

FIRST MOMENTS OF A POLYHEDRON CLIPPED BY A PARABOLOID

FABIEN EVRARD^{‡*}, ROBERT CHIODI[†], AUSTIN HAN^{*}, BEREND VAN WACHEM[‡],
AND OLIVIER DESJARDINS^{*}

Abstract. We provide closed-form expressions for the first moments (i.e., the volume and volume-weighted centroid) of a polyhedron clipped by a paraboloid, that is, of a polyhedron intersected with the subset of the three-dimensional real space located on one side of a paraboloid. These closed-form expressions are derived following successive applications of the divergence theorem and the judicious parametrization of the intersection of the polyhedron's faces with the paraboloid. We provide means for identifying ambiguous discrete intersection topologies, and propose a corrective procedure for preventing their occurrence. Finally, we put our proposed closed-form expressions and numerical approach to the test with millions of random and manually engineered polyhedron/paraboloid intersection configurations. The results of these tests show that we are able to provide robust machine-accurate estimates of the first moments at a computational cost that is within one order of magnitude of that of state-of-the-art half-space clipping algorithms.

1. Introduction. Many computational methods and applications, ranging from finite-element [26, 5], cut-cell discontinuous Galerkin [12], and immersed isometric analysis methods [18, 3], to the initialization [4, 19, 21, 22, 28] and transport [25] of interfaces for simulating gas-liquid flows, require estimating integrals over polyhedra that are clipped by curved surfaces. These applications have engendered multiple dedicated quadrature rules and integration strategies, most of which focusing on estimating the first few moments of these clipped polyhedra, thus considering polynomial integrands only. The numerical approaches employed to estimate these moments vary greatly in terms of accuracy, computational cost, and robustness. Monte-Carlo methods [13, 16] are extremely robust and straightforward to implement, however, they suffer from a poor convergence rate, hence their cost/accuracy ratio is significant. Approaches based on octree subdivision [2, 23, 10] or surface triangulation/tesselation [28] exhibit better convergence rates, yet their computational cost remains prohibitive for numerical applications requiring *on-the-fly* moment estimations. A number of recent approaches rely on successive applications of the divergence theorem, converting the first moments of a clipped polyhedron into two- and/or one-dimensional integrals. These integrals can then be numerically integrated at low computational cost or, for specific surface types, even be derived into closed-form expressions. Bnà *et al.* [4] estimate the volume (zeroth moment) of a cube clipped by an implicit surface represented with a level-set function through integrating the local height of the surface using a two-dimensional Gauss-Legendre quadrature rule. This work has been extended to the first moments of a clipped cuboid by Chierici *et al.* [6]. For the similar purpose of estimating the zeroth moment of a polyhedron clipped by an implicit surface, Jones *et al.* [19] decompose the clipped polyhedron into a set of simplices, themselves split into a reference polyhedron whose volume is computed analytically, and a set of fundamental curved domains whose volumes are estimated using a two-dimensional Gauss-Legendre quadrature rule. Chin and Sukumar [7] use a Duvant quadrature rule [11] for integrating over the faces of a polyhedron bounded by rational or non-rational Bézier and B-spline patches. For non-rational surface parametrizations, this

*Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, United States

[†]Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, United States

[‡]Lehrstuhl für Mechanische Verfahrenstechnik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany

yields exact integral estimations of polynomial integrands, provided that the order of the Duvanant rule is high enough. Kromer and Bothe [21, 22] estimate the zeroth moment of a polyhedron clipped by an implicit surface by locally approximating the implicit surface as a paraboloid and applying the divergence theorem twice, converting the clipped volume into a sum of one-dimensional integrals, which are then estimated with a Gauss-Legendre quadrature rule. Finally, using Bernstein basis functions instead of monomial ones, Antolin and Hirschler [3] recently showed that, following successive applications of the divergence theorem, polynomial integrands can be integrated in a straightforward and analytical manner over curved polyhedra bounded by non-rational Bézier or B-spline surfaces.

This manuscript is concerned with estimating the first moments of a specific type of curved polyhedra, that are planar non-convex polyhedra clipped by a paraboloid surface (as in [21, 22]). Moreover, we require this estimation to (a) reach machine accuracy, while (b) maintaining a computational expense that is low enough to enable its *on-the-fly* execution in typical numerical applications (e.g., the simulation of two-phase flows with finite-volumes), and (c) being robust to singular configurations (e.g., paraboloid surfaces being parabolic cylinder or planes, and/or ambiguous discrete intersection topologies). These choices and requirements, although mainly motivated by the use of these moments for simulating two-phase flows, may also find applications in the numerical fields listed above. A main difficulty in clipping a polyhedron with a paraboloid lies in the fact that the faces of the clipped polyhedron cannot systematically be represented with non-rational or rational Bézier patches [27, 24]. This prevents the use of recently proposed integration strategies designed for curved polyhedra bounded by Bézier or B-spline surfaces [7, 3]. By successive applications of the divergence theorem, we show that the first moments of the clipped polyhedron can be expressed as a sum of one-dimensional integrals over straight line segments and conic section arcs. With a parametrization of the latter into rational Bézier arcs, we derive closed-form expressions for the first moments, rendering their numerical estimation exact within machine accuracy. Implemented within the half-edge data structure of the open-source **Interface Reconstruction Library** [8]¹, the computational cost of these moment estimations is kept within an order of magnitude of that of clipping a polyhedron with a half-space. Finally, our choice of arc parametrization, in conjunction with the detection and treatment of ambiguous discrete topologies, allows for robust moment estimates even in degenerate configurations.

The remainder of this manuscript is organized as follows: Section 2 introduces the problem that we address and the notations employed throughout the manuscript. The closed-form expressions of the clipped polyhedron’s first moments are derived in Section 3. Section 4 touches upon the integration of quantities (e.g., the moments) of the clipped polyhedron’s curved face(s). Section 5 details the procedure employed for preventing ambiguous clipped polyhedron topologies. Section 6 discusses the extension of our approach to higher-order moments. Finally, the accuracy, efficiency, and robustness of our proposed integration strategy are assessed in Section 7, and we draw conclusions in Section 8.

2. Problem statement. Consider the two following subsets of \mathbb{R}^3 :

- A polyhedron \mathcal{P} delimited by $n_{\mathcal{F}}$ planar polygonal faces $\mathcal{F}_i, i \in \{1, \dots, n_{\mathcal{F}}\}$, each equipped with an outward-pointing normal vector \mathbf{n}_i (e.g., see Figures 1a and 1b). Neither \mathcal{P} nor its faces \mathcal{F}_i are required to be convex.

¹The Interface Reconstruction Library source code is available under Mozilla Public License 2.0 at <https://github.com/robert-chiodi/interface-reconstruction-library>.

- The region \mathcal{Q} , located on one side of a paraboloid \mathcal{S} (e.g., see [Figure 1c](#)).

Without loss of generality, we assume to be working in a Cartesian coordinate system equipped with the orthonormal basis $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$, within which the position vector reads $\mathbf{x} = [x \ y \ z]^\top$, and where \mathcal{Q} and \mathcal{S} are implicitly defined as

$$(2.1) \quad \mathcal{Q} = \{\mathbf{x} \in \mathbb{R}^3 : \phi(\mathbf{x}) \leq 0\},$$

$$(2.2) \quad \mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 : \phi(\mathbf{x}) = 0\},$$

with

$$(2.3) \quad \phi : \begin{cases} \mathbb{R}^3 & \rightarrow \mathbb{R} \\ \mathbf{x} & \mapsto \alpha x^2 + \beta y^2 + z, \end{cases} \quad (\alpha, \beta) \in \mathbb{R}^2.$$

These assumptions do not restrict \mathcal{Q} and \mathcal{S} since, for any paraboloid-bounded clipping region in \mathbb{R}^3 , there exists a combination of rotations and translations of the canonical coordinate system resulting in such implicit definitions of \mathcal{Q} and \mathcal{S} . For the sake of clarity and conciseness, we introduce the following notations:

- The subscript \square_i refers to a topological element or quantity related to the i th face of the polyhedron $\hat{\mathcal{P}}$.
- The superscript $\hat{\square}$ implies an intersection with the clipping region \mathcal{Q} , e.g., $\hat{\mathcal{P}} \equiv \mathcal{P} \cap \mathcal{Q}$ or $\hat{\mathcal{F}}_i \equiv \mathcal{F}_i \cap \mathcal{Q}$.
- The superscript $\tilde{\square}$ implies an intersection with the polyhedron \mathcal{P} , e.g., $\tilde{\mathcal{S}} \equiv \mathcal{S} \cap \mathcal{P}$. This means that $\hat{\mathcal{P}} \equiv \tilde{\mathcal{Q}}$.

As mentioned in [Section 1](#), we are interested in calculating the zeroth and first moments of $\hat{\mathcal{P}} = \mathcal{P} \cap \mathcal{Q}$ (e.g., see [Figure 1e](#)), i.e., its volume and volume-weighted centroid, given as

$$(2.4) \quad M_0^{\hat{\mathcal{P}}} = \int_{\hat{\mathcal{P}}} 1 \, d\mathbf{x}, \quad \text{and} \quad \mathbf{M}_1^{\hat{\mathcal{P}}} = \int_{\hat{\mathcal{P}}} \mathbf{x} \, d\mathbf{x}.$$

In the remainder of this work, we shall refer to these quantities as “the first moments” or “the moments” of $\hat{\mathcal{P}}$, which we group into the vector

$$(2.5) \quad \mathbf{M}^{\hat{\mathcal{P}}} = \begin{bmatrix} M_0^{\hat{\mathcal{P}}} \\ \mathbf{M}_1^{\hat{\mathcal{P}}} \end{bmatrix} = \int_{\hat{\mathcal{P}}} \boldsymbol{\Upsilon}(\mathbf{x}) \, d\mathbf{x},$$

where

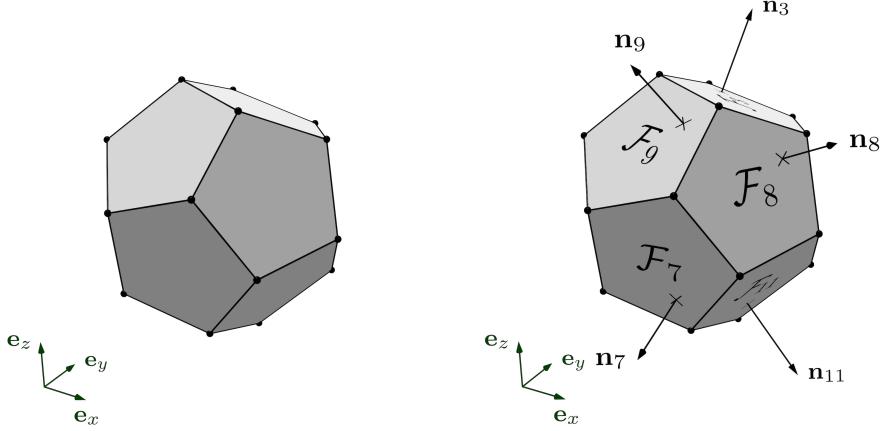
$$(2.6) \quad \boldsymbol{\Upsilon} : \begin{cases} \mathbb{R}^3 & \rightarrow \mathbb{R}^4 \\ \mathbf{x} & \mapsto [1 \ x \ y \ z]^\top \end{cases}.$$

3. Moments derivation. Using the divergence theorem, Eq. (2.5) can be rewritten as

$$(3.1) \quad \mathbf{M}^{\hat{\mathcal{P}}} = \int_{\hat{\mathcal{P}}} \nabla \cdot (\Phi(\mathbf{x}) \otimes \mathbf{e}_z) \, d\mathbf{x} = \int_{\partial \hat{\mathcal{P}}} \Phi(\mathbf{x}) (\mathbf{n}_{\partial \hat{\mathcal{P}}} \cdot \mathbf{e}_z) \, d\mathbf{a},$$

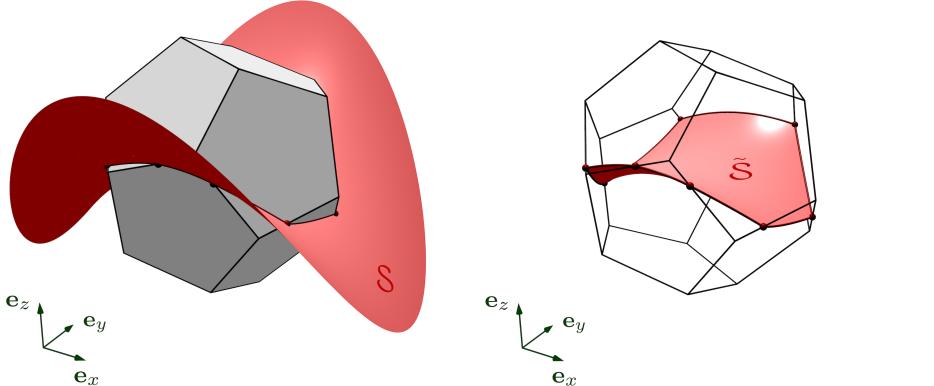
where $d\mathbf{a}$ is an infinitesimal surface element on $\partial \hat{\mathcal{P}} = \cup_i \hat{\mathcal{F}}_i \cup \tilde{\mathcal{S}}$, the boundary of $\hat{\mathcal{P}}$, $\mathbf{n}_{\partial \hat{\mathcal{P}}}$ is the normal to $\partial \hat{\mathcal{P}}$ pointing towards the outside of $\hat{\mathcal{P}}$, and Φ is defined as

$$(3.2) \quad \Phi : \begin{cases} \mathbb{R}^3 & \rightarrow \mathbb{R}^4 \\ \mathbf{x} & \mapsto \int_0^z \boldsymbol{\Upsilon}(\mathbf{x}) \, dz \quad (= [z \ xz \ yz \ \frac{1}{2}z^2]^\top) \end{cases}.$$



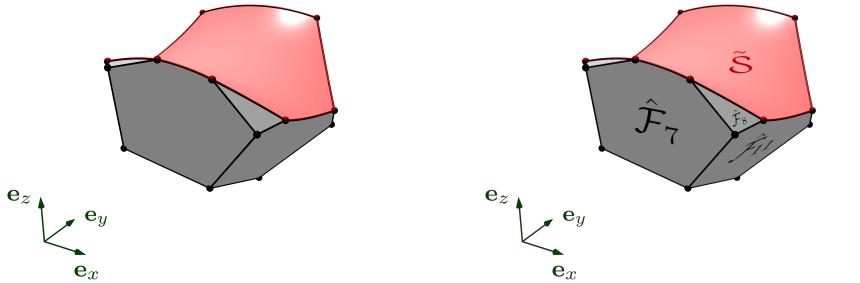
(a) A polyhedron $\mathcal{P} \subset \mathbb{R}^3$. In this example, \mathcal{P} is a regular dodecahedron with $n_{\mathcal{F}} = 12$ faces.

