Fast and Accurate Computation of Polyhedral Mass Properties

Brian Mirtich

University of California at Berkeley

Abstract

The location of a body's center of mass, and its moments and products of inertia about various axes are important physical quantities needed for any type of dynamic simulation or physical based modeling. We present an algorithm for automatically computing these quantities for a general class of rigid bodies: those composed of uniform density polyhedra. The mass integrals may be converted into volume integrals under these assumptions, and the bulk of the paper is devoted to the computation of these volume integrals. Our algorithm is based on a three step reduction of the volume integrals to successively simpler integrals. The algorithm is designed to minimize the numerical errors that can result from poorly conditioned alignment of polyhedral faces. It is also designed for efficiency. All required volume integrals of a polyhedron are computed together during a single walk over the boundary of the polyhedron; exploiting common subexpressions reduces floating point operations. We present numerical results detailing the speed and accuracy of the algorithm, and also give a complete low level pseudocode description.

1 Introduction

Dynamic simulation of rigid-body systems requires several parameters describing the mass distribution of rigid bodies: the total mass (a scalar), the location of the center of mass (3 parameters), and the moments and products of inertia about the center of mass (6 parameters). One can always find a *body frame*, with origin at the body's center of mass and axes aligned with its principle axes of inertia, in which the entire mass distribution can be described with a reduced set of four parameters. Nevertheless, the larger parameterization is still needed as a starting point.

This paper shows how to efficiently and accurately compute the needed data. The only restrictions are that the body in question be a disjoint union of uniform density polyhedra, given by a boundary representation. We assume one can enumerate over the faces of the polyhedra, and for each face, one can enumerate over the vertices in counter-clockwise order. The algorithm is exact, and linear in the number of vertices, edges, or faces of the polyhedra.

The problem of computing mass properties of solid objects has been studied previously. Lee and Requicha give an excellent survey of the various families of algorithms in existence [5]. Our approach is closest to that of Lien and Kajiya, who give an algorithm for computing integrals over arbitrary nonconvex polyhedra, based on a B-rep [6]. It is $O(n)$ in the polyhedron complexity, and fairly easy to code. In contrast to Lien's and Kajiya's algorithm, our algorithm is optimized for computation of mass parameters: it computes all needed mass quantities in parallel during a single traversal of the polyhedra, so that common subexpressions are exploited; it is very fast. In addition, our algorithm adaptively changes the projection direction, thereby reducing numerical errors over those in Lien's and Kajiya's and other algorithms.

2 Rigid body mass parameters

This section defines the rigid body mass parameters, and their relation to dynamic simulation; readers familiar with these topics may jump to Section 3. More detailed treatments of this topic may be found in any dynamics text, such as [4] or [7].

Key quantities in rigid body dynamics are a body's linear momentum $\mathbf{L}$ and angular momentum $\mathbf{H}$, given by

$$\mathbf{L} = m\mathbf{v} \tag{1}$$

$$\mathbf{H} = \mathbf{J}\boldsymbol{\omega} \tag{2}$$

Here, $\mathbf{v}$ and $\boldsymbol{\omega}$ are the linear velocity of the center of mass (which we denote $\mathbf{r}$) and the angular velocity of the body, respectively. The scalar m is the mass of the body, and $\mathbf{J}$ is the 3×3 inertia tensor (also called mass matrix) containing the moments and products of inertia:

$$\mathbf{J} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}. \tag{3}$$

In order to formulate the equations of motion for the body, the quantities m, $\mathbf{J}$, and $\mathbf{r}$ must be determined.

2.1 Computing mass parameters with volume integrals

The initial problem may be expressed as follows:

Problem 1 Given: *A rigid body comprising N parts, $B_1, \ldots B_N$, each a uniform density polyhedron. There are no restrictions on the convexity or genus of the polyhedra, nor on the shape of the bounding faces. For each polyhedron B_i, either its density ρ_i or mass m_i is specified, and the geometries of all of the polyhedra are specified relative to a single reference frame.* Compute: *The mass m, and the reference frame coordinates of the center of mass $\mathbf{r}$ and inertia tensor $\mathbf{J}$ for the entire rigid body.*

The mass m_i and density ρ_i of polyhedral part B_i are related by $m_i = \rho_i V_i$, where V_i is the volume of the polyhedron. Assuming one can compute

$$V_i = \int_{B_i} dV, \tag{4}$$

the masses and densities of each polyhedron can be found. The total mass is $m = \sum_{i=1}^{N} m_i$. The coordinates of the center of mass $\mathbf{r}$ for the entire body are

$$\mathbf{r} = \frac{1}{m} \sum_{i=1}^{N} \rho_i \left(\int_{B_i} x \; dV, \; \int_{B_i} y \; dV, \; \int_{B_i} z \; dV \right)^T. \tag{5}$$

The moments and products of the inertia are given by

$$I'_{xx} = \sum_{i=1}^{N} \rho_i \int_{B_i} (y^2 + z^2) \; dV \tag{6}$$

$$I'_{yy} = \sum_{i=1}^{N} \rho_i \int_{B_i} (z^2 + x^2) \; dV \tag{7}$$

$$I'_{zz} = \sum_{i=1}^{N} \rho_i \int_{B_i} (x^2 + y^2) \; dV \tag{8}$$

$$I'_{xy} = I'_{yx} = \sum_{i=1}^{N} \rho_i \int_{B_i} xy \; dV \tag{9}$$

$$I'_{yz} = I'_{zy} = \sum_{i=1}^{N} \rho_i \int_{B_i} yz \; dV \tag{10}$$

$$I'_{zx} = I'_{xz} = \sum_{i=1}^{N} \rho_i \int_{B_i} zx \; dV \tag{11}$$

2.2 Translating inertias to the center of mass

The inertia quantities in Equations (6-11) are primed because they are not exactly the values appearing in the inertia tensor (3). The values in (6-11) are computed relative to the given reference frame, but the values in the inertia tensor must be computed relative to a frame with origin at the center of mass. One way to accomplish this is to first compute the location of the center of mass in the given frame, using (5), and then to apply a translation to the body which brings the center of mass to the origin. After performing this transformation, the values computed in (6-11) can be directly inserted into the inertia tensor (3).

A better solution is to use the *transfer-of-axis* relations for transferring a moment or product of inertia about one axis to a corresponding one about a parallel axis. To transfer the values computed in (6-11) to a frame at the center of mass, one uses (see [7]):

$$I_{xx} = I'_{xx} - m(r_y^2 + r_z^2) \tag{12}$$
$$I_{yy} = I'_{yy} - m(r_z^2 + r_x^2) \tag{13}$$
$$I_{zz} = I'_{zz} - m(r_x^2 + r_y^2) \tag{14}$$
$$I_{xy} = I'_{xy} - m r_x r_y \tag{15}$$
$$I_{yz} = I'_{yz} - m r_y r_z \tag{16}$$
$$I_{zx} = I'_{zx} - m r_z r_x. \tag{17}$$

The unprimed quantities are inserted into the inertia tensor. If the transfer-of-axis relations are used, one doesn't have to explicitly transform the vertices of the polyhedron after computing the center of mass, hence all of the integrals can be computed simultaneously.

