Boundary integration over linear polyhedra

C Cattani and A Paoluzzi*

In order either to plan or control the static or dynamic behaviour of models in CAD applications, it is often necessary, to evaluate integral properties of solid models (i.e. volume, centroid, moments of inertia, etc.). This paper deals with the exact evaluation of inertial properties of homogeneous polyhedral objects. In particular, a finite integration method is developed for the computation of various order monomial integrals over polyhedral solids and surfaces in the 3-space. The integration method presented here can also be used for the exact evaluation of domain integrals of trivariate polynomial forms.

computer-aided design, integration, polynomial forms, polyhedra

Problem statement

The evaluation of area, volume, centroid and moments of inertia of rigid homogeneous solids frequently arises in a large number of engineering applications, in CAD as well as in robotics. In this paper, exact formulas are given for evaluating double and triple integrals of monomials over linear regular polyhedra in $\mathscr{R}^3$. In particular, the integration method presented here enables practical formulas to be obtained for the exact evaluation of the integrals

$$II_S \equiv \iint_S f(\mathbf{p})\, dS,\ III_P \equiv \iiint_P f(\mathbf{p})\, dV \qquad (1)$$

where S, and P are linear and regular 2- or 3-polyhedra in $\mathscr{R}^3$, dS and dV are the differential surface and the differential volume. The integrating function is a trivariate polynomial

$$f(\mathbf{p}) = \sum_{\alpha=0}^{n} \sum_{\beta=0}^{m} \sum_{\gamma=0}^{p} a_{\alpha\beta\gamma} x^\alpha y^\beta z^\gamma$$

where α, β, γ are non-negative integers. However, the paper is focused on the calculation of the following integrals of monomials

$$II_S^{\alpha\beta\gamma} \equiv \iint_S x^\alpha y^\beta z^\gamma\, dS,\ III_P^{\alpha\beta\gamma} \equiv \iiint_P x^\alpha y^\beta z^\gamma\, dV \qquad (2)$$

being the extension to $f(\mathbf{p})$ straightforwardly obtained by the linearity of integrals.

Previous work

Quadrature formulas for multiple integrals have always been of great interest in computer applications[1-3]. Various attempts have been made in order to get practical formulas over simplicial complexes. Numerical integration over simplices has been discussed by Hammer and Stroud[4], who obtained exact formulas for the quadratic and cubic polynomials over n-simplices in the n-space but the extension of them for the integration of higher degree polynomials offers, according to the authors, 'significantly greater complexity'. In a related paper, Hammer, *et al.*[5] gave quadrature formulas for the kth degree polynomial over the n-simplex, depending on the roots of a special class of polynomials. Afterwards, many papers on integration were related to solid modelling but as Lee and Requicha[6] pointed out most computational studies in multiple integration often deal with calculations over very simple domains, like a cube or a sphere, while the integrating function $f(\mathbf{p})$ is very complicated. On the contrary, in most of the engineering applications the converse problem usually arises. Lee and Requicha[6] also noticed that the particular representation of solids hardly constrains the design of integration algorithms, because the errors due to the particular representation usually dominate the numerical errors (round-off, truncation, and errors depending on the use of approximate integration formulae). An approximate boundary integration method for 2D objects – described as a bitmap – is given, as an alternative approach, by Fabrikant *et al.*[7]. In Lee and Requicha[8], a family of approximate algorithms for computing inertial properties of solids is outlined. Such algorithms are based on a representation conversion from CSG to octree, via recursive subdivision. With a different approach, O'Leary[9] developed quadrature formulas based on a quasidisjoint decomposition of the solid in volume elements of simple predefined shape. Wilson and Farrior[10] gave a large set of formulas for the computation of main geometrical and inertial properties of planar polygons and of rotational solids. Timmer and Stern[11] discussed a theoretical approach to the evaluation of domain integrals over a boundary representation of solids. Lien and Kajiya[12] showed a closed formula for volume integration, by decomposing the solid into a set of solid tetrahedra, but they did not handle the problem of

Dip. di Matematica, Università "La Sapienza", P.le A. Moro 2, 00185 Roma, Italy
*Dip. di Informatica e Sistemistica, Università "La Sapienza", Via Buonarroti 12, 00185 Roma, Italy
Paper received: 29 September 1988. Revised: 13 March 1989

 0010–4485/90/020130–06

surface integration. In this paper, a symbolic solution is given both to the surface and to the volume integration of polynomials, by using a triangulation of the volume boundary.

