA, IRISA, Inria, , Rennes, France; Paul Kry, kry@cs.mcgill.ca, McGill University, , Montreal, Canada; Sheldon Andrews, sheldon.andrews@etsmtl.ca, École de technologie supérieure, , Montreal, Canada; Christian Duriez, christian.duriez@inria.fr, Inria, Univ. Lille, CNRS, Centrale Lille, , Lille, France; Maud Marchal, maud.marchal@irisa.fr, Univ. Rennes, INSA, IRISA, Inria, , Rennes, France.

2022. XXXX-XXXX/2022/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Antonin Bernardin, Eulalie Coevoet, Paul Kry, Sheldon Andrews, Christian Duriez, and Maud Marchal. 2022. Constraint-based Simulation of Passive Suction Cups. 1, 1 (November 2022), 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Suction phenomena are important for many different physical interactions and tasks. For instance, we use suction cups in our daily lives to temporarily attach objects to surfaces, we have toys based on suction phenomena (see Figure 1), and robots use suction to grasp and manipulate objects. Suction is also a distinguishing feature of many animals, such as squids that use suckers to capture prey, or snails that rely on suction to attach to and move across surfaces. Simulation of suction is therefore of interest for numerous applications across several domains.

In computer graphics, we would like to simulate creatures with suckers in order to plausibly animate complex interactions with their environment. Whereas in robotics, being able to predict the real-world behavior of robots with suction cups is important for improving their control and their design, e.g., biologically inspired robots. Computational performance is a primary concern for both robotics and computer graphics applications, since it is often required to perform simulations at interactive or real-time frame rates. Fast models and methods are not only applicable to interactive video games and robot control, but also facilitate learning algorithms that require many simulations over a large variety of configurations. The benefits of an efficient approach for simulating suction is therefore a principal motivation for our work.

Simulating suction phenomena is challenging. Suction occurs when air is displaced or removed from an airtight cavity between two deformable surfaces. This creates a pressure difference with a low pressure volume on the inside of the cavity that causes the two objects to stick together due to higher pressure on the outside. The phenomenon therefore depends not only on the behavior of the fluid (air) inside the cavity, but also the elastic behavior of deformable bodies and the frictional contact between them. A coupled simulation involving the fluid and elastic bodies might provide a straightforward solution, but such an approach is unnecessarily expensive for capturing the suction phenomenon.

Instead, we avoid the complex modeling of air flow within cavities by a novel approach whereby suction is modeled as a constraint acting on the degrees of freedom (DOF) of bodies simulated using the finite element method (FEM). We simulate the suction effects that occur during contact between elastic bodies by tracking each cavity that forms between objects and applying pressure on the cavity surfaces according to the *ideal gas law*:

$$P\,V = n\,\mathrm{R}\,\mathrm{T}$$

where R is the ideal gas constant, and T is the temperature of the gas. In our dynamics simulation, we make the simplifying approximation that the system is iso-thermal, i.e., the surrounding environment acts as a large heat sink which holds the system at constant temperature. Thus, the pressure P and volume V are related based on the quantity of air n trapped in the cavity. Our simulations also permit the air quantity to decrease (i.e., be squeezed out) when the pressure exceeds a maximum limit, for instance, the ambient pressure. In this case, the pressure constraint functions as a unilateral constraint with complementarity conditions: the quantity of air can decrease, the pressure can be lower than the maximum pressure, but not both at the same time. We carefully treat contact and friction forces at the borders between cavities and contact patches, which leads to results that correctly account for geometric variation of object surfaces and their material properties. Finally, we note that modeling pressure forces between stiff elastic objects can be numerically challenging. We therefore use a linearized implicit integration that avoids these issues and our simulations proceed at interactive frame rates with only minor changes to an existing FEM simulation pipeline. Our principal contributions are the following:

- An implicit pressure constraint formulation that is based on the ideal gas law;
- The coupling of pressure and airflow being formulated as a complementarity constraint and allowing a global solution with contact and friction;
- A method to detect air cavities between bodies in contact.

We demonstrate the performances of our method with numerous simulated scenarios, and also provide a comparison of simulated suction forces with those produced by a real suction cup. We discuss the limitations of our approach at the end of the paper.

2 RELATED WORK

The simulation of suction phenomena is built on research on the simulation of elastic solids and frictional contact, and has related work in the areas of robotics, computer animation, and computational design for fabrication.

Modeling Suction Phenomena. Early work on the design of climbing robots includes physics-based modeling of suction cups to determine minimum pressure for safe wall attachment [Bahr et al. 1996]. More recent modeling work on window-cleaning robots makes the assumption of no-slip contact to simplify the analysis of suction cup behaviour [Liu et al. 2006].

Ge et al. [2015] present a quantitative study of the attachment and detachment of a passive suction cup and its basic mechanics. Based on their results, they model the pushing and pulling forces acting on the cup. Mahler et al. [2018] also propose a model to quantify the ability of a passive suction cup to stay attached to a target surface when lifting the object. Bernardin et al. [2019] propose a method to simulate active suction cups for robotics applications. In our work, we use similar geometric models, but our approach for the detection and monitoring of cavities is much more robust because we compute geometric intersections rather than the classification of vertices and flood-fill of surface regions. There is likewise the fundamental difference in the treatment of pressure within our physics simulations. The suction cups of Bernardin et al. [2019] are said to be *active* because the pressure in the cavity is a controlled quantity, as one would want in simulating a soft robot with the cavity of each suction cup connected to a vacuum pump and regulator. In contrast, the pressure in our simulations of *passive* suction cups is determined by a constraint based on the ideal gas law. To the best of our knowledge, there is no existing physics-based model of suction cup with a coupling between its deformation and the inner pressure. Such models require simulations of both deformable models and contact.

Simulating Deformable Objects. The simulation of elastic bodies has been well studied in computer graphics. Nealen et al. [2006] provide a survey of early work on the topic, and recent courses by Sifakis and Barbic [2012] and Bargteil et al. [2020] present popular techniques for soft-body simulation used in computer graphics. Performance is sought both in terms of accuracy and computation time. Efficient simulations may be achieved through the use of simplified models [Müller and Gross 2004; Nesme et al. 2005], whereas other work has focused on achieving performance by GPU implementations [Fratarcangeli et al. 2018]. Today, precise modeling of material behavior such as hyperelasticity [Liu et al. 2017] and elastoplasticity [Chentanez et al. 2016] can be obtained in real-time. However, the parameterization of complex constitutive law is often tedious and even futile if the boundary conditions cannot be defined correctly in the simulations [Metaxas 2012]. In our work, FEM using co-rotated linear elasticity Müller et al. [2002] is sufficient to simulate the deformations necessary to produce suction phenomena. While not a focus of our work, our method is compatible with more computationally expensive material models, and likewise, much faster simulation methods for co-rotated linear materials can be used [Kugelstadt et al. 2018].

Coupling Elastics with Air Pressure. In computer graphics, mixed simulations of elastic solids with air pressure have been used for computational design for fabrication. Skouras et al. [2012] design

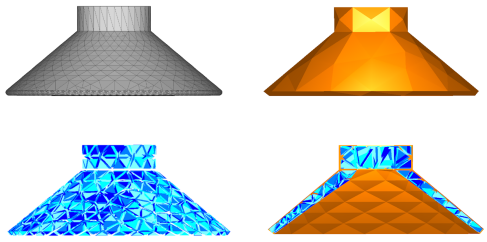

Fig. 2. Topological representations of a *suction cup*. Top left: triangle surface mesh used for visualisation. Top right: triangle surface mesh used for collision. Bottom left: tetrahedron volume mesh used for FEM. Bottom right: clipped view of the collision and FEM meshes.

balloon rest shapes that inflate and deform to take on a desired shape. Likewise, Skouras et al. [2014] examine the related problem of designing of inflatable structures from a set of flat panels attached at seams. More recently, Ma et al. [2017] optimize material distributions in heterogeneous printed 3D structures such that they can take on a set of previously defined target shapes when different cavities are inflated. Also related to the topic of simulating pressure forces in volumes is the application of volume constraints, which has been useful for a variety of problems, such as incompressible finite elements [Irving et al. 2007], and character animation [Rohmer et al. 2008].

Modeling Contact. While the deformation of suctions cups is obtained through an elastic soft body simulation, frictional contact is equally important for simulating suction phenomena. However, contact handling between deformable objects is challenging, especially if one wishes a precise modeling of the boundary conditions. In existing frameworks, the modeling of boundary conditions and the coupling with other phenomena is often achieved by penalty methods or constraint-based approaches. Explicit penalty methods [Marhefka and Orin 1996; Yamane and Nakamura 2006] require small time steps for stability, whereas implicit methods for penalty based contact have been successfully used in computer graphics applications [Xu et al. 2014]. Precise contact modeling may also be achieved by a constraint-based formulation of frictional contact, and typical approaches model contact as either a quadratic programming (QP) problem [Kaufman et al. 2008; Moreau 1966] or a complementarity problem (CP) [Erleben 2017; Stewart 2000]. Our simulations use the latter approach. The pressure constraints are similarly formulated as a complementarity problem, and a block Gauss-Seidel based solver [Duriez et al. 2005] is used to solve for both pressure and contact constraints. Contact simulation between soft bodies can generate a large number of collision points, which is often a bottleneck in real-time simulation as it leads to numerous contact constraints. However, techniques do exist to reduce the number of contact constraint, for instance by computing aggregate constraints [Talvas et al. 2015].

3 SIMULATING ELASTIC SOLIDS AND SUCTION

In this section we present our approach to model and simulate suction phenomena between deformable bodies in contact.