(b) The faces of $\partial\mathcal{P} = \cup_i \mathcal{F}_i$ and their outward-pointing normal vectors, \mathbf{n}_i , $i \in \{1, \dots, n_{\mathcal{F}}\}$.



(c) A paraboloid surface \mathcal{S} intersecting \mathcal{P} . The domain \mathcal{Q} is the subset of \mathbb{R}^3 located below \mathcal{S} , with respect to e_z .

(d) The intersection of the paraboloid \mathcal{S} with the polyhedron \mathcal{P} , $\tilde{\mathcal{S}} = \mathcal{S} \cap \mathcal{P}$.



(e) The clipped polyhedron $\hat{\mathcal{P}} = \mathcal{P} \cap \mathcal{Q}$, where \mathcal{Q} is the subset of \mathbb{R}^3 below \mathcal{S} .

(f) The faces of $\partial\hat{\mathcal{P}} = \cup_i \hat{\mathcal{F}}_i \cup \tilde{\mathcal{S}}$.

FIG. 1. A polyhedron $\mathcal{P} \subset \mathbb{R}^3$ intersected by the clipping region $\mathcal{Q} \subset \mathbb{R}^3$ located below a paraboloid surface \mathcal{S} .

Eq. (3.1) can be split into the following sum of integrals,

$$(3.3) \quad \mathbf{M}^{\hat{\mathcal{P}}} = \int_{\tilde{\mathcal{S}}} \Phi(\mathbf{x})(\mathbf{n}_{\tilde{\mathcal{S}}} \cdot \mathbf{e}_z) \, d\mathbf{a} + \sum_{i=1}^{n_{\mathcal{F}}} \int_{\hat{\mathcal{F}}_i} \Phi(\mathbf{x})(\mathbf{n}_i \cdot \mathbf{e}_z) \, d\mathbf{a},$$

where $\tilde{\mathcal{S}} = \mathcal{S} \cap \mathcal{P}$ is the portion of the paraboloid \mathcal{S} inside the polyhedron \mathcal{P} (e.g., see Figure 1d), $\mathbf{n}_{\tilde{\mathcal{S}}}$ is the normal to $\tilde{\mathcal{S}}$ pointing outwards of \mathcal{Q} (i.e., $\mathbf{n}_{\tilde{\mathcal{S}}} \cdot \mathbf{e}_z \geq 0$), and $\hat{\mathcal{F}}_i = \mathcal{F}_i \cap \mathcal{Q}$ is the portion of the face \mathcal{F}_i inside the clipping region \mathcal{Q} (e.g., see Figure 1f). Owing to the definitions of \mathcal{Q} and \mathcal{S} , as given in Eqs. (2.1) and (2.2), the normal to \mathcal{S} pointing outwards of \mathcal{Q} reads as

$$(3.4) \quad \mathbf{n}_{\mathcal{S}} = \frac{\nabla \phi}{\|\nabla \phi\|},$$

yielding

$$(3.5) \quad \mathbf{M}^{\hat{\mathcal{P}}} = \int_{\tilde{\mathcal{S}}} \Phi(\mathbf{x}) \left(\frac{\nabla \phi(\mathbf{x}) \cdot \mathbf{e}_z}{\|\nabla \phi(\mathbf{x})\|} \right) \, d\mathbf{a} + \sum_{i=1}^{n_{\mathcal{F}}} \int_{\hat{\mathcal{F}}_i} \Phi(\mathbf{x})(\mathbf{n}_i \cdot \mathbf{e}_z) \, d\mathbf{a}.$$

The surface $\tilde{\mathcal{S}}$ can be expressed in the parametric form

$$(3.6) \quad \tilde{\mathcal{S}} = \left\{ \begin{bmatrix} x \\ y \\ -\alpha x^2 - \beta y^2 \end{bmatrix}, (x, y) \in \tilde{\mathcal{S}}^\perp \right\},$$

with α and β the coefficients introduced in Eq. (2.3), and with $\tilde{\mathcal{S}}^\perp$ the projection of $\tilde{\mathcal{S}}$ onto the xy -plane. Under the assumption that $\mathbf{n}_i \cdot \mathbf{e}_z \neq 0$, each face $\hat{\mathcal{F}}_i$, $i \in \{1, \dots, n_{\mathcal{F}}\}$, can also be expressed in the parametric form

$$(3.7) \quad \hat{\mathcal{F}}_i = \left\{ \begin{bmatrix} x \\ y \\ \delta_i - \lambda_i x - \tau_i y \end{bmatrix}, (x, y) \in \hat{\mathcal{F}}_i^\perp \right\},$$

with

$$(3.8) \quad \delta_i = \frac{\mathbf{n}_i \cdot \mathbf{x}_{\mathcal{F}_i}}{\mathbf{n}_i \cdot \mathbf{e}_z}, \quad \text{for any } \mathbf{x}_{\mathcal{F}_i} \in \mathcal{F}_i,$$

$$(3.9) \quad \lambda_i = \frac{\mathbf{n}_i \cdot \mathbf{e}_x}{\mathbf{n}_i \cdot \mathbf{e}_z},$$

$$(3.10) \quad \tau_i = \frac{\mathbf{n}_i \cdot \mathbf{e}_y}{\mathbf{n}_i \cdot \mathbf{e}_z},$$

and with $\hat{\mathcal{F}}_i^\perp$ the projection of $\hat{\mathcal{F}}_i$ onto the xy -plane. These explicit parametrizations yield

$$(3.11) \quad \mathbf{M}^{\hat{\mathcal{P}}} = \int_{\tilde{\mathcal{S}}^\perp} \Phi_{\mathcal{S}}(x, y) \, d\mathbf{a}^\perp + \sum_{i=1}^{n_{\mathcal{F}}} \text{sign}(\mathbf{n}_i \cdot \mathbf{e}_z) \int_{\hat{\mathcal{F}}_i^\perp} \Phi_{\mathcal{F}_i}(x, y) \, d\mathbf{a}^\perp,$$

where $\Phi_{\mathcal{S}}$ and $\Phi_{\mathcal{F}_i}$ are the function vectors

$$(3.12) \quad \Phi_{\mathcal{S}}(x, y) = \begin{bmatrix} -\alpha x^2 - \beta y^2 \\ -\alpha x^3 - \beta x y^2 \\ -\alpha y x^2 - \beta y^3 \\ \frac{1}{2} (\alpha x^2 + \beta y^2)^2 \end{bmatrix}, \quad \Phi_{\mathcal{F}_i}(x, y) = \begin{bmatrix} \delta_i - \lambda_i x - \tau_i y \\ \delta_i x - \lambda_i x^2 - \tau_i x y \\ \delta_i y - \lambda_i x y - \tau_i y^2 \\ \frac{1}{2} (\delta_i - \lambda_i x - \tau_i y)^2 \end{bmatrix},$$

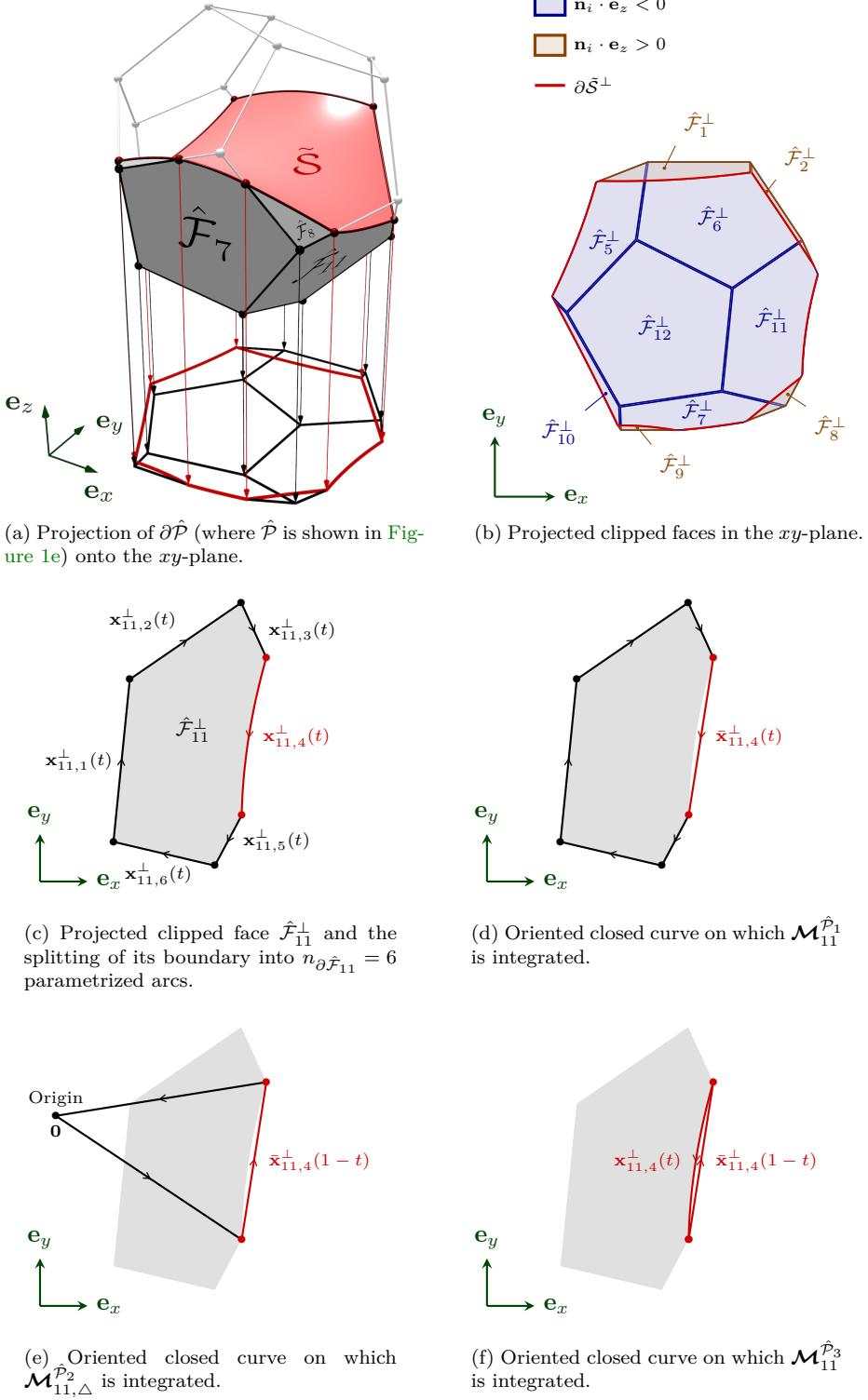


FIG. 2. Illustration of the one-dimensional integration domains used for integrating the first, second, and third contributions to the moments.

and $d\mathbf{a}^\perp = dx dy$ is an infinitesimal surface element on the xy -plane. Note that in order to simplify Eq. (3.5) into (3.11), we have used the fact that $\|\nabla\phi(\mathbf{x})\|$ is the determinant of the parametrization (3.6) of $\tilde{\mathcal{S}}$, and that $|\mathbf{n}_i \cdot \mathbf{e}_z|^{-1}$ is the determinant of the parametrization (3.7) of $\hat{\mathcal{F}}_i$. The projected integration domains $\tilde{\mathcal{S}}^\perp$ and $\hat{\mathcal{F}}_i^\perp$ corresponding to the configuration of Figure 1 are illustrated in Figure 2.

Eq. (3.11) can be rewritten as

$$(3.13) \quad \mathcal{M}^{\hat{\mathcal{P}}} = \int_{\tilde{\mathcal{S}}^\perp} \nabla \cdot (\Psi_{\mathcal{S}}(x, y) \otimes \mathbf{e}_x) \, d\mathbf{a}^\perp + \sum_{i=1}^{n_{\mathcal{F}}} \text{sign}(\mathbf{n}_i \cdot \mathbf{e}_z) \int_{\hat{\mathcal{F}}_i^\perp} \nabla \cdot (\Psi_{\mathcal{F}_i}(x, y) \otimes \mathbf{e}_x) \, d\mathbf{a}^\perp,$$

where $\Psi_{\mathcal{S}}$ and $\Psi_{\mathcal{F}_i}$ are defined as

$$(3.14) \quad \Psi_{\mathcal{S}} : \begin{cases} \mathbb{R}^2 & \rightarrow \mathbb{R}^4 \\ (x, y) & \mapsto \int_0^x \Phi_{\mathcal{S}}(x, y) \, dx \end{cases} .$$

and $\forall i \in \{1, \dots, n_{\mathcal{F}}\}$,

$$(3.15) \quad \Psi_{\mathcal{F}_i} : \begin{cases} \mathbb{R}^2 & \rightarrow \mathbb{R}^4 \\ (x, y) & \mapsto \int_0^x \Phi_{\mathcal{F}_i}(x, y) \, dx \end{cases} .$$

Note that the choice made here of integrating $\Phi_{\mathcal{S}}$ and $\Phi_{\mathcal{F}_i}$ with respect to x is arbitrary, and that we could have equivalently integrated them with respect to y , requiring to replace \mathbf{e}_x by \mathbf{e}_y in Eq. (3.13). Using the divergence theorem once again, this gives

$$(3.16) \quad \mathcal{M}^{\hat{\mathcal{P}}} = \int_{\partial\tilde{\mathcal{S}}^\perp} \Psi_{\mathcal{S}}(x, y) (\mathbf{n}_{\partial\tilde{\mathcal{S}}^\perp} \cdot \mathbf{e}_x) \, dl + \sum_{i=1}^{n_{\mathcal{F}}} \int_{\partial\hat{\mathcal{F}}_i^\perp} \Psi_{\mathcal{F}_i}(x, y) (\mathbf{n}_{\partial\hat{\mathcal{F}}_i^\perp} \cdot \mathbf{e}_x) \, dl ,$$

where dl is an infinitesimal line element on the integration domains $\partial\tilde{\mathcal{S}}^\perp$ and $\partial\hat{\mathcal{F}}_i^\perp$, which are the boundaries of the projections of the faces of $\hat{\mathcal{P}}$ onto the xy -plane. As such, they consist of closed curves in the xy -plane, that are successions of conic section arcs and/or line segments (e.g., see Figures 2b and 2c). Note that the term “ $\text{sign}(\mathbf{n}_i \cdot \mathbf{e}_z)$ ” is now implicitly accounted for, as the closed curves $\partial\hat{\mathcal{F}}_i$ (and therefore their projection onto the xy -plane, $\partial\hat{\mathcal{F}}_i^\perp$) are oriented so as to produce a normal vector pointing towards the outside of $\hat{\mathcal{P}}$. It should also be noted that the integration domains $\partial\tilde{\mathcal{S}}^\perp$ and $\partial\hat{\mathcal{F}}_i^\perp$ do not necessarily consist of one closed curve each – they may each be the union of several non-intersecting oriented closed curves.