Overview

This paper may be a specialization of Timmer and Stern's general method[11], consisting of the transformation of a volume integral into a surface integral and then into a parametric line integral. Timmer and Stern suggest using polynomial approximations (splines) of the object edges, whereas explicit exact formulas for linear domains are given here. The evaluation of surface and volume integrals is achieved by transforming them into line integrals over the boundary of every 2-simplex of a domain triangulation. Two functions for the exact evaluation of monomial integrals over the boundary surface or the volume of regular linear polyhedra in 3-space are derived. There results an inertia discretization, such that all geometrical and intertial properties – of any order – can be exactly computed (without approximation errors) by means of a finite number of algebraic operations on the coordinates of vertices.

The paper is organized as follows. In the next section, a methodology is suggested for computing surface integrals of the form (2) as a summation of integrals over a triangulation of the surface. Any triangle is mapped into the unit triangle in the 2-space of parameters, and it will be seen that integrals of monomials evaluated over this special triangle become particularly simple. In the third section, formulas for integrals over polyhedral volumes are given. They are easily derived by transforming volume integrals into surface integrals. In the fourth section, it is shown that such integrals are computable in polynomial time, and that inertia moments are computable in $O(E)$ time, E being the number of edges of the integration domain. In Appendix 1, the cumbersome proofs of some formulas are given. In Appendix 2, an implementation of the proposed integration methodology is presented.

SURFACE INTEGRATION

A structure product is the integral of a monomial over a simplicial complex. Exact formulas for structure products over n-sided polygons in 2-space, the unit triangle in 2-space, and an arbitrary triangle in 3-space, are derived in the following. Structure products are a generalization of the usual products and moments of inertia, that can be obtained from formula (2) by assuming $\alpha + \beta + \gamma \leqslant 2$.

Polygon integrals

A structure product over a polygon π in the plane xy is

$$\Pi_{\pi}^{\alpha\beta} = \iint_{\pi} x^{\alpha} y^{\beta}\, dS, \quad \alpha, \beta \geqslant 0, \text{ integers} \tag{3}$$

Such integrals can be exactly expressed, when π is a polygon with n oriented edges, as

$$\Pi_{\pi}^{\alpha\beta} = \frac{1}{\alpha+1} \sum_{i=1}^{n} \sum_{h=0}^{\alpha+1} \binom{\alpha+1}{h} x_i^{\alpha+1-h} X_i^h \times \sum_{k=0}^{\beta} \frac{\binom{\beta}{k}}{h+k+1} y_i^{\beta-k} Y_i^{k+1} \tag{4}$$

where $\mathbf{p}_i = (x_i, y_i)$, $X_i = x_{i+1} - x_i$, $Y_i = y_{i+1} - y_i$ and $\mathbf{p}_{n+1} = \mathbf{p}_1$. The derivation of formula (4) is based on the application of Green's theorem and on Newton's expression for binomial powers. The proof is shown in the first section of Appendix 1.

Unit triangle integrals

The general formula (4) can be specialized for the unit triangle $\tau' = \langle \mathbf{w}_o, \mathbf{w}_a, \mathbf{w}_b \rangle$, with vertices

$$\mathbf{w}_o = (0, 0), \quad \mathbf{w}_a = (1, 0), \quad \mathbf{w}_b = (0, 1), \tag{5}$$

getting a very simplified expression. With some algebraic manipulations, shown in the second section of Appendix 1, we obtain*

$$\Pi^{\alpha\beta} = \frac{1}{\alpha+1} \sum_{h=0}^{\alpha+1} \binom{\alpha+1}{h} \frac{(-1)^h}{h+\beta+1} \tag{6}$$

which reduces, for $\alpha = \beta = 0$, to the area of the triangle (5): $\Pi^{00} = 1/2$.