Algorithm 1 General pipeline of our suction simulation.

procedure STEP
  solve $\mathbf{A}\Delta\dot{\mathbf{x}} = \mathbf{b}$
  // Update configuration
  $\mathbf{x}_0 \leftarrow \mathbf{x} + h\,(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})$
  $\mathbf{H}_c \leftarrow \text{CollisionDetection}(O_1, O_2)$
  $\mathbf{H}_p, n, V \leftarrow \text{CavityDetection}(O_1, O_2)$
  $\boldsymbol{\lambda}, P \leftarrow \text{SolveContactPressure}(\mathbf{H}_c, \mathbf{H}_p, n, V)$
  update $\Delta\dot{\mathbf{x}} \leftarrow \Delta\dot{\mathbf{x}} + h\mathbf{A}^{-1}\mathbf{H}_c^T\boldsymbol{\lambda} + h\mathbf{A}^{-1}\mathbf{H}_p^T P$
  $\dot{\mathbf{x}} \leftarrow \dot{\mathbf{x}} + \Delta\dot{\mathbf{x}}$
  $\mathbf{x} \leftarrow \mathbf{x} + h\dot{\mathbf{x}}$
end procedure

3.1 General Overview

The deformable objects are represented by FEM meshes composed of tetrahedra or hexahedra. Additional triangle meshes representing the surface of each object are used for visualization and a collision detection step. We use the mapping mechanism described by Faure et al. [2012] to transfer positions, velocities and forces between the mechanical, visualization, and collision models. These three topological representations are illustrated in Figure 2.

A collision detection step determines the vertices on each object where contact occurs. This detection is based on minimal distances computation using an implementation of the algorithm described by Johnson and Willemsen [2003], adapted to deformable meshes.

Contacts and pressure are modeled as constraints using Lagrange multipliers and two constraint matrices that respectively encode friction and non-interpenetration for the first one, and pressure constraints for the second one. We use a reduced system of equations involving only the constraint variables which defines a complementarity problem. This problem accounts for the bounds of the non-interpenetration, friction, and cavity pressure constraints. As the gas law is non-linear, we favored the use of a PGS solver (see Sections 3.3 and 3.4) that allow iteratively solving small blocks of non-linear constraints while taking into account the coupling between them.

Our simulation pipeline proceeds by first simulating the unconstrained motion of the two deformable objects O_1 and O_2, which can result in a penetration configuration. In a second step, we apply the correction forces resulting from the constraint solve to the nodal DOFs which finally results in a penetration-free configuration of the FEM meshes, and also accounts for the depressurization and suction of the cavities. This simulation pipeline is summarized in Algorithm 1 and we present each step in detail below.

3.2 Equations of Motion

We suppose that the state of the mechanical system is given by the position $\mathbf{x}$ and the velocity $\dot{\mathbf{x}}$ of a total of N nodes across the finite element models. Following the semi-implicit Backward Euler stepping of Baraff and Witkin [1998], the unconstrained elastic system involves computing updates to the nodal velocities $\Delta\dot{\mathbf{x}}$ by

solving

$$\underbrace{\left(\mathbf{M} - h\mathbf{B} - h^2\mathbf{K}\right)}_{\mathbf{A}} \Delta\dot{\mathbf{x}} = \underbrace{h\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) + h^2\mathbf{K}\dot{\mathbf{x}} + h\mathbf{f}_{\text{ext}}}_{\mathbf{b}} . \tag{1}$$

Here, h is the time step, $\mathbf{M}$, $\mathbf{B}$, and $\mathbf{K}$ are respectively the mass, damping, and stiffness matrices of the finite element models, $\dot{\mathbf{x}}$ are the nodal velocities, and elastic forces are given by $\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}})$, and external forces, such as gravity, given by $\mathbf{f}_{\text{ext}}$.

Since contacts are an essential component of suction simulation, we constrain the elastic bodies by introducing non-interpenetration and friction constraints. The matrix $\mathbf{H}_c \in \mathbb{R}^{3k\times 3\mathrm{N}}$, encodes the frictional contact constraint gradient at k discrete contact locations that form a seal around each cavity. We also observe that this constraint matrix provides important information about the separation and tangent slip velocity at contacts $\mathbf{H}_c\dot{\mathbf{x}}$. Further details on constructing the contact constraint matrix can be found in Section 3.3.

Suction is also modeled as a constraint in our simulations. Letting $\mathbf{H}_p \in \mathbb{R}^{1\times 3\mathrm{N}}$ be the volume gradient of a sealed cavity, we observe that this row vector only has non-zero components at nodes that participate in defining the volume of the cavity. This constraint matrix also provides the rate of volume change in the cavity based on nodal velocities as

$$\dot{V} = \mathbf{H}_p\dot{\mathbf{x}} . \tag{2}$$

With the system in Equation 1 being solved to obtain $\Delta\dot{\mathbf{x}}$, that is, at the velocity level, we have constraint forces acting according to Lagrange multipliers $\boldsymbol{\lambda}$ and P, which respectively enforce the contact and suctions constraints. These forces act in constraint directions determined by $\mathbf{H}_c^T$ and $\mathbf{H}_p^T$, which are visualized in Figure 3, and a time step factor of h is used to make these constraint forces act as impulses. An update to the unconstrained motion computed in Equation 1 can then be obtained by

$$\Delta\dot{\mathbf{x}} \leftarrow \Delta\dot{\mathbf{x}} + h\mathbf{A}^{-1}\mathbf{H}_c^T\boldsymbol{\lambda} + h\mathbf{A}^{-1}\mathbf{H}_p^T P . \tag{3}$$

We note that the terms $\mathbf{A}^{-1}\mathbf{H}_c^T$ and $\mathbf{A}^{-1}\mathbf{H}_p^T$ can be computed efficiently by using a sparse linear solve.

3.3 Solving Contact Constraints

For each contact point, a block is added to the matrix $\mathbf{H}_c$ that encodes the contact normal and friction directions. We use $\boldsymbol{\lambda}_\mathrm{n} \in \mathbb{R}$ to denote the non-interpenetration constraint force of a contact, and $\boldsymbol{\lambda}_\mathrm{t} \in \mathbb{R}^2$ for the frictional forces. The inner loop of the PGS solver in Algorithm 2 computes contact forces block-wise by solving for the tuple of constraint forces that satisfy Signorini and Coulomb laws. This involves solving three non-linear equations with complementarity conditions,

$$0 \le \boldsymbol{\lambda}_\mathrm{n} \perp \mathbf{H}_{c,\mathrm{n}}(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}}) \ge 0 \tag{4}$$

$$\left(\|\boldsymbol{\lambda}_\mathrm{t}\|_2 < \mu\boldsymbol{\lambda}_\mathrm{n}\right) \oplus \left(\boldsymbol{\lambda}_\mathrm{t} = -\mu\boldsymbol{\lambda}_\mathrm{n}\frac{\mathbf{H}_{c,\mathrm{t}}(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})}{\|\mathbf{H}_{c,\mathrm{t}}(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})\|_2}\right) , \tag{5}$$

where $\oplus$ denotes exclusive disjunction, and μ is the friction coefficient. In Equation 4, the term $\mathbf{H}_{c,\mathrm{n}}(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})$ is the velocity in the normal direction at the contact at the end of the time step. Similarly, in Equation 5, $\mathbf{H}_{c,\mathrm{t}}(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})$ is the 2D velocity in the tangent plane at the contact, and when $\|\boldsymbol{\lambda}_\mathrm{t}\|_2 < \mu\boldsymbol{\lambda}_\mathrm{n}$, this tangential velocity is constrained to be zero. The Schur complement is used to solve for contact constraint forces, accounting for the unconstrained motion update $\Delta\dot{\mathbf{x}}$ and the current estimate of the suction pressure P:

$$h\mathbf{H}_c\mathbf{A}^{-1}\mathbf{H}_c^T\boldsymbol{\lambda} = \mathbf{H}_c\left(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}} - h\mathbf{A}^{-1}\mathbf{H}_p^T P\right) . \tag{6}$$

When multiplied by the constraint matrix, we obtain the nominal constraint gap function velocity $\dot{\mathbf{g}}_{cb} = \mathbf{H}_c(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})$ due to the known forcing terms and the current velocity. Likewise, we solve $h\mathbf{A}^{-1}\mathbf{H}_p^T$ in advance, allowing us to write the compliance, i.e., rate at which the constraint velocity changes due to pressure, as $\mathbf{W}_{cp} = h\mathbf{H}_c\mathbf{A}^{-1}\mathbf{H}_p^T$. Thus,

$$h\mathbf{H}_c\mathbf{A}^{-1}\mathbf{H}_c^T\boldsymbol{\lambda} = -\dot{\mathbf{g}}_{cb} - \mathbf{W}_{cp}P . \tag{7}$$

Note that $\mathbf{H}_p$ and $\mathbf{H}_c$ are fixed at the beginning of the time step. Thus if the vertices that define the inner boundary of the cavity were to change during a solve, it will only be handled at the beginning of the next step.

Algorithm 2 Solver algorithm for pressure and contact forces.

```
procedure SolveContactPressure(H_c, H_p, n, V)
    Δn = 0
    for i < max iterations do
        // Solve for contacts
        for each contact
            solve and project λ_n, λ_t   // Equations 4-7
        end for
        // Solve for pressure
        P = (−b + √d)/2a                 // Equations 11 and 12
        if P > P_max then
            P = P_max
            Δn = (a P_max^2 + b P_max)/(RT) − n   // Equation 13
        end if
        i = i + 1
    end for
return λ, P, Δn
end procedure
```

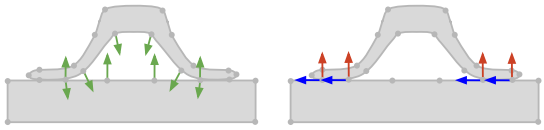

Fig. 3. When a cavity exists due to collision, three types of rows exist in the constraint matrix: non-penetration constraint directions (red), friction constraint directions (blue), and the uniform negative pressure distribution in the cavity (green).