Let us assume that a parametrization

$$(3.17) \quad \mathbf{x}_{i,j}(t) = x_{i,j}(t)\mathbf{e}_x + y_{i,j}(t)\mathbf{e}_y + z_{i,j}(t)\mathbf{e}_z, \quad t \in [0, 1], \quad j \in \{1, \dots, n_{\partial\hat{\mathcal{F}}_i}\},$$

is known for each of the $n_{\partial\hat{\mathcal{F}}_i}$ arcs of $\partial\hat{\mathcal{F}}_i$, where the functions $x_{i,j}$, $y_{i,j}$, and $z_{i,j}$ belong to $C^1([0, 1])$. Moreover, let us note that each parametrized conic section arc belonging to $\partial\tilde{\mathcal{S}}$ is necessarily present in one and only one of the clipped face boundaries $\partial\hat{\mathcal{F}}_i$, where it is traversed in the opposite direction for integrating $\Psi_{\mathcal{F}_i}$. Eq. (3.16) can then be written as

$$(3.18) \quad \mathcal{M}^{\hat{\mathcal{P}}} = \sum_{i=1}^{n_{\mathcal{F}}} \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} \int_0^1 \left(\Psi_{\mathcal{F}_i}(x_{i,j}(t), y_{i,j}(t)) - 1_{i,j}^{\partial\tilde{\mathcal{S}}} \Psi_{\mathcal{S}}(x_{i,j}(t), y_{i,j}(t)) \right) y'_{i,j}(t) \, dt ,$$

where

$$(3.19) \quad 1_{i,j}^{\partial\tilde{\mathcal{S}}} = \begin{cases} 1 & \text{if the } j\text{th arc of } \partial\hat{\mathcal{F}}_i \text{ also belongs to } \partial\tilde{\mathcal{S}} \\ 0 & \text{otherwise} \end{cases}.$$

Note that in Eq. (3.18) and in the remainder of this manuscript, the superscript \square' indicates that a function has been differentiated with respect to its unique variable. A closed-form expression can be derived for the integral in Eq. (3.18), however, its use for numerically calculating the moments is undesirable for two main reasons:

1. the expression contains many terms, rendering its numerical calculation expensive;
2. the expression depends on δ_i , λ_i , and τ_i , which all tend towards infinity as $\mathbf{n}_i \cdot \mathbf{e}_z$ tends towards zero, leading to large round-off errors in the context of floating-point arithmetics.

Instead, the introduction of a twin parametrization of the arcs of $\partial\hat{\mathcal{P}}$ and the judicious splitting of the integral in Eq. (3.18) can both reduce the complexity of its closed-form expression and remove its direct dependency on the potentially singular coefficients δ_i , λ_i , and τ_i . Let us then introduce the parametrization

$$(3.20) \quad \bar{\mathbf{x}}_{i,j}(t) = \bar{x}_{i,j}(t)\mathbf{e}_x + \bar{y}_{i,j}(t)\mathbf{e}_y + \bar{z}_{i,j}(t)\mathbf{e}_z, \quad t \in [0, 1], \quad j \in \{1, \dots, n_{\partial\hat{\mathcal{F}}_i}\},$$

which links $\mathbf{x}_{i,j}(0)$ to $\mathbf{x}_{i,j}(1)$ by a straight line. For the sake of conciseness, we shall refer to these two points as $\mathbf{x}_{i,j}(0) = \mathbf{x}_{i,j,0}$ and $\mathbf{x}_{i,j}(1) = \mathbf{x}_{i,j,1}$ in the remainder of this work. This twin parametrization of each arc is simply given as

$$(3.21) \quad \bar{\mathbf{x}}_{i,j}(t) = (1-t)\mathbf{x}_{i,j,0} + t\mathbf{x}_{i,j,1}, \quad t \in [0, 1], \quad j \in \{1, \dots, n_{\partial\hat{\mathcal{F}}_i}\}.$$

If the j th arc of $\partial\hat{\mathcal{F}}_i$ does not belong to $\tilde{\mathcal{S}}$, then $x_{i,j} \equiv \bar{x}_{i,j}$, $y_{i,j} \equiv \bar{y}_{i,j}$ and $z_{i,j} \equiv \bar{z}_{i,j}$. We can then re-organize Eq. (3.18) as

$$(3.22) \quad \begin{aligned} \mathcal{M}^{\hat{\mathcal{P}}} = & \sum_{i=1}^{n_{\mathcal{F}}} \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} \left[\int_0^1 \Psi_{\mathcal{F}_i}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) \bar{y}'_{i,j}(t) dt \right. \\ & - 1_{i,j}^{\partial\tilde{\mathcal{S}}} \int_0^1 \Psi_{\mathcal{S}}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) \bar{y}'_{i,j}(t) dt \\ & - 1_{i,j}^{\partial\tilde{\mathcal{S}}} \int_0^1 \left(\Psi_{\mathcal{F}_i}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) - \Psi_{\mathcal{S}}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) \right) \bar{y}'_{i,j}(t) dt \\ & \left. + \left(\Psi_{\mathcal{S}}(x_{i,j}(t), y_{i,j}(t)) - \Psi_{\mathcal{F}_i}(x_{i,j}(t), y_{i,j}(t)) \right) y'_{i,j}(t) dt \right], \end{aligned}$$

The moments can thus be described as the sum of three distinct contributions, i.e.,

$$(3.23) \quad \mathcal{M}^{\hat{\mathcal{P}}} = \mathcal{M}^{\hat{\mathcal{P}}_1} + \mathcal{M}^{\hat{\mathcal{P}}_2} + \mathcal{M}^{\hat{\mathcal{P}}_3} = \sum_{i=1}^{n_{\mathcal{F}}} \mathcal{M}_i^{\hat{\mathcal{P}}_1} + \sum_{i=1}^{n_{\mathcal{F}}} \mathcal{M}_i^{\hat{\mathcal{P}}_2} + \sum_{i=1}^{n_{\mathcal{F}}} \mathcal{M}_i^{\hat{\mathcal{P}}_3},$$

where

$$(3.24) \quad \mathcal{M}_i^{\hat{\mathcal{P}}_1} = \int_0^1 \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} \Psi_{\mathcal{F}_i}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) \bar{y}'_{i,j}(t) dt,$$

$$(3.25) \quad \mathcal{M}_i^{\hat{\mathcal{P}}_2} = \int_0^1 \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} -1_{i,j}^{\partial\tilde{\mathcal{S}}} \Psi_{\mathcal{S}}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) \bar{y}'_{i,j}(t) dt,$$

$$(3.26) \quad \mathcal{M}_i^{\hat{\mathcal{P}}_3} = \int_0^1 \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} 1_{i,j}^{\partial\tilde{\mathcal{S}}} \left((\Psi_{\mathcal{S}}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t)) - \Psi_{\mathcal{F}_i}(\bar{x}_{i,j}(t), \bar{y}_{i,j}(t))) \bar{y}'_{i,j}(t) \right. \\ \left. + (\Psi_{\mathcal{F}_i}(x_{i,j}(t), y_{i,j}(t)) - \Psi_{\mathcal{S}}(x_{i,j}(t), y_{i,j}(t))) y'_{i,j}(t) \right) dt.$$

The contributions $\mathcal{M}_i^{\hat{\mathcal{P}}_1}$ and $\mathcal{M}_i^{\hat{\mathcal{P}}_2}$ require the integration of the paraboloid and plane primitives over straight lines only (e.g., see Figures 2d and 2e), hence are straightforward to derive. The contribution $\mathcal{M}_i^{\hat{\mathcal{P}}_3}$, on the other hand, requires the parametrization of the conic section arcs in $\partial\tilde{\mathcal{S}}$ (e.g., see Figure 2f). It should also be noted that each arc of the clipped faces $\partial\hat{\mathcal{F}}_i$ contributes to $\mathcal{M}_i^{\hat{\mathcal{P}}_1}$, whereas only the conic section arcs in those faces (originating from the intersection of $\partial\mathcal{P}$ with \mathcal{S}) contribute to $\mathcal{M}_i^{\hat{\mathcal{P}}_2}$ and $\mathcal{M}_i^{\hat{\mathcal{P}}_3}$, owing to the presence of the coefficient $1_{i,j}^{\partial\tilde{\mathcal{S}}}$.

3.1. First term: $\mathcal{M}_i^{\hat{\mathcal{P}}_1}$. Let us be reminded that the boundary of each clipped face, $\partial\hat{\mathcal{F}}_i$, is a succession of $n_{\partial\hat{\mathcal{F}}_i}$ conic section arcs and/or straight line segments that each link a start point $\mathbf{x}_{i,j,0}$ to an end point $\mathbf{x}_{i,j,1}$. Now recall that we aim to derive expressions that are free of the coefficients δ_i , λ_i , and τ_i , so as to avoid round-off errors in the numerical calculation of the moments. To do so, we assign to each face $\hat{\mathcal{F}}_i$ a reference point $\mathbf{x}_{i,\text{ref}}$ whose only requirement is to belong to the plane containing $\hat{\mathcal{F}}_i$, e.g., $\mathbf{x}_{i,\text{ref}} = \mathbf{x}_{i,1,0}$. For each arc of each clipped face, rather than integrating on the straight line linking $\mathbf{x}_{i,j,0}$ to $\mathbf{x}_{i,j,1}$, we integrate instead on the oriented triangle $T_{i,j}^{(1)} = (\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,\text{ref}})$. Since $\partial\hat{\mathcal{F}}_i$ is the union of closed curves, the start point of each of its constituting arcs is necessarily the end point of another arc, and the sum of all these triangle integrals is equal to the sum of the straight arc integrals. In other terms, substituting Eq. (3.21) into Eq. (3.24), the latter can be rewritten as

(3.27)

$$\mathcal{M}_i^{\hat{\mathcal{P}}_1} = \int_0^1 \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} \Psi_{\mathcal{F}_i}((1-t)x_{i,j,0} + tx_{i,j,1}, (1-t)y_{i,j,0} + ty_{i,j,1})(y_{i,j,1} - y_{i,j,0}) dt, \\ = \int_0^1 \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} \left(\begin{array}{l} \Psi_{\mathcal{F}_i}((1-t)x_{i,j,0} + tx_{i,j,1}, (1-t)y_{i,j,0} + ty_{i,j,1})(y_{i,j,1} - y_{i,j,0}) \\ + \Psi_{\mathcal{F}_i}((1-t)x_{i,j,1} + tx_{i,\text{ref}}, (1-t)y_{i,j,1} + ty_{i,\text{ref}})(y_{i,\text{ref}} - y_{i,j,1}) \\ + \Psi_{\mathcal{F}_i}((1-t)x_{i,\text{ref}} + tx_{i,j,0}, (1-t)y_{i,\text{ref}} + ty_{i,j,0})(y_{i,j,0} - y_{i,\text{ref}}) \end{array} \right) dt.$$

Making use of Eqs. (3.12) and (3.15), and of the fact that

$$(3.28) \quad z_{i,\text{ref}} = \delta_i - \lambda_i x_{i,\text{ref}} - \tau_i y_{i,\text{ref}}$$

$$(3.29) \quad z_{i,j} = \delta_i - \lambda_i x_{i,j} - \tau_i y_{i,j}, \quad \forall j \in \{1, \dots, n_{\partial\hat{\mathcal{F}}_i}\}$$

it follows that

$$(3.30) \quad \mathcal{M}_i^{\hat{\mathcal{P}}_1} = \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} \mathcal{A}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,\text{ref}}) \mathcal{B}^{(1)}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,\text{ref}}),$$

where \mathcal{A} is the operator for calculating the signed projected area of a triangle from the knowledge of its three corners, i.e.,

$$(3.31) \quad \mathcal{A} : \begin{cases} \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 & \rightarrow \mathbb{R} \\ (\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) & \mapsto \frac{1}{2}(x_a(y_b - y_c) + x_b(y_c - y_a) + x_c(y_a - y_b)) \end{cases},$$

and $\mathcal{B}^{(1)} : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^4$ reads as

$$(3.32) \quad \mathcal{B}^{(1)}(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) = \frac{1}{12} \begin{bmatrix} 4(z_a + z_b + z_c) \\ (z_a + z_b + z_c)(x_a + x_b + x_c) + x_a z_a + x_b z_b + x_c z_c \\ (z_a + z_b + z_c)(y_a + y_b + y_c) + y_a z_a + y_b z_b + y_c z_c \\ z_a^2 + z_b^2 + z_c^2 + z_b z_c + z_a z_b + z_a z_c \end{bmatrix}.$$