Triangle integrals

In the following, the general expression is derived for structure products evaluated on an arbitrary triangle $\tau = \langle \mathbf{v}_o, \mathbf{v}_a, \mathbf{v}_b \rangle$ of the 3-space xyz, defined by $\mathbf{v}_o = (x_o, y_o, z_o)$ and by the vectors $\mathbf{a} = \mathbf{v}_a - \mathbf{v}_o$ and $\mathbf{b} = \mathbf{v}_b - \mathbf{v}_o$. The parametric equation of its embedding plane is

$$\mathbf{p} = \mathbf{v}_o + u\mathbf{a} + v\mathbf{b} \tag{7}$$

where the area element is

$$d\tau = |J|\, du\, dv = \left| \frac{\partial \mathbf{p}}{\partial u} \times \frac{\partial \mathbf{p}}{\partial v} \right| du\, dv = |\mathbf{a} \times \mathbf{b}|\, du\, dv \tag{8}$$

A structure product over a triangle τ in 3-space can be transformed by a coordinates transformation, as follows

$$\Pi_{\tau}^{\alpha\beta\gamma} = \iint_{\tau} x^{\alpha} y^{\beta} z^{\gamma}\, d\tau = |\mathbf{a} \times \mathbf{b}| \iint_{\tau'} x^{\alpha}(u, v) y^{\beta}(u, v) z^{\gamma}(u, v)\, du\, dv \tag{9}$$

* $\Pi^{\alpha\beta}$ is substituted, when referred to the unit triangle, by the symbol $\Pi^{\alpha\beta}$

where τ' is the uv domain that corresponds to τ under the coordinate transformation (7). In this case we have (the proof is given in the third section of Appendix 1)

$$III_\tau^{\alpha\beta\gamma} = |\mathbf{a}\times\mathbf{b}| \sum_{h=0}^{\alpha}\binom{\alpha}{h} x_0^{\alpha-h} \sum_{k=0}^{\beta}\binom{\beta}{k} y_0^{\beta-k} \times \sum_{m=0}^{\gamma}\binom{\gamma}{m} z_0^{\gamma-m} \sum_{i=0}^{h}\binom{h}{i} a_x^{h-i} b_x^i \sum_{j=0}^{k}\binom{k}{j} a_y^{k-j} b_y^j \times \sum_{l=0}^{m}\binom{m}{l} a_z^{m-l} b_z^l III^{\mu\nu} \tag{10}$$

where $\mu = (h+k+m)-(i+j+l)$, $\nu = (i+j+l)$, and $III^{\mu\nu}$ is a structure produce over triangle (5), as given by formula (6). Of course the area of a triangle τ is

$$III_\tau^{000} = \iint_\tau d\tau = |\mathbf{a}\times\mathbf{b}|\, III^{00} = \frac{|\mathbf{a}\times\mathbf{b}|}{2} \tag{11}$$

Surface integrals

In conclusion, a structure product over a polyhedral surface S, open or closed, is a summation of structure products (10) over the 2-simplices of a triangulation K_2 of S

$$III_S^{\alpha\beta\gamma} = \iint_S x^\alpha y^\beta z^\gamma\, dS = \sum_{\tau\in K_2} III_\tau^{\alpha\beta\gamma} \tag{12}$$

VOLUME INTEGRATION

Let P be a three-dimensional polyhedron bounded by a polyhedral surface ∂P. The regularity of the integration domain and the continuity of the integrating function enable the application of the Divergence theorem, which can be briefly summarized, for a vector field $\mathbf{F} = \mathbf{F}(\mathbf{p})$ as

$$\iiint_P \nabla\cdot\mathbf{F}\, dx\, dy\, dz = \iint_{\partial P} \mathbf{F}\cdot\mathbf{n}\, dS = \sum_{\tau\in K_2} \iint_\tau \mathbf{F}\cdot\mathbf{n}_\tau\, d\tau \tag{13}$$

where $\mathbf{n}$ is the outward vector normal to the surface portion dS, and hence $\mathbf{n}_\tau = \mathbf{a}\times\mathbf{b}/|\mathbf{a}\times\mathbf{b}|$. As the function $x^\alpha y^\beta z^\gamma$ equates the divergence of the vector field $\mathbf{F} = (x^{\alpha+1}y^\beta z^\gamma/\alpha+1,\ 0,\ 0)$, there is easily derived an expression for $III_P^{\alpha\beta\gamma}$, which depends only on the 1-simplices of a triangulation of the domain boundary and on the structure products over its 2-simplices. As a matter of fact

$$\begin{aligned} III_P^{\alpha\beta\gamma} &= \iiint_P x^\alpha y^\beta z^\gamma\, dx\, dy\, dz \\ &= \iiint_P \frac{\partial}{\partial x}\left(\frac{1}{\alpha+1}x^{\alpha+1}y^\beta z^\gamma\right) dx\, dy\, dz \\ &= \frac{1}{\alpha+1}\sum_{\tau'\in K_2'} (\mathbf{a}\times\mathbf{b})_x \iint_{\tau'} x^{\alpha+1}y^\beta z^\gamma\, du\, dv \end{aligned} \tag{14}$$

Taking into account equations (8) and (9), the integral can be substituted in the previous equation to give

$$III_P^{\alpha\beta\gamma} = \frac{1}{\alpha+1}\sum_{\tau\in K_2}\left[\frac{(\mathbf{a}\times\mathbf{b})_x}{|\mathbf{a}\times\mathbf{b}|}\right]_\tau III_\tau^{\alpha+1,\beta,\gamma} \tag{15}$$

where the surface integrals are evaluable by using formula (10).