3.4 Solving Pressure Constraints

As mentioned above, the PGS solve for frictional contact follows the well proven approach [Jourdan et al. 1998] of solving in sequence blocks of 3 equations (contact and friction) for each contact. The corresponding Lagrange multipliers $\boldsymbol{\lambda}_\mathrm{n}$ and $\boldsymbol{\lambda}_\mathrm{t}$ are updated in turn and projected to bounds as necessary. The pressure constraint, however,

is non-linear and requires closer inspection to ensure the correct solution.

The initial quantity of air in a cavity is set to n, and at each time step we compute the pressure P and a change in air quantity $\Delta n \leq 0$. If the gas law constraint can be satisfied, the cavity remains airtight and the quantity of trapped air remains constant, i.e., $\Delta n = 0$. However, if the gas law cannot be satisfied without increasing the pressure above the maximum $P_{\max}$, then we allow an air quantity change $\Delta n < 0$ during the time step (see Figure 4).

The pressure computation must account for how the volume will change over the next time step due to contact forces, the change in the amount of trapped air, and the pressure itself. Thus we solve for P in

$$P\,(V + h\dot{V}) = (n + \Delta n)\mathrm{RT}\,, \tag{8}$$

and using Equation 2 with the velocity at the next time step we obtain

$$P\left(V + h\mathbf{H}_p(\dot{\mathbf{x}} + \Delta\dot{\mathbf{x}})\right) = (n + \Delta n)\mathrm{RT}\,. \tag{9}$$

Recall that before starting the contact PGS solve, we compute solutions for the velocity changes due to $\mathbf{b}$ and the pressure constraint direction $\mathbf{H}_p^T$, to multiply by $\mathbf{H}_c$ and thus prepare the contact constraint velocities $\dot{\mathbf{g}}_{cb}$ and $\mathbf{W}_{cp}$ used in Equation 7. Here, we similarly prepare in advance the *volume velocity* due to forcing terms $\mathbf{b}$ and the current FEM velocity as $\dot{\mathbf{g}}_{pb} = \mathbf{H}_p(\mathbf{A}^{-1}\mathbf{b} + \dot{\mathbf{x}})$, and the compliance, i.e., volume velocity produced by pressure, $\mathbf{W}_{pp} = \mathbf{H}_p h\mathbf{A}^{-1}\mathbf{H}_p^T$ (note that $\mathbf{W}_{pp}$ is a scalar if there is only one cavity). Furthermore, pre-computing (at the start of time step) compliance $\mathbf{W}_{pc} = (\mathbf{H}_p h\mathbf{A}^{-1})\mathbf{H}_c^T$, Equation 9 can be expanded as

$$P\left(V + h\dot{\mathbf{g}}_{pb} + h\mathbf{W}_{pc}\boldsymbol{\lambda} + h\mathbf{W}_{pp}P\right) = (n + \Delta n)\mathrm{RT}\,. \tag{10}$$

Collecting terms on the left hand side yields

$$\underbrace{\left(h\mathbf{W}_{pp}\right)}_{\mathrm{a}>0}P^2 + \underbrace{\left(V + h\dot{\mathbf{g}}_{pb} + h\mathbf{W}_{pc}\boldsymbol{\lambda}\right)}_{\mathrm{b}>0}P + \underbrace{\left(-\,(n+\Delta n)\,\mathrm{RT}\right)}_{\mathrm{c}<0} = 0. \tag{11}$$

This quadratic equation in P is easy to solve for one cavity (and cavity by cavity with Gauss-Seidel in case of multiple cavities), but consider briefly the impossibility of complex roots. First, note that $\mathbf{W}_{pp}$, and quadratic term coefficient a, are positive, provided that $\mathbf{A}$ and its inverse are symmetric positive definite (which is typically the case, and can be guaranteed by choosing a small enough time step even if the simulation state involves a large and indefinite elastic stiffness $\mathbf{K}$). Furthermore, the value of c must be negative (or zero) because $n + \Delta n$ must always be a positive (or zero) quantity of air trapped in the cavity at the next time step, and constants R and T are positive. Thus the discriminant $\mathrm{d} = \mathrm{b}^2 - 4\mathrm{ac}$ is always positive, and it is also greater or equal to b^2.

Now let us understand the impossibility of multiple positive solutions for P. Notice that the linear term coefficient b measures the volume at the next time step accounting for the current velocity, external and elastic forces, and the contact forces, while ignoring the influence of pressure on the cavity. The total will be non-negative provided that we can rely on the resolution of contacts to ensure contact forces that prevent the cavity going past a zero volume state with interpenetration between the surfaces forming the cavity. Thus,

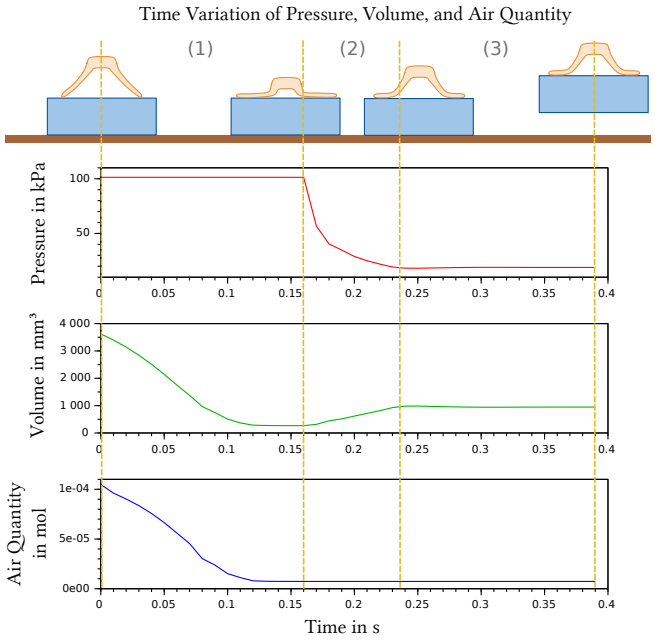

Fig. 4. Time-variation of the pressure, volume, and air quantity of a cavity during a scenario wherein we lift a rigid cuboid with a suction cup. The timeline is split in 3 time periods: (1) pushing the cup onto the object and hold for few time steps, (2) pulling the object while it is still on the ground, and (3) finally lifting the object.

with $\mathrm{b} > 0$, we have one negative solution $P = (-\mathrm{b} - \sqrt{\mathrm{d}})/(2\mathrm{a})$, and with the discriminant $\mathrm{d} \geq \mathrm{b}^2$, we will also always have a positive (or zero) solution,

$$P = (-\mathrm{b} + \sqrt{\mathrm{d}})/(2\mathrm{a}). \tag{12}$$

This is the solution we choose because we are computing absolute pressure and the value must be non-negative.

Recall that if the gas law constraint can be satisfied with P not exceeding $P_{\max}$, then the cavity remains airtight and the quantity of trapped air remains constant, i.e., $\Delta n = 0$. When the solution for P exceeds $P_{\max}$, then P must be clamped at maximum value, and a change in air quantity over this time step is computed by evaluating the quadratic Equation 11 with the pressure set to $P_{\max}$, that is,

$$\Delta n = \frac{\mathrm{a}P_{\max}^2 + \mathrm{b}P_{\max}}{\mathrm{RT}} - n. \tag{13}$$

Here, Δn does not depend on P computed by Equation 12 because it is clamped, and furthermore, we can observe that Δn is zero if P is exactly $P_{\max}$ (i.e., the pressure does not exceeded the limit, does not need to be clamped, and the gas law is satisfied). When the solution P is clamped to $P_{\max}$, notice that Δn becomes negative, because $\mathrm{a}P_{\max}^2 + \mathrm{b}P_{\max} < \mathrm{a}P^2 + \mathrm{b}P$ when $P > P_{\max}$. That is, the air quantity will never increase when P is clamped.

Finally, notice also that this first term is positive because coefficients a and b, and the maximum pressure are positive (the air quantity can never be reduced below zero over a time step). Thus

we arrive at a complementarity condition:

$$0 \leq (P_{\max} - P) \perp -\Delta n \geq 0. \tag{14}$$

3.5 Air Leakage

The amount of air n that is trapped under the cavity is a parameter that allows to actively control the suction cups. In nature for instance, the upper chamber of the octopus' suckers plays this role of reducing n in the lower chamber (i.e. in the suckers) to improve the tightening of the grip. Also, the grain of a surface can lead to slow leakage into the cavity, increasing n, and eventually breaking the air-tightness of the cavity.

We propose to allow such controllable parameter n_{leak}, which is a variation of quantity of air per second, and add it to n at the beginning of each time step,

$$n \leftarrow n + hn_{\text{leak}} \tag{15}$$

This allows interesting scenarios such as the Monster pop-up toy example shown in Figure 1, where slow leakage into the cavity releases the spring forces after some time, which makes the whole structure to jump.

3.6 Maximum Pressure

The geometry of the elastic objects and contacts ultimately determine the necessary pressure for there to be a break in the seal to let air escape when the suction object is being pressed onto a surface. For simplicity, we set this pressure limit to a value at or above atmospheric pressure. Thus, the pressure in the cavity P, must always be between zero and some upper limit $P_{\max} > P_{\text{atm}}$.

Alternatively, it is also reasonable to set $P_{\max}$ by searching for the minimum total traction path connecting an active contact vertex on the inner boundary of the seal to one on the outer boundary of the seal, where the traction at each vertex is computed as a third of the area of adjacent triangular faces times the normal contact force. We can then set $P_{\max}$ to P_{atm} plus the minimum total traction.

In another simplification, it may also be possible to map the total elastic potential energy of the suction cup to a reasonable upper pressure limit.

3.7 Practical Observations and Implementation

Once the pressure is solved and clamped, we perform further PGS iterations to refine non-interpenetration, friction, and pressure forces. This occurs in the for-loop of Algorithm 2.