Rigid body dynamics can be computed more efficiently with a reduced set of mass parameters, based on a *body frame*. Computing the body frame amounts to diagonalizing the inertia tensor, a classical problem of linear algebra. The Jacobi method [8] works quite well for this application since $\mathbf{J}$ is real, symmetric, and of moderate size.

3 Overview of volume integration

Equations (4–11) show that all required mass properties can be found from ten integrals over volume, for each of the individual polyhedral components. To simplify notation, we drop the polyhedron index and consider only a single polyhedral body. We write the domain of integration as $\mathcal{V}$ as reminder that it is a volume. The remainder of this paper describes an efficient, exact algorithm for calculating these ten integrals:

$$T_1 = \int_{\mathcal{V}} 1 \, dV \tag{18}$$
$$T_x = \int_{\mathcal{V}} x \, dV \tag{19}$$
$$T_y = \int_{\mathcal{V}} y \, dV \tag{20}$$
$$T_z = \int_{\mathcal{V}} z \, dV \tag{21}$$
$$T_{x^2} = \int_{\mathcal{V}} x^2 \, dV \tag{22}$$
$$T_{y^2} = \int_{\mathcal{V}} y^2 \, dV \tag{23}$$
$$T_{z^2} = \int_{\mathcal{V}} z^2 \, dV \tag{24}$$
$$T_{xy} = \int_{\mathcal{V}} xy \, dV \tag{25}$$
$$T_{yz} = \int_{\mathcal{V}} yz \, dV \tag{26}$$
$$T_{zx} = \int_{\mathcal{V}} zx \, dV \tag{27}$$

Note that each is an integral of a monomial in x, y, and z. The basic idea is to use the divergence theorem to reduce each of the volume integrals (18–27) to a sum of surface integrals over the individual faces of the polyhedron. Each of these surface integrals are evaluated in terms of integrals over a planar projection of the surface. For polygons in the plane, Green's theorem reduces the planar integration to a sum of line integrals around the edges of the polygon. Finally, these line integrals are evaluated directly from the coordinates of the projected vertices of the original polyhedron. Figure 1 illustrates the approach: the volume integral is ultimately reduced to a collection of line integrals in the plane, and the values from these integrations are propagated back into the value of the desired volume integration.

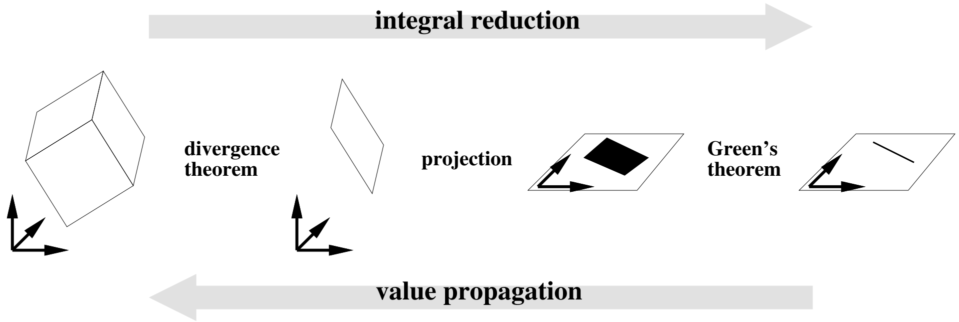

Figure 1: *Strategy for evaluating volume integrals. Complicated integrals are decomposed into successively simpler ones, and the values from evaluating the simplest integrals are combined and propagated back to evaluate the original ones.*

4 Reduction to surface integrals

The first reduction is from an integral over the three-dimensional polyhedral volume to a sum of integrals over its two-dimensional planar faces. These is easily accomplished with the divergence theorem [9]:

Theorem 1 (Divergence) *Let $\mathcal{V}$ be a region in space bounded by the surface $\partial\mathcal{V}$. Let $\hat{\mathbf{n}}$ denote the exterior normal of $\mathcal{V}$ along its boundary $\partial\mathcal{V}$. Then*

$$\int_{\mathcal{V}} \nabla \cdot \mathbf{F} \; dV = \int_{\partial\mathcal{V}} \mathbf{F} \cdot \hat{\mathbf{n}} \; dA \tag{28}$$

for any vector field $\mathbf{F}$ defined on $\mathcal{V}$.

For a polyhedral volume, the right hand side of (28) can be broken up into a summation over faces of constant normal, and so $\hat{\mathbf{n}}$ can be pulled outside the integral. The integrals to be computed, for example $\int_{\mathcal{V}} x \; dV$, do not immediately appear to be of the form in the theorem. But one can find many vector fields whose divergence is the function x; a particularly simple choice is $\mathbf{F}(x, y, z) = (\frac{1}{2}x^2, 0, 0)^T$. This choice has the added advantage that two of its components are identically zero, so that the dot product on the right hand side of (28) becomes a scalar multiplication. By making similar choices for the other integrals we wish to evaluate, and applying the divergence theorem, equations (18–27) become:

$$T_1 = \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_x \int_{\mathcal{F}} x \; dA \tag{29}$$

$$T_x = \frac{1}{2} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_x \int_{\mathcal{F}} x^2 \; dA \tag{30}$$

$$T_y = \frac{1}{2} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_y \int_{\mathcal{F}} y^2 \; dA \tag{31}$$

$$T_z = \frac{1}{2} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_z \int_{\mathcal{F}} z^2 \; dA \tag{32}$$

$$T_{x^2} = \frac{1}{3} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_x \int_{\mathcal{F}} x^3 \; dA \tag{33}$$

$$T_{y^2} = \frac{1}{3} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_y \int_{\mathcal{F}} y^3 \; dA \tag{34}$$

$$T_{z^2} = \frac{1}{3} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_z \int_{\mathcal{F}} z^3 \; dA \tag{35}$$

$$T_{xy} = \frac{1}{2} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_x \int_{\mathcal{F}} x^2 y \; dA \tag{36}$$

$$T_{yz} = \frac{1}{2} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_y \int_{\mathcal{F}} y^2 z \; dA \tag{37}$$

$$T_{zx} = \frac{1}{2} \sum_{\mathcal{F}\in\partial\mathcal{V}} \hat{n}_z \int_{\mathcal{F}} z^2 x \; dA \tag{38}$$

5 Reduction to projection integrals

Green's theorem reduces an integral over a planar region to an integral around its one-dimensional boundary, however one must start with a region *in the plane*. Although the planar faces of the polyhedron are in three-space, one can project them onto one of the coordinate planes. The next theorem relates integrations over the original face to integrations over the projection.