COMPUTATIONAL COMPLEXITY

Surface and volume integrals over linear polyhedra are computable in polynomial time. In particular, surface and volume integrals of a monomial $x^\alpha y^\beta z^\gamma$ over a linear 2-or 3-polyhedron are computable in $O(\alpha^3\beta^2\gamma^2 E)$ time, E being the number of edges of the polyhedron. In fact, for the surface and volume integrals it is very easy to see, from inspection of equations (6), (10), (12) and (15), that both integrals may be evaluated in $O(\alpha^3\beta^2\gamma^2 T)$ time, T being the number of triangles of a minimal triangulation of the domain. The following shows that the relation $T = 2E - 2F < 2E$ holds between the number T of triangles of a minimal triangulation of a polyhedron boundary and the number E and F of its original edges and faces respectively. When all triangle faces are triangular, the relation reduces to $T = 2/3E$. In both cases the existence of such linear relationships will complete the proof.

It is well known (see for example Woo[13]) that $V_{i_1} + V_{i_2} + \ldots + V_{i_F} = 2E$, where V_{i_k} is the number of vertices belonging to the boundary of the face f_{i_k}. When triangulating a polygon face f_{i_k} to get a minimal tringulation it is necessary to add a number $E_{a_k} = V_{i_k} - 3$ of new edges. The total number E_a of added edges is

$$E_a = \sum_{k=1}^{F} E_{a_k} = \sum_{k=1}^{F} V_{i_k} - 3F = 2E - 3F \tag{16}$$

From the Euler formula applied both to the original and to the triangulated polyhedron, it is $V - E + F = 2$ and $V - (E + E_a) + T = 2$, and hence $T = F + E_a$. So substituting equation (16) in the last equation gives $T = 2E - 2F$, and hence the assertion holds.

The following property is very important for CAD and robotics applications: it shows that the inertia tensor of a linear polyhedral solid is easy to compute. The inertia tensor of a linear polyhedron is computable in $O(E)$ time. As a matter of fact the elements of the inertia matrix of a 1-, 2- or 3-dimensional homogeneous object, namely its mass M, first moments M_x, M_y, M_z, products of inertia M_{xy}, M_{yz}, M_{zx}, and second moments M_{xx}, M_{yy}, M_{zz}, can all be expressed as

$$\rho\int_{\mathscr{D}} x^\alpha y^\beta z^\gamma\, d\mathscr{D} \tag{17}$$

where $\mathscr{D}$ is a 1-, 2- or 3-dimensional domain of 3-space, ρ is the mass (constant) density, and where $0 \leqslant \alpha+\beta+\gamma \leqslant 2$. Being α, β, γ bounded, the assertion follows from the previous claim.

CONCLUSIONS

A very important feature of the integration formulas presented here is that they can also be used with a partial model of a polyhedron, consisting of the collection of its face loops. Loops are oriented counter-clockwise if external, clockwise if internal to another loop. Such a kind of model, without explicit memory of face adjacencies, is very frequently adopted in computer graphics. In this case it is sufficient to consider any $(n+1)$-sided (also unconnected or multiply connected) face as the topological sum of $n-1$ oriented triangles t_i, with vertices $\langle v_0, v_i, v_{i+1} \rangle$, where $1 \leqslant i \leqslant n-1$. In applying formulas (12) or (15) to such a set of triangles, any edge that does not belong to the original polygon will be computed twice, in the two opposite directions. These contributions to the whole integral will mutually cancel, as they correspond to pairs of line integrals evaluated along opposite paths. The method presented here may eventually suggest how such outstanding problems in CAD/CAE applications as the integration over nonlinear polyhedra could be approached with boundary surfaces described by implicit equations.

ACKNOWLEDGEMENTS

The work of the second author was supported, during his visit at the Comuter Science Department of Cornell University, by the Mathematical Science Institute of Cornell and by the Italian Research Council.

