



HAL
open science

High-Order Elements in Position Based Dynamics

Bastien Saillant, Florence Zara, Guillaume Damiand, Fabrice Jaillet

► **To cite this version:**

Bastien Saillant, Florence Zara, Guillaume Damiand, Fabrice Jaillet. High-Order Elements in Position Based Dynamics. *The Visual Computer*, 2024, 40, pp.4737-4749. 10.1007/s00371-024-03467-3 . hal-04569181

HAL Id: hal-04569181

<https://hal.science/hal-04569181v1>

Submitted on 6 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High-Order Elements in Position Based Dynamics

Bastien Saillant* · Florence Zara · Guillaume Damiand · Fabrice Jaillet

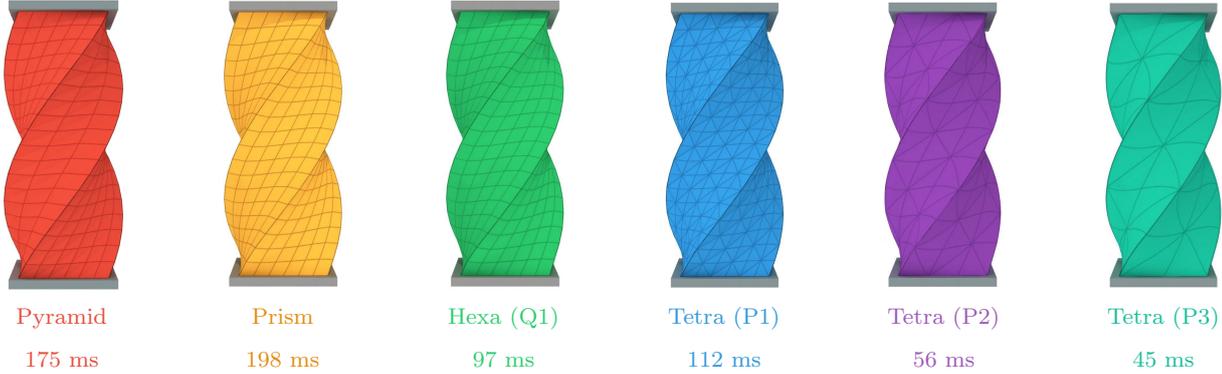


Fig. 1: 180-degree torsion of beams composed of different types of elements: pyramids, prisms, hexahedra and tetrahedra, considering linear (P1), quadratic (P2) and cubic (P3) tetrahedra. The timing indicates the average computation time for each XPBD simulation loop.

Abstract The simulation of deformable objects has been the subject of a great deal of work in the field of computer graphics. The constraint-based PBD (Position Based Dynamics) approach has been proven to be effective in this field for real-time and stable deformable objects simulation. Finite Element Method with linear tetrahedron discretization are the most widely used in Computer Graphics despite producing less accurate results than hexahedral or higher order elements. In this context, our proposal is to integrate higher degree elements within the PBD framework. In addition, we propose a solution to improve convergence of unstable energies (like Neo-Hooke) when used as constraints. We show that our approach improves accuracy compared to linear tetrahedra. We also highlight the time savings, since fewer elements are needed.

Keywords Physical Based Simulation · Position Based Dynamics · High-order elements · Elasticity · Realtime Physics.

1 Introduction

In Computer Graphics, a lot of effort have been invested in the field of simulation of deformable objects. One of the challenges has been to develop physically based models that are sufficiently accurate to be used in medical applications [31,29,9,10,30], while also offering real-time interaction with dynamic simulation.

For this purpose, well-known physical models have been adapted to achieve real time performances. For example, adaptive Finite Element methods [41,28,11] have been proposed, or the realism of discrete mass-spring model [38] has been improved by linking the springs' mechanical parameters to material properties such as Young's modulus or Poisson's ratio [32,17]. The interested reader can refer to these surveys [26,19,3] for a better understanding of these physical models for dynamic simulation.

More recently, the position-based dynamics (PBD) approach [27,6] has become popular. Due to its simplicity and robustness, many physical models have been defined within this framework [6]. To bring this approach closer to physics, the extension of PBD, called XPBD [23], provides a direct correlation with elastic energy and dissipation potentials. An adaptation has also been proposed incorporating a stable Neo-Hooke en-

B. Saillant
E-mail: bastien.saillant@liris.cnrs.fr, INSA Lyon, UCBL, CNRS, LIRIS, UMR5205, F-69622 Villeurbanne

F. Zara
E-mail: florence.zara@liris.cnrs.fr, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne

F. Jaillet
E-mail: fabrice.jaillet@liris.cnrs.fr, IUT Lyon 1, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne

G. Damiand
E-mail: guillaume.damiand@liris.cnrs.fr, CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69622 Villeurbanne

ergy [22] to ensure near-perfect preservation of the volumes. However, the various formulations proposed over the years, to solve continuum mechanics problems with PBD approaches, are restricted to linear tetrahedron discretization [5,22], which are known to be less accurate than hexahedra or second-order tetrahedra [34]. Not to mention the fact that using high-order elements makes it easier to represent curved surfaces and to describe the mesh more finely with fewer elements [18].

In this context, we propose two main contributions: (i) a constraint formulation in the XPBD approach based on iso-parametric elements permitting to use higher-order finite elements in real-time; (ii) a solution improving convergence of unstable energies, such as Neo-Hooke, in constraint formulation. The interest of these two contributions, in dealing both with different topologies and using higher-order elements than linear ones for fast simulations, is demonstrated by our results.

In the following, Section 2 presents previous work on physical simulation of deformable bodies using constraint-based approaches. Section 3 gives a brief overview on soft body dynamics with a focus on the XPBD approach. Section 4 presents our method to integrate high order elements in XPBD to improve the precision of simulations. In addition, we provide detailed explanation on how to use elements of any topology. Also, we present our method that avoid convergence loss derived from using unstable energies, as for Neo-Hookean materials. Finally, Section 5 provides classical mechanical tests showing that our approach reaches better accuracy and lower runtime.

2 Previous Work

Simulation of deformable objects has been intensively studied over the last decades since the pioneering work of Terzopoulos [39]. Recently, Kim and Eberle [19] provide a survey on this topic.

Constrained Dynamics. Constraint based approach [37, 24] gained popularity due to its efficiency, simplicity and capacity to represent a large scale of different phenomena within the same framework. It has been used in many software (PhysiX, Bullet, Havok) [13] for real-time and interactive applications like game engines (Unity, Unreal, Godot), robotic and medical simulator [12, 7, 1]. Also, the compliant constraint formulation has been shown to allow robust Finite Element simulation [35, 40, 22, 15].

Quasi-Newton Methods. To achieve real-time running, constraint dynamics are often solved using quasi-Newton methods [21, 8, 23]. These solvers are based on implicit time integration [4] and a Newton solver while replacing the geometric stiffness with a fast approximation.

The counterpart is slower convergence as a function of the approximation and, in some cases, a deterioration in simulation stability [40]. However in practice, quasi-Newton methods permit greater accuracy with limited resources compared to a fully Newton solver.

XPBD. The Position Based Dynamics (PBD) [27] approach is a popular method for real time simulation. It has been applied on numerous types of simulations [6] due to its simplicity and robustness. The initial problems of iteration count and time-step dependent stiffness have been addressed within XPBD [23]. This is based on a quasi-Newton solver where the geometric stiffness computation is withdrawn in exchange for an error of $O(\Delta t^2)$ order, deteriorating its convergence rate. Macklin [25] showed that XPBD approach converges faster and yields more accurate results using small-time steps with only one iteration rather than a large time step with several iterations. In addition, an advantage of the XPBD method is its ability to handle rigid materials and extreme deformations while remaining stable. Thus, XPBD [25, 22] has demonstrated its robustness, accuracy and efficacy while keeping its simplicity. Moreover, this approach is interesting as it only requires the computation of constraint's gradient. Furthermore, as a local solver, no system assembly is required. This circumvents two of the main bottlenecks of high-order elements, which in other solvers considerably increase computational and memory costs.