Theorem 2 *Let $\mathcal{F}$ be a polygonal region in α-β-γ space, with surface normal $\hat{\mathbf{n}}$, and lying in the plane*

$$\hat{n}_\alpha \alpha + \hat{n}_\beta \beta + \hat{n}_\gamma \gamma + w = 0. \tag{39}$$

Let Π be the projection of $\mathcal{F}$ into the α-β plane. Then

$$\int_{\mathcal{F}} f(\alpha, \beta, \gamma)\, dA = \frac{1}{|\hat{n}_\gamma|} \int_\Pi f(\alpha, \beta, h(\alpha, \beta))\, d\alpha\, d\beta, \tag{40}$$

where

$$h(\alpha, \beta) = -\frac{1}{\hat{n}_\gamma}(\hat{n}_\alpha \alpha + \hat{n}_\beta \beta + w). \tag{41}$$

Proof: The points $(\alpha, \beta, h(\alpha, \beta))$ lie in the plane of $\mathcal{F}$, so $\mathcal{F}$ is the graph of h over Π. From [1] [Section 17.5, Formula (6)],

$$\int_{\mathcal{F}} f(\alpha, \beta, \gamma)\, dA = \int_\Pi f(\alpha, \beta, h(\alpha, \beta)) \sqrt{1 + \left(\frac{\partial h}{\partial \alpha}\right)^2 + \left(\frac{\partial h}{\partial \beta}\right)^2}\, d\alpha\, d\beta. \tag{42}$$

For our h, the square root in the integrand reduces to $|\hat{n}_\gamma|^{-1}$; the theorem follows. $\square$

This theorem provides the reduction of the integral of a polynomial in α, β, and γ over the face $\mathcal{F}$ to the integral of a polynomial in α and β over the projected region Π. From (39), the constant w can be computed: $w = -\hat{\mathbf{n}} \cdot \mathbf{p}$, where $\mathbf{p}$ is any point on the face $\mathcal{F}$.

Numerical inaccuracy or floating point errors can occur when the face normal $\hat{\mathbf{n}}$ has little or no component in the projection direction; in the extreme situation ($\hat{n}_\gamma = 0$), the face projects to a line segment. To reduce such errors, for a given face the α-β-γ coordinates are always chosen as a right-handed[1] permutation of the of the x-y-z coordinates such that $|\hat{n}_\gamma|$ is maximized. This maximizes the area of the projected shadow in the α-β plane (see Figure 2). Note that a choice can always be found such that $|\hat{n}_\gamma| \geq \sqrt{3}^{-1}$.

Recall the integrals we need over the region $\mathcal{F}$ given in (29–38). Independent of the three possible correspondences between x-y-z and α-β-γ coordinates, they all can be found by computing the following twelve integrals:

$$\int_{\mathcal{F}} \alpha\, dA \quad (43) \qquad \int_{\mathcal{F}} \alpha^2\, dA \quad (46) \qquad \int_{\mathcal{F}} \alpha^3\, dA \quad (49) \qquad \int_{\mathcal{F}} \alpha^2 \beta\, dA \quad (52)$$

$$\int_{\mathcal{F}} \beta\, dA \quad (44) \qquad \int_{\mathcal{F}} \beta^2\, dA \quad (47) \qquad \int_{\mathcal{F}} \beta^3\, dA \quad (50) \qquad \int_{\mathcal{F}} \beta^2 \gamma\, dA \quad (53)$$

$$\int_{\mathcal{F}} \gamma\, dA \quad (45) \qquad \int_{\mathcal{F}} \gamma^2\, dA \quad (48) \qquad \int_{\mathcal{F}} \gamma^3\, dA \quad (51) \qquad \int_{\mathcal{F}} \gamma^2 \alpha\, dA \quad (54)$$

Using Theorem 2, these twelve face integrals can all be reduced to integrals over the projection region Π. For instance,

$$\begin{aligned} \int_{\mathcal{F}} \beta^2 \gamma\, dA &= |\hat{n}_\gamma|^{-1} \int_\Pi \beta^2 \frac{\hat{n}_\alpha \alpha + \hat{n}_\beta \beta + w}{-\hat{n}_\gamma}\, d\alpha\, d\beta \\ &= -|\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-1} \left(\hat{n}_\alpha \int_\Pi \alpha \beta^2\, d\alpha\, d\beta + \hat{n}_\beta \int_\Pi \beta^3\, d\alpha\, d\beta + w \int_\Pi \beta^2\, d\alpha\, d\beta \right) \\ &= -|\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-1} (\hat{n}_\alpha \pi_{\alpha\beta^2} + \hat{n}_\beta \pi_{\beta^3} + w \pi_{\beta^2}), \end{aligned} \tag{55}$$

[1] We require $\hat{\alpha} \times \hat{\beta} = \hat{\gamma}$.

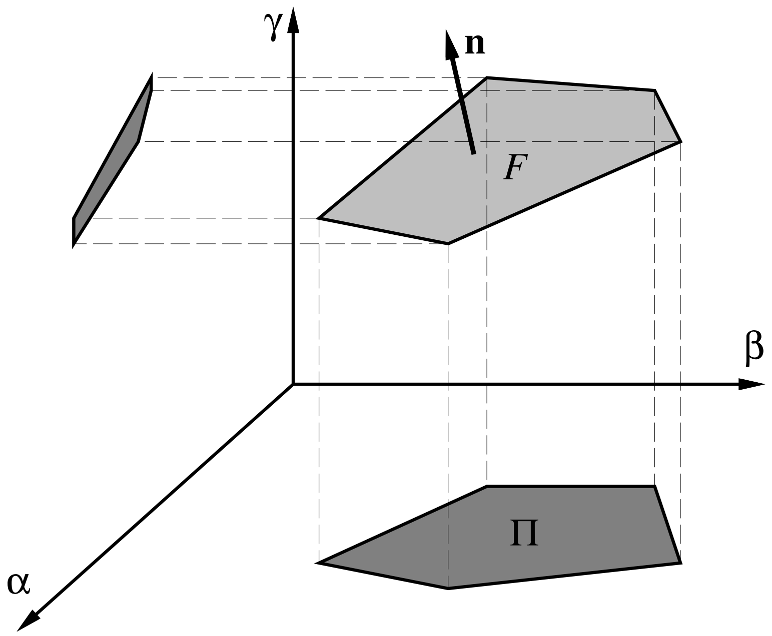

Figure 2: *The α-β-γ axes are a right-handed permutation of the x-y-z axes chosen to maximize the size of the face's projected shadow in the α-β plane.*

where

$$\pi_f = \int_\Pi f \; dA. \tag{56}$$

The complete set of face integrals, reduced to projection integrals with Theorem 2, is shown below:

$$\begin{aligned}
\int_\mathcal{F} \alpha \; dA &= |\hat{n}_\gamma|^{-1} \pi_\alpha & (57)\\
\int_\mathcal{F} \beta \; dA &= |\hat{n}_\gamma|^{-1} \pi_\beta & (58)\\
\int_\mathcal{F} \gamma \; dA &= -|\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-1} (\hat{n}_\alpha \pi_\alpha + \hat{n}_\beta \pi_\beta + w \pi_1) & (59)\\
\int_\mathcal{F} \alpha^2 \; dA &= |\hat{n}_\gamma|^{-1} \pi_{\alpha^2} & (60)\\
\int_\mathcal{F} \beta^2 \; dA &= |\hat{n}_\gamma|^{-1} \pi_{\beta^2} & (61)\\
\int_\mathcal{F} \gamma^2 \; dA &= |\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-2} (\hat{n}_\alpha^2 \pi_{\alpha^2} + 2\hat{n}_\alpha \hat{n}_\beta \pi_{\alpha\beta} + \hat{n}_\beta^2 \pi_{\beta^2} + 2\hat{n}_\alpha w \pi_\alpha + 2\hat{n}_\beta w \pi_\beta + w^2 \pi_1) & (62)\\
\int_\mathcal{F} \alpha^3 \; dA &= |\hat{n}_\gamma|^{-1} \pi_{\alpha^3} & (63)\\
\int_\mathcal{F} \beta^3 \; dA &= |\hat{n}_\gamma|^{-1} \pi_{\beta^3} & (64)\\
\int_\mathcal{F} \gamma^3 \; dA &= -|\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-3} (\hat{n}_\alpha^3 \pi_{\alpha^3} + 3\hat{n}_\alpha^2 \hat{n}_\beta \pi_{\alpha^2\beta} + 3\hat{n}_\alpha \hat{n}_\beta^2 \pi_{\alpha\beta^2} + \hat{n}_\beta^3 \pi_{\beta^3} + \\
& \qquad 3\hat{n}_\alpha^2 w \pi_{\alpha^2} + 6\hat{n}_\alpha \hat{n}_\beta w \pi_{\alpha\beta} + 3\hat{n}_\beta^2 w \pi_{\beta^2} + 3\hat{n}_\alpha w^2 \pi_\alpha + 3\hat{n}_\beta w^2 \pi_\beta + w^3 \pi_1) & (65)\\
\int_\mathcal{F} \alpha^2 \beta \; dA &= |\hat{n}_\gamma|^{-1} \pi_{\alpha^2\beta} & (66)\\
\int_\mathcal{F} \beta^2 \gamma \; dA &= -|\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-1} (\hat{n}_\alpha \pi_{\alpha\beta^2} + \hat{n}_\beta \pi_{\beta^3} + w \pi_{\beta^2}) & (67)
\end{aligned}$$