REFERENCES

1 **Krylov, V I** *Approximate calculation of integrals* MacMillan (1962)

2 **Stroud, A H** 'Numerical integration' in **Klerer, M and Korn, G A (eds)** *Digital Computer User's Handbook* McGraw-Hill (1967) pp 117–143

3 **Stroud, A H** 'Approximate multiple integration' in **Ralston, A and Wilf, H S (eds),** *Mathematical Methods for Digital Computers* Wiley (1967) pp 145–155

4 **Hammer, P C and Stroud, A H** 'Numerical integration over simplexes' *Math. Tables Aids Comput.* Vol 10 (1956) pp 137–139

5 **Hammer, P C, Marlowe, O J and Stroud, A H** 'Numerical integration over simplexes and cones' *Math. Tables Aids Comput.* Vol 10 (1956) pp 130–137

6 **Lee, Y T and Requicha, A A G** 'Algorithms for computing the volume and other integral properties of solids I: known methods and open issues' *Comm. ACM* Vol 25 No 9 (September 1982) pp 635–641

7 **Fabrikant, V I, Latinovic, V and Sankar, T S** 'Contour integration on the graphics screen and its applications in CADCAM' *Comput.-Aided Des.* Vol 17 No 2 (March 1985) pp 60–67

8 **Lee, Y T and Requicha, A A G** 'Algorithms for computing the volume and other integral properties of solids II: a family of algorithms based on representation conversion and on cellular approximation' *Comm. ACM* Vol 25 No 9 (September 1982) pp 642–650

9 **O'Leary, J R** 'Evaluation of mass properties by finite elements' *J. Guidance and Control* Vol 3 No 2 (1980) pp 188–190

10 **Wilson, H B, jr and Farrior, D S** 'Computation of geometrical and inertial properties for general areas and volumes of revolution' *Comput.-Aided Des.* Vol 8 No 4 (October 1976) pp 257–263

11 **Timmer, H G and Stern, J M** 'Computation of glocal geometric properties of solid objects' *Comput.-Aided Des.* Vol 12 No 6 (1980) pp 301–304

12 **Lien, S and Kajiya, J T** 'A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra' *IEEE Comput. Graph. Applic.* Vol 4 No 9 (October 1984) pp 35–41

13 **Woo, T C** 'A combinatorial analysis of boundary data structure schemata' *IEEE Comput. Graph. Applic.* Vol 5 (1985)

14 **Amazigo, J C and Rubenfeld, L A** *Advanced Calculus* Wiley (1962)

15 **Apostol, T M** *Calculus* Blaisdell (1980)

16 **Requicha, A A G** 'Representations for rigid solids: theory, methods and systems' *ACM Comput. Surv.* Vol 12 No 4 (1980) pp 437–464

17 **Paoluzzi, A, Ramella, M and Santarelli, A** 'A Boolean algebra over linear polyhedra' *Comput.-Aided Des.* Vol 21 No 9 (November 1989) pp 474–484

18 **Huang, T C** *Engineering Mechanics* (Vols 1 and 2) Addison-Wesley (1967)

APPENDIX 1: PROOFS

Structure products on a polygon π in $\mathscr{R}^2$

It is possible to write

$$II_{\pi}^{\alpha\beta} \equiv \iint_{\pi} x^{\alpha} y^{\beta} \, dS = \iint_{\pi} \frac{\partial}{\partial x} \left(\frac{x^{\alpha+1} y^{\beta}}{\alpha+1} \right) dS = \iint_{\pi} \nabla \cdot \mathbf{F} \, dS$$

where $\mathbf{F} = (x^{\alpha+1} y^{\beta}/(\alpha+1), 0)$. Then, by applying the Divergence theorem, it follows

$$II_{\pi}^{\alpha\beta} = \oint_{\partial\pi} \mathbf{F} \cdot \mathbf{n} \, dl$$

where $\partial\pi$ is the polygon boundary, counterclockwise oriented. It is well known (see, for example Amazigo and Rubenfeld[14] and Apostol[15]) that

$$\oint_{\partial\pi} \mathbf{F} \cdot \mathbf{n} \, dl = \int_{\partial\pi} -F_y \, dx + F_x \, dy$$

and hence

$$\oint_{\partial\pi} \mathbf{F}\cdot\mathbf{n}\,dl = \int_{\partial\pi} F_x\,dy = \frac{1}{\alpha+1}\int_{\partial\pi} x^{\alpha+1}\,dy$$