High-Order Elements. Simulation methods are based on the discretization of the object into unit elements. Linear tetrahedral elements are predominant in computer graphics [5, 22, 19] due to their simplicity. However, a high Poisson's ratio (for near-perfect volume preservation) is accompanied by locking artifacts for linear tetrahedra as these elements do not have enough Degree of Freedom (DoF) to handle deformation while preserving their volume, resulting in artificial rigidity. They also highly depend on the quality of the topology. However, the energy-compliant formulation softens constraints [16, 15] avoiding the locking artifact.

The locking and discretization biases do not occur (or are considerably reduced) with high-order elements [33] that are gaining in popularity. Although expensive compared to their linear counterparts, they can naturally represent curved surfaces and thus depict complex shapes with fewer elements [18]. A consistent study proposed by Schneider *et al.* [34] compared the efficiency and accuracy of linear and quadratic tetrahedra (called P1, P2, respectively) as well as hexahedra (called Q1, Q2, respectively). They showed that P2 elements offer a better discretization error reduction for a given cost than other classical elements. Overall, the

use of high-order coarse meshes achieves similar results to those of dense tetrahedral meshes for a reduced simulation time [20, 14].

3 Reminder on Soft Body Dynamics

Before presenting our method, we continue by giving an overview of continuum mechanics for soft-body dynamics with a focus on the XPBD method.

Continuous materials. The deformation function ϕ describes the object's displacement from its initial position $X \in R^3$, defined in the rest state Ω_0 , to its current position $x(t) \in R^3$, defined in a deformed state Ω_t . The object's continuous displacement $u(X, t)$ is relied on deformation ϕ by:

$$\phi(X) = X + u(X, t) = x(X, t). \quad (1)$$

The deformation gradient \mathbb{F} corresponds to the Jacobian of the displacement function. It represents the mapping between Ω_0 and Ω_t :

$$\mathbb{F} = \frac{\partial \phi(X)}{\partial X} = \frac{\partial x}{\partial X}. \quad (2)$$

The strain tensor (noted $\epsilon(x)$) quantifies the object's deformation. It is defined according to the considered material. For example, the Green-Lagrange tensor is defined by $\epsilon(x) = \frac{1}{2}(\mathbb{F}^T \mathbb{F} - \mathbb{1})$ (see Table 2 in Annex A).

The energy density field function, noted $\Psi(x)$, measures the object's deformation. For an isotropic Saint-Venant Kirchhoff material, we have:

$$\Psi_{StVK}(x) = \frac{\lambda}{2} \text{tr}(\epsilon(x))^2 + \mu \text{tr}(\epsilon(x)^2), \quad (3)$$

where λ and μ are the Lamé coefficients characterizing the material stiffness. The total strain energy $U(x)$ of an object is then obtained by integrating the energy density $\Psi(x)$ over its domain Ω with:

$$U(x) = \int_{\Omega} \Psi(x) dX. \quad (4)$$

Considering that forces are derived from potential energy $U(x)$, the equation of motion governs the object's movement with $M \ddot{x} = -\nabla_x U^T(x)$.

Forces computation. The internal forces of an object correspond to an energy gradient:

$$f_{int}(x) = -\nabla_x U(x) = - \int_{\Omega} \frac{\partial \Psi(x)}{\partial x} dX. \quad (5)$$

Chain rule can then be applied to simplify their formulations and computation with:

$$f_{int}(x) = - \int_{\Omega} \frac{\partial \Psi(x)}{\partial \mathbb{F}} \frac{\partial \mathbb{F}}{\partial x} dX = - \int_{\Omega} \mathbb{P}(x) \frac{\partial \mathbb{F}}{\partial x} dX, \quad (6)$$

where $\mathbb{P}(x)$ corresponds to the first Piola-Kirchhoff stress tensor defined for this material by:

$$\mathbb{P}_{StVK}(x) = \lambda \text{tr}(\epsilon(x)) \mathbb{F} + 2 \mu \mathbb{F} \epsilon(x). \quad (7)$$

The chain rule on the energy density separates the material from object geometry. In the case of Finite Element Method (FEM), it allows any type of energy to be defined without knowing in advance on which geometric element it will be applied and *vice versa*.

Finite Element Method. The Finite Element Method (FEM) is a popular method based on the discretization of the continuous object into geometrical primitives (called elements) for which some shape functions N_i are associated. These functions are defined at each node i . Then, considering u_i the displacement at node i , the displacement can be evaluated at any position inside the element's domain with:

$$u(X) = \sum_i^n u_i N_i(X), \quad (8)$$

where n is the number of vertices in the element. If the shape functions are built in the rest state, they will be different for each element and difficult to define for complex elements. To avoid that, iso-parametric elements can be used. The method consists of defining the same reference state Ω_r for all elements. The map between any position x (defined in the deformed state) and the corresponding position r (in the reference state) corresponds to :

$$J(x, r) = \frac{\partial x}{\partial r} = \sum_i^n x_i \frac{\partial N_i(r)}{\partial r}^T, \quad (9)$$

where r_i, x_i are positions of node i in reference and deformed state, respectively. Therefore, the deformation gradient \mathbb{F} is defined as:

$$\mathbb{F} = \frac{\partial x}{\partial X} = \frac{\partial x}{\partial r} \frac{\partial r}{\partial X} = J(x, r) J(X, r)^{-1}. \quad (10)$$

Thus, the deformation gradient can be calculated for any element as long as the shape functions derivatives are known. Using this formulation, the energy gradient correspond to:

$$\frac{\partial \Psi(x)}{\partial x} = \frac{\partial \Psi(x)}{\partial \mathbb{F}} \frac{\partial \mathbb{F}}{\partial x} = \mathbb{P}(x) J(X, r)^{-T} \frac{\partial N(r)}{\partial r}. \quad (11)$$

The reference element is built in such way that numerical integration can be applied effectively. First, the forces integral is scaled to match the reference element size with:

$$f_{int}(r) = - \int_{\Omega_r} \frac{\partial \Psi(x)}{\partial x} |det(J(X, r))| dr. \quad (12)$$

Then, this enables the use of Gaussian quadrature to approximate the integrals. For polynomial functions, like shape functions, the analytical solution is found when enough integration points are used. The weight w_i of each integration point r_i is then combined with the scale factor to express the quadrature volume part $V_i = w_i |\det(J_i(X))|$. The weights and integration points coordinates are known beforehand. We obtain:

$$f_{int}(r) = - \sum_i^m V_i \mathbb{P}(x_i) J(X, r_i)^{-T} \frac{\partial N(r_i)}{\partial r}. \quad (13)$$

Using shape functions of linear tetrahedron's gives the same formulations of energy and forces presented by Bender *et al.* [5] (*a supplementary material is provided which gives a demonstration*).

XPBD. Macklin *et al.* [23] proposed the Extended Position Based Dynamics (XPBD) to improve the initial PBD approach in order to efficiently simulate arbitrary elastic energy. In constrained-dynamics, an energy potential function $U(x)$ can be expressed in terms of constraint vector $C(x)$ [40,23], with:

$$U_C(x) = \frac{1}{2} \alpha^{-1} C(x)^2, \quad (14)$$

where the compliance α is the inverse of the stiffness. As forces correspond to the negative gradient of the energy potential, they can be expressed as a combination of constraint functions and their gradients with:

$$f_{int}(x) = - \int_{\Omega} \frac{\partial \Psi(x)}{\partial x} dX = - \nabla C(x) \alpha^{-1} C(x). \quad (15)$$

Algorithm 1 XPBD simulation loop [25]

```

1:  $\Delta t_s = \Delta t_f / n_{substeps}$ 
2: while  $s < n_{substeps}$  do
3:   predict  $\tilde{x} = x_n + \Delta t_s v_n + \Delta t_s^2 M^{-1} f_{ext}(x_n)$ 
4:   while  $i < n_{it}$  do
5:     for all constraints do
6:       compute  $C(x)$  and  $\nabla C(x)$ 
7:       compute  $\Delta \lambda$  using Eq (17)
8:       compute  $\Delta x$  using Eq (16)
9:       update  $\lambda_{i+1} \leftarrow \lambda_i + \Delta \lambda$ 
10:      update  $x_{i+1} \leftarrow x_i + \Delta x$ 
11:     end for
12:      $i \leftarrow i + 1$ 
13:   end while
14:   update velocities  $v_{n+1} \leftarrow \frac{1}{\Delta t_s} (x_{n+1} - x_n)$ 
15:    $s \leftarrow s + 1$ 
16: end while