Note that if the quantity of air $n + \Delta n$ vanishes to zero during a time step, then we will have arrived in a $PV = 0$ state that would lead to behavior in total vacuum: the pressure would always be $P = 0$ regardless the volume of the cavity (i.e., the gas law is trivially solved).

In practice, the pressures applied in the simulation are shifted by P_{atm} such that it is not necessary to apply pressures to the outer surfaces of the finite element models. Thus it is only on the interior surface of cavities that we apply pressures, both positive, and negative down to $-P_{\text{atm}}$.

Our implementation is in C++ and runs within SOFA, an open-source simulation framework for deformable objects [Faure et al. 2012]. In order to be able to simulate deformable objects undergoing large displacements, we use the co-rotational FEM method [Müller and Gross 2004]. When necessary, we can use an asynchronous preconditioner [Courtecuisse et al. 2010] with a GPU implementation in order to accelerate the construction of the compliance matrices, $\mathbf{W}_{pp}$, $\mathbf{W}_{cp}$, $\mathbf{W}_{pc}$, and $\mathbf{W}_{cc} = \mathbf{H}_c\mathbf{A}^{-1}\mathbf{H}_c^T$.

4 CAVITIES

In this section, we present how we detect closed cavities formed between bodies in contact, and how we process these cavities to apply the pressure on the bodies' wall. In [Bernardin et al. 2019] the cavities are detected using a classification of vertices and flood-fill of surface regions. The main drawback of this approach is that the users have to define a point on all the meshes that will always be outside the cavity. We propose a more practical and robust approach based on geometric operations.

4.1 Detection and Meshing of Cavities

A cavity can be formed between multiple bodies in contact. Let us first consider the simple case of two bodies in contact. We start the detection method by computing unions of meshes using the CGAL library. At each time step of the simulation, if a new potential contact point is detected between two bodies, we compute the union of the corresponding collision meshes and identify if there is only one or several connected components (see Figure 5). This step is fast and easy to perform with the connectivity of the mesh, which is computed only once at its creation. If the union is composed of a single connected component, it means that there is no closed cavity and the process stops here. Otherwise, if the union is composed of two or more connected components, we identify the components with vertices on both bodies, remove the outer mesh (gray in Figure 5), and select the remaining meshes for suction (green in Figure 5), if any. Indeed, only the cavities shared by the two bodies can be empty and thus give suction.

The outer mesh is identified by looking at the volumes; the volume of the outer mesh will always be greater than the volume of a cavity. We compute the volume of the surface meshes using Gauss's divergence theorem. Note that to detect that a cavity has vertices on both objects, we need a map between the nodes of the original meshes and the nodes of the resulting union, and this map is easily retrieved from CGAL's outputs. The collision meshes are also coarse to lower the computational expense of the collision response, which also benefits the computational expense of the union.

We use the CGAL library because it provides a robust and fast algorithm allowing us to maintain real time performance (see Table 1). For this algorithm to work, each mesh must be closed with no self-intersections (i.e., no intersection with itself). These two conditions are generally satisfied in simulations of solids, and are furthermore required to handle self-collision. If the two meshes are perfectly in contact, i.e., the distance between the meshes at the contact points is exactly zero, the union will contain the contact surface. Therefore, we let the two meshes slightly intersect in order to isolate cavities. Since our collision detection method does not allow for this offset, we slightly translate the points of the mesh that are in contact in the direction of the contact's normal before computing the union. We call this offset the *sealing distance*.

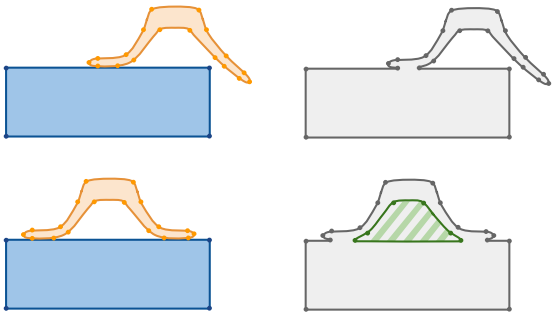

Fig. 5. Unions (right) of two meshes (left). Top: the union is composed by one connected component, there is no cavity. Bottom: the union is composed by two connected components, the cavity (hashed green) and the outer mesh (gray).

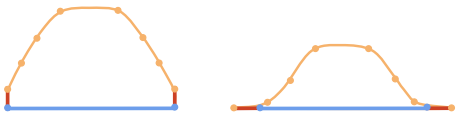

Fig. 6. Duplication of the inner boundary points. The orange and blue points are mapped to a distinct object. Left: the suction cup moves away from the object, the duplicated points separate vertically. Right: the suction cup slides on the object, the duplicated points separate horizontally.

The current implementation only handles cavities formed by two bodies, but it is straightforward to extend the approach to more bodies. Indeed, once we have identified groups of bodies connected through contacts, we can compute the union of each group, adding the contribution of each body one by one. In the remaining of the document we will continue to only consider the case with two bodies. Note that this implementation allows for multiple cavities between two bodies.

4.2 Inner Boundary and Mapping of Cavities

Once we have the mesh of a cavity, we map each of its vertices to the mechanical model of the body it lies on. We use the same mapping technique than for the visualization and collision meshes, except that the points are distributed among two different models. By doing so, we allow the cavity mesh to follow the deformation of the bodies, if any, and the pressure forces computed at each node of the cavity mesh to be transferred to the mechanical model of the two bodies.

In order for the inner boundary of a cavity to follow the deformation of both bodies, before mapping, we duplicate the points of the boundary and also add the triangles that connect the duplicated points to the mesh. This allows the cavity to stretch or break up according to the bodies deformation (see Figure 6 for an illustration in 2D). The connectivity of the inner boundary points is easily retrieved from CGAL's output, making it simple to create these new triangles.

When the area of the boundary triangles is greater than 1% of the cavity area, we decide that the current cavity mesh is no longer a good match and we recompute the union. By allowing the cavity to deform with the suction cup, and remeshing only when slippage or breakage occurs, the cavity mesh is free to flatten and then it can apply some pressure on the part of contact surface. The model therefore follows the phenomenon that occurs with a real suction cup. We recall that we also recompute the union if we detect a new potential contact point. For instance if a model is composed of several suction cups, and each cup comes in contact with the object at different times of the simulation, this allow us to capture the apparitions of new cavities.

4.3 Tracking Cavities Over Time

Re-meshing can lead to a loss of information about the cavities. Yet, it is important to keep track of the amount of air in each cavity to avoid discontinuities. For each cavity found in the remeshing, we find its corresponding in the previous step. In that matter, we compute the geometric center of each cavity of a new set, and compare them against those of the previous set. We first pair the cavities based on the proximity of their center. Each cavity being in a unique pair, which allow us to handle the situations where the two sets have different size (i.e some cavities vanished or newly appeared). Because this proximity check alone does not account for situations where some cavities vanished and other newly appeared at the same time, we also check that the previous center is inside the cavity of the new set. Once we have paired the cavities, for each cavity of the new set, we assign the air quantity of the corresponding previous cavity.

5 RESULTS

In this section we present different simulations which illustrate the performances of our approach (see also the supplementary video). In all these simulations the Young's modulus of the suction cup is set to 4 MPa and the Poisson's ratio to 0.45, which corresponds to soft silicone. While our experiments include suction cups of different shapes, the first four scenarios below involve a standard suction cup of 35×15 mm size and 20 g weight.

Ground. In the first row of Figure 7, the suction cup is pushed onto the ground until the cavity is almost empty, and then pulled out. The suction cup first sticks on the surface before being released when the pressure constraints are not as strong as the elasticity forces and the pulling force. Note that because the union only works with closed meshes, the ground is in fact a flat cuboid.

Soft Cube. In the second row of Figure 7, the suction cup is used to lift a compliant cube of 100 g. The Young's modulus of the cube is 0.05 MPa and its Poisson's ratio is set to 0, allowing the material to expend horizontally when being pushed vertically. We can see both objects deform when we flatten the suction cup onto the cube.

Bunny. In the last row of Figure 7, the suction cup is used to lift a rigid bunny of 147 g. Note that the visual model of the bunny shows a non smooth surface, which should probably prevent the sealing of the cup, but the collision model, which is used to compute the