3.2. Second term: $\mathcal{M}_{i,\Delta}^{\hat{\mathcal{P}}_2}$. For computing $\mathcal{M}_{i,\Delta}^{\hat{\mathcal{P}}_2}$, similarly as in Subsection 3.1, we choose a reference point \mathbf{x}_S belonging to \mathcal{S} . An obvious choice for this reference point is the origin of our coordinate system, i.e., $\mathbf{x}_S = \mathbf{0}$. For each conic section arc of each clipped face, rather than integrating on the straight line linking $\mathbf{x}_{i,j,0}$ to $\mathbf{x}_{i,j,1}$, we integrate instead on the oriented triangle $T_{i,j}^{(2)} = (\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_S)$. For each i th face of \mathcal{P} , this yields the moment contribution

$$(3.33) \quad \mathcal{M}_{i,\Delta}^{\hat{\mathcal{P}}_2} = \int_0^1 \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} -1_{i,j}^{\partial\hat{\mathcal{S}}} \begin{pmatrix} \Psi_S((1-t)x_{i,j,0} + tx_{i,j,1}, (1-t)y_{i,j,0} + ty_{i,j,1})(y_{i,j,1} - y_{i,j,0}) \\ +\Psi_S((1-t)x_{i,j,1} + tx_S, (1-t)y_{i,j,1} + ty_S)(y_S - y_{i,j,1}) \\ +\Psi_S((1-t)x_S + tx_{i,j,0}, (1-t)y_S + ty_{i,j,0})(y_{i,j,0} - y_S) \end{pmatrix} dt,$$

Using Eqs. (3.12) and (3.14), substituting $\mathbf{x}_S = \mathbf{0}$, and making use of the fact that

$$(3.34) \quad 1_{i,j}^{\partial\hat{\mathcal{S}}} \neq 0 \Leftrightarrow \begin{cases} z_{i,j,0} = -\alpha x_{i,j,0}^2 - \beta y_{i,j,0}^2 \\ z_{i,j,1} = -\alpha x_{i,j,1}^2 - \beta y_{i,j,1}^2 \end{cases}, \quad \forall j \in \{1, \dots, n_{\partial\hat{\mathcal{F}}_i}\}$$

it follows that

$$(3.35) \quad \mathcal{M}_{i,\Delta}^{\hat{\mathcal{P}}_2} = \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} 1_{i,j}^{\partial\hat{\mathcal{S}}} \mathcal{A}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{0}) \mathcal{B}^{(2)}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}),$$

where \mathcal{A} is the signed projected triangle area operator defined in Eq. (3.31) and the operator $\mathcal{B}^{(2)} : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^4$ reads as

$$(3.36) \quad \mathcal{B}^{(2)}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{90} \begin{bmatrix} 15(\alpha x_a x_b + \beta y_a y_b - z_a - z_b) \\ 6\beta(y_a - y_b)(x_b y_a - x_a y_b) + 9(x_a + x_b)(z_a + z_b) \\ 6\alpha(x_a - x_b)(y_b x_a - y_a x_b) + 9(y_a + y_b)(z_a + z_b) \\ 2\alpha\beta(x_b y_a - x_a y_b)^2 + 3(z_a + z_b)(\alpha x_a x_b + \beta y_a y_b - z_a - z_b) + 3z_a z_b \end{bmatrix}.$$

Note that $\mathcal{M}_{i,\Delta}^{\hat{\mathcal{P}}_2} \neq \mathcal{M}_i^{\hat{\mathcal{P}}_2}$, owing to our choice of integrating over triangles rather than the individual arcs, however their sum over all faces is equal, yielding

$$(3.37) \quad \mathcal{M}^{\hat{\mathcal{P}}_2} = \sum_{i=1}^{n_{\mathcal{F}}} \mathcal{M}_{i,\Delta}^{\hat{\mathcal{P}}_2}.$$

3.3. Third term: $\mathcal{M}^{\hat{\mathcal{P}}_3}$. To derive a closed-form expression for $\mathcal{M}^{\hat{\mathcal{P}}_3}$, a parametrization of the conic section arcs in $\partial\hat{\mathcal{S}}$ must be provided. For the elliptic and hyperbolic cases, traditional parametrizations using trigonometric functions are obvious choices, however they can yield significant round-off errors due to very large values of their constitutive parameters (e.g., the semi-major and semi-minor axes). This is

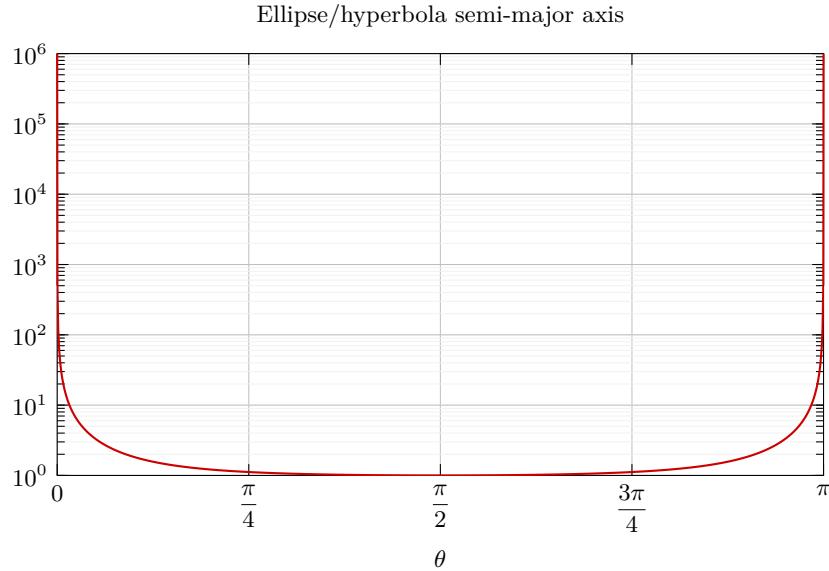


FIG. 3. Semi-major axis of the conic section generated by the intersection of the paraboloid \mathcal{S} , as defined in Eq. (2.2), when $\alpha = |\beta| = 1$, with the plane implicitly defined by $\mathbf{n} \cdot \mathbf{x} = -\mathbf{n} \cdot \mathbf{e}_z$, with $\mathbf{n} = [\cos(\theta) \ 0 \ \sin(\theta)]^\top$, as a function of the angle θ . The semi-major axis of the conic section tends to infinity when θ approaches a multiple of π , meaning that a parametrization of any arc of this conic section with trigonometric functions becomes singular if $\mathbf{n} \cdot \mathbf{e}_z \rightarrow 0$.

illustrated in Figure 3 where the semi-major axis of the conic section generated by the intersection of a paraboloid with a plane is plotted as this plane is rotated about the \mathbf{e}_y basis vector. To avoid singular arc parametrizations, we express each conic section arc as a rational Bézier curve [15]. This provides a general parametrization that is valid over all conic section cases (i.e., elliptic, hyperbolic, and parabolic) and allows a seamless and smooth transition between cases. A conic section arc linking a start point $\mathbf{x}_{i,j,0}$ to an end point $\mathbf{x}_{i,j,1}$ can be exactly represented by the rational quadratic Bézier curve parametrically defined as

$$(3.38) \quad \mathbf{x}_{i,j}(t) = \frac{B_0(t) \mathbf{x}_{i,j,0} + w_{i,j} B_1(t) \mathbf{x}_{i,j}^* + B_2(t) \mathbf{x}_{i,j,1}}{B_0(t) + w_{i,j} B_1(t) + B_2(t)}, \quad t \in [0, 1],$$

where

$$(3.39) \quad B_0(t) = (1-t)^2,$$

$$(3.40) \quad B_1(t) = 2(1-t)t,$$

$$(3.41) \quad B_2(t) = t^2,$$

are the Bernstein polynomials of degree 2, and $w_{i,j}$ is a weight associated with the control point $\mathbf{x}_{i,j}^*$. This control point is located at the intersection of the tangents to the conic section at the start and end points. In the case of the intersection of a planar face \mathcal{F}_i with a paraboloid surface \mathcal{S} , these tangents are obtained by the intersection of the planes tangent to \mathcal{S} at the start and end points, with the plane containing \mathcal{F}_i . The point $\mathbf{x}_{i,j}\left(\frac{1}{2}\right)$ is, by definition, located at the intersection of \mathcal{S} with the segment linking $\bar{\mathbf{x}}_{i,j}\left(\frac{1}{2}\right)$ to $\mathbf{x}_{i,j}^*$. Substituting t by $\frac{1}{2}$ in Eq. (3.38), it follows that

$$(3.42) \quad w_{i,j} (\mathbf{x}_{i,j}\left(\frac{1}{2}\right) - \mathbf{x}_{i,j}^*) = \bar{\mathbf{x}}_{i,j}\left(\frac{1}{2}\right) - \mathbf{x}_{i,j}\left(\frac{1}{2}\right),$$

from which $w_{i,j}$ can be deducted. If $|w_{i,j}| < 1$, the rational Bézier curve is an arc of an ellipse, if $|w_{i,j}| = 1$, the rational Bézier curve is an arc of a parabola, and if $|w_{i,j}| > 1$, the rational Bézier curve is an arc of a hyperbola (e.g., see Figure 4).

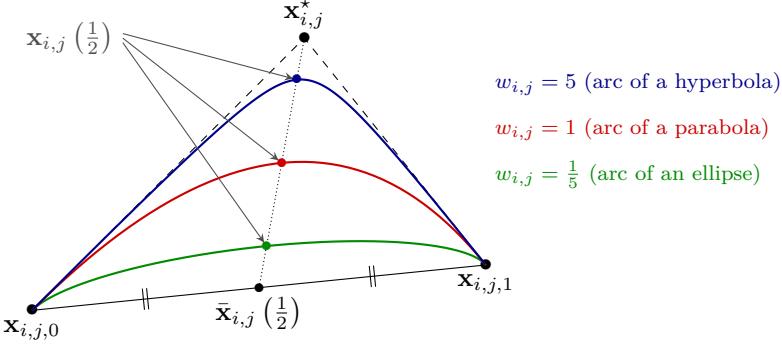


FIG. 4. Three rational Bézier arcs linking a start point $\mathbf{x}_{i,j,0}$ to an end point $\mathbf{x}_{i,j,1}$. Also shown are the control point $\mathbf{x}_{i,j}^*$ and weight $w_{i,j}$, along with the points $\bar{\mathbf{x}}_{i,j}(\frac{1}{2})$ and $\mathbf{x}_{i,j}(\frac{1}{2})$ that can be used for determining $w_{i,j}$.

Alternatively, the weight $w_{i,j}$ (when positive) relates to the absolute curvature κ of the conic section arc at its extremities $\mathbf{x}_{i,j,0}$ and $\mathbf{x}_{i,j,1}$ following [14]

$$(3.43) \quad w_{i,j} = \sqrt{\frac{|\mathcal{A}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,j}^*)|}{\kappa(\mathbf{x}_{i,j,0}) \|\mathbf{x}_{i,j}^* \mathbf{x}_{i,j,0}\|^3}} = \sqrt{\frac{|\mathcal{A}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,j}^*)|}{\kappa(\mathbf{x}_{i,j,1}) \|\mathbf{x}_{i,j}^* \mathbf{x}_{i,j,1}\|^3}},$$

where κ at any point \mathbf{x} on the conic section is given by [17]

$$(3.44) \quad \kappa(\mathbf{x}) = \frac{|(\mathbf{n}_i \times \nabla \phi(\mathbf{x})) \cdot (\mathbf{H}_\phi(\mathbf{n}_i \times \nabla \phi(\mathbf{x})))|}{\|\mathbf{n}_i \times \nabla \phi(\mathbf{x})\|},$$

with \mathbf{H}_ϕ the Hessian matrix

$$(3.45) \quad \mathbf{H}_\phi = 2 \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Note that, in order to prevent round-off errors in the numerical calculation of $\mathcal{M}^{\hat{\mathcal{P}}_3}$, we limit our implementation to cases where $w_{i,j} \geq 0$. As a consequence, conic section arcs that would result in negative rational Bézier weights are recursively split until positive weights are found. With such a parametrization of the conic section arcs, it can be shown that

$$(3.46) \quad \mathcal{M}_i^{\hat{\mathcal{P}}_3} = \sum_{j=1}^{n_{\partial \mathcal{F}_i}} 1_{i,j}^{\partial \tilde{\mathcal{S}}} \mathcal{A}(\mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,j}^*) \mathcal{B}^{(3)}(w_{i,j}, \mathbf{x}_{i,j,0}, \mathbf{x}_{i,j,1}, \mathbf{x}_{i,j}^*) ,$$

where \mathcal{A} is the signed projected triangle area operator defined in Eq. (3.31) and the operator $\mathcal{B}^{(3)} : \mathbb{R}^+ \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^4$ is given in Appendix A.

3.4. A special elliptic case. When $\alpha\beta > 0$ (i.e., \mathcal{S} is an elliptic paraboloid) and the normal to \mathcal{F}_i is such that $\mathbf{n}_i \cdot \mathbf{e}_z \neq 0$, the intersection of the plane containing the face \mathcal{F}_i with the surface \mathcal{S} is an ellipse. The intersection of \mathcal{F}_i with \mathcal{S} can then be: empty, a collection of arcs of this ellipse, or the entire ellipse. In the latter case, the

sum of the contributions $\mathcal{M}_i^{\hat{\mathcal{P}}_2}$ and $\mathcal{M}_i^{\hat{\mathcal{P}}_3}$ can be directly calculated by integrating Ψ_S and $\Psi_{\mathcal{F}_i}$ over the full ellipse, which yields the more concise expression

$$(4.47) \quad \mathcal{M}_i^{\hat{\mathcal{P}}_2} + \mathcal{M}_i^{\hat{\mathcal{P}}_3} = -\text{sign}(\mathbf{n}_i \cdot \mathbf{e}_z) \pi \frac{(\tau_i^2 \alpha + \lambda_i^2 \beta - 4\alpha\beta\delta_i)^2}{32(\alpha\beta)^{5/2}} \begin{bmatrix} 1 \\ \frac{\lambda_i}{2\alpha} \\ \frac{\tau_i}{2\beta} \\ \frac{5\tau_i^2}{12\beta} + \frac{5\lambda_i^2}{12\alpha} - \frac{2\delta_i}{3} \end{bmatrix},$$

where δ_i , λ_i , and τ_i have been defined in Eqs. (3.8)–(3.10). Note that this case only occurs for $\mathbf{n}_i \cdot \mathbf{e}_z \neq 0$, hence these coefficients are here non-singular.