```

Algorithm 1 gives the simulation's loop involved by XPBD [25] approach. Each XPBD simulation loop is subdivided in sub-steps (denoted $n_{substeps}$) with smaller time step. Each sub-step starts by using the explicit Euler integration scheme to predict positions according

to velocities. Let's note that these latter were obtained by only considering the external forces (and not the internal ones). Then, solver iterations (denoted n_{it}) are carried out to project positions with a correction Δx . This correction is obtained by iterating over a set of previously defined constraints, with:

$$\Delta x = M^{-1} \nabla C(x)^T \Delta \lambda. \quad (16)$$

Thus, the correction applied to particle i is defined through the direction of constraint gradient $\nabla_{x_i} C(x_i)$. Its magnitude depends on its inverse mass m_i^{-1} and a scalar Lagrange multiplier $\Delta \lambda_j$ defined for constraint j . If we consider Δt the time step size and $\tilde{\alpha} = \alpha / \Delta t^2$ a time-step scaled compliance parameter, it gives:

$$\Delta \lambda_j = \frac{-C_j(x_i) - \tilde{\alpha}_j \lambda_{ij}}{\nabla C_j(x) M^{-1} \nabla C_j(x)^T + \tilde{\alpha}_j}. \quad (17)$$

For each iteration, position and Lagrange multiplier are updated using $\lambda_{i+1} = \lambda_i + \Delta \lambda$ and $x_{i+1} = x_i + \Delta x$. Setting compliance to zero corresponds to original PBD [27] formulation for infinitely stiff constraint. Remember that in practice, as shown by Macklin [25], there are several sub-steps (with a smaller time step) but only one solver iteration. Reducing the time step size leads to a quadratic reduction in error, making it much more efficient.

Stable energies. To get a correct behavior, no forces should appear when the object is not deformed. This is naturally the case when using constitutive laws such as Saint-Venant Kirchhoff or Hooke. But this is no more true if we consider a Neo-Hookean material. In this context, Smith [36] proposed a method for correcting these unstable energies. To stabilize the energy, the first Piola-Kirchhoff tensor must be checked to be a null matrix if $\mathbb{F} = \mathbb{1}$. If this is not the case, a correction can be deduced. Smith put this correction into practice for a Neo-Hookean material, involving this following energy's definition :

$$\Psi_{SNH} = \frac{\mu}{2} (I_2 - 3) + \frac{\lambda}{2} (I_3 - \gamma)^2, \quad (18)$$

where $\gamma = 1 + \mu \lambda^{-1}$ is a constant factor added to the energy's hydrostatic term; I_2, I_3 correspond to their set of invariants defined by $I_2 = \text{tr}(\mathbb{F}^T \mathbb{F})$, $I_3 = \det(\mathbb{F})$.

In 2021, Macklin [22] proposed to take into account this corrected Neo-Hookean energy in the PBD formulation. Macklin split the energy in two sub-energy terms to obtain a constraint according to the Lamé coefficients (λ, μ) , as a single stiffness per constraint can be defined. But these constraints no longer ensure that their corresponding first Piola-Kirchhoff tensor is a null matrix if $\mathbb{F} = \mathbb{1}$ (see Table 2). In the end, the deformation gradient is not zero in the resting state resulting into unwanted corrections at rest.

4 Our Method

We propose a new method for high order elements in the XPBD framework. The constraint is derived from the definition of elastic energy potential expressed in terms of iso-parametric elements. In addition, we define a stable strain energy constraint applied on Neo-Hookean materials. For this purpose, the constraint is constructed in such a way that the stability condition presented by Smith [36] is also valid for constraints.

4.1 Iso-parametric element constraint

In the XPBD approach, energies are defined as compliant constraint function. From Eq.(14), we have:

$$C(x) = \left(\int_{\Omega} 2 \alpha \Psi(x) dX \right)^{1/2}. \quad (19)$$

By defining $\tilde{\Psi}(x) = 2 \alpha \Psi(x)$, we obtain

$$C(x) = \left(\int_{\Omega} \tilde{\Psi}(x) dX \right)^{1/2}, \quad (20)$$

with the following constraint gradient

$$\nabla C(x) = \frac{1}{2 C(x)} \int_{\Omega} \frac{\partial \tilde{\Psi}(x)}{\partial x} dX. \quad (21)$$

As the stiffness α can be expressed by several terms according to the constitutive law, the constraint can also be expressed by several terms [22,15]. For Saint-Venant Kirchhoff's law remembered in Eq.(3), its two stiffness components (μ, λ) can thus be linked to its strain energy terms $(\tilde{\Psi}_A, \tilde{\Psi}_B)$ (see Table 2) involving two terms for its constraint.

Now that we have a continuous formulation, we can apply finite element discretization (as in Eq.(12-13)). The function is mapped onto reference elements, then integrated with Gaussian quadrature. We obtain the following formulations (with m being the number of Gaussian quadrature nodes):

$$C(x) = \left[\sum_i^m V_i \tilde{\Psi}(x_i) \right]^{1/2}, \quad (22)$$

$$\nabla C = \frac{1}{2 C(x)} \sum_i^m V_i \tilde{\mathbb{P}}(x_i) J(X, r_i)^{-T} \frac{\partial N(r_i)}{\partial r}. \quad (23)$$

To sum up, Algorithm 2 details how we now compute the constraint and its gradient, which are used in line 6 of Algorithm 1. Moreover, Table 2 gives formulations of $\tilde{\Psi}$ and $\tilde{\mathbb{P}}$ considering three constitutive laws. And the supplementary material of the article remembers the shape functions formulations and how their derivatives are computed.

Algorithm 2 Iso-parametric element constraint

```

1:  $C = 0; \nabla C = 0;$ 
2: for all  $i$  in QuadraturePoints do
3:   compute  $J(x, r_i)$  using Eq.(9)
4:   compute  $\mathbb{F}_i = J(x, r_i) J(X, r_i)^{-1}$ 
5:   compute  $C += \tilde{\Psi}(\mathbb{F}_i) V_i$ 
6:   compute  $\nabla C += \tilde{\mathbb{P}}(\mathbb{F}_i) \frac{\partial N(r_i)}{\partial r} V_i$ 
7: end for
8:  $C = |C|^{1/2}; \nabla C = (2 C)^{-1} \nabla C;$ 

```

At the end, these formulations can be used for any type of element. We only have to define shape functions partial derivatives $\partial N(r_i)/\partial r$ for the corresponding element, which is used for the Jacobian and constraint's gradient calculations. Then, as in previous work, we can use any type of elastic material by defining energy density function $\tilde{\Psi}(F_i)$ and first Piola-Kirchhoff stress tensor $\tilde{\mathbb{P}}(F_i)$. Let's note that these two formulas are both topology-independent. They only require the deformation gradient \mathbb{F} which is a 3×3 matrix.

Furthermore, the quadrature volume term V_i and the initial position Jacobian $J(X, r_i)$ can be pre-computed at initialization for each element, regardless of the element types (topology and order) using Algorithm 3. In addition, the partial derivatives of the shape functions are identical for all elements of the same type and are evaluated at a predefined set of quadrature points. It can therefore be evaluated only once and shared by all elements of the same type. As a result, we can greatly reduce the number of operations to be performed at a small memory cost.

Algorithm 3 Iso-parametric element constraint (Init)

```

1:  $w$  : quadrature weights
2:  $r$  : quadrature coordinates
3:  $\partial N(r)/\partial r$  : shape functions partial derivatives
4: for all  $i$  in QuadraturePoints do
5:    $J_i(X, r_i) = \sum_j^n X_j (\partial N_j(r_i)/\partial r)^T$ 
6:   Save  $V_i = |\det(J_i(X, r_i))| w_i$ 
7:   Save  $J_i(X, r_i)^{-1}$ 
8: end for

```

Let's note that the constraint and its gradient are not well-defined when the constraint is close to 0 corresponding to very small corrections. Therefore, we can assume that it can be ignored when the value of the constraint is sufficiently small, *i.e.* close to the floating-point arithmetic resolution. Moreover, by definition, a compliant constraint can only represent non-negative energies. If the energy can take negative values, like for Neo-Hooke's stable material, Macklin [22] proposed adding a constant factor in order to shift the function to positive half-space. This constant disappears in the energy gradient, giving a *force equivalent energy*.