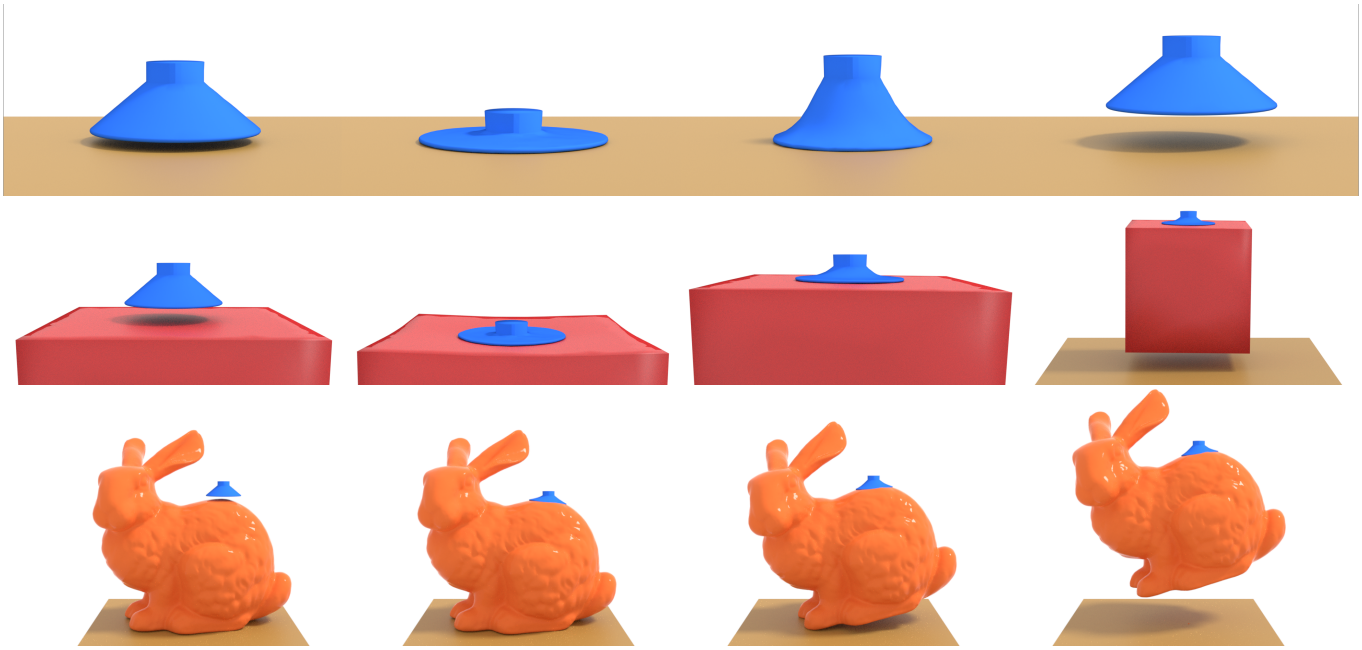

Fig. 7. Different scenarios illustrating our approach for simulating suction phenomena. From top to bottom: (1) a suction cup is stuck to the ground before being released, (2) the same suction cup lifts a highly compliant cube, (3) and a rigid bunny.

union, is coarse and smooth at this location, allowing a cavity to form and the suction cup to lift the bunny.

Toy. In Figure 1, we show a jumping toy that combines a spring and a suction cup. The toy has a solid base (dark brown) attached to a suction cup (blue) by a spring. The spring is physically simulated with a finite element model using a wrapped deformable model composed of hexahedra. A rigid monster head is placed above the suction cup and can be pushed, sticking the suction cup on the basis. With the right spring stiffness, and a bit of air leakage into the cavity, the toy jumps into the air after some time.

Pick and Place. In the first row of Figure 8, we show a pick and place scenario where a suction cup is fixed to the end effector of an articulated rigid arm. The robot is able to pick an apple and place it into a compartmentalized box. The apple is released when being gently pushed onto the box's wall.

Octopus Tentacle. In the second row of Figure 8, an octopus tentacle falls gently onto a can, making some of its suckers flatten on the curved object. Once some cavities are sealed, the tentacle is able to lift the can up in the air. Note that a real octopus uses some muscles to contract its suction cups and empty the cavities.

Cube Halves. All the examples presented so far involve suction cups with classic shapes, while one does not necessarily require a so-called suction cup to observe suction phenomena. In the last row of Figure 8, the two halves of a jelly cube are pressed against each other. One half having a hole which creates a cavity when the other object comes in contact. By pushing the one on top, we allow some quantity of air to go out of the cavity, and then the two objects stay stick when we raise the one on top. Note that the hole has a random shape.

5.1 Convergence and Performance

Below we present a convergence test involving different mesh resolutions and different time steps. We also report timings for the different scenarios presented in the previous section. We performed timings on a laptop with an Intel Core i9-9980HK CPU at 2.40 GHz × 16, 32GB of DDR4 system memory, and a NVIDIA GeForce Quadro RTX 3000 Mobile graphics card.

Convergence. In Figure 9 we show the time-variation of the pressure, volume, and air quantity of a cavity with different FEM mesh resolutions of the suction cup (434, 816, 1787, 2676, and 5197 nodes), and different time steps (0.03, 0.01, and 0.001). The simulation corresponds to a scenario wherein we push a suction cup on the ground and then pull it off until release. We can see that the pressure is a bit lower in the case of a large time step (i.e., 0.03), but overall the method shows consistent results even with coarse meshes.

Computation Time Performances. Table 1 summarizes the computation time performances of our approach for different scenarios. We measured the total time needed to achieve the entire simulation with respect to the complexity of the scenario. In addition to the total time, we report the corresponding percentage of time spent for the computation of the free motion configuration, the collision detection, the cavities detection, the computation of the compliance matrices, and the system solve. For all scenarios, we used a time-stepping value of 10 ms. The maximum number of Gauss-Seidel iteration was clamped to 50. All the results were acquired using

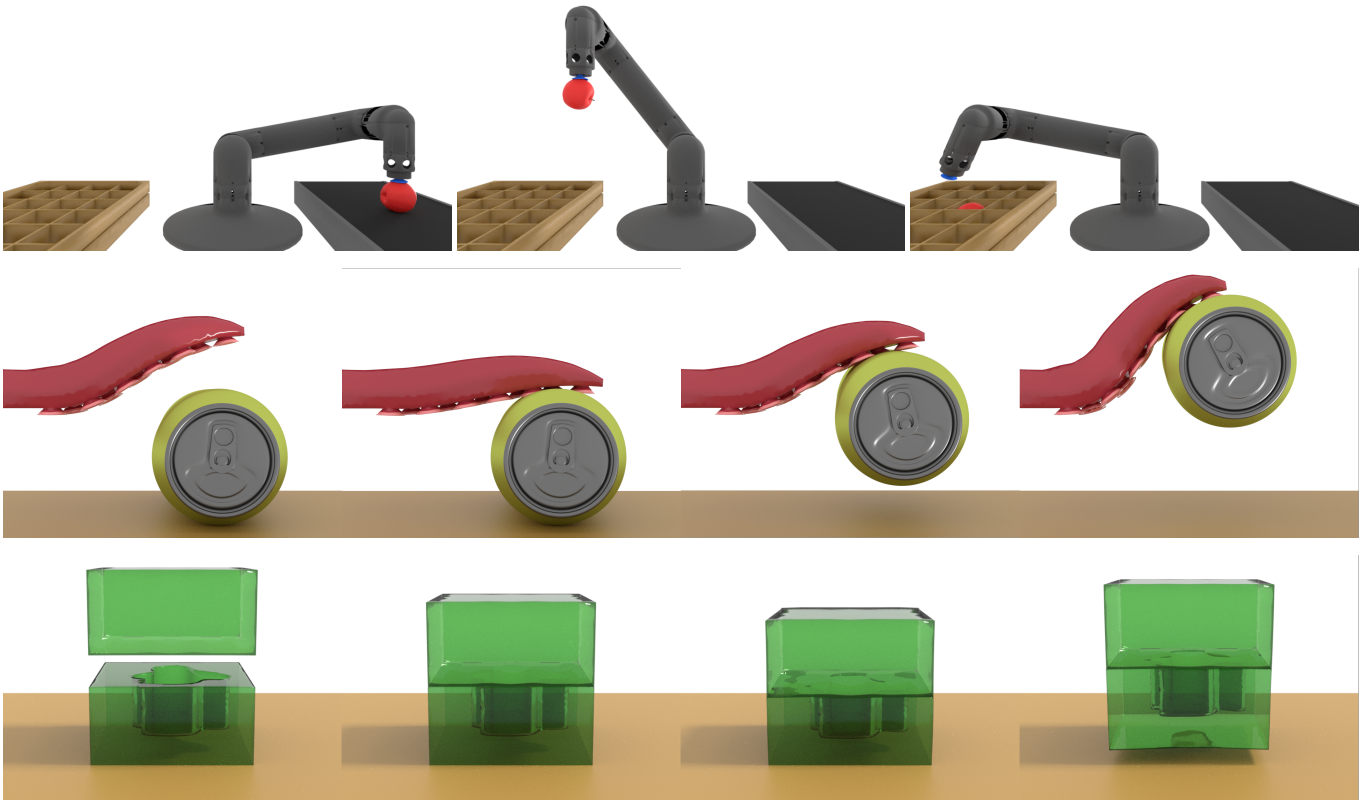

Fig. 8. Top: A suction cup is fixed on a rigid arm's end effector to perform a pick and place task. The robot drops the apple off in a compartmentalized box by gently pushing it against the box's wall. Middle: An octopus tentacle lifts a can using its suction cups. Bottom: The two halves of a jelly cube are pressed against the other. One half has a hole which creates a cavity. When pushing the half on top, air escapes from the cavity which makes the object behave like a suction cup.

Table 1. Computation time performances of the different scenarios. We report, the total number of nodes of the FEM models, the total number of nodes of the collision models which are also used for the union, the average number of constraints observed during the simulation, the average computation time per time step (in milliseconds), and the detailed computation times among the main different components of the simulation, we additionally indicate in parenthesis the portion of total simulation time as a percentage. The remaining portion of the simulation that is not reported includes visualization and general bookkeeping.

Scenario	Nb of nodes FEM meshes	Nb of nodes coll. meshes	Avg. Nb of constraints	Avg. time per step	Free motion	Collision detection	Cavities detection	Compliance build	System solve
ground	434	135	122.75	38.51	9.02 (23.44%)	0.40 (1.04%)	3.13 (8.14%)	21.31 (55.34%)	1.33 (3.47%)
soft cube	820	325	111.96	56.55	16.52 (29.23%)	0.57 (1.02%)	6.91 (12.22%)	26.58 (47.01%)	1.96 (3.47%)
bunny	434	572	182.00	67.59	9.10 (13.47%)	0.94 (1.40%)	6.33 (9.38%)	41.67 (61.66%)	3.01 (4.46%)
toy	628	212	234.72	86.48	19.12 (22.12%)	0.76 (0.89%)	1.62 (1.88%)	56.22 (65.01%)	2.17 (2.51%)
robot	434	507	262.64	87.96	15.24 (17.18%)	2.44 (2.75%)	9.06 (10.21%)	51.17 (57.68%)	3.15 (3.55%)
cube halves	644	801	705.16	390.04	14.31 (3.67%)	1.91 (0.49%)	2.92 (0.75%)	343.85 (88.16%)	18.21 (4.67%)
tentacle	1585	2044	249.70	427.61	36.56 (8.55%)	1.83 (0.43%)	98.30 (22.99%)	263.15 (61.54%)	3.24 (0.76%)

averages onto a sample of 150 time-step iterations, taken around a suction phenomena. For instance in the robot scenario, it corresponds to when the robot picks up the apple.