4. Integrating on $\tilde{\mathcal{S}}$. Although this manuscript is concerned with estimating the first moments of $\hat{\mathcal{P}}$, the integration domains and parametrizations introduced in Section 3 can also be used for integrating quantities associated with the clipped surface $\tilde{\mathcal{S}}$, e.g., its moments. The area of $\tilde{\mathcal{S}}$, for instance, given as

$$(4.1) \quad M_0^{\tilde{\mathcal{S}}} = \int_{\tilde{\mathcal{S}}} 1 \, d\mathbf{x},$$

also reads after application of the divergence theorem as

$$(4.2) \quad M_0^{\tilde{\mathcal{S}}} = \sum_{i=1}^{n_{\mathcal{F}}} \sum_{j=1}^{n_{\partial\hat{\mathcal{F}}_i}} -1 \partial_{i,j}^{\tilde{\mathcal{S}}} \int_0^1 \gamma(x_{i,j}(t), y_{i,j}(t)) y'_{i,j}(t) \, dt,$$

where

$$(4.3) \quad \gamma : \begin{cases} \mathbb{R}^2 & \rightarrow \mathbb{R} \\ (x, y) & \mapsto \int_0^x \sqrt{1 + 4\alpha^2 x^2 + 4\beta^2 y^2} \, dx \end{cases}.$$

The first moments of $\tilde{\mathcal{S}}$, as well as the average normal vector, average Gaussian curvature and average mean curvature of $\tilde{\mathcal{S}}$, for example, can be expressed similarly as sums of one-dimensional integrals over the parametric arcs of $\partial\tilde{\mathcal{S}}^\perp$. Contrary to the first moments of $\hat{\mathcal{P}}$, however, they need to be estimated using a numerical quadrature rule, as closed-form expressions cannot be derived.

5. On floating-point arithmetics and robustness. In the context of floating-point arithmetics, there exist cases for which the computed topology of the clipped faces $\hat{\mathcal{F}}_i$ may be ill-posed, preventing the accurate calculation of the moments of $\hat{\mathcal{P}}$. This occurs when, in the discrete sense:

1. The surface \mathcal{S} is tangent to one or more edges of the polyhedron \mathcal{P} ;

2. At least one corner or vertex of the polyhedron \mathcal{P} belongs to the surface \mathcal{S} .

For any given face of \mathcal{P} , the former case is numerically detected by computing the absolute value of the dot product between the normalized tangent to \mathcal{S} and the normalized edge from which the tangent originates, and checking whether it lies within $\epsilon_{\text{tangent}}$ of unity. The latter case is detected by checking whether the intersection of an edge with \mathcal{S} lies within ϵ_{corner} of a corner or vertex of the polyhedron \mathcal{P} . When any of these configurations is detected, the polyhedron \mathcal{P} is randomly translated and rotated about its centroid by a distance and an angle equal to ϵ_{nudge} , and the clipped face discrete topologies are re-computed. Moreover, we also switch to a higher-accuracy floating-point format, e.g., from a 64-bit to a 128-bit format. In the current work, where we aim to produce results in “double precision”, we find the values $\epsilon_{\text{nudge}} = 10^{10} \times \epsilon_{128}$, $\epsilon_{\text{corner}} = 10^2 \times \epsilon_{64}$, and $\epsilon_{\text{tangent}} = 10^6 \times \epsilon_{64}$, with $\epsilon_{64} = 2^{-52}$ and

$\epsilon_{128} = 2^{-112}$ the upper bounds of the relative approximation error in 64-bit and 128-bit floating-point arithmetics, respectively, to prevent the computation of any ill-posed topologies in all tests presented in [Section 7](#) (for which more than 5×10^7 occurrences of the current “nudging” procedure are forced to occur). Choosing lower values for these tolerances may result in the generation of non-valid discrete topologies and/or erroneous moments. It should be noted that, for a polyhedron \mathcal{P} with volume $M_0^{\mathcal{P}} = \mathcal{O}(1)$, the rate of occurrence of the cases triggering this correction procedure is extremely small (that is, for non-engineered intersection configurations).

6. Higher-order moments. Second- and higher-order moments of the clipped polyhedron $\hat{\mathcal{P}}$ can be derived using the same procedure as presented in [Section 3](#) for the calculation of the zeroth and first moments. This merely entails appending higher-order monomials to the function vector Υ defined in Eq. [\(2.6\)](#). The arc parametrizations introduced in [Section 3](#) and used for calculating the three moment contributions $\mathcal{M}^{\mathcal{P}_1}$, $\mathcal{M}^{\hat{\mathcal{P}}_2}$, and $\mathcal{M}^{\hat{\mathcal{P}}_3}$ would remain unchanged, while the operators $\mathcal{B}^{(1)}$, $\mathcal{B}^{(2)}$, and $\mathcal{B}^{(3)}$ would then contain additional components respectively corresponding to the monomials appended to Υ . We hypothesize that closed-form expressions similar to those given in [Section 3](#) can be derived for moments of $\hat{\mathcal{P}}$ of arbitrary order. However, these would involve an ever increasing amount of high-order monomials of α , β , and of the components of the vertices of $\hat{\mathcal{P}}$ and of the control points of the conic section arcs in $\partial\partial\hat{\mathcal{P}}$, rendering the computation of the moments more sensitive to round-off errors.

7. Verification. In this section, the closed-form expressions derived in [Section 3](#), the approach of [Section 4](#) for integrating on the clipped surface, and the corrective procedure of [Section 5](#) are tested on a wide variety of engineered and random intersection configurations. When analytical expressions of the exact moments are not available, we recursively split the faces of the polyhedron \mathcal{P} so as to approximate $\partial\mathcal{P} \cap \mathcal{Q}$ and $\partial\mathcal{P} \cap (\mathbb{R}^3 \setminus \mathcal{Q})$ by collections of oriented triangles. We refer to this procedure as the adaptive mesh refinement (AMR) of the faces of the polyhedron \mathcal{P} . We then exactly integrate Φ_S on the triangulated approximation of $\partial\mathcal{P} \cap (\mathbb{R}^3 \setminus \mathcal{Q})$ and $\Phi_{\mathcal{F}_i}$ on the triangulated approximation of each clipped face $\hat{\mathcal{F}}_i$, effectively approximating Eq. [\(3.11\)](#). For each case, we ensure that enough levels of recursive refinement are employed in order to reach machine-zero. Accumulated errors due to the summation of the contributions of all triangles are avoided by the use of compensated summation, also known as Kahan summation [\[20\]](#).

7.1. Unit cube translating along \mathbf{e}_z . In a first test, we consider the elliptic paraboloid defined by Eq. [\(2.2\)](#) with $\alpha = \beta = 1$, intersecting with the unit cube centered at $\mathbf{x}_c = [1/2 \ 1/2 \ 1/2 - k]^\top$ (as illustrated in [Figure 5](#)). For this case, the zeroth and first moments of $\hat{\mathcal{P}}$, as well as the zeroth moment of $\tilde{\mathcal{S}}$, can be derived as analytical functions of k^2 . We compare these exact moments against those computed using the closed-form expressions derived in [Section 3](#) for estimating the moments of $\hat{\mathcal{P}}$, and by integrating Eq. [\(4.2\)](#) numerically with an adaptive Gauss-Legendre quadrature rule for estimating the zeroth moment of $\tilde{\mathcal{S}}$. The parameter k is regularly sampled on $[0, 3]$ with a uniform spacing $\Delta k = 10^{-3}$.

The left-hand graph of [Figure 6](#) shows the exact moments of $\hat{\mathcal{P}}$, scaled by their maximum values, as well as the exact zeroth moment of $\tilde{\mathcal{S}}$, scaled by its value at $k = 1$. The right-hand side of [Figure 6](#) shows the errors associated with their estimation, scaled similarly. These errors are all contained within an order of magnitude of

²A C++ implementation of these functions can be found at the beginning of `/tests/src/paraboloid_intersection_test.cpp` in the open-source **Interface Reconstruction Library**.

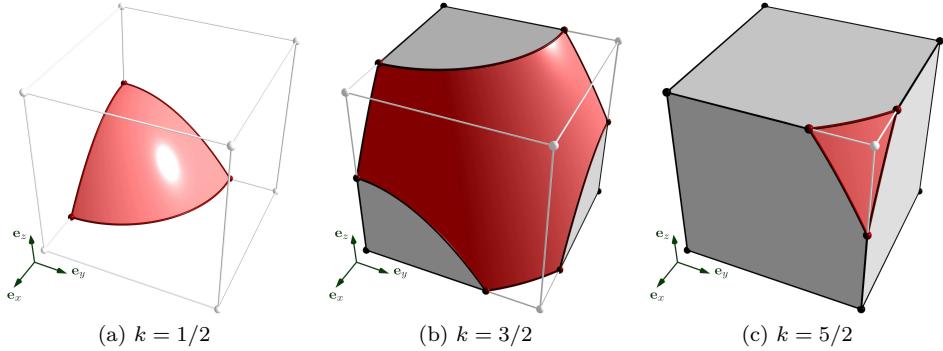


FIG. 5. Unit cube centred at $[1/2 \ 1/2 \ 1/2 - k]^\top$ clipped by the elliptic paraboloid parametrically defined as $z = -x^2 - y^2$.

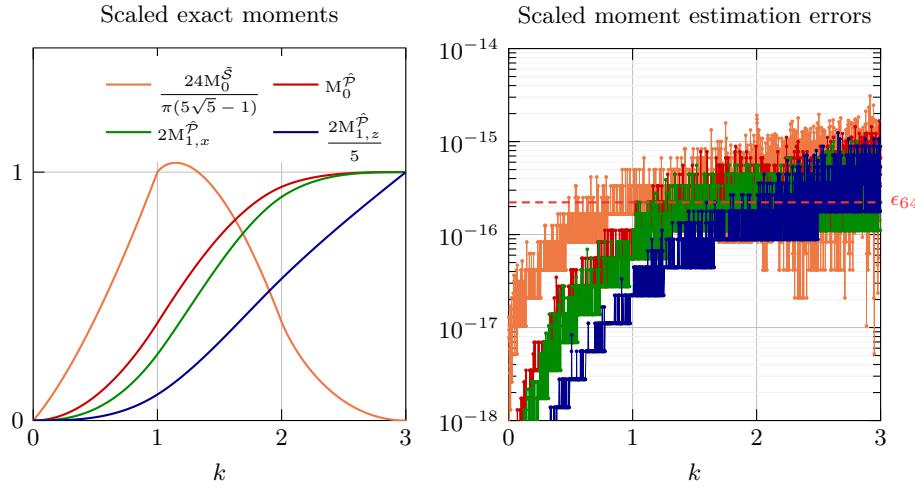


FIG. 6. Moments of the unit cube centred at $[1/2 \ 1/2 \ 1/2 - k]^\top$ clipped by the elliptic paraboloid parametrically defined as $z = -x^2 - y^2$, and the error in their estimation with 64-bit floating point arithmetics. The volume moments and their estimation errors are scaled with respect to the moments at $k = 3$. The surface area and its estimation error is scaled with respect to the surface area at $k = 1$. The volume moments are computed from the analytical expressions derived in Section 3, whereas the surface area is computed from Eq. (4.2) using an adaptive Gauss-Legendre quadrature rule. The 64-bit machine-epsilon $\epsilon_{64} = 2^{-52}$ is shown as the dashed red line.

$\epsilon_{64} = 2^{-52}$, the upper bound of the relative approximation error in 64-bit floating-point arithmetics.

7.2. Parameter sweep for several geometries. In a second test, we consider a selection of convex polyhedra (a regular tetrahedron, a cube, and a regular dodecahedron) and non-convex polyhedra (a hollow cube and the triangulated Stanford bunny [29]). These polyhedra, whose properties are summarized in Table 1, are scaled so as to have a unit volume (i.e., $M_0^P = 1$), and a centroid initially located at the origin (i.e., $M_1^P = \mathbf{0}$). They are then translated along $\mathbf{t} = [t_x \ t_y \ t_z]^\top$, and rotated about the three basis vectors ($\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$) with the angles θ_x, θ_y , and θ_z , respectively. Throughout this section, (t_x, t_y, t_z) are varied in $[-\frac{1}{2}, \frac{1}{2}]^3$, $(\theta_x, \theta_y, \theta_z)$ are varied in $[-\pi, \pi]^3$, and the paraboloid coefficients (α, β) are varied in $[-5, 5]^2$.

A random parameter sweep is first conducted by uniform random sampling of the

TABLE 1

Five polyhedra are considered: a regular tetrahedron, a cube, a regular dodecahedron, a hollow cube, and the triangulated Stanford bunny [29]. The first three polyhedra are convex, whereas the last two are not. The hollow cube contains non-convex faces.

Geometry	Tetrahedron	Cube	Dodecahedron	Hollow cube	Stanford bunny
Number of vertices	4	8	20	16	167,891
Number of faces	4	6	12	12	335,778
Are all faces convex?	Yes	Yes	Yes	No	Yes
Is polyhedron convex?	Yes	Yes	Yes	No	No
Snapshot					

eight parameters $(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, \alpha, \beta)$ in the parameter space $[-\frac{1}{2}, \frac{1}{2}]^3 \times [-\pi, \pi]^3 \times [-\frac{1}{2}, \frac{1}{2}]^2$, totalling more than 2×10^8 realizations. A graded parameter sweep is then conducted, in which the eight parameters are chosen in the discrete parameter space $\{-\frac{1}{2}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}\}^3 \times \{-\pi, -\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi\}^3 \times \{-5, -4, \dots, 4, 5\}^2$, resulting in $5^3 \times 5^3 \times 11^2 = 1,890,625$ distinct realizations for each geometry³. The graded parameter sweep differs from the random one in that it raises many singular intersection configurations (e.g., degenerate conic sections that consist of parallel or intersecting line segments, or conic sections that are parabolas) and/or ambiguous discrete topologies that arise from polyhedron vertices lying on the paraboloid or edges of the polyhedron being tangent to the paraboloid, therefore testing the robustness of our implementation as well as of the procedure described in Section 5. For each case of the random and graded parameter sweeps, the reference value of \mathcal{M}^P for calculating the moment errors is obtained by adaptive mesh refinement (AMR) of the faces of P into triangles, followed by the (exact) integration of Φ_S and Φ_{F_i} , given in Eq. (3.12), over the triangles above and below the paraboloid, respectively. Examples of random intersection cases and their associated AMR are shown in Figure 7, whereas examples of singular cases raised during the graded parameter sweep are shown in Figure 8.