Thus, by using the parametric equations of the edges

$$\begin{aligned}
\Pi_\pi^{\alpha\beta} &= \frac{1}{\alpha+1}\sum_{i=1}^{n}\int_0^1 (x_i + X_i t)^{\alpha+1}(y_i + Y_i t)^{\beta} Y_i\,dt \\
&= \frac{1}{\alpha+1}\sum_{i=1}^{n}\sum_{h=0}^{\alpha+1}\int_0^1 \binom{\alpha+1}{h} x_i^{\alpha+1-h}(X_i t)^h \\
&\quad \times \sum_{k=0}^{\beta}\binom{\beta}{k} y_i^{\beta-k}(Y_i t)^k Y_i\,dt \\
&= \frac{1}{\alpha+1}\sum_{i=1}^{n}\sum_{h=0}^{\alpha+1}\binom{\alpha+1}{h} x_i^{\alpha+1-h}X_i^h \\
&\quad \times \sum_{k=0}^{\beta}\binom{\beta}{k} y_i^{\beta-k}Y_i^{k+1}\int_1^1 t^{h+k}\,dt
\end{aligned}$$

Hence, formula (4) easily follows.

Structure product on unit triangle in $\mathscr{R}^2$

It is desirable to simplify equation (4) by the following substitutions

$$\begin{aligned}
&x_1 = 0, \quad x_2 = 1, \quad x_3 = 0 \\
&y_1 = 0, \quad y_2 = 0, \quad y_3 = 1 \\
&X_1 = 1, \quad X_2 = -1, \quad X_3 = 0 \\
&Y_1 = 0, \quad Y_2 = 1, \quad Y_3 = -1
\end{aligned} \tag{18}$$

In order to correctly evaluate the derived expression, it is necessary to separate the terms with null exponent. From equation (4), with $\alpha, \beta \neq 0$, we have, by breaking the k summation

$$\begin{aligned}
\Pi^{\alpha\beta} &= \frac{1}{\alpha+1}\sum_{i=1}^{3}\sum_{h=0}^{\alpha+1}\binom{\alpha+1}{h} x_i^{\alpha-h+1}X_i^h \\
&\quad \times\left[\sum_{k=0}^{\beta-1}\frac{\binom{\beta}{k}}{h+k+1} y_i^{\beta-k}Y_i^{k+1} + \frac{1}{h+\beta+1}Y_i^{\beta+1}\right]
\end{aligned} \tag{19}$$

Similarly, it is necessary to break the summation with respect to the h index. Then by substituting the values given by (18), the desired expression for the monomial integral over the unit triangle (5) follows*

$$\begin{aligned}
\Pi^{\alpha\beta} &= \frac{1}{\alpha+1}\left[\sum_{h=1}^{\alpha}\binom{\alpha+1}{h}\frac{(-1)^h}{h+\beta+1} + \frac{1}{\beta+1}\right. \\
&\quad \left. + \frac{(-1)^{\alpha+1}}{\alpha+\beta+2}\right] = \frac{1}{\alpha+1}\sum_{h=0}^{\alpha+1}\binom{\alpha+1}{h}\frac{(-1)^h}{h+\beta+1}
\end{aligned} \tag{20}$$

Structure products on triangle in $\mathscr{R}^3$

Finally, the integral $\Pi_\tau^{\alpha\beta\gamma}$ is explicitly derived. Starting from equations (9)–(7), gives

$$\begin{aligned}
\Pi_\tau^{\alpha\beta\gamma} &= |\mathbf{a}\times\mathbf{b}|\iint_{\tau'}(x_o + a_x u + b_x v)^{\alpha}(y_o + a_y u + b_y v)^{\beta} \\
&\quad \times (z_o + a_z u + b_z v)^{\gamma}\,du\,dv
\end{aligned}$$