Most of the cavities detection's computation is spent on the union. Note that we use non-uniform meshes for the collision model, for instance, a flat surface which does not deform very much will typically have less points than a highly deformable one or a curved one. This allow us to reduce the computational expense of the collision process, which also benefits the computation of unions. For the tentacle example, the resolution of the FEM and collision meshes is high because of the numerous small suckers, which increases considerably the computation time of the cavity detection and the building of the compliance matrices. As for the cavities detection, mainly the computation of the union, different approaches can be

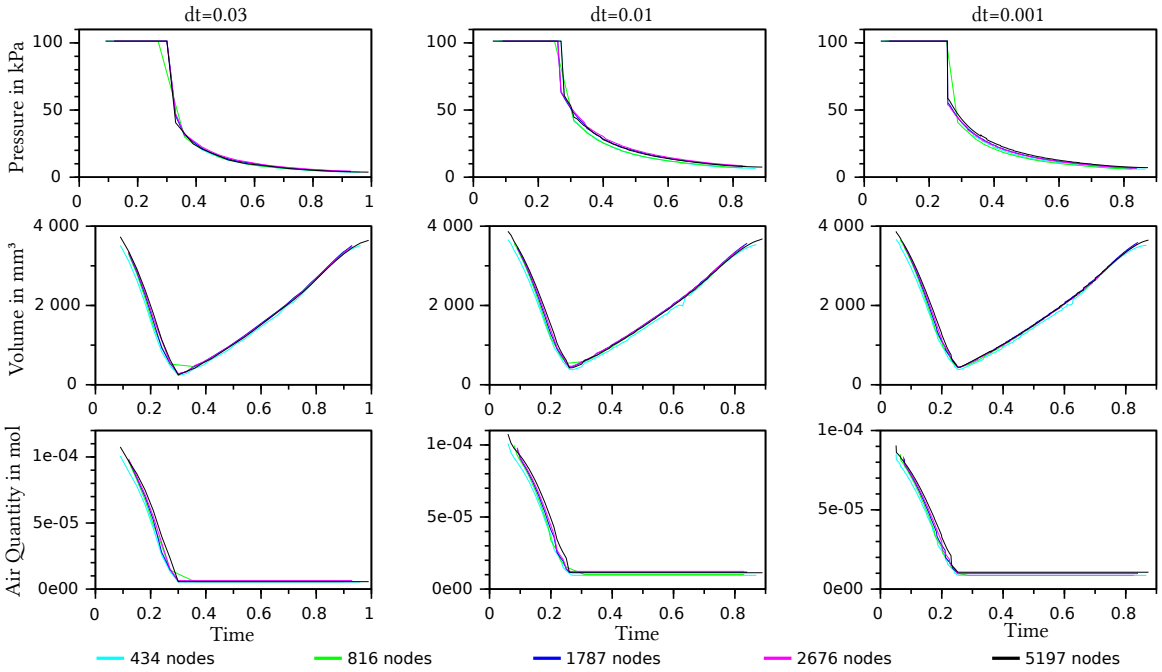

Fig. 9. Time-variation of the pressure, volume, and air quantity of a cavity for different FEM mesh resolutions for the suction cup and different time steps, during a scenario where the suction cup is pushed downward against the ground and then pulled upward until it detaches.

Table 2. Performance of the rigid cube grasping scenario with different mesh resolutions of the cube collision model. We report, the total number of nodes of the collision models which are also used for the union. The average compute time for a time step is listed in milliseconds and for each component of the algorithm we additionally indicate in parenthesis the portion of total simulation time as a percentage. The FEM mesh of the suction cup is the same in each case and has 434 nodes.

Nb of nodes coll. meshes	Avg. Nb of constraints	Avg. time per step	Free motion	Collision detection	Cavities detection	Compliance build	System solve
135	123.52	37.17	8.99 (24.20%)	0.42 (1.15%)	1.07 (2.90%)	22.56 (60.71%)	0.76 (2.06%)
225	229.22	74.00	9.15 (12.37%)	0.59 (0.80%)	1.56 (2.11%)	56.81 (76.78%)	2.41 (3.26%)
360	322.91	127.42	9.65 (7.58%)	0.75 (0.59%)	2.95 (2.32%)	105.17 (82.54%)	4.63 (3.64%)
991	674.60	475.36	9.26 (1.95%)	1.18 (0.25%)	8.98 (1.89%)	430.39 (90.54%)	21.48 (4.52%)
1817	1022.85	1185.12	9.59 (0.81%)	2.13 (0.18%)	24.17 (2.04%)	1082.72 (91.36%)	61.74 (5.21%)

investigated to lower the computation expense. For instance, the collision mesh can be divided into several pieces, one for each sucker and one for the main body, and if several unions happen at the same time, it is also straightforward to compute them concurrently.

We also ran a rigid cube grasping scenario with several resolutions of the cube's collision mesh in order to illustrate how it impacts the computation of the collision process and the cavities detection. The results are shown in Table 2. We can see that the main loss in terms of performance is due to the building of the compliance matrices. This can be improved by using the GPU approach introduced in Section 3.7.

5.2 Benchmarks

In order to challenge the behavior of the simulation model, we reproduced some scenarios several times by varying parameters and shapes.

Curved Surfaces. The first row of Figure 10 shows an experiment in which we vary the curvature of a cropped cylinder until the suction cup is no longer able to lift the object. The second row of Figure 10 shows the same curvatures but on concave objects. We can see that the suction cup rapidly fails to create a sealed cavity, which is mainly due to the design of the cup (more precisely its diameter), as it can be seen in Figure 11 and in the supplementary video.

Different Shapes. Figure 12 shows three different designs of a suction cup, all three succeeding at lifting a rigid cube of 190 g. From left to right, the cup is inspired by the plunger, the suckers of the octopus, and a cup that can be found in a specialized store. In opposition to the suction cups we have shown so far (e.g. the suction cup of Figure 14), these suction cups do not empty their cavity by flattening on the object, but rather by deforming their top.

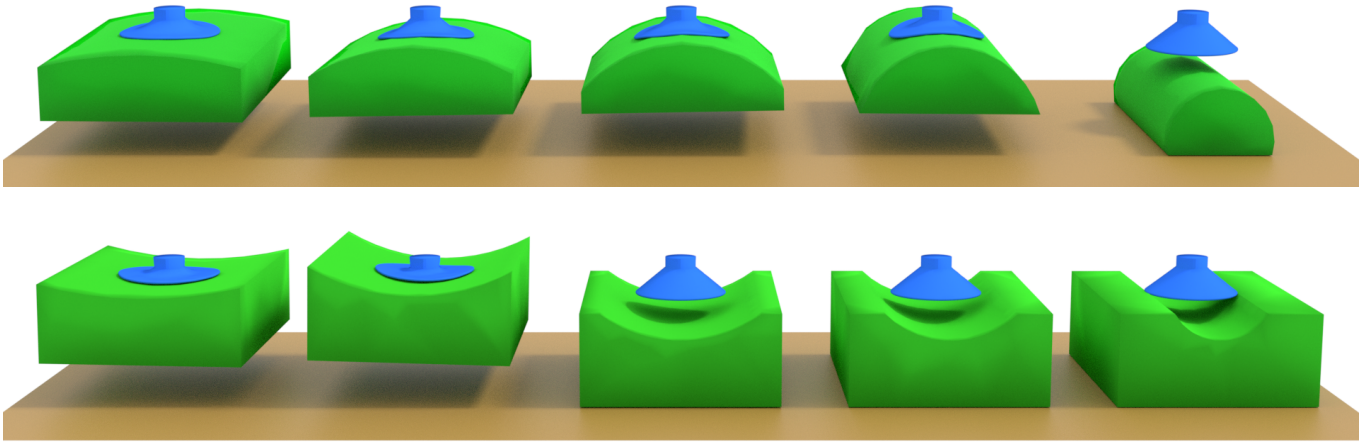

Fig. 10. The suction cup is able to lift objects with curved surfaces, but fails if the curvature is too high.

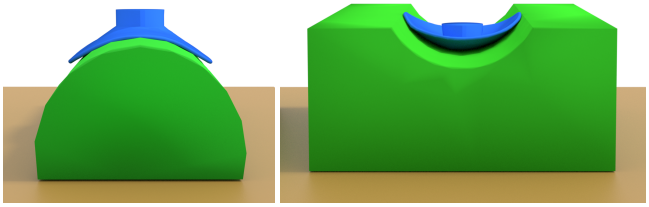

Fig. 11. Two curved surfaces presented in Figure 10 (last on the right) when the suction cup is flattened onto the object, showing that no sealed cavity can be created between them due to the design of the cup.

Air Tunnel. In the scenario of Figure 13, we test different configurations of suction on a deformable cube composed of a tunnel connecting two holes on its top surface. The suction cup has to simultaneously cover the two holes to lift the cube. If the two holes are not entirely covered (as shown in the left images of Figure 13), it is detected by our method and the suction cup is unable to lift the cube. On the right images of Figure 13, the holes are covered by the two suction cups.

Different Masses. Figure 14 shows six identical suction cups used to lift rigid cubes of different masses; 1.25 kg, 2.5 kg, and 5 kg. We can see the effect of the weight on the deformation of the cup. When the weight reaches a certain limit, the suction cup finally fails to lift the cube, while using two suction cups on the same cube works.