4.2 Stable constraint

We now propose a new method to build strain energy constraints that ensure stability at rest. This means that no corrections will be applied when no deformation is measured (*i.e.* $\mathbb{F} = \mathbb{1}$). A natural solution is to ensure that the correction proposed by Smith [36] is applied directly to the unstable term of the energy. At constraint level, this means having the same stiffness parameter. As an example, we consider the developed version of stable Neo-Hookean material (Eq.(18)). This permits to group together the correction and the term to be corrected under the same stiffness parameter, giving:

$$\Psi_{SNH_2} = \frac{\mu}{2}(I_2 - 3) - \mu(I_3 - 1) + \frac{\lambda}{2}(I_3 - 1)^2 \quad (24)$$

$$\begin{aligned} &= \frac{\mu}{2}(I_2 - 2I_3 - 1) + \frac{\lambda}{2}(I_3 - 1)^2 \\ &= \frac{\mu}{2}\tilde{\Psi}_A + \frac{\lambda}{2}\tilde{\Psi}_B \end{aligned} \quad (25)$$

Let's define the Piola-Kirchoff stress tensor of Smith's invariants [36] remembering that:

$A^{-1} = \det(A)^{-1} \text{com}(A)^T$. We have:

$$\frac{\partial I_2}{\partial \mathbb{F}} = 2 \mathbb{F}; \quad \frac{\partial I_3}{\partial \mathbb{F}} = \det(\mathbb{F})(\mathbb{F}^{-1})^T = \text{com}(\mathbb{F}). \quad (26)$$

Next, we define $\mathbb{F} = [f_0, f_1, f_2]$ in the same way as [22, 19] to compute the co-factor matrix of the deformation gradient, which gives $\text{com}(\mathbb{F}) = [f_1 \times f_2, f_2 \times f_0, f_0 \times f_1]$. In the end, the Piola-Kirchoff stress tensor of $\tilde{\Psi}_A$ and $\tilde{\Psi}_B$ are:

$$\tilde{P}_A = \frac{\partial \tilde{\Psi}_A}{\partial \mathbb{F}} = \frac{\partial I_2}{\partial \mathbb{F}} - 2 \frac{\partial I_3}{\partial \mathbb{F}} - \frac{\partial 1}{\partial \mathbb{F}} = 2(\mathbb{F} - \text{com}(\mathbb{F})); \quad (27)$$

$$\begin{aligned} \tilde{P}_B &= \frac{\partial \tilde{\Psi}_B}{\partial \mathbb{F}} = \frac{\partial}{\partial \mathbb{F}}(I_3 - 1)^2 = 2(I_3 - 1) \frac{\partial}{\partial \mathbb{F}}(I_3 - 1) \\ &= 2(I_3 - 1) \text{com}(\mathbb{F}). \end{aligned} \quad (28)$$

These two Piola-Kirchoff stress tensors comply with Smith's stability condition as they are a null matrix when $\mathbb{F} = \mathbb{1}$. This property ensures the stability of the constraints gradients defined in Eq.(23). Moreover, Section 5 shows that this stability condition greatly improves the convergence rate of the XPBD approach.

5 Experimental results

We have implemented our approach using high-order elements inside the XPBD framework into our own C++ software. All experiments were run on an Intel I5-10300H CPU with 16 GB RAM. Rendering have been made using Paraview [2]. Let's note that display time was not taken into account in our measurements.

The aim of our experiments is to show that, although XPBD makes major simplifications, this has little effect on the properties of high-order elements and

remains stable. In particular, we want to prove that conclusions made by [34] are still valid in XPBD. Normally, higher-order elements are assumed to be faster than linear elements for a given accuracy. To highlight this, we need to study the failure cases of linear tetrahedra and compare them with higher-order elements. Let's remember that linear elements suffer from two main artifacts. Firstly, their accuracy is highly dependent on topology, as shown by [33]. Secondly, they exhibit a locking artefact creating artificial rigidity when volume conservation is almost perfect ($\nu \approx 0.5$). Let's recall that one of the main features of XPBD is its ability to represent this type of material thanks to its compliant stress formulation coupled to the Neo-Hooke material [22]. Moreover, we will show that, when the mesh is not well constructed, high Poisson's ratios still strongly deteriorate the result for linear elements.

5.1 Mechanical validation tests

We first carried out conventional mechanical tests (tension, compression, torsion and bending) using several types of elements to highlight the impact of the type of elements chosen on the quality of results and performances. We used the Neo-Hooke constitutive law (see Section 4.2) to simulate quasi-perfect volume conservation material. Similarly to [34], we considered as reference, results obtained on linear hexahedral meshes (see Table 5 for details) computed using a semi-explicit Euler approach with a small step size ($dt = 5.6e^{-5}s$).

In the following, we call P_1 , P_2 and P_3 , the meshes composed of linear, quadratic and cubic tetrahedra, respectively and Q_1 (or Ref) those composed of linear hexahedra. Moreover, as linear tetrahedra are highly dependent on the quality of the mesh, two types of mesh (called biased and unbiased) were used to highlight the impact of a topological bias in the mesh. For this purpose, all beam meshes were built from a hexahedral regular mesh. Then, biased meshes correspond to a subdivision of each hexahedron into 2 prisms, then each prism into 3 tetrahedra. Thus, prism subdivision creates a regular pattern that makes a global direction emerge. While, unbiased meshes were directly obtained by splitting each hexahedron into 5 tetrahedra. For all details, Table 3 sums up parameters used in our tests and Table 4 gives information on meshes used, including computational times. Simulations were carried out up to the equilibrium state.

Tension. For the first mechanical test, we considered biased beams having 637 vertices. The two sides of beams were fixed and then displaced in opposite direction until beams reached twice their initial length. Young's modulus was fixed to 100 KPa, Poisson ratio to 0.499 and there was no gravity. Fig. 2 presents the

simulation results for P_1 , P_2 , P_3 meshes with respect to the *Ref* solution corresponding to Q_1 mesh. P_1 beam did not converge to the right result and followed the preferred direction induced by the biased mesh. High-order elements matched closely the reference and can be used to simulate objects with strong volume preservation constraints.

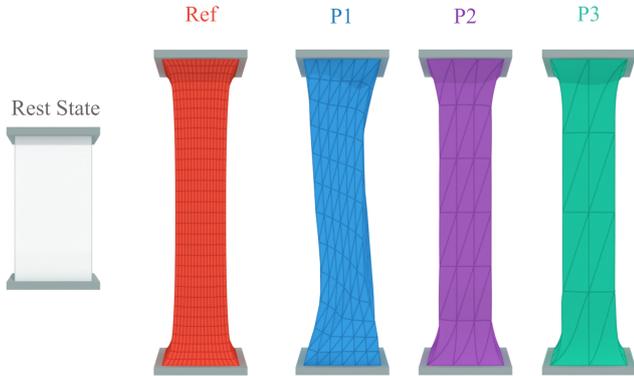


Fig. 2: Tension test performed on P_1 , P_2 and P_3 biased beams having 637 vertices using a Poisson ratio ν set to 0.499. P_1 elements did not behave correctly and moved along the privileged diagonal while high-order elements matched to the reference corresponding to Q_1 mesh.

Compression. To highlight the impact of a Poisson ratio close to 0.5 on the behavior of linear tetrahedra, we now performed a compression test on biased cube meshes (having 12,001 elements) for ν set to 0.49, 0.499 and 0.4999. The cubes were crushed by half. Young's modulus was fixed to 100KPa and there was no gravity. Fig. 3 presents the displacement obtained in the middle of the cubes for P_1 , P_2 , P_3 (top to bottom) for the different values of ν (left to right). The behavior of linear elements was deteriorated. And higher the Poisson coefficient was, more the privileged direction was visible. Behavior of the high-order elements stays unchanged whatever the Poisson ratio, further demonstrating their ability to simulate incompressible objects.