$$\int_{\mathcal{F}} \gamma^2 \alpha \, dA \quad = \quad |\hat{n}_\gamma|^{-1} \hat{n}_\gamma^{-2} (\hat{n}_\alpha^2 \pi_{\alpha^3} + 2\hat{n}_\alpha \hat{n}_\beta \pi_{\alpha^2 \beta} + \hat{n}_\beta^2 \pi_{\alpha \beta^2} + 2\hat{n}_\alpha w \pi_{\alpha^2} + 2\hat{n}_\beta w \pi_{\alpha\beta} + w^2 \pi_\alpha) \qquad (68)$$

6 Reduction to line integrals

The final step is to reduce an integral over a polygonal projection region in the α-β plane to a sum of line integrals over the edges bounding the region. We adopt the following notation (Figure 3). The

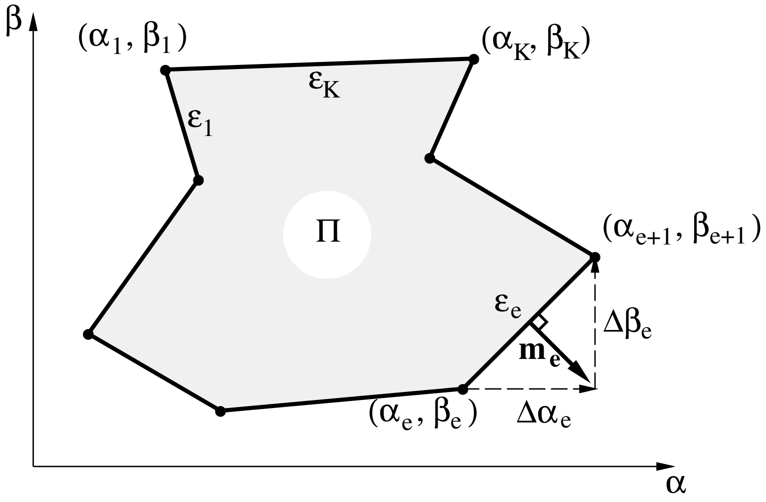

Figure 3: *Notation for computing a projection integral as a sum of line integrals.*

edges of Π are labeled $\mathcal{E}_1$ through $\mathcal{E}_K$. Edge $\mathcal{E}_e$ is the directed line segment from (α_e, β_e) to $(\alpha_{e+1}, \beta_{e+1})$, $\Delta\alpha_e = \alpha_{e+1} - \alpha_e$, and $\Delta\beta_e = \beta_{e+1} - \beta_e$ [note that $(\alpha_{K+1}, \beta_{K+1}) = (\alpha_1, \beta_1)$]. Finally, edge $\mathcal{E}_e$ has length L_e and exterior unit normal $\hat{\mathbf{m}}_e$.

Green's theorem[2] [9] provides the final integration reduction:

Theorem 3 (Green's) *Let Π be a region in the plane bounded by the single curve $\partial\Pi$. Let $\hat{\mathbf{m}}$ denote the exterior normal along the boundary. Then*

$$\int_\Pi \nabla \cdot \mathbf{H} \, dA = \oint_{\partial\Pi} \mathbf{H} \cdot \hat{\mathbf{m}} \, ds \qquad (69)$$

for any vector field $\mathbf{H}$ defined on Π, where the line integral traverses the boundary counter-clockwise.

This is a two dimensional version of the divergence theorem, and our special case again provides simplification. Since Π is polygonal, the right hand side of (69) may be broken into a summation over edges of constant normal, and by always choosing $\mathbf{H}$ so that one component is identically zero, the dot product becomes a scalar multiplication. From (56) and (57–68), all integrals of the form

$$\pi_{\alpha^p \beta^q} = \int_\Pi \alpha^p \beta^q \, dA \qquad (70)$$

are needed for nonnegative integers p and q with $p + q \leq 3$. Consider first the case $q = 0$. By choosing $\mathbf{H} = (\frac{1}{p+1}\alpha^{p+1}, 0)^T$, and applying Green's theorem to the polygonal region Π, we have

$$\int_\Pi \alpha^p \, dA = \frac{1}{p+1} \sum_{e=1}^{K} \hat{m}_{e_\alpha} \int_{\mathcal{E}_e} \alpha(s)^{p+1} \, ds \qquad (71)$$

[2] Sometimes more formally called *Green's theorem in the plane*. Additionally, some texts call this *Green's Lemma*, reserving *Green's Theorem* for a more general 3D result [10].

In the right hand integral, the integration variable s is arc length, and runs from 0 to L_e, the length of the edge; $\alpha(s)$ is the α-coordinate of the point on the edge that is a distance s from the starting point. Letting the variable λ be s/L_e, one can change integration variables ($ds = L_e d\lambda$) to get

$$\int_\Pi \alpha^p \, dA = \frac{1}{p+1} \sum_{e=1}^{K} \hat{m}_{e_\alpha} L_e \int_0^1 \alpha(L_e\lambda)^{p+1} \, d\lambda. \tag{72}$$

Now $\hat{m}_{e_\alpha} L_e = \pm\Delta\beta_e$, where the plus or minus depends on whether the vertices Π are indexed in counter-clockwise or clockwise order, respectively. By convention, we assume that the vertices of the original face $\mathcal{F}$ are indexed in counter-clockwise order, thus the vertices of Π will be indexed in counter-clockwise order exactly when the sign of $\hat{n}_\gamma$ is positive (Figure 2 helps in visualizing this). Hence, $\hat{m}_{e_\alpha} L_e = (\mathrm{sgn}\hat{n}_\gamma)\Delta\beta_e$, and

$$\int_\Pi \alpha^p \, dA = \frac{\mathrm{sgn}\hat{n}_\gamma}{p+1} \sum_{e=1}^{K} \Delta\beta_e \int_0^1 \alpha(L_e\lambda)^{p+1} \, d\lambda. \tag{73}$$