The maximum and average moment errors obtained during the random parameter sweep are given in Table 2. For the tetrahedron, cube, dodecahedron, and hollow cube, the average moment error is of the order $\epsilon_{64} = 2^{-52} \sim \mathcal{O}(10^{-16})$, whereas the maximum error is about one to two orders of magnitude larger. The average and maximum moment errors for the Stanford bunny, which contains $\mathcal{O}(10^5)$ faces, are each about one order of magnitude larger than for the other polyhedra but still close to machine-zero. The graded parameter sweep, whose results are given in Table 3, exhibits similar moment errors as for the random parameter sweep, even though, by design, it raises many more singular intersection configurations and ambiguous discrete topologies than the random parameter sweep, therefore requiring a more frequent use of the nudging procedure described in Section 5.

7.3. Parameter sweep with nudge. In order to further test the robustness of our implementation and of the nudging procedure described in Section 5, we consider the same random parameter sweep for the cube geometry as described in Sub-

³We do not present the results of a graded parameter sweep on the Stanford bunny, since the “organic” nature of this polyhedron renders a graded parameter sweep equivalent to a random one.

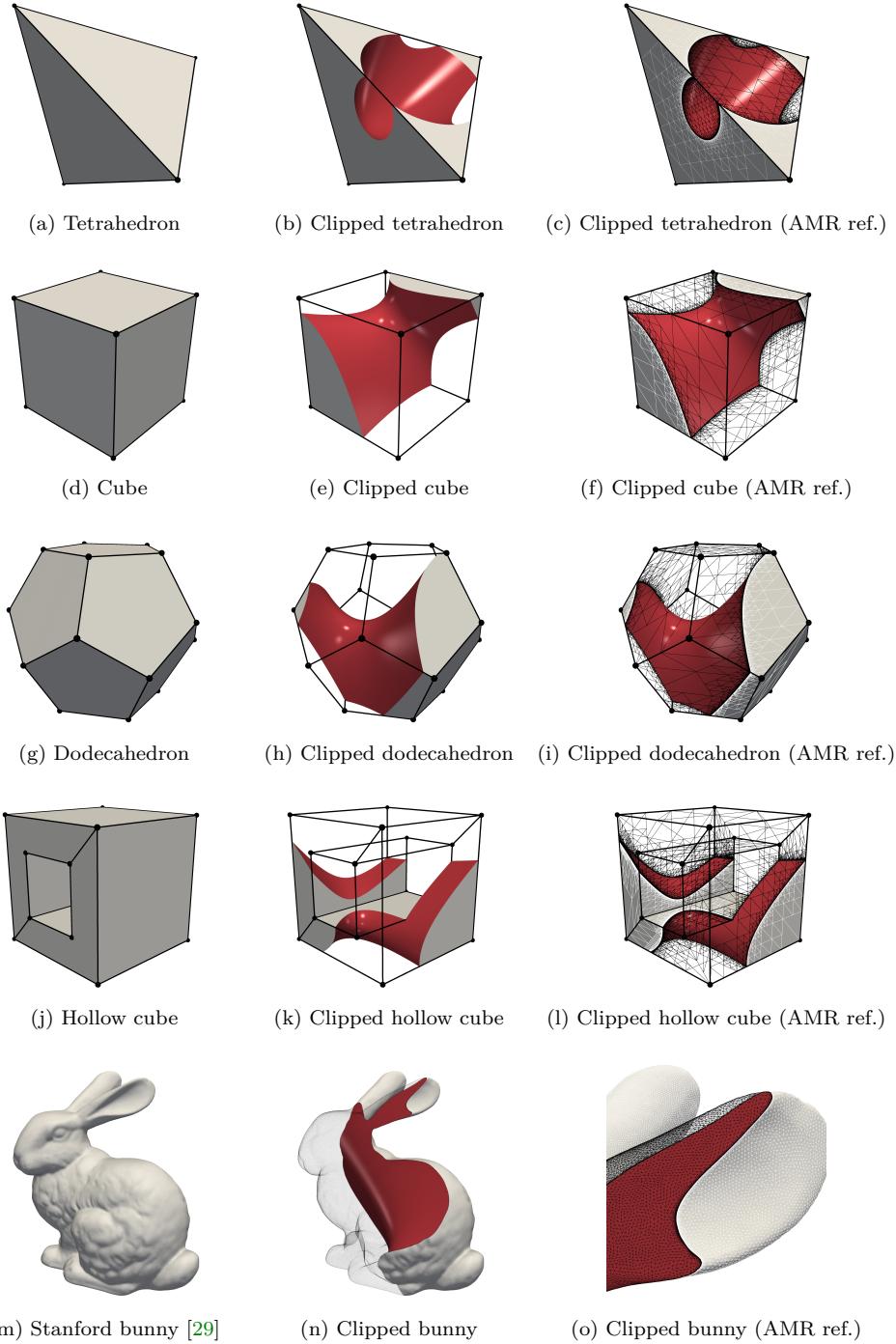


FIG. 7. Examples of random intersection cases between the five considered polyhedra and a paraboloid. The left column shows the full polyhedra; the central column shows the polyhedra clipped by a paraboloid; the right column shows the edges of the AMR of the polyhedra used for calculating the reference moments.

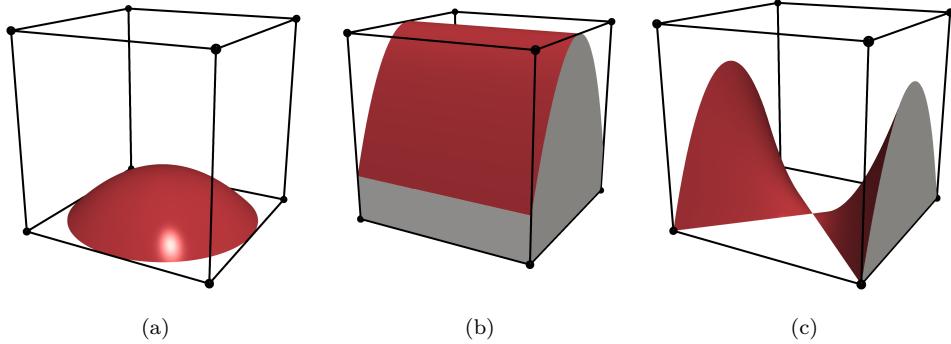


FIG. 8. Examples of cases with singular intersection configurations and/or ambiguous topologies covered by the graded parameter sweep: (a) the elliptic paraboloid is tangent to four edges of the polyhedron; (b) the parabolic cylinder is tangent to one face of the polyhedron, resulting in a degenerate conic section intersection that is made of two overlapping parallel lines; (c) the intersection of the hyperbolic paraboloid with one face of the polyhedron results in a degenerate hyperbola that is the intersection of two straight lines.

TABLE 2

Random parameter sweep results. For each geometry, we provide: the number of tests conducted, the number of recursive levels used for the AMR reference moment calculation, as well as the average and maximum errors in the estimation of the zeroth and first moments.

Geometry	Number of tests	AMR levels	Zeroth moment error		First moments error	
			Average	Maximum	Average	Maximum
Tetrahedron	5×10^7	17	2.3×10^{-16}	3.8×10^{-15}	1.6×10^{-16}	6.9×10^{-14}
Cube	5×10^7	17	2.4×10^{-16}	2.5×10^{-15}	8.7×10^{-17}	2.1×10^{-14}
Dodecahedron	5×10^7	17	3.5×10^{-16}	1.9×10^{-15}	6.8×10^{-17}	6.2×10^{-15}
Hollow cube	5×10^7	17	2.0×10^{-16}	3.2×10^{-15}	1.3×10^{-16}	3.4×10^{-14}
Stanford bunny	1×10^3	13	5.8×10^{-15}	4.8×10^{-14}	2.7×10^{-15}	3.1×10^{-14}

section 7.2, but with the addition of a post-sample translation of the polyhedron along \mathbf{e}_z so as for one of its vertices to lie exactly on the paraboloid. At least one iteration of the nudging procedure of Section 5 is therefore required for each random case. The results of this random parameter sweep are given in Table 4, displaying errors of a similar order of magnitude as for the random parameter sweep presented in Subsection 7.2.

7.4. Timings. In order to assess the performances of our implementation, the time required for each moment estimation of the random parameter sweeps presented in Subsections 7.2 and 7.3 has been measured using OpenMP’s `omp_get_wtime()` function [9], which has a precision of 1 nanosecond on the workstation that we used. The characteristics of this workstation are summarized in Table 5. The C++ code implementing the closed-form expressions presented in Section 3 has been compiled with the GNU 10.3.0 suite of compilers [1], using the flags given in Table 5.

The timings are summarized in Table 6, which shows the average moment calculation time for the zeroth moment only, and for both the zeroth and first moments. Overall, the average time for calculating the first moments of a polyhedron clipped by a paraboloid is less than 1 μ s per face of the original polyhedron. A direct comparison can be made with the half-space clipping of [8], which used the same workstation as the current work. It transpires that the clipping of a cube by a paraboloid is on average less than 6 times more expensive than its clipping by a plane. When the

TABLE 3

Graded parameter sweep results. For each geometry, we provide: the number of tests conducted, the number of recursive levels used for the AMR reference moment calculation, as well as the average and maximum errors in the estimation of the zeroth and first moments.

Geometry	Number of tests	AMR levels	Zeroth moment error		First moments error	
			Average	Maximum	Average	Maximum
Tetrahedron	$5^6 \times 11^2$	22	2.7×10^{-16}	4.9×10^{-15}	2.1×10^{-16}	3.9×10^{-14}
Cube	$5^6 \times 11^2$	22	2.6×10^{-16}	4.7×10^{-15}	1.0×10^{-16}	2.1×10^{-14}
Dodecahedron	$5^6 \times 11^2$	22	3.7×10^{-16}	1.7×10^{-15}	8.3×10^{-17}	5.4×10^{-15}
Hollow cube	$5^6 \times 11^2$	22	2.0×10^{-16}	4.2×10^{-15}	1.6×10^{-16}	4.8×10^{-14}

TABLE 4

Random parameter sweep with one vertex of the polyhedron lying exactly on the paraboloid. We provide: the number of tests conducted, the number of recursive levels used for the AMR reference moment calculation, as well as the average and maximum errors in the estimation of the zeroth and first moments.

Geometry	Number of tests	AMR levels	Zeroth moment error		First moments error	
			Average	Maximum	Average	Maximum
Cube (vertex on \mathcal{S})	5×10^7	17	2.4×10^{-16}	7.8×10^{-15}	1.8×10^{-16}	4.5×10^{-14}

nudging procedure of Section 5 is required, as is the case for each realization of the modified random parameter sweep presented in Subsection 7.3, the timings of which are shown in the last row of Table 6, an increase in cost by a factor of about 200 is observed. This is mostly due to the switch to quadruple precision that is operated in conjunction with the random translation and rotation of the polyhedron by a value of ϵ_{nudge} . It should be noted that such cases occur extremely rarely, unless they are manually engineered as in Subsection 7.3.

8. Conclusions. We have derived closed-form expressions for the first moments of a polyhedron clipped by a paraboloid, enabling their robust machine-accurate estimation at a computational cost that is considerably lower than with any other available approach. These expressions have been obtained by consecutive applications of the divergence theorem, transforming the three-dimensional integrals that are the zeroth and first moments of the clipped polyhedron into a sum of one-dimensional integrals. This requires parametrizing the conic section arcs resulting from the intersection of the paraboloid with the polyhedron’s faces, which we have chosen to express as rational quadratic Bézier curves. The moments of the clipped polyhedron can, as a result, be expressed as the sum of three main contributions that are function of the polyhedron vertices and of the coefficients of the paraboloid. These expressions do not differ based on the type of paraboloid that is considered (elliptic, hyperbolic, or parabolic). Making use of this parametrization, we also show how to express integrals over the curved faces of the clipped polyhedron as the sums of one-dimensional integrals. When ambiguous discrete intersection topologies are detected, e.g., when the paraboloid is tangent to an edge of the polyhedron or intersects the polyhedron at the location of one of its vertices, a nudging procedure is triggered so as to guarantee robust moment estimations. A series of millions of intersection configurations that are randomly chosen, as well as manually engineered so as to raise singular intersection configurations and/or ambiguous discrete topologies, have been tested. These showcase an average moment estimation error that is of the order of the machine-zero, and a maximum error that is about one order of magnitude larger. The timing of these moment estimations shows that the clipping of a polyhedron by a paraboloid, with our approach, is on average about 6 times more expensive than its clipping by a

TABLE 5

Characteristics of the workstation used for the timing results presented in Subsection 7.4 (this is the same workstation as used in [8]).

CPU	vendor_id	GenuineIntel
	CPU family	6
	Model	158
	Model name	Intel(R) Core(TM) i7-8700K CPU 3.70 GHz
	Stepping	10
	Microcode	0xca
	Min/max clock CPU frequency	800 MHz – 4.70 GHz
	CPU asserted frequency	4.0 GHz
	Cache size	12288 KB
Compiler	Suite	GNU
	Version	10.3.0
	Flags	-O3 -march=native -DNDEBUG -DNDEBUG_PERF*

* -DNDEBUG_PERF is an IRL-specific compiler flag that disables additional debugging assertions [8].

TABLE 6

Timings for the random parameter sweeps presented in Subsections 7.2 and 7.3. The timings measured in [8] for the clipping of a cube by a plane are provided for reference (they were measured on the same workstation as used for the current work).