Low Friction. Figure 15 shows a case of sliding between the suction cup and the object. In this scenario, the suction cup is slightly shifted to the left to break the symmetry from the start, and the coefficient of friction is set very low. The suction cup is able to lift the cube but slides to one of its corners at the same time. Once the suction cup reaches the corner of the cube, the cavity breaks and the cube is released. Note that without the shift, the shape of the cavity rapidly breaks the symmetry.

Table 3. Force (in N) required to detach a suction cup from cylinders of different radius (in mm). Comparison between the averaged sensed force (10 measurements) using a real suction cup and its simulated twin.

Radius	Real (wet)	Real (dry)	Simulation
50.0	34.56	32.34	28.19
31.5	28.29	25.54	24.39
25.0	23.60	22.83	22.60
20.0	31.65	31.50	-

Experiments. In the following experiments, we collect the force required to detach a real suction cup from a series of curved surfaces, and compare with the force we obtain in simulation. The suction cup we used is a commercially available 31.5 mm diameter neoprene suction cup. The cup is attached to a force-torque sensor (HEX-E QC by OnRobot) mounted on the end effector of a manipulator robot (UR3 by Universal Robots). Using the manipulator, we press the cup against the curved surface of PVC tubes until it is attached to the surface by vacuum. Then, we lift the end effector of the robot until the cup is detached from the tube (see Figure 16). We repeat the sequence 10 times per surface and report the average force limit in Table 3. Each surface has a different curvature radius ranging from 20 mm up to 50 mm. For each surface, the test was performed twice; the first time while the surface was wet, with few drops of water on the surface, and the second one while the surface was completely dry. We modeled the suction cup to visually match the real one as much as possible. In the simulation we set the Young's modulus of the cup to 1 MPa and its Poisson's ratio to 0.45.

We can see that both the curvature and the wetness of the surface influence the resistance of the suction. Indeed the liquid helps to seal the cavity. In simulation, we obtain values that are fairly close to those of the dry condition; except for the highest curvature where no sealed cavity can be created as air gaps, similar to those in the

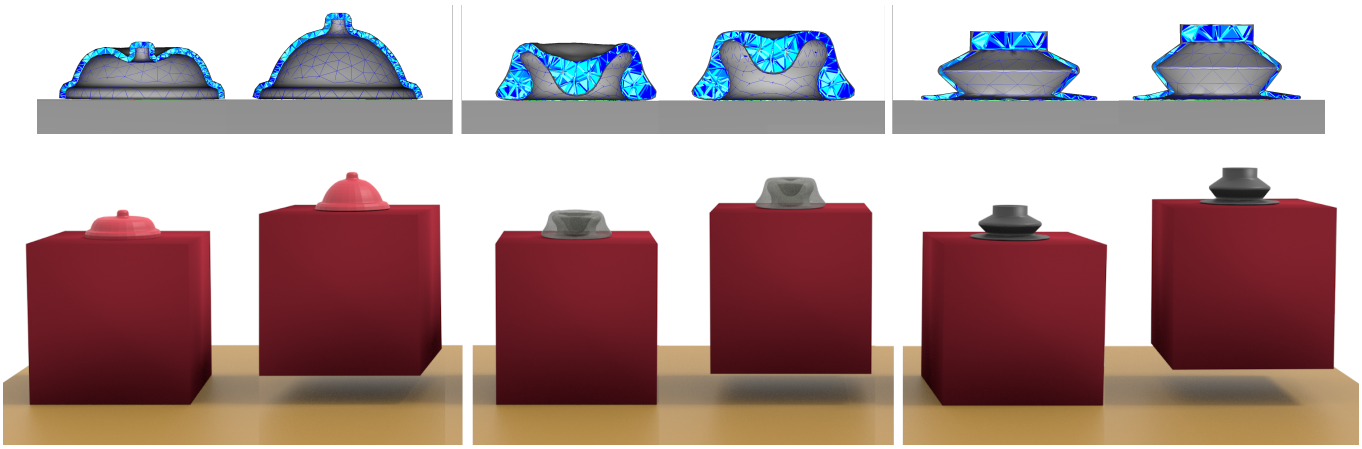

Fig. 12. Three suction cups with different shape succeeding at lifting a rigid cube of 190 g. Top: clipped views showing the shape of the three cups. The gray triangles show the surface mesh used for visualization, and the blue elements show the volume tetrahedral mesh used for the FEM model. Bottom: corresponding global view of the simulation.

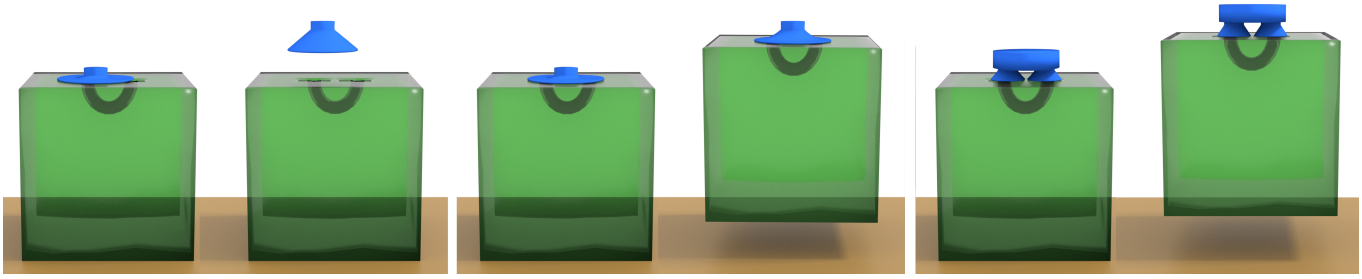

Fig. 13. In this experiment, a deformable cube includes a tunnel connecting two holes. Left-to-right shows the progression of the scenarios. Left: if the suction cup is pushed down onto only one hole, we correctly identify the air passage and an airtight cavity is not formed. Middle: in contrast, we can lift the cube by suction when the rim of the suction cup covers both holes. In this case the tunnel is part of the cavity. Right: both holes are covered by a different suction cup forming a closed cavity.

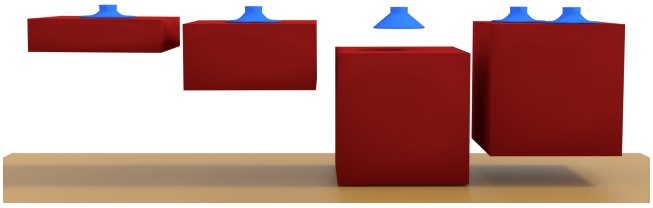

Fig. 14. The same suction cup is used to lift cubes of different masses: from left to right, 1.25 kg, 2.5 kg, 5 kg, and 5 kg.

left image of Figure 11, can be seen. We think this is due to errors in the modeling of the cup which are really hard to catch.

6 DISCUSSION AND LIMITATIONS

Our approach is able to simulate various suction phenomena, such as those demonstrated in the different scenarios in Section 5. We are able to stick or slide suction cups on different types of surfaces (flat and curved surfaces, rigid and deformable objects), and to lift objects thanks to the pressure forces that result in the air cavities formed by suction cups. When combined with other types of phenomena, such as leakage with the spring in the toy scenario, our method efficiently handles the different constraints and produces realistic behaviors.

In our current implementation, cavity detection is only between two objects. As such, we do not handle the case where a suction cup creates a cavity with several other objects participating to form an airtight seal. Given our focus is on simulating suction cups, this limitation does not prevent us from handling the common cases in which our technique can be useful, such as simulation of soft robotics, or computational design applications. However, it is straightforward to extend our approach to handle more complex cavities. Indeed, as already mentioned in section 4.1, once groups of bodies connected through contacts have been identified, one can compute the union of each group, adding the contribution of each body one by one. Then, for each pair of bodies in contact in a group, the duplication of the inner boundaries and the mapping of the cavities remain the same.

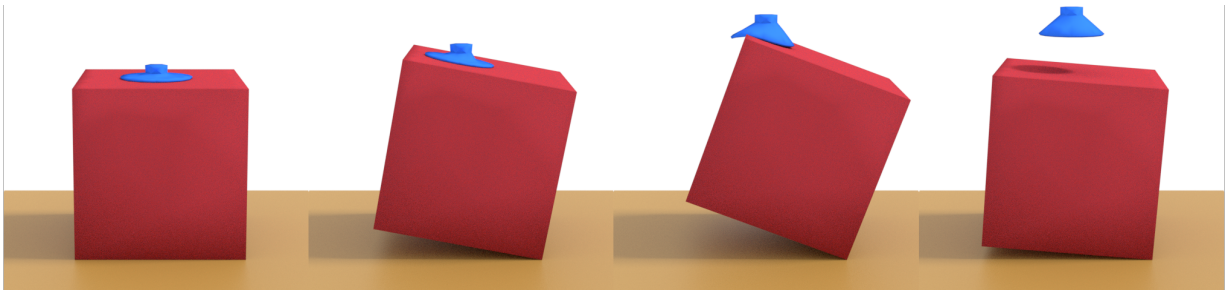

Fig. 15. With a low coefficient of friction the suction cup slides on the object.

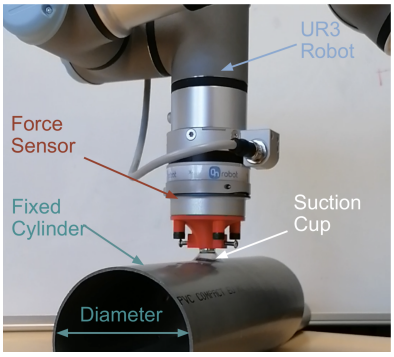

Fig. 16. Snapshot of the experimental setup. The robot, equipped with a force sensor, repeats an adhesion test of the suction cup on cylinders of various diameters.