then, grouping terms and applying Newton's formula for binomial powers, gives

$$\begin{aligned}
\Pi_\tau^{\alpha\beta\gamma} &= |\mathbf{a}\times\mathbf{b}|\iint_{\tau'}\sum_{h=0}^{\alpha}\binom{\alpha}{h}x_o^{\alpha-h}(a_x u + b_x v)^h \\
&\quad \times \sum_{k=0}^{\beta}\binom{\beta}{k}y_o^{\beta-k}(a_y u + b_y v)^k \sum_{m=0}^{\gamma}\binom{\gamma}{m}z_o^{\gamma-m} \\
&\quad \times (a_z u + b_z v)^m\,du\,dv \\
&= |\mathbf{a}\times\mathbf{b}|\sum_{h=0}^{\alpha}\binom{\alpha}{h}x_o^{\alpha-h}\sum_{k=0}^{\beta}\binom{\beta}{k}y_o^{\beta-k}\sum_{m=0}^{\gamma}\binom{\gamma}{m} \\
&\quad \times z_o^{\gamma-m}\iint_{\tau'}(a_x u + b_x v)^h(a_y u + b_y v)^k \\
&\quad \times (a_z u + b_z v)^m\,du\,dv \\
&= |\mathbf{a}\times\mathbf{b}|\sum_{h=0}^{\alpha}\binom{\alpha}{h}x_o^{\alpha-h}\sum_{k=0}^{\beta}\binom{\beta}{k}y_o^{\beta-k}\sum_{m=0}^{\gamma}\binom{\gamma}{m} \\
&\quad \times z_o^{\gamma-m}\iint_{\tau'}\sum_{i=0}^{h}\binom{h}{i}(a_x u)^{h-i}(b_x v)^i\sum_{j=0}^{k}\binom{k}{j} \\
&\quad \times (a_y u)^{k-i}(b_y v)^j\sum_{l=0}^{m}\binom{m}{l}(a_z u)^{m-l}(b_z v)^l\,du\,dv \\
&= |\mathbf{a}\times\mathbf{b}|\sum_{h=0}^{\alpha}\binom{\alpha}{h}x_o^{\alpha-h}\sum_{k=0}^{\beta}\binom{\beta}{k}y_o^{\beta-k}\sum_{m=0}^{\gamma}\binom{\gamma}{m} \\
&\quad \times z_o^{\gamma-m}\sum_{i=0}^{h}\binom{h}{i}a_x^{h-i}b_x^i\sum_{j=0}^{k}\binom{k}{j}a_y^{k-j}b_y^j\sum_{l=0}^{m}\binom{m}{l} \\
&\quad \times a_z^{m-l}b_z^l\iint_{\tau'}u^{(h+k+m)-(i+j+l)}v^{(i+j+l)}\,du\,dv
\end{aligned} \tag{21}$$

APPENDIX 2: IMPLEMENTATION

A representation scheme is defined by Requicha[16] as a mapping from a space of geometrical models to a set of symbolic representations of objects. Boundary representation schemes are mappings from the set of represented objects to some set of topological and geometrical relationships among their vertices, edges and faces. Many different data structures may be used to give a complete representation of the object topology[13]. The winged-triangle (wt) representation scheme[17] is based on the storage of a simplicial complex associated with the represented object. The scheme is a two-step mapping between linear polyhedra and data structures stored in a computer memory. The former mapping associates a polyhedron P with the set of manifold surfaces whose support space (union of simplices) coincides with the polyhedron boundary. Any of such surfaces, that we call an envelope of P, is a manifold, even if P is a nonmanifold polyhedron. It turns out an easier implementation of many algorithms. The latter mapping associates an envelope with a data structure describing one of its triangulations.

* Although this equation looks asymmetric, one can check by numerical computation that $\Pi^{\alpha\beta} = \Pi^{\beta\alpha}$

An implementation of the formulas shown in the paper is given in order to alow the reader to verify the proposed integration method. The authors express the integral properties of homogeneous polyhedra, making only use of the usual definitions from a mechanical textbook[18]. A programming trick is extensively used, with the aim of improving readability of vector components. It consists of indexing arrays with the labels *x*, *y*, *z* instead of using the first three integers. The language used is standard Pascal. The data structure for describing polyhedra and the procedure Read-Polyhedron refer to the geometric modeller Minerva developed at the University of Rome 'La Sapienza'[17].

```
Type
  index = (x,y,z);
  vet3 = array[index] of real;
  triangle = record vo, va, vb : vet3 end;
var
  P : polyhedron;
  Surface, Volume: real;
  FirstMoment, Centroid, SecondMoment,
  InertiaProduct, InertiaMoment : vet3;
begin
  ReadPolyhedron(P);
  Surface := II(P, 0, 0, 0);
  Volume := III(P, 0, 0, 0);
  FirstMoment[x] := III(P, 1, 0, 0);
  FirstMoment[y] := III(P, 0, 1, 0);
  FirstMoment[z] := III(P, 0, 0, 1);
  Centroid[x] := FirstMoment[x]/Volume;
  Centroid[y] := FirstMoment[y]/Volume;
  Centroid[z] := FirstMoment[z]/Volume;
  SecondMoment[x] := III(P, 2, 0, 0);
  SecondMoment[y] := III(P, 0, 2, 0);
  SecondMoment[z] := III(P, 0, 0, 2);
  InertiaProduct[x] := III(P, 0, 1, 1);
  InertiaProduct[y] := III(P, 1, 0, 1);
  InertiaProduct[z] := III(P, 1, 1, 0);
  InertiaMoment[x] := SecondMoment[y] + SecondMoment[z];
  InertiaMoment[y] := SecondMoment[x] + SecondMoment[z];
  InertiaMoment[z] := SecondMoment[x] + SecondMoment[y];
end.
```