Torsion. To go further on the comparison between elements behavior, we performed a torsion's test. It consists of rotating the top side of beams by 180 degrees, while the lower side was fixed. Poisson ratio was set to 0.35. Young modulus was equal to 1 MPa. There was no gravity. Similar to [34], Fig. 5 shows the absolute difference between the angle of each cross-section and a linear interpolation. Moreover, to highlight results in comparison to Q_1 , Fig. 4 shows a cut through the middle of the P_1 , P_2 , P_3 biased or unbiased beams. We observe that unbiased-mesh P_1 beam gave a good overall approximation but reconstructed poorly the reference results, while high-order results were indissociable from the reference.

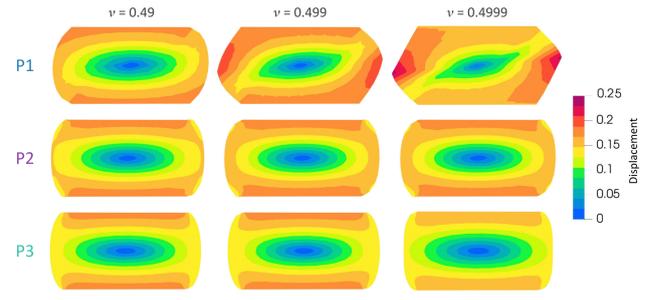


Fig. 3: Visualization of the displacement norm for a compression test on P_1 , P_2 and P_3 cubes considering three values of Poisson ratios (0.49, 0.499, 0.4999). P_1 followed the privileged direction, while high-order elements had a consistent behavior.

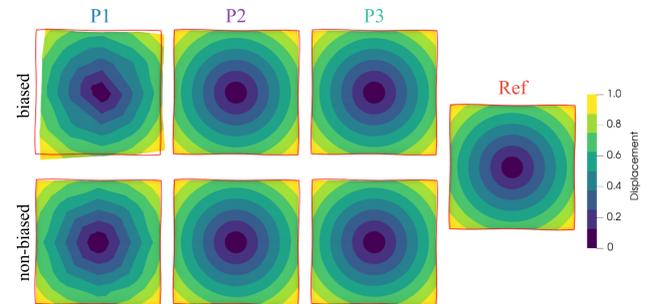


Fig. 4: Torsion test. Visualization of the cross-section of biased and unbiased beams colored according to the displacement. P_1 did not match reference silhouette (in red) or was a rough approximation of the reference, while high-order meshes closely matched the reference for biased or unbiased meshes.

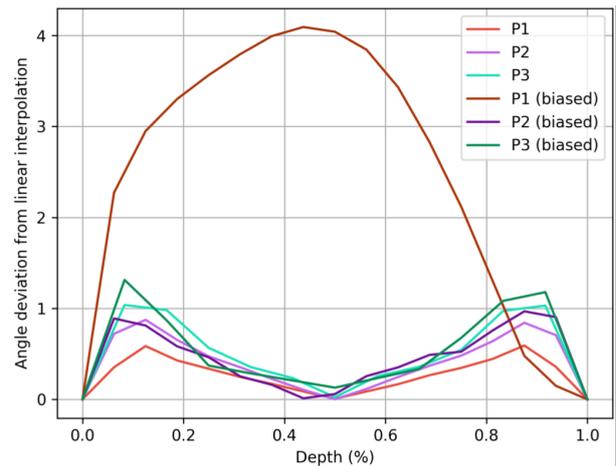


Fig. 5: Cross-section angle deviation from a linearly interpolated angle along the depth of the beam for P_1 , P_2 and P_3 elements considering biased and non-biased meshes. Linear tetrahedra exhibit a meaningful change compared to other elements when the mesh was biased.

Torsion of irregular meshes. We performed a second torsion test by rotating both sides of irregularly meshes beams by 180 degrees. Poisson ratio was set to 0.49. Young modulus was equal to 100KPa. There was no gravity. Moreover, we compared the time required to achieve the same deformation for P_1, P_2, P_3 beams having 1,279 elements. Fig. 6 shows our results. As expected, higher-order elements were more costly with 314 ms. However, to achieve a similar result with P_1 elements, a refined mesh had to be used with 64 times more elements involving a simulation of 2184ms. Thus, P_1 loses its advantage in terms of efficiency. Overall, we can expect a speed-up of at least one order of magnitude with higher-order elements for the same precision.

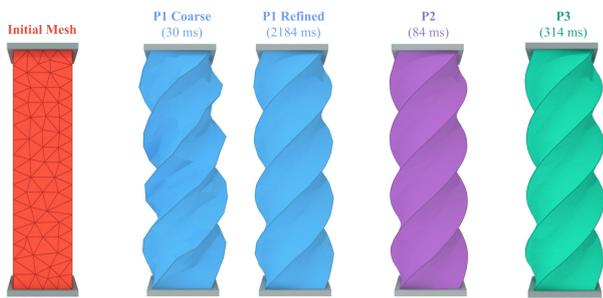


Fig. 6: Torsion of P_1, P_2, P_3 irregular beams having 1,279 elements. High-order elements were more time-consuming, but they achieved the same deformation quality than refined P_1 beam (with 64 times more elements) with a speed-up of one order of magnitude.

Bending. For a fair comparison, we performed a bending test with gravity considering a relatively small Poisson ratio ($\nu = 0.35$) for unbiased P_1, P_2, P_3 beams. The Young modulus was equal to 1MPa. The errors shown in Table 1 correspond to the difference in displacement between the point at the center of the end of each beam and the same point on the reference solution. Fig. 7 shows the beams at rest, but colored according to the amplitude of displacement relative to the initial state. Even if the P_1 mesh was in the right configuration, it was still far from the reference solution. By increasing the number of elements dramatically, it became closer but still did not reach the same precision as high-order elements with coarser meshes.

Analysis. We tested high-order elements through classical mechanical tests for some failure cases of linear tetrahedra. We found that, for the same number of vertices, higher-order elements produced visually satisfactory results, closer to the reference solution and faster than linear elements. The use of the XPBD approach does not therefore change this well-known result in mechanics. In addition, higher-order elements

require fewer elements for the same number of vertices than their linear counterpart. A difference in computation time for a given error can therefore be expected, particularly with linear tetrahedra.

Beam	#vertices	#elements	time (ms)	e (m)
P_1	425	1,280	43.8	$3.9e^{-1}$
	2,417	10,240	346	$1.8e^{-1}$
	16,625	81,920	3,055	$5.6e^{-2}$
P_2	393	160	24.8	$8.0e^{-3}$
P_3	573	80	29.4	$7.7e^{-3}$

Table 1: Bending test. Displacement error of the middle point at the end of each beam compared to the reference solution obtained for a beam composed of Q_1 elements.

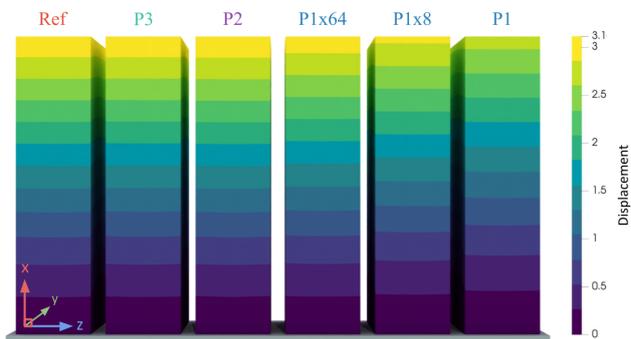


Fig. 7: Top view of the beams considered in the unreformed state colored by the displacement induced by the bending test. A good solution must have all colors aligned with the reference. The greatest offset was for the beam composed of 1,280 P_1 elements.

5.2 Scaling

We now compared P_1, P_2, P_3 elements in terms of performance and memory to observe if we come back to the fact that high-order elements cost more than linear elements or not, due to the need of fewer elements.