Geometry	Number of tests	Average moment calculation time			
		Zeroth moment only		Zeroth and first moments	
		$\mu\text{s}/\text{test}$	$\mu\text{s}/\text{test/face}$	$\mu\text{s}/\text{test}$	$\mu\text{s}/\text{test/face}$
Tetrahedron	5×10^7	1.07	0.27	1.89	0.47
Cube	5×10^7	1.57	0.26	2.53	0.42
Cube (half-space clipping) [8]	15×10^6	0.27	0.05	--	--
Dodecahedron	5×10^7	2.66	0.22	3.96	0.33
Hollow cube	5×10^7	3.27	0.27	5.12	0.43
Stanford bunny	1×10^3	6.22×10^4	0.19	6.41×10^4	0.19
Cube (vertex on \mathcal{S})	5×10^7	236	39.3	541	90.2

plane.

The code used to produce the results presented in this manuscript is openly available as part of the Interface Reconstruction Library at <https://github.com/robert-chiodi/interface-reconstruction-library>⁴.

Acknowledgments. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101026017 . Robert Chiodi was sponsored by the Office of Naval Research (ONR) as part of the Multidisciplinary University Research Initiatives (MURI) Program, under grant number N00014-16-1-2617. The views and conclusions contained herein are those of the authors only and should not be interpreted as representing those of ONR, the U.S. Navy or the U.S. Government.

Appendix A. Third contribution to the moments. The third contribution to the moments of face \mathcal{F}_i , $\mathcal{M}_i^{\hat{P}_3}$, is obtained by calculating the integral on the right-hand side of Eq. (3.26) using the definitions in Eqs. (3.12), (3.14), and (3.15), as well as the facts that $\forall j \in \{1, \dots, n_{\partial\hat{\mathcal{F}}_i}\}$,

$$(A.1) \quad z_{i,j} = \delta_i - \lambda_i x_{i,j} - \tau_i y_{i,j},$$

⁴The commit hash corresponding to the version of the code used in this manuscript is `ad6ecbd`.

$$(A.2) \quad 1_{i,j}^{\partial\tilde{S}} \neq 0 \Leftrightarrow \begin{cases} z_{i,j,0} = -\alpha x_{i,j,0}^2 - \beta y_{i,j,0}^2 \\ z_{i,j,1} = -\alpha x_{i,j,1}^2 - \beta y_{i,j,1}^2 \end{cases},$$

and that the control point $\mathbf{x}_{i,j}^*$ belongs to the plane containing the face \mathcal{F}_i hence

$$(A.3) \quad z_{i,j}^* = \delta_i - \lambda_i x_{i,j}^* - \tau_i y_{i,j}^*.$$

After substitution of these expressions in Eq. (3.26) and some simplification, $\mathcal{M}_i^{\hat{\mathcal{P}}_3}$ can be shown to read as in Eq. (3.46), with

$$(A.4) \quad \mathcal{B}^{(3)}(w, \mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) = \text{diag}(\mathcal{E}(w)) (\mathcal{C}(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) \mathcal{K}\mathcal{D}(w)),$$

where: \mathcal{C} is a 4×12 matrix whose non-zero coefficients contributing to $\mathbf{M}_0^{\hat{\mathcal{P}}}$ are given as

$$(A.5) \quad \mathcal{C}_{1,1} = \alpha(x_a^2 + 2x_a x_b + x_b^2) + \beta(y_a^2 + 2y_a y_b + y_b^2) - 2z_a - 2z_b,$$

$$(A.6) \quad \begin{aligned} \mathcal{C}_{1,2} = & \alpha(x_a^2 + 2x_a x_b + x_b^2 + 4x_a x_c + 4x_b x_c + 4x_c^2) \\ & + \beta(y_a^2 + 2y_a y_b + y_b^2 + 4y_a y_c + 4y_b y_c + 4y_c^2) \\ & - 4z_a - 4z_b - 8z_c, \end{aligned}$$

$$(A.7) \quad \mathcal{C}_{1,3} = \alpha x_c^2 + \beta y_c^2 - z_c,$$

whose non-zero coefficients contributing to $\mathbf{e}_x \cdot \mathbf{M}_1^{\hat{\mathcal{P}}}$ are given as

$$(A.8) \quad \begin{aligned} \mathcal{C}_{2,4} = & \beta(-12x_a y_a y_b + 12x_a y_b^2 + 12x_b y_a^2 - 12x_b y_a y_b) \\ & - 6x_a z_a + 6x_a z_b + 6x_b z_a - 6x_b z_b, \end{aligned}$$

$$(A.9) \quad \begin{aligned} \mathcal{C}_{2,5} = & \alpha(-12x_a x_c^2 + 60x_a x_c x_b - 12x_c^2 x_b) \\ & + \beta(28x_a y_a y_c - 8x_a y_a y_b - 4x_a y_c^2 + 20x_a y_c y_b + 8x_a y_b^2 \\ & - 28x_c y_a^2 - 8x_c y_a y_c + 20x_c y_a y_b - 8x_c y_c y_b - 28x_c y_b^2 \\ & + 8x_b y_a^2 + 20x_b y_a y_c - 8x_b y_a y_b - 4x_b y_c^2 + 28x_b y_c y_b) \\ & + 10x_a z_a + 20x_a z_c + 14x_a z_b - 22x_c z_a - 8x_c z_c \\ & - 22x_c z_b + 14x_b z_a + 20x_b z_c + 10x_b z_b, \end{aligned}$$

$$(A.10) \quad \begin{aligned} \mathcal{C}_{2,6} = & \alpha(-36x_a x_c^2 + 12x_a x_c x_b + 12x_c^3 - 36x_c^2 x_b) \\ & + \beta(-12x_a y_c^2 + 4x_a y_c y_b - 24x_c y_a y_c + 4x_c y_a \\ & y_b + 12x_c y_c^2 - 24x_c y_c y_b + 4x_b y_a y_c - 12x_b y_c^2) \\ & - 10x_a z_c + 2x_a z_b - 10x_c z_a - 12x_c z_c - 10x_c z_b \\ & + 2x_b z_a - 10x_b z_c, \end{aligned}$$

$$(A.11) \quad \mathcal{C}_{2,7} = 2\alpha x_c^3 + 2\beta x_c y_c^2 + 2x_c z_c,$$

whose non-zero coefficients contributing to $\mathbf{e}_y \cdot \mathbf{M}_1^{\hat{\mathcal{P}}}$ are given as

$$(A.12) \quad \begin{aligned} \mathcal{C}_{3,4} = & \alpha(12x_a^2 y_b - 12x_a x_b y_a - 12x_a x_b y_b + 12x_b^2 y_a) \\ & - 6y_a z_a + 6y_a z_b + 6y_b z_a - 6y_b z_b \end{aligned}$$

$$(A.13) \quad \begin{aligned} \mathcal{C}_{3,5} = & \alpha(-28x_a^2 y_c + 8x_a^2 y_b + 28x_a x_c y_a - 8x_a x_c y_c + 20x_a x_c y_b \\ & - 8x_a x_b y_a + 20x_a x_b y_c - 8x_a x_b y_b - 4x_c^2 y_a - 4x_c^2 y_b \\ & + 20x_c x_b y_a - 8x_c x_b y_c + 28x_c x_b y_b + 8x_b^2 y_a - 28x_b^2 y_c) \\ & + \beta(-12y_a y_c^2 + 60y_a y_c y_b - 12y_c^2 y_b) \end{aligned}$$

$$\begin{aligned}
& + 10y_a z_a + 20y_a z_c + 14y_a z_b - 22y_c z_a - 8y_c z_c \\
& - 22y_c z_b + 14y_b z_a + 20y_b z_c + 10y_b z_b, \\
(A.14) \quad \mathcal{C}_{3,6} = & \alpha (-24x_a x_c y_c + 4x_a x_c y_b + 4x_a x_b y_c - 12x_c^2 y_a \\
& + 12x_c^2 y_c - 12x_c^2 y_b + 4x_c x_b y_a - 24x_c x_b y_c) \\
& + \beta (-36y_a y_c^2 + 12y_a y_c y_b + 12y_c^3 - 36y_c^2 y_b) \\
& - 10y_a z_c + 2y_a z_b - 10y_c z_a - 12y_c z_c - 10y_c z_b \\
& + 2y_b z_a - 10y_b z_c,
\end{aligned}$$

$$(A.15) \quad \mathcal{C}_{3,7} = 2\alpha x_c^2 y_c + 2\beta y_c^3 + 2y_c z_c,$$

and whose non-zero coefficients contributing to $\mathbf{e}_z \cdot \mathbf{M}_1^{\hat{P}}$ are given as

$$\begin{aligned}
(A.16) \quad \mathcal{C}_{4,8} = & +\alpha\beta (-42y_a^2 x_a^2 - 10y_b^2 x_a^2 - 28y_a y_b x_a^2 - 28x_b y_a^2 x_a \\
& - 28x_b y_b^2 x_a - 40x_b y_a y_b x_a - 10x_b^2 y_a^2 - 42x_b^2 y_b^2 - 28x_b^2 y_a y_b) \\
& + \alpha^2 (-21x_a^4 - 28x_b x_a^3 - 30x_b^2 x_a^2 - 28x_b^3 x_a - 21x_b^4) \\
& + \beta^2 (-21y_a^4 - 28y_b y_a^3 - 30y_b^2 y_a^2 - 28y_b^3 y_a - 21y_b^4) \\
& 40z_a^2 + 40z_b^2 + 48z_a z_b,
\end{aligned}$$

$$\begin{aligned}
(A.17) \quad \mathcal{C}_{4,9} = & \alpha\beta (-7y_c^2 x_a^2 - 10y_b^2 x_a^2 + 63y_a y_c x_a^2 - 21y_a y_b x_a^2 + 35y_c y_b x_a^2 \\
& + 63x_c y_a^2 x_a - 21x_b y_a^2 x_a - 10x_b y_c^2 x_a + 35x_c y_b^2 x_a \\
& - 21x_b y_b^2 x_a - 28x_c y_a y_c x_a + 70x_b y_a y_c x_a + 70x_c y_a y_b x_a \\
& - 40x_b y_a y_b x_a - 20x_c y_c y_b x_a + 70x_b y_c y_b x_a - 7x_c^2 y_a^2 \\
& - 10x_b^2 y_a^2 + 35x_c x_b y_a^2 - 7x_b^2 y_c^2 - 7x_c^2 y_b^2 + 63x_c x_b y_b^2 \\
& + 35x_b^2 y_a y_c - 20x_c x_b y_a y_c - 10x_c^2 y_a y_b - 21x_b^2 y_a y_b \\
& + 70x_c x_b y_a y_b + 63x_b^2 y_c y_b - 28x_c x_b y_c y_b) \\
& + \alpha^2 (63x_c x_a^3 - 21x_b x_a^3 - 21x_c^2 x_a^2 - 30x_b^2 x_a^2 + 105x_c x_b x_a^2 \\
& - 21x_b^3 x_a + 105x_c x_b^2 x_a - 30x_c^2 x_b x_a + 63x_c x_b^3 - 21x_c^2 x_b^2) \\
& + \beta^2 (63y_c y_a^3 - 21y_b y_a^3 - 21y_c^2 y_a^2 - 30y_b^2 y_a^2 + 105y_c y_b y_a^2 \\
& - 21y_b^3 y_a + 105y_c y_b^2 y_a - 30y_c^2 y_b y_a + 63y_c y_b^3 - 21y_c^2 y_b^2) \\
& - 30z_a^2 + 12z_c^2 - 30z_b^2 - 60z_a z_c - 24z_a z_b - 60z_c z_b,
\end{aligned}$$

$$\begin{aligned}
(A.18) \quad \mathcal{C}_{4,10} = & \alpha\beta (-56y_c^2 x_a^2 - 2y_b^2 x_a^2 + 28y_c y_b x_a^2 + 84x_c y_c^2 x_a \\
& - 92x_b y_c^2 x_a + 28x_c y_b^2 x_a - 224x_c y_a y_c x_a + 56x_b y_a y_c x_a \\
& + 56x_c y_a y_b x_a - 8x_b y_a y_b x_a - 184x_c y_c y_b x_a \\
& + 56x_b y_c y_b x_a - 56x_c^2 y_a^2 - 2x_b^2 y_a^2 + 28x_c x_b y_a^2 \\
& - 12x_c^2 y_c^2 - 56x_b^2 y_c^2 + 84x_c x_b y_c^2 - 56x_c^2 y_b^2 \\
& + 84x_c^2 y_a y_c + 28x_b^2 y_a y_c - 184x_c x_b y_a y_c \\
& - 92x_c^2 y_a y_b + 56x_c x_b y_a y_b + 84x_c^2 y_c y_b - 224x_c x_b y_c y_b) \\
& + \alpha^2 (-6x_c^4 + 84x_a x_c^3 + 84x_b x_c^3 - 168x_a^2 x_c^2 - 168x_b^2 x_c^2 \\
& - 276x_a x_b x_c^2 + 84x_a x_b^2 x_c + 84x_a^2 x_b x_c - 6x_a^2 x_b^2) \\
& + \beta^2 (-6y_c^4 + 84y_a y_c^3 + 84y_b y_c^3 - 168y_a^2 y_c^2 - 168y_b^2 y_c^2 \\
& - 276y_a y_b y_c^2 + 84y_a y_b^2 y_c + 84y_a^2 y_b y_c - 6y_a^2 y_b^2) \\
& + 15z_a^2 + 24z_c^2 + 15z_b^2 + 120z_a z_c - 6z_a z_b + 120z_c z_b,
\end{aligned}$$

$$(A.19) \quad \begin{aligned} \mathcal{C}_{4,11} = & \alpha\beta(-12y_c^2x_c^2 + 14y_ay_cx_c^2 - 2y_ay_bx_c^2 \\ & + 14y_cy_bx_c^2 + 14x_ay_c^2x_c + 14x_by_c^2x_c \\ & - 4x_by_ay_cx_c - 4x_ay_bx_c - 2x_ax_by_c^2) \\ & + \alpha^2(-6x_c^4 + 14x_ax_c^3 + 14x_bx_c^3 - 6x_ax_bx_c^2) \\ & + \beta^2(-6y_c^4 + 14y_ay_c^3 + 14y_by_c^3 - 6y_ay_by_c^2) \\ & - 7z_c^2 - 5z_az_c + z_az_b - 5z_cz_b, \end{aligned}$$

$$(A.20) \quad \mathcal{C}_{4,12} = -2\alpha\beta y_c^2x_c^2 - \alpha^2x_c^4 - \beta^2y_c^4 + z_c^2;$$