We think that the detection and meshing steps based on the union operation are really robust, even with very small cavities. Tracking on the other hand is not straightforward. The heuristic we propose gives satisfactory results in most cases, but can be challenged by adjacent very small cavities since we look at the proximity to the cavities center. For instance, it can happen that those very small cavities merge together or with a bigger cavity, and we do not detect that.

The computation time of our simulations are close to real time. The use of GPU approaches and model order reduction techniques for the deformable objects can further improve the total time of the simulation. For instance, a way to reduce the cost would be to implement a different constraint-solver with a full-GPU implementation. Yet another solution would be to decrease the number of contacts points in the manner described by Talvas et al. [2015].

Note that for collision detection purpose, the objects are meshes that are faceted with a relatively coarse resolution, which does not always produce ideal sealing conditions. To counter these difficulties, we can use the *sealing distance* introduced in Section 4.1. When this distance is increased, the cavity detection algorithm becomes more tolerant to small leaks that can occur due to the geometric discretization. For instance, a sealing distance of 0.05 mm is used in most of the simulations we have presented, although a distance of 0.3 mm is used for some of the curved surfaces. The sealing distance also avoids expensive collision computations, since highly curved objects would otherwise require a high resolution mesh.

The current implementation of our method does not completely handle the case of zero volume. That is, if the suction cup is perfectly flattened, and no remeshing is triggered, the method will work even if the volume of the remaining cavities is zero. However, if a remeshing is triggered, resulting in a union with no cavity, the current implementation will have no surface on which applying the pressure and will then fail by releasing the suction cup. One solution is to keep the previous mesh in such case, but it often happens that very small cavities form and vanish during the simulation, which makes it difficult to identify which one to keep.

Our constraint-based suction is based on the ideal gas law and assumes that suction is an isothermal process. In other words, $PV = n\mathrm{RT}$ and T is constant since we assume that thermal gains due to volume or pressure changes are immediately transferred to the surrounding environment. Other than inertial effects, simulations behave the same regardless if the suction cup is pulled slowly or quickly. Conversely, an adiabatic process is insulated from the environment, and the temperature inside the suction cavity fluctuates since for adiabatic compression, PV^{γ} is constant and pressure-volume changes according to γ the adiabatic index of the gas (e.g., $\gamma = 1.4$ for dry air). The pressure inside the cavity will therefore be lower in an adiabatic process compared to an isothermal process with the same volume. This would effectively increase the force of a suction cup and could explain, in part, why the real-world robot requires a larger force to detach the suction cup (see Table 3). However, the robotic suction cup is not exactly an adiabatic system, since some thermal transfer is possible. Rapid changes in the pressure or volume also mean that it is not exactly an isothermal process either, since there is little time for thermal transfer. It would be interesting to extend our approach in the future to allow modeling such phenomena.

The suction phenomenon is a multi-scale phenomenon. The micro scale roughness of the surface is idealized in our macro-scale approach. We have introduced a parameter of air leak which can represent at the macro-scale the effects that appear at micro-scale, but it is an assumption of our model.

7 CONCLUSION

We present a novel physics-based model for handling simulations of suction phenomenon between elastic solids. We chose a constraint-based formulation to model both pressure and contact between objects. The dynamic deformations of the suction cup are achieved using the finite element method. Our approach is composed of several important components: (1) the modeling of the pressure as a nonlinear constraint based on ideal gas law and its coupling with interpenetration and friction constraints, (2) the formulation of a complementarity constraint that couples pressure and airflow to model the state of the suction phenomenon. (3) a method to detect and process cavities, We have experimented various scenarios illustrating the ability of our method to simulate different suction phenomena coupled with other dynamic behaviors. We believe that the efficient computation time of our approach makes it useful for applications such as the fast design of suction cups for soft robots, or for computer animation of common household suction cups, as well as animals that use suction, such as snails and octopuses.

REFERENCES

B. Bahr, Y. Li, and M. Najafi. 1996. Design and suction cup analysis of a wall climbing robot. *Computers & electrical engineering* 22, 3 (1996), 193–209.

D. Baraff and A. Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54. https://doi.org/10.1145/280814.280821

A. W. Bargteil, T. Shinar, and P. G. Kry. 2020. An Introduction to Physics-Based Animation. In *SIGGRAPH Asia 2020 Courses*. Association for Computing Machinery, New York, NY, USA, Article 5, 57 pages. https://doi.org/10.1145/3415263.3419147

A. Bernardin, C. Duriez, and M. Marchal. 2019. An Interactive Physically-based Model for Active Suction Phenomenon Simulation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1466–1471.

N. Chentanez, M. Müller, and M. Macklin. 2016. Real-time simulation of large elasto-plastic deformation with shape matching. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 159–167.

H. Courtecuisse, J. Allard, C. Duriez, and S. Cotin. 2010. Asynchronous preconditioners for efficient solving of non-linear deformations. In *Proc. of Eurographics Symposium VRIPHYS-Virtual Reality Interaction and Physical Simulation*. 59–68.

C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. 2005. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE transactions on visualization and computer graphics* 12, 1 (2005), 36–47.

K. Erleben. 2017. Rigid body contact problems using proximal operators. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*. 1–12.

F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin. 2012. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Y. Payan (Ed.). Studies in Mechanobiology, Tissue Engineering and Biomaterials, Vol. 11. Springer, 283–321.

M. Fratarcangeli, H. Wang, and Y. Yang. 2018. Parallel iterative solvers for real-time elastic deformations. In *Proc. of ACM SIGGRAPH Asia Courses*. 14.

D. Ge, T. Matsuno, Y. Sun, C. Ren, Y. Tang, and S. Ma. 2015. Quantitative study on the attachment and detachment of a passive suction cup. *Vacuum* 116 (2015), 13–20.

G. Irving, C. Schroeder, and R. Fedkiw. 2007. Volume Conserving Finite Element Simulations of Deformable Models. *ACM Transactions on Graphics* 26, 3 (2007), 13–es.

D. E. Johnson and P. Willemsen. 2003. Six degree-of-freedom haptic rendering of complex polygonal models. In *Proc. of IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. 229–235.

F. Jourdan, P. Alart, and M. Jean. 1998. A Gauss-Seidel like algorithm to solve frictional contact problems. *Computer methods in applied mechanics and engineering* 155, 1-2 (1998), 31–47.

D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai. 2008. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics* 27, 5 (2008), 164:1–164:11.

T. Kugelstadt, D. Koschier, and J. Bender. 2018. Fast Corotated FEM using Operator Splitting. *Computer Graphics Forum* 37, 8 (2018), 149–160. https://doi.org/10.1111/cgf.13520 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13520

J. Liu, K. Tanaka, L. Bao, and I. Yamaura. 2006. Analytical modelling of suction cups used for window-cleaning robots. *Vacuum* 80, 6 (2006), 593–598.

T. Liu, S. Bouaziz, and L. Kavan. 2017. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Transactions on Graphics* 36, 3 (2017), 23.

L.-K. Ma, Y. Zhang, Y. Liu, K. Zhou, and X. Tong. 2017. Computational Design and Fabrication of Soft Pneumatic Objects with Desired Deformations. *ACM Transactions on Graphics* 36, 6, Article 239 (Nov. 2017), 12 pages.

J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg. 2018. Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *Proc. of IEEE international conference on robotics and automation*. 1–8.

D. W. Marhefka and D. E. Orin. 1996. Simulation of contact using a nonlinear damping model. In *Proceedings of IEEE international conference on robotics and automation*, Vol. 2. IEEE, 1662–1668.

D. N. Metaxas. 2012. *Physics-based deformable models: applications to computer vision, graphics and medical imaging*. Vol. 389. Springer Science & Business Media.

J. J. Moreau. 1966. Quadratic programming in mechanics: dynamics of one-sided constraints. *SIAM Journal on control* 4, 1 (1966), 153–158.

M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. 2002. Stable Real-Time Deformations. (2002), 49–54. https://doi.org/10.1145/545261.545269

M. Müller and M. Gross. 2004. Interactive virtual materials. In *Proc. of Graphics Interface 2004*. 239–246.

A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. 2006. Physically based deformable models in computer graphics. 25, 4 (2006), 809–836.

M. Nesme, Y. Payan, and F. Faure. 2005. Efficient, Physically Plausible Finite Elements. In *Proc. of EG Short Presentations*, J. Dingliana and F. Ganovelli (Eds.). 1–4.

D. Rohmer, S. Hahmann, and M.-P. Cani. 2008. Local Volume Preservation for Skinned Characters. *Computer Graphics Forum* 27 (10 2008).

E. Sifakis and J. Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses*. Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages. https://doi.org/10.1145/2343483.2343501

M. Skouras, B. Thomaszewski, B. Bickel, and M. Gross. 2012. Computational Design of Rubber Balloons. *Computer Graphics Forum* 31, 2pt4 (2012), 835–844.

M. Skouras, B. Thomaszewski, P. Kaufmann, A. Garg, B. Bickel, E. Grinspun, and M. Gross. 2014. Designing Inflatable Structures. *ACM Transactions on Graphics* 33, 4, Article 63 (2014), 10 pages.

D. E. Stewart. 2000. Rigid-body dynamics with friction and impact. *SIAM review* 42, 1 (2000), 3–39.

A. Talvas, M. Marchal, C. Duriez, and M. A. Otaduy. 2015. Aggregate Constraints for Virtual Manipulation with Soft Fingers. *IEEE Transactions on Visualization and Computer Graphics* 21, 4 (2015), 452–461. https://doi.org/10.1109/TVCG.2015.2391863

H. Xu, Y. Zhao, and J. Barbic. 2014. Implicit Multibody Penalty-Based Distributed Contact. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 20, 9 (2014), 1266–1279.

K. Yamane and Y. Nakamura. 2006. Stable penalty-based model of frictional contacts. In *Proceedings IEEE International Conference on Robotics and Automation*. 1904–1909.