The two main integration functions over linear homogeneous polyhedra and over linear homogeneous surfaces are given. The first function, II, is the summation of the integrals (10) over the whole set of triangles of the surface, that may be either the boundary of a solid or an open surface. The second function, III, is a direct implementation of formula (15). Of course, in the case of integration over a surface, all the 2-simplices may be coplanar – if their union is a planar polygon (arbitrarily oriented in $\mathscr{R}^3$). As it is possible to see, the function II for computation of structure products over surfaces, as well as the function III for the computation of structure products over solids, make use of the same primitive function *T* for the computation of structure products over a triangle in $\mathscr{R}^3$. The procedure GetTriangle returns the *i*th 2-simplex from the polyhedron *P*.

```
function II(P : polyhedron; alpha, beta, gamma : integer): real;
var W : real; i : integer, tau : triangle;
begin
  w := 0;
  for i := 1 to P.NumberOfTriangles do begin
    GetTriangle(P, i, tau);
    w := w + T (tau, alpha, beta, gamma);
  end;
  II := w;
  end;

function III(P : polyhedron; alpha, beta, gamma : integer): real;
var W : real; i : integer; tau : triangle; a, b, c : vet3;
begin
  w := 0;
  for i := 1 to P.NumberOfTriangles do begin
    GetTriangle(P, i, tau);
    Difference(tau.va, tau.vo, a);
    Difference(tau.vb, tau.vo, b);
    Times(a, b, c);
    w := w + (c[x]/Magnitude(c)) * T(tau, alpha + 1, beta, gamma);
  end;
  III := w/(alpha + 1);
end;
```

M and *T* are the most internal integration functions, which compute, respectively, the requested structure product over the unit 2-simplex (5) of the plane and over any 2-simplex in $\mathscr{R}^3$. Both are direct implementation of formulas (6) and (10). The function BinCoeff (*i*, *k*) gives the binomial coefficient $\binom{i}{k}$.

```
function M(alpha, beta : integer): real;
var a : real; i : integer;
begin;
  a := 0;
  for l := 0 to alpha + 1 do
    a := a + BinCoeff (alpha + 1, 1) * Power(-1, l)/(l + beta + 1);
    M := a/(alpha + 1);
end;
function T(tau : triangle; alpha, beta, gamma : integer): real;
var s1, s2, s3, s4 : real; i, j, l, h, k, m : integer; a, b, c : vet3;
begin;
  with tau do begin
    Difference(va, vo, a); Difference(vb, vo, b);
    s1 := 0;
    for h := 0 to alpha do
    for k := 0 to beta do
    for m := 0 to gamma do begin
      s2 := 0;
      for i := 0 to h do begin
        s3: = 0;
        for j := 0 to k do begin
          s4: = 0;
          for l := 0 to m do
            s4 := s4 + BinCoeff (m, l)
              *Power(a[z], m - l)
              *Power(b[z], l) * s4
              *M(k + k + m - i - j - l, i + j + l);
          s3: = s3 + BinCoeff (k, j)
            *Power(a[y], k - j)
            *Power(b[y], j) * s4;
        end;
        s2 := s2 + BinCoeff (h, i)
          *Power(a[x], h - i)
          *Power(b[x], i) * s3;
      end;
      s1 := s1 + BinCoeff (alpha, h)
        *BinCoeff (beta, k)
        *BinCoeff (gamma, m)
        *Power(vo[x], alpha - h)
        *Power(vo[y], beta - k)
        *Power(vo[z], gamma - m) * s2;
    end
    Times(a, b, c);
    T := s1 * Magnitude(c);
  end;
end;
```