For this purpose, we considered a unit cube fixed at its base with only gravity and 50 sub-step. We put $E = 1\text{MPa}$, $\nu = 0.49$ and $\rho = 1000\text{kg/m}^3$. We increased the number of elements until reaching 10,000 vertices for each mesh (corresponding to 55,566 elements for P_1 , 7,986 for P_2 and 2,058 for P_3).

Fig. 8 (up) gives times and Fig. 8 (down) gives required memory. We observe that linear tetrahedra were more expensive and took more memory than higher-order for the same number of vertices. It comes from the fact that XPBD approach avoids the main bottlenecks of high-order elements having only first derivatives to compute for constraint gradients which reduces their cost. Moreover, this cost is mostly due to the number of needed integration. So even if a P_2 element requires 4 integrations instead of 1 for P_1 , we need 8 linear tetrahedra to obtain the same number of vertices which ex-

plain the factor 2 between P_1 and P_2 for the same number of vertices. Thus, the scaling changes from classical approach and makes high-order elements cost less in memory and calculation time for the same number of vertices compared to their linear counter-part. This result demonstrates the interest of proposing the use of higher-order elements in this approach.

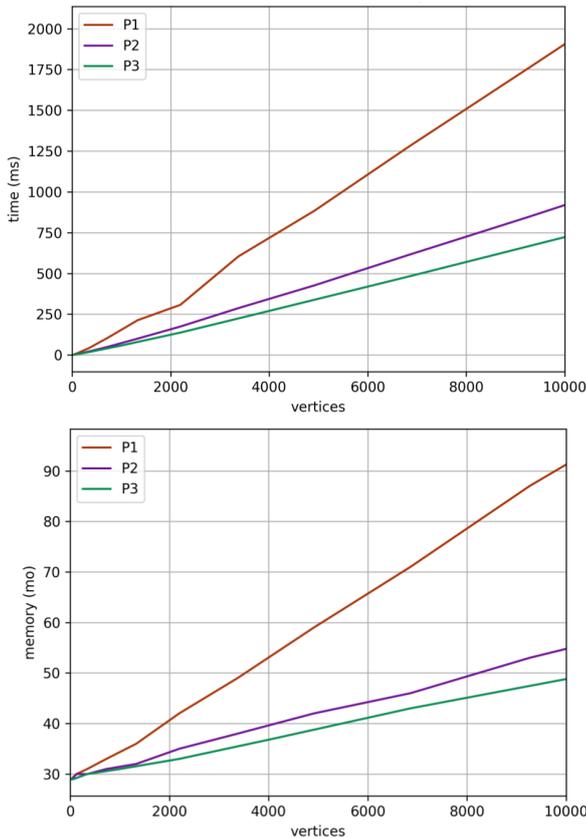


Fig. 8: Comparison in (up) time and (down) memory of P_1 , P_2 , P_3 elements by progressively increasing the number of vertices of a cube subjected to gravity.

5.3 Convergence

We performed a second bending test with P_2 elements to compare our stable Neo-Hooke constraints with those proposed by Macklin [22] against a reference solution obtained by performing a force based FEM simulation considering a stable Neo-Hookean material [36].

Fig. 9 shows results obtained with 20, 50 and 100 sub-steps. Our solution (purple) gives results close to the FEM reference solution (red outline) with only 20 sub-steps. Macklin’s constraints (blue) produce poor solution and require many more steps to converge. Even with 100 sub-steps, the displacement error remains significant. As anticipated, the proposed stability condition greatly improves the convergence rate of the XPBD approach.

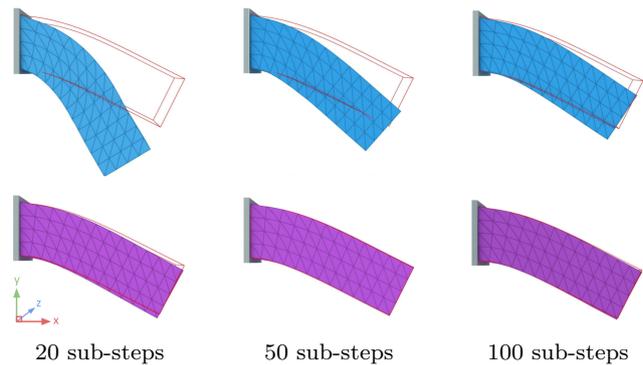


Fig. 9: Comparison between P_2 elements using Neo-Hooke’s constraint (in blue) and our approach (in purple). The red outline corresponds to a detailed FEM solution considering a stable Neo-Hookean material. Stable constraint requires fewer sub-ste to achieve a result close to reference.

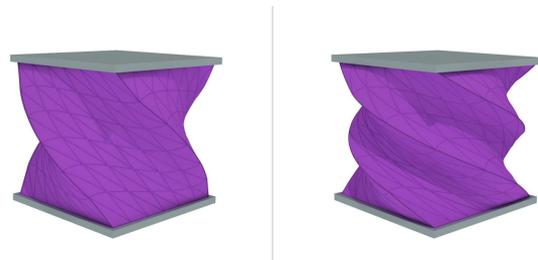


Fig. 10: The mesh is an unit cube made 2592 P_2 elements. The upper part of the cube has two consecutive twists of 90° and the lower part is fixed. Even in this extreme scenario, the simulation goes without inversion nor instabilities.

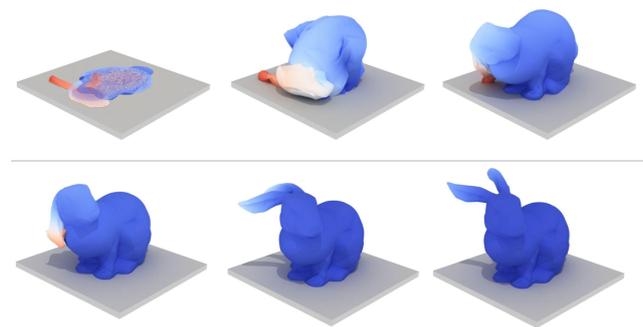


Fig. 11: Shape recovery of a crushed bunny mesh discretized with P_3 elements and colored by its displacement (small shift in blue to large shift in red). With extreme deformation and inverted elements, the rabbit unfolds and regains its rest shape.

5.4 Robustness

One of the problems with high-order finite elements is that they are more prone to inversion and may not recover. To check the robustness of our method in the face of this problem, we carried out two experiments. The first one consists of a fully inverted mesh and checking whether it can recover its original shape. The second one consists of an extreme twist similar to [36, 22], where element inversion is difficult to avoid.

Stanford bunny test. We generated a Stanford bunny with P_3 elements using the Jiang high-order tetrahedral coarsing library [18]. The bunny is fixed on the plate and then flattened at the first frame resulting into a fully inverted mesh. After 1 s of simulation time, the rabbit recovered its shape quickly and robustly. Fig. 11 shows this experiment where colors correspond to displacement's magnitude.

Twist test. Fig. 10 shows our results reproducing the test proposed by Smith [36] on a unit cube initially composed of 6^3 hexahedra, where each cell was subdivided into 12 P_2 elements. Then, the bottom side of the mesh is fixed while the top side perform two consecutive twists by 90° . The simulation was carried out with 50 sub-steps. There is no element inversions nor instabilities which illustrates the robustness of the simulation.

Analysis. Our method stays robust despite the tendency of high-order elements to inversion. However, we have shown that, in practice, their use in the XPBD approach makes it possible to deal with complex situations while remaining stable and recover from highly inverted mesh.

6 Conclusion and Perspectives

We presented our new method, which easily takes into account all types of topology (tetrahedra, hexahedra, prisms, pyramids), as well as high-order finite elements within the framework of the XPBD (Extended Position Based Dynamics) formulation, which has proved its worth in terms of efficiency and accuracy.

Our approach is based on the formulation of constraints for iso-parametric elements from their force and energy formulations. The use of higher-order finite elements has more accurate results than linear elements with coarse meshes. They also reduce the computation time required for simulations, by reducing the number of elements needed. In addition, we have proposed a stable constraint to consider the Neo-Hookean material, enabling us to obtain accurate results with a reduced number of iterations.