The case with $q = 1$ is similar, except one chooses $\mathbf{H} = (\frac{1}{p+1}\alpha^{p+1}\beta, 0)^T$. Finally, one can derive analogous equations for the cases when $p = 0$ or $p = 1$. The results are:

$$\int_\Pi \beta^q \, dA = -\frac{\mathrm{sgn}\hat{n}_\gamma}{q+1} \sum_{e=1}^{K} \Delta\alpha_e \int_0^1 \beta(L_e\lambda)^{q+1} \, d\lambda \tag{74}$$

$$\int_\Pi \alpha^p\beta \, dA = \frac{\mathrm{sgn}\hat{n}_\gamma}{p+1} \sum_{e=1}^{K} \Delta\beta_e \int_0^1 \alpha(L_e\lambda)^{p+1}\beta(L_e\lambda) \, d\lambda \tag{75}$$

$$\int_\Pi \alpha\beta^q \, dA = -\frac{\mathrm{sgn}\hat{n}_\gamma}{q+1} \sum_{e=1}^{K} \Delta\alpha_e \int_0^1 \alpha(L_e\lambda)\beta(L_e\lambda)^{q+1} \, d\lambda \tag{76}$$

7 Evaluation of integrals from vertex coordinates

We have successively reduced the original volume integrals to face integrals, projection integrals, and finally line integrals. The latter can be directly evaluated in terms of vertex coordinates, with the help of the following theorem.

Theorem 4 *Let L_e be the length of the directed line segment from (α_e, β_e) to $(\alpha_{e+1}, \beta_{e+1})$. Let $\alpha(s)$ and $\beta(s)$ be the α- and β-coordinates of the point on this segment a distance s from the starting point. Then for nonnegative integers p and q,*

$$\int_0^1 \alpha(L_e\lambda)^p \beta(L_e\lambda)^q \, d\lambda = \frac{1}{p+q+1} \sum_{i=0}^{p} \sum_{j=0}^{q} \frac{\binom{p}{i}\binom{q}{j}}{\binom{p+q}{i+j}} \alpha_{e+1}^i \alpha_e^{p-i} \beta_{e+1}^j \beta_e^{q-j}. \tag{77}$$

Proof: Denoting the integral on the left hand side of (77) by I,

$$I = \int_0^1 [(1-\lambda)\alpha_e + \lambda\alpha_{e+1}]^p \, [(1-\lambda)\beta_e + \lambda\beta_{e+1}]^q \, d\lambda \tag{78}$$

$$= \int_0^1 \left[\sum_{i=0}^{p} B_i^p(\lambda)\alpha_{e+1}^i \alpha_e^{p-i}\right] \left[\sum_{j=0}^{q} B_j^q(\lambda)\beta_{e+1}^j \beta_e^{q-j}\right] \, d\lambda, \tag{79}$$

(80)

where the B's are *Bernstein Polynomials*:

$$B_k^n(\lambda) = \begin{pmatrix} n \\ k \end{pmatrix} \lambda^k (1-\lambda)^{n-k}. \tag{81}$$

Two important properties of these polynomials are [3, 2]:

$$\begin{aligned} B_i^p(\lambda) B_j^q(\lambda) &= \frac{\begin{pmatrix} p \\ i \end{pmatrix}\begin{pmatrix} q \\ j \end{pmatrix}}{\begin{pmatrix} p+q \\ i+j \end{pmatrix}} B_{i+j}^{p+q}(\lambda), & (82) \\ \int_0^1 B_k^n(\lambda)\, d\lambda &= \frac{1}{n+1}. & (83) \end{aligned}$$

Expanding the product in (79) and applying (82) gives

$$I = \sum_{i=0}^{p} \sum_{j=0}^{q} \frac{\begin{pmatrix} p \\ i \end{pmatrix}\begin{pmatrix} q \\ j \end{pmatrix}}{\begin{pmatrix} p+q \\ i+j \end{pmatrix}} \alpha_{e+1}^i \alpha_e^{p-i} \beta_{e+1}^j \beta_e^{q-j} \int_0^1 B_{i+j}^{p+q}(\lambda)\, d\lambda. \tag{84}$$

Evaluating the integrals using (83) proves the theorem. □

All of the line integrals appearing in (73–76) are special cases of Theorem 4, with either p or q set to 0 or 1. Specifically,

$$\begin{aligned} \int_0^1 \alpha^{p+1}\, d\lambda &= \frac{1}{p+2} \sum_{i=0}^{p+1} \alpha_{e+1}^i \alpha_e^{p+1-i} & (85) \\ \int_0^1 \beta^{q+1}\, d\lambda &= \frac{1}{q+2} \sum_{j=0}^{q+1} \beta_{e+1}^j \beta_e^{q+1-j} & (86) \\ \int_0^1 \alpha^{p+1}\beta\, d\lambda &= \frac{1}{(p+2)(p+3)} \left[\beta_{e+1} \sum_{i=0}^{p+1} (i+1) \alpha_{e+1}^i \alpha_e^{p+1-i} + \beta_e \sum_{i=0}^{p+1} (p+2-i) \alpha_{e+1}^i \alpha_e^{p+1-i} \right] & (87) \\ \int_0^1 \alpha\beta^{q+1}\, d\lambda &= \frac{1}{(q+2)(q+3)} \left[\alpha_{e+1} \sum_{j=0}^{q+1} (j+1) \beta_{e+1}^j \beta_e^{q+1-j} + \alpha_e \sum_{j=0}^{q+1} (q+2-j) \beta_{e+1}^j \beta_e^{q+1-j} \right] & (88) \end{aligned}$$

Substituting (85–88) into (73–76), respectively give all of the needed projection integrals in terms of the coordinates of the projection vertices:

$$\pi_1 = \int_\Pi 1\, dA = +\frac{\operatorname{sgn} \hat{n}_\gamma}{2} \sum_{e=1}^{K} \Delta\beta_e (\alpha_{e+1} + \alpha_e) \tag{89}$$

$$\pi_\alpha = \int_\Pi \alpha\, dA = +\frac{\operatorname{sgn} \hat{n}_\gamma}{6} \sum_{e=1}^{K} \Delta\beta_e (\alpha_{e+1}^2 + \alpha_{e+1}\alpha_e + \alpha_e^2) \tag{90}$$

$$\pi_\beta = \int_\Pi \beta\, dA = -\frac{\operatorname{sgn} \hat{n}_\gamma}{6} \sum_{e=1}^{K} \Delta\alpha_e (\beta_{e+1}^2 + \beta_{e+1}\beta_e + \beta_e^2) \tag{91}$$

$$\pi_{\alpha^2} = \int_\Pi \alpha^2 \, dA \quad = \quad +\frac{\operatorname{sgn}\hat{n}_\gamma}{12} \sum_{e=1}^{K} \Delta\beta_e(\alpha_{e+1}^3 + \alpha_{e+1}^2 a_e + \alpha_{e+1}\alpha_e^2 + \alpha_e^3) \tag{92}$$

$$\pi_{\alpha\beta} = \int_\Pi \alpha\beta \, dA \quad = \quad +\frac{\operatorname{sgn}\hat{n}_\gamma}{24} \sum_{e=1}^{K} \Delta\beta_e \left[\beta_{e+1}(3\alpha_{e+1}^2 + 2\alpha_{e+1}\alpha_e + \alpha_e^2) + \beta_e(\alpha_{e+1}^2 + 2\alpha_e\alpha_{e+1} + 3\alpha_e^2)\right] \tag{93}$$