\mathcal{K} is the 12×10 matrix given as

$$(A.21) \quad \mathcal{K} = \left[\begin{array}{cccccccccc} -\frac{3}{8} & \frac{31}{48} & -\frac{7}{8} & -\frac{1}{16} & 0 & \frac{1}{24} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{6} & \frac{5}{8} & -\frac{3}{16} & 0 & \frac{1}{24} & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & -3 & \frac{11}{6} & -2 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{32} & \frac{93}{2240} & 0 & -\frac{163}{3360} & 0 & \frac{5}{168} & 0 & -\frac{1}{140} & 0 & 0 \\ 0 & \frac{1}{70} & -\frac{1}{16} & \frac{29}{1120} & 0 & -\frac{19}{1680} & 0 & \frac{1}{420} & 0 & 0 \\ 0 & -\frac{1}{210} & 0 & \frac{1}{21} & -\frac{1}{8} & \frac{13}{560} & 0 & -\frac{1}{280} & 0 & 0 \\ 0 & \frac{1}{35} & 0 & -\frac{16}{105} & 0 & \frac{56}{105} & -1 & \frac{1}{14} & 0 & 0 \\ -\frac{1}{128} & \frac{193}{16128} & 0 & -\frac{149}{8064} & 0 & \frac{19}{1120} & 0 & \frac{41}{5040} & 0 & \frac{1}{630} \\ 0 & \frac{4}{945} & -\frac{1}{48} & \frac{65}{6048} & 0 & -\frac{1}{144} & 0 & \frac{11}{3780} & 0 & -\frac{1}{1890} \\ 0 & -\frac{1}{1890} & 0 & \frac{13}{1890} & -\frac{1}{48} & \frac{11}{2016} & 0 & -\frac{5}{3024} & 0 & \frac{1}{3780} \\ 0 & \frac{1}{315} & 0 & -\frac{1}{45} & 0 & \frac{4}{35} & -\frac{1}{4} & \frac{17}{504} & 0 & -\frac{1}{252} \\ 0 & -\frac{1}{63} & 0 & \frac{29}{315} & 0 & -\frac{26}{105} & 0 & \frac{194}{315} & -1 & \frac{1}{18} \end{array} \right];$$

\mathcal{D} is the vector given as

$$(A.22) \quad \begin{aligned} \mathcal{D}(w) = & \left[0 \quad w^2 \quad 0 \quad w^4 \quad 0 \quad w^6 \quad 0 \quad w^8 \quad 0 \quad w^{10} \right]^\top \\ & + \Theta(w) \left[w \quad 0 \quad w^3 \quad 0 \quad w^5 \quad 0 \quad w^7 \quad 0 \quad w^9 \quad 0 \right]^\top; \end{aligned}$$

and \mathcal{E} is the vector given as

$$(A.23) \quad \mathcal{E}(w) = \left[\Lambda(w)^3 \quad \Lambda(w)^4 \quad \Lambda(w)^4 \quad \Lambda(w)^5 \right]^\top,$$

with

$$(A.24) \quad \Theta(w) = \begin{cases} \arctan\left(\frac{1-w}{\sqrt{1-w^2}}\right) \frac{1}{\sqrt{1-w^2}} & 0 < w < 1 \\ \operatorname{arctanh}\left(\frac{w-1}{\sqrt{w^2-1}}\right) \frac{1}{\sqrt{w^2-1}} & 1 \leq w \end{cases},$$

$$(A.25) \quad \Lambda(w) = \frac{1}{(w-1)(w+1)}.$$

It should be noted that the naive implementation of these expressions may lead to significant round-off errors when w is in the vicinity of 1. To avoid such problems, we resort to the Taylor series expansion of $\mathcal{B}^{(3)}$ around $w = 1$ for its numerical estimation. Using 64-bit floating-point arithmetics, we have found that the Taylor series expansion of $\mathcal{B}^{(3)}$ to order 40 for $w \in [0.35, 1.7]$ is sufficient for producing near machine-zero estimates. This implementation has been used for producing the results presented in Section 7.

REFERENCES

- [1] *The GNU Compiler Collection 10.3*, <https://gcc.gnu.org/onlinedocs/gcc-10.3.0> (accessed 2022-12-09).
- [2] A. ABEDIAN, J. PARVIZIAN, A. DÜSTER, H. KHADEMZYADEH, AND E. RANK, *Performance of different integration schemes in facing discontinuities in the Finite Cell Method*, International Journal of Computational Methods, 10 (2013), p. 1350002, <https://doi.org/10.1142/S0219876213500023>, <https://www.worldscientific.com/doi/abs/10.1142/S0219876213500023> (accessed 2022-07-27).
- [3] P. ANTOLIN AND T. HIRSCHLER, *Quadrature-free Immersed Isogeometric Analysis*, July 2021, <http://arxiv.org/abs/2107.09024> (accessed 2022-07-21). arXiv:2107.09024 [cs, math].
- [4] S. BNÀ, S. MANSERVISI, R. SCARDOVELLI, P. YECKO, AND S. ZALESKI, *Numerical integration of implicit functions for the initialization of the VOF function*, Computers & Fluids, 113 (2015), pp. 42–52, <https://doi.org/10.1016/j.compfluid.2014.04.010>, <http://www.sciencedirect.com/science/article/pii/S0045793014001480>.
- [5] E. BURMAN, S. CLAUS, P. HANSBO, M. G. LARSON, AND A. MASSING, *CutFEM: Discretizing geometry and partial differential equations*, International Journal for Numerical Methods in Engineering, 104 (2015), pp. 472–501, <https://doi.org/10.1002/nme.4823>, <https://onlinelibrary.wiley.com/doi/10.1002/nme.4823> (accessed 2022-07-21).
- [6] A. CHIERICI, L. CHIRCO, V. LE CHENADEC, R. SCARDOVELLI, P. YECKO, AND S. ZALESKI, *An optimized Vof library to initialize the volume fraction field*, Computer Physics Communications, (2022), p. 108506, <https://doi.org/10.1016/j.cpc.2022.108506>, <https://linkinghub.elsevier.com/retrieve/pii/S0010465522002259> (accessed 2022-08-25).
- [7] E. B. CHIN AND N. SUKUMAR, *An efficient method to integrate polynomials over polytopes and curved solids*, Computer Aided Geometric Design, 82 (2020), p. 101914, <https://doi.org/10.1016/j.cagd.2020.101914>, <https://linkinghub.elsevier.com/retrieve/pii/S0167839620301011> (accessed 2022-07-21).
- [8] R. CHIODI AND O. DESJARDINS, *General, robust, and efficient polyhedron intersection in the interface reconstruction library*, Journal of Computational Physics, 449 (2022), p. 110787, <https://doi.org/10.1016/j.jcp.2021.110787>, <https://linkinghub.elsevier.com/retrieve/pii/S0021999121006823> (accessed 2021-10-20).
- [9] L. DAGUM AND R. MENON, *OpenMP: an industry standard API for shared-memory programming*, IEEE Computational Science and Engineering, 5 (1998), pp. 46–55, <https://doi.org/10.1109/99.660313>, <http://ieeexplore.ieee.org/document/660313/> (accessed 2022-09-12).
- [10] S. C. DIVI, C. V. VERHOOSEL, F. AURICCHIO, A. REALI, AND E. H. VAN BRUMMELEN, *Error-estimate-based adaptive integration for immersed isogeometric analysis*, Computers & Mathematics with Applications, 80 (2020), pp. 2481–2516, <https://doi.org/10.1016/j.camwa.2020.03.026>, <https://linkinghub.elsevier.com/retrieve/pii/S0898122120301590> (accessed 2022-07-27).
- [11] D. A. DUNAVANT, *High degree efficient symmetrical Gaussian quadrature rules for the triangle*, International Journal for Numerical Methods in Engineering, 21 (1985), pp. 1129–1148, <https://doi.org/10.1002/nme.1620210612>, <https://onlinelibrary.wiley.com/doi/10.1002/nme.1620210612> (accessed 2022-07-29).
- [12] C. ENGWER, S. MAY, A. NÜSSING, AND F. STREITBÜRGER, *A Stabilized DG Cut Cell Method for Discretizing the Linear Transport Equation*, SIAM Journal on Scientific Computing, 42 (2020), pp. A3677–A3703, <https://doi.org/10.1137/19M1268318>, <https://epubs.siam.org/doi/10.1137/19M1268318> (accessed 2023-03-02).
- [13] M. EVANS AND T. SWARTZ, *Approximating integrals via Monte Carlo and deterministic methods*, no. 20 in Oxford statistical science series, Oxford University Press, Oxford ; New York, 2000.
- [14] G. E. FARIN, *From conics to NURBS: A tutorial and survey*, IEEE Computer Graphics and Applications, 12 (1992), pp. 78–86, <https://doi.org/10.1109/38.156017>, <http://ieeexplore.ieee.org/document/156017/> (accessed 2023-02-05).
- [15] G. E. FARIN, *Curves and surfaces for CAGD: a practical guide*, Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kaufmann, San Francisco, CA, 5th ed ed., 2001.
- [16] T. HAHN, *Cuba—a library for multidimensional numerical integration*, Computer Physics Communications, 168 (2005), pp. 78–95, <https://doi.org/10.1016/j.cpc.2005.01.010>, <https://linkinghub.elsevier.com/retrieve/pii/S0010465505000792> (accessed 2022-07-28).
- [17] E. HARTMANN, *\$G^2\$ interpolation and blending on surfaces*, The Visual Computer, 12 (1996), pp. 181–192, <https://doi.org/10.1007/BF01782321>, <http://link.springer.com/10.1007/BF01782321> (accessed 2023-02-05).
- [18] T. HUGHES, J. COTTRELL, AND Y. BAZILEVS, *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*, Computer Methods in Applied Mechanics

- and Engineering, 194 (2005), pp. 4135–4195, <https://doi.org/10.1016/j.cma.2004.10.008>, <https://linkinghub.elsevier.com/retrieve/pii/S0045782504005171> (accessed 2022-07-21).
- [19] B. W. JONES, A. G. MALAN, AND N. A. ILANGAKOON, *The initialisation of volume fractions for unstructured grids using implicit surface definitions.*, Computers & Fluids, 179 (2019), pp. 194–205, <https://doi.org/10.1016/j.compfluid.2018.10.021>, <https://linkinghub.elsevier.com/retrieve/pii/S0045793018308338> (accessed 2020-02-05).
- [20] W. KAHAN, *Pracniques: further remarks on reducing truncation errors*, Communications of the ACM, 8 (1965), p. 40, <https://doi.org/10.1145/363707.363723>, <https://dl.acm.org/doi/10.1145/363707.363723> (accessed 2022-09-12).
- [21] J. KROMER AND D. BOTHE, *Highly accurate computation of volume fractions using differential geometry*, Journal of Computational Physics, 396 (2019), pp. 761–784, <https://doi.org/10.1016/j.jcp.2019.07.005>, <http://www.sciencedirect.com/science/article/pii/S0021999119304899> (accessed 2019-10-02).
- [22] J. KROMER AND D. BOTHE, *Third-order accurate initialization of VOF volume fractions on unstructured grids with arbitrary polyhedral cells*, arXiv:2111.01073 [cs, math], (2021), <http://arxiv.org/abs/2111.01073> (accessed 2021-11-02). arXiv: 2111.01073.
- [23] L. KUDELA, N. ZANDER, S. KOLLMANNSSBERGER, AND E. RANK, *Smart octrees: Accurately integrating discontinuous functions in 3D*, Computer Methods in Applied Mechanics and Engineering, 306 (2016), pp. 406–426, <https://doi.org/10.1016/j.cma.2016.04.006>, <https://linkinghub.elsevier.com/retrieve/pii/S0045782516301578> (accessed 2022-07-27).
- [24] S. LODHA AND J. WARREN, *Bézier representation for quadric surface patches*, Computer-Aided Design, 22 (1990), pp. 574–579, [https://doi.org/10.1016/0010-4485\(90\)90042-B](https://doi.org/10.1016/0010-4485(90)90042-B), <https://linkinghub.elsevier.com/retrieve/pii/001044859090042B> (accessed 2022-02-15).
- [25] Y. RENARDY AND M. RENARDY, *PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method*, Journal of Computational Physics, 183 (2002), pp. 400–421, <https://doi.org/10.1006/jcph.2002.7190>, <http://linkinghub.elsevier.com/retrieve/pii/S0021999102971901>.
- [26] R. SEVILLA, S. FERNÁNDEZ-MÉNDEZ, AND A. HUERTA, *NURBS-enhanced finite element method (NEFEM)*, International Journal for Numerical Methods in Engineering, 76 (2008), pp. 56–83, <https://doi.org/10.1002/nme.2311>, <https://onlinelibrary.wiley.com/doi/10.1002/nme.2311> (accessed 2022-07-20).
- [27] S. J. TELLER AND C. H. SÉQUIN, *Modeling Implicit Quadrics and Free-form Surfaces With Trimmed Rational Quadratic Bezier Patches*. and Carlo H. Séquin. (CSD-90-577), tech. report, Computer Science Division, U.C. Berkeley, June 1990, <https://www2.eecs.berkeley.edu/Pubs/TechRpts/1990/5528.html>.
- [28] T. TOLLE, D. GRÜNDING, D. BOTHE, AND T. MARIĆ, *triSurfaceImmersion: Computing volume fractions and signed distances from triangulated surfaces immersed in unstructured meshes*, Computer Physics Communications, 273 (2022), p. 108249, <https://doi.org/10.1016/j.cpc.2021.108249>, <https://linkinghub.elsevier.com/retrieve/pii/S0010465521003611> (accessed 2021-12-20).
- [29] G. TURK, *Stanford 3D Scanning Repository*, 1994, <http://graphics.stanford.edu/data/3Dscanrep/> (accessed 2022-12-09).