Through various mechanical tests (bending, tension, torsion, compression), we validated our approach and proved its advantages in terms of accuracy and calculation time. We also illustrated its robustness.

Overall, our method provides a simple, stable and efficient generic simulation. This framework has been developed for elastic and hyperelastic materials. Extending it to more complex behaviors such as anisotropy or plasticity, should be a great step forward. In particular, plastic deformations have been little studied in computer graphics and, to our knowledge, never with high-order elements. For our future work, we would like to implement these extensions and propose a fix for high stiffness artifacts.

A Appendix

For better readability,

- Table 2 recalls some well-known energy of deformation formulations for usual constitutive laws and associated Piola-Kirchhof tensors;
- Table 3 sums up the parameters of our experiments;
- Table 4 gives information about meshes used;
- Table 5 gives information about meshes used for the reference solution.

References

1. Adagolodjo, Y., Goffin, L., De Mathelin, M., Courteu-cuisse, H.: Robotic insertion of flexible needle in deformable structures using inverse finite-element simulation. *IEEE Transactions on Robotics* **35**(3), 697–708 (2019). DOI 10.1109/TRO.2019.2897858
2. Ahrens, J., Geveci, B., Law, C.: *Visualization Handbook*, chap. ParaView: An End-User Tool for Large Data Visualization, pp. 717–731. Elsevier Inc., Burlington, MA, USA (2005). URL <https://www.sciencedirect.com/book/9780123875822/visualization-handbook>
3. Andrews, S., Erleben, K., Ferguson, Z.: Contact and friction simulation for computer graphics. In: *ACM SIGGRAPH 2022 Courses, SIGGRAPH '22*. Association for Computing Machinery, New York, NY, USA (2022). DOI 10.1145/3532720.3535640. URL <https://doi.org/10.1145/3532720.3535640>
4. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, p. 43–54. Association for Computing Machinery, New York, NY, USA (1998). DOI 10.1145/280814.280821. URL <https://doi.org/10.1145/280814.280821>
5. Bender, J., Koschier, D., Charrier, P., Weber, D.: Position-based simulation of continuous materials. *Computers & Graphics* **44**, 1–10 (2014). DOI <https://doi.org/10.1016/j.cag.2014.07.004>
6. Bender, J., Müller, M., Macklin, M.: A Survey on Position Based Dynamics. In: A. Bousseau, D. Gutierrez (eds.) *EG 2017 - Tutorials*. The Eurographics Association (2017). DOI 10.2312/egt.20171034
7. Berndt, I., Torchelsen, R., Maciel, A.: Efficient surgical cutting with position-based dynamics. *IEEE Computer Graphics and Applications* **37**(3), 24–31 (2017). DOI 10.1109/MCG.2017.45
8. Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph.* **33**(4) (2014). DOI 10.1145/2601097.2601116

	Saint-Venant Kirchhoff	Neo-Hooke [22]	Developped Neo-Hooke
Ψ	$\mu \operatorname{tr}(\epsilon(x)^2) + \frac{\lambda}{2} \operatorname{tr}(\epsilon(x))^2$ $\epsilon(x) = \frac{1}{2}(\mathbb{F}^T \mathbb{F} - I)$	$\frac{\mu}{2} (I_2 - 3) + \frac{\lambda}{2} (I_3 - \gamma)^2$ $\gamma = 1 + \mu \lambda^{-1}, I_2 = \operatorname{tr}(\mathbb{F}^T \mathbb{F}), I_3 = \det(\mathbb{F})$	$\frac{\mu}{2} (I_2 - 2I_3 - 1) + \frac{\lambda}{2} (I_3 - 1)^2$ $I_2 = \operatorname{tr}(\mathbb{F}^T \mathbb{F}), I_3 = \det(\mathbb{F})$
$\tilde{\Psi}_A$	$2 \operatorname{tr}(\epsilon(x)^2)$	$\operatorname{tr}(\mathbb{F}^T \mathbb{F}) - 3$	$I_2 - 2I_3 - 1$
$\tilde{\Psi}_B$	$\operatorname{tr}(\epsilon(x))^2$	$(\det(\mathbb{F}) - \gamma)^2$	$(I_3 - 1)^2$
$\tilde{\mathbb{P}}_A(\mathbb{F})$	$2 \mathbb{F} \epsilon(x)$	$2 \mathbb{F}$	$2(\mathbb{F} - \operatorname{com}(\mathbb{F}))$
$\tilde{\mathbb{P}}_B(\mathbb{F})$	$2 \operatorname{tr}(\epsilon(x)) \mathbb{F}$	$2 (\det(\mathbb{F}) - \gamma) \operatorname{com}(\mathbb{F})$	$2(I_3 - 1) \operatorname{com}(\mathbb{F})$

Table 2: Formulations of the Ψ strain energy, its various terms for the constraints formulations (with $\tilde{\Psi} = 2 \alpha \Psi$) and their associated Piola-Kirchhof tensors for different constitutive laws.

Experiment	picture	size	gravity	n _{it}	n _{substeps}	ρ	ν	E
Torsion (teaser)	Fig. 1	(3,1,1)	no	1	30	1,000	0.49	1×10^6
Tension	Fig. 2	(3,1,1)	no	1	40	100	0.499	1×10^5
Compression	Fig. 3	(1,1,1)	no	1	50	1,000	{0.49, 0.499, 0.4999}	1×10^5
Torsion	Fig. 5, 4	(4,1,1)	no	1	50	1,000	0.35	1×10^6
Torsion of irregular meshes	Fig. 6	(1,1,4)	no	1	30	100	0.49	1×10^5
Bending	Fig. 7	(4,1,1)	(0, -9.8, 0)	1	50	1,000	0.35	1×10^6
Stable constraint	Fig. 9	(3,1,1)	(0, -9.8, 0)	1	{20, 50, 100}	100	0.49	1×10^5
Bunny	Fig. 11	(1,1,1)	no	1	50	1,000	0.49	1×10^6
Hard twist	Fig. 10	(1,1,1)	no	1	50	1,000	0.49	1×10^5

Table 3: Summary of experimental parameters: size corresponds to mesh's size in $m \times m \times m$; ρ is the density in kg/m^3 ; ν is the Poisson's ratio; E is the Young modulus in Pa; n_{it} is the number of Gauss-Seidel solver iteration; $n_{substeps}$ is the number of sub-step per XPBD simulation loop (see Algorithm 1).

Experiment	#vertices			#elements			time (ms)		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
Mesh type									
Torsion (teaser)	931	850	835	3,240	405	120	112	56	45
Tension (biased)	637	637	637	2,592	324	96	79	41	32
Compression (biased)	2,331	17,261	56,791	12,001	12,001	12,001	677	1,984	6,465
Torsion (biased)	425	425	637	1,536	192	96	51	25	35
Torsion (unbiased)	425	393	573	1,280	160	80	53	28	30
Torsion of irregular meshes	{356;15,779}	2,252	6,967	{1,279;81,793}	1,279	1,279	{30;2,184}	89	314
Bending (unbiased)	425	393	573	1,280	160	80	44	25	29
Bunny	-	-	7,842	-	-	1,487	-	-	524

Table 4: Summary for experiments: number of vertices and elements of the meshes used; computation times in ms.

Experiment	resolution
Tension	64x16x16
Bending	64x16x16
Torsion	16x32x32
Stable Constraint	48x16x16

Table 5: Resolution of meshes used for reference's tests.