$$\pi_{\beta^2} = \int_\Pi \beta^2 \, dA \quad = \quad -\frac{\operatorname{sgn}\hat{n}_\gamma}{12} \sum_{e=1}^{K} \Delta\alpha_e(\beta_{e+1}^3 + \beta_{e+1}^2\beta_e + \beta_{e+1}\beta_e^2 + \beta_e^3) \tag{94}$$

$$\pi_{\alpha^3} = \int_\Pi \alpha^3 \, dA \quad = \quad +\frac{\operatorname{sgn}\hat{n}_\gamma}{20} \sum_{e=1}^{K} \Delta\beta_e(\alpha_{e+1}^4 + \alpha_{e+1}^3\alpha_e + \alpha_{e+1}^2\alpha_e^2 + \alpha_{e+1}\alpha_e^3 + \alpha_e^4) \tag{95}$$

$$\begin{aligned}\pi_{\alpha^2\beta} = \int_\Pi \alpha^2\beta \, dA \quad = \quad & +\frac{\operatorname{sgn}\hat{n}_\gamma}{60} \sum_{e=1}^{K} \Delta\beta_e \Big[\beta_{e+1}(4\alpha_{e+1}^3 + 3\alpha_{e+1}^2\alpha_e + 2\alpha_{e+1}\alpha_e^2 + \alpha_e^3) + \\ & \beta_e(\alpha_{e+1}^3 + 2\alpha_{e+1}^2\alpha_e + 3\alpha_{e+1}\alpha_e^2 + 4\alpha_e^3)\Big]\end{aligned} \tag{96}$$

$$\begin{aligned}\pi_{\alpha\beta^2} = \int_\Pi \alpha\beta^2 \, dA \quad = \quad & -\frac{\operatorname{sgn}\hat{n}_\gamma}{60} \sum_{e=1}^{K} \Delta\alpha_e \Big[\alpha_{e+1}(4\beta_{e+1}^3 + 3\beta_{e+1}^2\beta_e + 2\beta_{e+1}\beta_e^2 + \beta_e^3) + \\ & \alpha_e(\beta_{e+1}^3 + 2\beta_{e+1}^2\beta_e + 3\beta_{e+1}\beta_e^2 + 4\beta_e^3)\Big]\end{aligned} \tag{97}$$

$$\pi_{\beta^3} = \int_\Pi \beta^3 \, dA \quad = \quad -\frac{\operatorname{sgn}\hat{n}_\gamma}{20} \sum_{e=1}^{K} \Delta\alpha_e(\beta_{e+1}^4 + \beta_{e+1}^3\beta_e + \beta_{e+1}^2\beta_e^2 + \beta_{e+1}\beta_e^3 + \beta_e^4) \tag{98}$$

8 Algorithm

Based on the derivation in Sections 4–7, we give a complete algorithm for computing the ten desired volume integrals (18–27).

The algorithm comprises three routines:

1. `CompVolumeIntegrals(`$\mathcal{V}$`)` (Figure 4) computes the required volume integrals for a polyhedron by summing surface integrals over its faces, as detailed in Equations (29–38).

2. `CompFaceIntegrals(`$\mathcal{F}$`)` (Figure 5) computes the required surface integrals over a polyhedral face from the integrals over its projection, as detailed in Equations (57–68).

3. `CompProjectionIntegrals(`$\mathcal{F}$`)` (Figure 6) computes the required integrals over a face projection from the coordinates of the projections vertices, as detailed in Equations (89–98).

The algorithm contains a slight simplification over the presented derivation. When equations (89–98) are substituted into (57–68), the computation of $\operatorname{sgn}\hat{n}_\gamma$ and $|\hat{n}_\gamma|$ becomes unnecessary, since these terms always appear together in a product, giving simply $\hat{n}_\gamma$. Thus, no signs or absolute values are computed in routines `CompFaceIntegrals` and `CompProjectionIntegrals`.

9 Test Results

We now analyze the speed and accuracy of the algorithm for various test cases. These test were run on an *SGI Indigo II* with an R4400 CPU, and double precision floating point numbers were used for the calculations.

The set of polyhedral objects that have volume integrals which are commonly tabulated or easy to compute by hand is rather limited. We ran our algorithm on two such objects: an axes-aligned cube, 20 units on a side, and centered at the origin; and a tetrahedron defined by the convex hull of the origin, and the vectors $5\hat{\mathbf{i}}$, $4\hat{\mathbf{j}}$, and $3\hat{\mathbf{k}}$. The theoretical values for these objects are shown in Table 1. For these two

object	T_1	T_x	T_y	T_z	T_{x^2}	T_{y^2}	T_{z^2}	T_{xy}	T_{yz}	T_{zx}
cube	8000	0	0	0	2.67×10^5	2.67×10^5	2.67×10^5	0	0	0
tetrahedron	10	12.5	10	7.5	25	16	9	10	6	7.5

Table 1: Theoretical values of volume integrals for simple test polyhedra.

examples, all values computed by the algorithm were correct to at least 15 significant figures. The total time required to compute all ten volume integrals was 64 μs for the tetrahedron, and 110 μs for the cube.

For a more interesting test, the algorithm was applied to several polyhedral approximations of a unit radius sphere, centered at the origin. In this case there are two sources of error: numerical errors from the algorithm, and approximation errors inherent in the geometric model, which is not a true sphere. These latter errors should not be attributed to the algorithm itself. For a perfect unit sphere, the integrals $T_x, T_y, T_z, T_{xy}, T_{yz}$, and T_{zx} should vanish, while $T_1 = \frac{4}{3}\pi$ and $T_{x^2} = T_{y^2} = T_{z^2} = \frac{4}{15}\pi$. We applied our algorithm to a series of successive approximations to the sphere, beginning with an icosahedron, and obtaining each finer approximation by projecting the midpoint of each polyhedral edge onto the unit sphere, and taking a convex hull. The computed values of a representative set of volume integrals for each polyhedron are shown in Table 2.

Without numerical error, the integrals T_x and T_{yz} would vanish for all six polyhedral approximations of the sphere, due to symmetry. From Table 2, the absolute values of these computed values are all less than 10^{-15}. The theoretical values in the table correspond to the sphere which circumscribes the polyhedra. For each polyhedron, we have also determined corresponding values for the inscribed sphere, and verified that the computed values for T_1 and T_{x^2} for the polyhedron lie between the bounding values for these two spheres. For approximation 6, the difference in values for the inscribed and circumscribed sphere is 2.8×10^{-3} for T_1 and 9.5×10^{-4} for T_{x^2}. These are upper bounds on the numerical errors of the algorithm. Note that the deviations between theoretical and computed values for T_1 and T_{x^2} are reduced as the complexity of the polyhedron increases, while numerical error from the algorithm should increase with complexity. In light of the very small errors incurred in the computation of T_x and T_{yz}, the deviations between the computed and theoretical values of T_1 and T_{x^2} are almost certainly due mainly to the polyhedral approximation rather than to numerical errors.

approx.	verts	edges	faces	T_1	T_x	T_{x^2}	T_{yz}	time
1	12	30	20	2.536	-2.8×10^{-17}	0.3670	-3.1×10^{-17}	500 μs
2	42	120	80	3.659	$+1.4 \times 10^{-16}$	0.6692	$+1.5 \times 10^{-17}$	1.2 ms
3	162	480	320	4.047	-3.2×10^{-16}	0.7911	-6.1×10^{-18}	4.9 ms
4	642	1920	1280	4.153	$+3.0 \times 10^{-16}$	0.8258	$+7.8 \times 10^{-18}$	21 ms
5	2562	7680	5120	4.180	-3.8×10^{-17}	0.8347	$+2.1 \times 10^{-17}$	82 ms
6	10242	30720	20480	4.187	$+5.6 \times 10^{-16}$	0.8370	$+6.4 \times 10^{-18}$	350 ms
theoretical values for sphere				4.189	0.0	0.8378	0.0	–

Table 2: Data for successive approximations of a unit sphere.