- Chentanez, N., Alterovitz, R., Ritchie, D., Cho, L., Hauser, K.K., Goldberg, K., Shewchuk, J.R., O'Brien, J.F.: Interactive simulation of surgical needle insertion and steering **28**(3) (2009). DOI 10.1145/1531326.1531394. URL <https://doi.org/10.1145/1531326.1531394>
- Chentanez, N., Alterovitz, R., Ritchie, D., Cho, L., Hauser, K.K., Goldberg, K., Shewchuk, J.R., O'Brien, J.F.: Interactive simulation of surgical needle insertion and steering. ACM Trans. Graph. **28**(3) (2009). DOI 10.1145/1531326.1531394. URL <https://doi.org/10.1145/1531326.1531394>

- Courtecuisse, H., Allard, J., Kerfriden, P., Bordas, S.P., Cotin, S., Duriez, C.: Real-time simulation of contact and cutting of heterogeneous soft-tissues. Medical Image Analysis **18**(2), 394–410 (2014). DOI <https://doi.org/10.1016/j.media.2013.11.001>
- Duriez, C., Guébert, C., Marchal, M., Cotin, S., Grisoni, L.: Interactive simulation of flexible needle insertions based on constraint models. In: G.Z. Yang, D. Hawkes, D. Rueckert, A. Noble, C. Taylor (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009, pp. 291–299. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
- Erez, T., Tassa, Y., Todorov, E.: Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 4397–4404 (2015). DOI 10.1109/ICRA.2015.7139807
- Ferguson, Z., Jain, P., Zorin, D., Schneider, T., Panozzo, D.: High-order incremental potential contact for elastodynamic simulation on curved meshes. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings, SIGGRAPH

- '23. ACM (2023). DOI 10.1145/3588432.3591488. URL <http://dx.doi.org/10.1145/3588432.3591488>
15. Frâncu, M., Asgeirsson, A., Erleben, K., Rønnow, M.J.L.: Locking-proof tetrahedra. *ACM Trans. Graph.* **40**(2) (2021). DOI 10.1145/3444949
 16. Frâncu, M., Moldoveanu, F.: Unified Simulation of Rigid and Flexible Bodies Using Position Based Dynamics. In: F. Jaillet, F. Zara (eds.) *Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association (2017). DOI 10.2312/vriphys.20171083
 17. Golec, K., Palierne, J.F., Zara, F., Nicolle, S., Damiand, G.: Hybrid 3D mass-spring system for simulation of isotropic materials with any Poisson's ratio. *The Visual Computer* **36**(4), 809–825 (2020). DOI 10.1007/s00371-019-01663-0. URL <https://hal.science/hal-02124471>
 18. Jiang, Z., Zhang, Z., Hu, Y., Schneider, T., Zorin, D., Panozzo, D.: Bijective and coarse high-order tetrahedral meshes. *ACM Trans. Graph.* **40**(4) (2021). DOI 10.1145/3450626.3459840. URL <https://doi.org/10.1145/3450626.3459840>
 19. Kim, T., Eberle, D.: Dynamic deformables: Implementation and production practicalities (now with code!). In: *ACM SIGGRAPH 2022 Courses, SIGGRAPH '22*. Association for Computing Machinery, New York, NY, USA (2022). DOI 10.1145/3532720.3535628. URL <https://doi.org/10.1145/3532720.3535628>
 20. Le, Q., Deng, Y., Bu, J., Zhu, B., Du, T.: Second-order finite elements for deformable surfaces. In: *SIGGRAPH Asia 2023 Conference Papers, SA '23*. Association for Computing Machinery, New York, NY, USA (2023). DOI 10.1145/3610548.3618186. URL <https://doi.org/10.1145/3610548.3618186>
 21. Liu, T., Bouaziz, S., Kavan, L.: Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Trans. Graph.* **36**(4) (2017). DOI 10.1145/3072959.2990496. URL <https://doi.org/10.1145/3072959.2990496>
 22. Macklin, M., Müller, M.: A constraint-based formulation of stable neo-hookean materials. In: *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games, MIG '21*. Association for Computing Machinery, New York, NY, USA (2021). DOI 10.1145/3487983.3488289. URL <https://doi.org/10.1145/3487983.3488289>
 23. Macklin, M., Müller, M., Chentanez, N.: Xpbd: Position-based simulation of compliant constrained dynamics. In: *Proceedings of the 9th International Conference on Motion in Games, MIG '16*, p. 49–54. Association for Computing Machinery, New York, NY, USA (2016). DOI 10.1145/2994258.2994272
 24. Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-time applications. *ACM Trans. Graph.* **33**(4) (2014). DOI 10.1145/2601097.2601152
 25. Macklin, M., Storey, K., Lu, M., Terdiman, P., Chentanez, N., Jeschke, S., Müller, M.: Small steps in physics simulation. In: *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '19*. Association for Computing Machinery, New York, NY, USA (2019). DOI 10.1145/3309486.3340247. URL <https://doi.org/10.1145/3309486.3340247>
 26. Müller, M., Stam, J., James, D., Thürey, N.: Real time physics: class notes. In: *ACM SIGGRAPH 2008 Classes, SIGGRAPH '08*. Association for Computing Machinery, New York, NY, USA (2008). DOI 10.1145/1401132.1401245. URL <https://doi.org/10.1145/1401132.1401245>
 27. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. *Journal of Visual Communication and Image Representation* **18**(2), 109–118 (2007). DOI <https://doi.org/10.1016/j.jvcir.2007.01.005>
 28. Nesme, M., Faure, F., Payan, Y.: Hierarchical multi-resolution finite element model for soft body simulation. In: M. Harders, G. Székely (eds.) *3rd International Symposium on Biomedical Simulation, ISBMS 2006*, July, 2006, *Lecture Notes in Computer Science*, vol. 4072, pp. 40–47. Zurich, Suisse (2006)
 29. Nesme, M., Marchal, M., Promayon, E., Chabanas, M., Payan, Y., Faure, F.: Physically realistic interactive simulation for biological soft tissues. *Recent Research Developments in Biomechanics* **2**, 1–22 (2005). URL <https://hal.archives-ouvertes.fr/hal-00080378>
 30. Perrusi, P.H.S., Baksic, P., Courtecuisse, H.: Interactive Finite Element Model of Needle Insertion and Laceration. In: H. Theisel, M. Wimmer (eds.) *Eurographics 2021 - Short Papers*. The Eurographics Association (2021). DOI 10.2312/egs.20211020
 31. Picinbono, G., Delingette, H., Ayache, N.: Nonlinear and anisotropic elastic soft tissue models for medical simulation. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, pp. 1370–1375 vol.2 (2001). DOI 10.1109/ROBOT.2001.932801
 32. San-Vicente, G., Aguinaga, I., Tomás Celigueta, J.: Cubical Mass-Spring Model design based on a tensile deformation test and nonlinear material model. *IEEE transactions on visualization and computer graphics* **18**(2), 228–41 (2012). DOI 10.1109/TVCG.2011.32
 33. Schneider, T., Hu, Y., Dumas, J., Gao, X., Panozzo, D., Zorin, D.: Decoupling simulation accuracy from mesh quality. *ACM Trans. Graph.* **37**(6) (2018). DOI 10.1145/3272127.3275067. URL <https://doi.org/10.1145/3272127.3275067>
 34. Schneider, T., Hu, Y., Gao, X., Dumas, J., Zorin, D., Panozzo, D.: A large-scale comparison of tetrahedral and hexahedral elements for solving elliptic pdes with the finite element method. *ACM Transactions on Graphics* **41**(3), 1–14 (2022). DOI 10.1145/3508372. URL <http://dx.doi.org/10.1145/3508372>
 35. Servin, M., Lacoursière, C., Melin, N.: Interactive simulation of elastic deformable materials. *Proc. SIGRAD* (2006)
 36. Smith, B., Goes, F.D., Kim, T.: Stable neo-hookean flesh simulation. *ACM Trans. Graph.* **37**(2) (2018). DOI 10.1145/3180491. URL <https://doi.org/10.1145/3180491>
 37. Stam, J.: Nucleus: Towards a unified dynamics solver for computer graphics. In: *2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pp. 1–11 (2009). DOI 10.1109/CADCG.2009.5246818
 38. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. *SIGGRAPH Comput. Graph.* **21**(4), 205–214 (1987)
 39. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. *SIGGRAPH Comput. Graph.* **21**(4), 205–214 (1987). DOI 10.1145/37402.37427. URL <https://doi.org/10.1145/37402.37427>
 40. Tournier, M., Nesme, M., Gilles, B., Faure, F.: Stable constrained dynamics **34**(4) (2015). DOI 10.1145/2766969
 41. Wu, X., Downes, M., Goktekin, T., Tendick, F.: adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In: *Eurographics*, pp. 349–358 (2001)