The execution times shown in Table 2 are the total times for computing all ten volume integrals for each polyhedron. The $O(n)$ nature of the algorithm is evident: from approximation 2 on, the time ratios between successive refinements very closely follow the 4:1 ratio in the number of faces. The algorithm is also very fast: all ten integrals are computed for a polyhedron with over 20,000 faces in only 350 ms.

10 Available Code

ANSI C source code for the algorithm described in this paper, and detailed in Figures 4–6, is publicly available from

`http://www.acm.org/jgt`

Also included is an interface to build up polyhedra (using a simple data structure) from ASCII specifications; examples are provided. The code is public domain, and may be used as is or in modified form.

Acknowledgements. We thank Aristides Requicha for a valuable literature survey on this topic, and David Baraff for useful comments on the initial draft of this paper. We especially thank John Hughes for his detailed review and many suggestions for improving the paper.

`CompVolumeIntegrals(`$\mathcal{V}$`)`

$T_1, T_x, T_y, T_z, T_{x^2}, T_{y^2}, T_{z^2}, T_{xy}, T_{yz}, T_{zx} \leftarrow 0$

`for each face` $\mathcal{F}$ `on the boundary of` $\mathcal{V}$
  `choose` α-β-γ `as a right-handed permutation of` x-y-z `that maximizes` $|\hat{n}_\gamma|$
  `compFaceIntegrals(`$\mathcal{F}$`)`
  `if (`$\alpha = x$`)` $T_1 \leftarrow T_1 + \hat{n}_\alpha F_\alpha$
  `else if (`$\beta = x$`)` $T_1 \leftarrow T_1 + \hat{n}_\beta F_\beta$
  `else` $T_1 \leftarrow T_1 + \hat{n}_\gamma F_\gamma$
  $T_\alpha \leftarrow T_\alpha + \hat{n}_\alpha F_{\alpha^2}$
  $T_\beta \leftarrow T_\beta + \hat{n}_\beta F_{\beta^2}$
  $T_\gamma \leftarrow T_\gamma + \hat{n}_\gamma F_{\gamma^2}$
  $T_{\alpha^2} \leftarrow T_{\alpha^2} + \hat{n}_\alpha F_{\alpha^3}$
  $T_{\beta^2} \leftarrow T_{\beta^2} + \hat{n}_\beta F_{\beta^3}$
  $T_{\gamma^2} \leftarrow T_{\gamma^2} + \hat{n}_\gamma F_{\gamma^3}$
  $T_{\alpha\beta} \leftarrow T_{\alpha\beta} + \hat{n}_\alpha F_{\alpha^2\beta}$
  $T_{\beta\gamma} \leftarrow T_{\beta\gamma} + \hat{n}_\beta F_{\beta^2\gamma}$
  $T_{\gamma\alpha} \leftarrow T_{\gamma\alpha} + \hat{n}_\gamma F_{\gamma^2\alpha}$

$(T_x, T_y, T_z) \leftarrow (T_x, T_y, T_z)\ /\ 2$
$(T_{x^2}, T_{y^2}, T_{z^2}) \leftarrow (T_{x^2}, T_{y^2}, T_{z^2})\ /\ 3$
$(T_{xy}, T_{yz}, T_{zx}) \leftarrow (T_{xy}, T_{yz}, T_{zx})\ /\ 2$

Figure 4: `CompVolumeIntegrals(`$\mathcal{V}$`)`. Compute the required volume integrals for a polyhedron. See Equations (29–38).

```
CompFaceIntegrals(𝓕)

    computeProjectionIntegrals(𝓕)
    w ← −n̂·p for some point p on 𝓕
    k1 ← n̂γ^(-1);  k2 ← k1 * k1;  k3 ← k2 * k1;  k4 ← k3 * k1
```

$$
\begin{aligned}
&F_\alpha \leftarrow k_1 \pi_\alpha \\
&F_\beta \leftarrow k_1 \pi_\beta \\
&F_\gamma \leftarrow -k_2(\hat{n}_\alpha \pi_\alpha + \hat{n}_\beta \pi_\beta + w\pi_1) \\
\\
&F_{\alpha^2} \leftarrow k_1 \pi_{\alpha^2} \\
&F_{\beta^2} \leftarrow k_1 \pi_{\beta^2} \\
&F_{\gamma^2} \leftarrow k_3(\hat{n}_\alpha^2 \pi_{\alpha^2} + 2\hat{n}_\alpha \hat{n}_\beta \pi_{\alpha\beta} + \hat{n}_\beta^2 \pi_{\beta^2} + 2\hat{n}_\alpha w \pi_\alpha + 2\hat{n}_\beta w \pi_\beta + w^2 \pi_1) \\
\\
&F_{\alpha^3} \leftarrow k_1 \pi_{\alpha^3} \\
&F_{\beta^3} \leftarrow k_1 \pi_{\beta^3} \\
&F_{\gamma^3} \leftarrow -k_4(\hat{n}_\alpha^3 \pi_{\alpha^3} + 3\hat{n}_\alpha^2 \hat{n}_\beta \pi_{\alpha^2\beta} + 3\hat{n}_\alpha \hat{n}_\beta^2 \pi_{\alpha\beta^2} + \hat{n}_\beta^3 \pi_{\beta^3} + \\
&\qquad 3\hat{n}_\alpha^2 w \pi_{\alpha^2} + 6\hat{n}_\alpha \hat{n}_\beta w \pi_{\alpha\beta} + 3\hat{n}_\beta^2 w \pi_{\beta^2} + 3\hat{n}_\alpha w^2 \pi_\alpha + 3\hat{n}_\beta w^2 \pi_\beta + w^3 \pi_1) \\
\\
&F_{\alpha^2\beta} \leftarrow k_1 \pi_{\alpha^2\beta} \\
&F_{\beta^2\gamma} \leftarrow -k_2(\hat{n}_\alpha \pi_{\alpha\beta^2} + \hat{n}_\beta \pi_{\beta^3} + w\pi_{\beta^2}) \\
&F_{\gamma^2\alpha} \leftarrow k_3(\hat{n}_\alpha^2 \pi_{\alpha^3} + 2\hat{n}_\alpha \hat{n}_\beta \pi_{\alpha^2\beta} + \hat{n}_\beta^2 \pi_{\alpha\beta^2} + 2\hat{n}_\alpha w \pi_{\alpha^2} + 2\hat{n}_\beta w \pi_{\alpha\beta} + w^2 \pi_\alpha)
\end{aligned}
$$

Figure 5: `CompFaceIntegrals(`$\mathcal{F}$`)`. Compute the required surface integrals over a polyhedral face. See Equations (57–68).

```
CompProjectionIntegrals(F)
```

$\pi_1, \pi_\alpha, \pi_b, \pi_{\alpha^2}, \pi_{\alpha\beta}, \pi_{\beta^2}, \pi_{\alpha^3}, \pi_{\alpha^2\beta}, \pi_{\alpha\beta^2}, \pi_{\beta^3} \leftarrow 0$

for each edge $\mathcal{E}$ in CCW order around $\mathcal{F}$

$\alpha_0 \leftarrow \alpha$-coordinate of start point of $\mathcal{E}$
$\beta_0 \leftarrow \beta$-coordinate of start point of $\mathcal{E}$
$\alpha_1 \leftarrow \alpha$-coordinate of end point of $\mathcal{E}$
$\beta_1 \leftarrow \beta$-coordinate of end point of $\mathcal{E}$
$\Delta\alpha \leftarrow \alpha_1 - \alpha_0$
$\Delta\beta \leftarrow \beta_1 - \beta_0$
$\alpha_0^2 \leftarrow \alpha_0 * \alpha_0$; $\alpha_0^3 \leftarrow \alpha_0^2 * \alpha_0$; $\alpha_0^4 \leftarrow \alpha_0^3 * \alpha_0$
$\beta_0^2 \leftarrow \beta_0 * \beta_0$; $\beta_0^3 \leftarrow \beta_0^2 * \beta_0$; $\beta_0^4 \leftarrow \beta_0^3 * \beta_0$
$\alpha_1^2 \leftarrow \alpha_1 * \alpha_1$; $\alpha_1^3 \leftarrow \alpha_1^2 * \alpha_1$
$\beta_1^2 \leftarrow \beta_1 * \beta_1$; $\beta_1^3 \leftarrow \beta_1^2 * \beta_1$

$C_1 \leftarrow \alpha_1 + \alpha_0$
$C_\alpha \leftarrow \alpha_1 C_1 + \alpha_0^2$; $C_{\alpha^2} \leftarrow \alpha_1 C_\alpha + \alpha_0^3$; $C_{\alpha^3} \leftarrow \alpha_1 C_{\alpha^2} + \alpha_0^4$
$C_\beta \leftarrow \beta_1^2 + \beta_1\beta_0 + \beta_0^2$; $C_{\beta^2} \leftarrow \beta_1 C_\beta + \beta_0^3$; $C_{\beta^3} \leftarrow \beta_1 C_{\beta^2} + \beta_0^4$
$C_{\alpha\beta} \leftarrow 3\alpha_1^2 + 2a_1a_0 + a_0^2$; $K_{\alpha\beta} \leftarrow \alpha_1^2 + 2\alpha_1\alpha_0 + 3\alpha_0^2$
$C_{\alpha^2\beta} \leftarrow \alpha_0 C_{\alpha\beta} + 4\alpha_1^3$; $K_{\alpha^2\beta} \leftarrow \alpha_1 K_{\alpha\beta} + 4\alpha_0^3$
$C_{\alpha\beta^2} \leftarrow 4\beta_1^3 + 3\beta_1^2\beta_0 + 2\beta_1\beta_0^2 + \beta_0^3$; $K_{\alpha\beta^2} \leftarrow \beta_1^3 + 2\beta_1^2\beta_0 + 3\beta_1\beta_0^2 + 4\beta_0^3$

$\pi_1 \leftarrow \pi_1 + \Delta\beta C_1$
$\pi_\alpha \leftarrow \pi_\alpha + \Delta\beta C_\alpha$; $\pi_{\alpha^2} \leftarrow \pi_{\alpha^2} + \Delta\beta C_{\alpha^2}$; $\pi_{\alpha^3} \leftarrow \pi_{\alpha^3} + \Delta\beta C_{\alpha^3}$
$\pi_\beta \leftarrow \pi_\beta + \Delta\alpha C_\beta$; $\pi_{\beta^2} \leftarrow \pi_{\beta^2} + \Delta\alpha C_{\beta^2}$; $\pi_{\beta^3} \leftarrow \pi_{\beta^3} + \Delta\alpha C_{\beta^3}$
$\pi_{\alpha\beta} \leftarrow \pi_{\alpha\beta} + \Delta\beta(\beta_1 C_{\alpha\beta} + \beta_0 K_{\alpha\beta})$
$\pi_{\alpha^2\beta} \leftarrow \pi_{\alpha^2\beta} + \Delta\beta(\beta_1 C_{\alpha^2\beta} + \beta_0 K_{\alpha^2\beta})$
$\pi_{\alpha\beta^2} \leftarrow \pi_{\alpha\beta^2} + \Delta\alpha(\alpha_1 C_{\alpha\beta^2} + \alpha_0 K_{\alpha\beta^2})$

$\pi_1 \leftarrow \pi_1/2$
$\pi_\alpha \leftarrow \pi_\alpha/6$; $\pi_{\alpha^2} \leftarrow \pi_{\alpha^2}/12$; $\pi_{\alpha^3} \leftarrow \pi_{\alpha^3}/20$
$\pi_\beta \leftarrow -\pi_\beta/6$; $\pi_{\beta^2} \leftarrow -\pi_{\beta^2}/12$; $\pi_{\beta^3} \leftarrow -\pi_{\beta^3}/20$
$\pi_{\alpha\beta} \leftarrow \pi_{\alpha\beta}/24$
$\pi_{\alpha^2\beta} \leftarrow \pi_{\alpha^2\beta}/60$
$\pi_{\alpha\beta^2} \leftarrow -\pi_{\alpha\beta^2}/60$

Figure 6: `CompProjectionIntegrals(F)`. Compute the required integrals over a face projection. See equations (89–98).

References

[1] C. H. Edwards, Jr. and David E. Penney. *Calculus and Analytic Geometry.* Prentice-Hall, Inc., Englewood Cliffs, second edition, 1986.

[2] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, Inc., San Diego, second edition, 1990.

[3] R.T. Farouki and V.T. Rajan. Algorithms for polynomials in Bernstein form. *Computer Aided Geometric Design*, 5(1):1–26, June 1988.

[4] Donald T. Greenwood. *Principles of Dynamics.* Prentice-Hall, Inc., Englewood Cliffs, second edition, 1988.

[5] Yong Tsui Lee and Aristides A. G. Requicha. Algorithms for computing the volume and other integral properties of solids. I. Known methods and open issues. *Communications of the ACM*, 25(9):635–41, September 1982.

[6] Sheue-ling Lien and James T. Kajiya. A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. *IEEE Computer Graphics and Applications*, 4(10):35–41, October 1984.

[7] J.L. Meriam and L.G. Kraige. *Engineering Mechanics Volume 2: Dynamics.* John Wiley & Sons, Inc., New York, 1986.

[8] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian R. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, second edition, 1992.

[9] Sherman K. Stein. *Calculus and Analytic Geometry.* McGraw-Hill, Inc., New York, third edition, 1982.

[10] C. Ray Wylie and Louis C. Barrett. *Advanced Engineering Mathematics.* McGraw-Hill, Inc., New York, fifth edition, 1982.