ARTIFICIAL SATELLITES, Vol. 53, No. 1 – 2018
DOI: 10.2478/arsa-2018-0002

SELF-SHADOWING OF A SPACECRAFT IN THE COMPUTATION OF SURFACE FORCES.
AN EXAMPLE IN PLANETARY GEODESY

G. Balmino (1) (✉), J.C. Marty (2)
Centre National d'Etudes Spatiales
Groupe de Recherches de Geodesie Spatiale
18, Avenue Edouard Belin, 31401 Toulouse Cedex 9, France
(1) Tel.: +33 (0)5 61 33 2889
E-mail: georges.balmino@get.omp.eu
(2) Tel.: +33 (0)5 61 33 2996
E-mail: jean-charles.marty@cnes.fr

ABSTRACT

We describe in details the algorithms used in modelling the self-shadowing between spacecraft components, which appears when computing the surface forces as precisely as possible and especially when moving parts are involved. This becomes necessary in planetary geodesy inverse problems using more and more precise orbital information to derive fundamental parameters of geophysical interest. Examples are given with two Mars orbiters, which show significant improvement on drag and solar radiation pressure model multiplying factors, a prerequisite for improving in turn the determination of other global models.

Keywords : Surface forces. Self-shadowing. Polyhedral decomposition. Pixelation. Mars Planet.

1 INTRODUCTION

The modelling of disturbing forces acting on any spacecraft, Earth or planetary artificial satellite, has been the subject of great attention since decades. Not only is this necessary for precise navigation in the environment of a studied body, but it is essential in the analysis of physical phenomena at stake in space dynamics and in the derivation of underlying parameters, such as gravity field and atmospheric density model coefficients, from observations (e.g. distance, Doppler measurements) of trajectories, a well-known inverse problem of celestial mechanics and satellite geodesy.

The surface forces (atmospheric drag, direct, rediffused and infrared solar radiation pressure, and thermal effects) are especially difficult to model all the more as they depend on the different elements of a spacecraft which may be geometrically and physically complex and may vary with time. Not only the spacecraft orientation in space (its attitude) is usually time dependent but spacecraft elements themselves may also vary their position with respect to each other. In this context an approximate approach is usually adopted which consists in treating each surface element (a so-called plate) independently, also in simplifying their relative motions if any. In so doing it has been shown (Mazarico, 2008) that the occultation of some surface elements facing a given direction (of the drag, or of the direct solar radiation pressure for instance) by others, which is called *self-shadowing*, may significantly impact the value of the surface forces, and in turn affect the accuracy with which some parameters may be determined. Examples of Martian atmosphere drag coefficients retrieved from orbit observations of Mars Odyssey (ODY) and Mars Reconnaissance Orbiter (MRO) with and without accounting for self-shadowing, are given in Mazarico et al. (2009). In turn these uncertainties may affect the determination of other parameters such as the time variable gravity field spherical harmonics (Konopliv et al., 2011).

The computation of self-shadowing effects has not been often documented in the literature. It seems that the first significant work was done by Ziebart (2001) who solved the problem via ray tracing methods; it was then applied to GPS and altimetry satellites (Ziebart et al., 2005). His approach adopts a two steps strategy. First, the ray tracing method is used to accurately determine the effects (assuming an approximate orbit) and to fit an analytical model. This model is then introduced in the orbit determination step. Then Mazarico (2008) and Mazarico et al.(2009) used a geometrical (projective) approach, not described in every detail but results show its efficiency albeit at the expense of high computational costs – this being improved by finally implementing the method in two steps.

On our side a first study of self-shadowing was made by Clavier (1991) but was discontinued. A novel analysis was performed and algorithms were designed and implemented by Balmino (2007). They were applied to planetary geodesy studies conducted since many years, and in particular in our efforts to model the gravity field of Mars and its seasonal variations (Marty et al., 2009). Their extensive use over the years makes them worthy of this article. The core of this paper is a detailed description of the algorithms - which requires many figures for clarity, their overall performance, followed by examples of data analysis from two Martian missions performed with the improved computation of surface forces.

2 SELF-SHADOWING IN THE CONTEXT OF ORBIT DETERMINATION

The POD (precise orbit determination) software used in our satellite dynamics works and subsequent computation is GINS (Geodesie par Integrations Numeriques Simultanees), documented in Marty et al. (2011).

Our strategy may differ significantly from others, in the sense that the method was developed from the beginning with the aim to compute exactly in GINS all geometrical elements and self-shadowing conditions at each time step of the numerical integration process of the POD, once per epoch in the predictor phase of the integrator (we use the pseudo-correction technique in the corrector, which consists in not recomputing some parts of the disturbing accelerations). No call to outside and predetermined results is necessary, which makes the method quite flexible especially in cases of highly mobile spacecraft parts.

Figure 1 shows a spacecraft composed of several elements – some of them being mobile, in orbit around a given (central) body. Each element can be decomposed into, or approximated by several plates. The four main types of surface forces are atmospheric drag, direct solar radiation pressure, indirect solar radiation pressure (reflected by the central body and function of its albedo), and infrared radiation pressure (emitted by the central body). They result in perturbing accelerations which depend on physical models (e.g., density of the atmosphere), plate surface properties and characteristic vectors of these forces, respectively $\vec{V}$, $\vec{s}$, $\vec{a}$, $\vec{i}$ (for $\vec{a}$ and $\vec{i}$ we use mean directions not accounting for albedo or emissivity spatial variations). We assume that there is no self-shadowing for disturbing forces due to thermal effects on each element plate.

Our goal is to determine all along the orbit if any spacecraft element plate is in the shadow of another one with respect to each of the surface force characteristic vectors and, if so, to compute the reduced surface on which the force acts, the resulting accelerations applied to the plate being proportional to this surface.

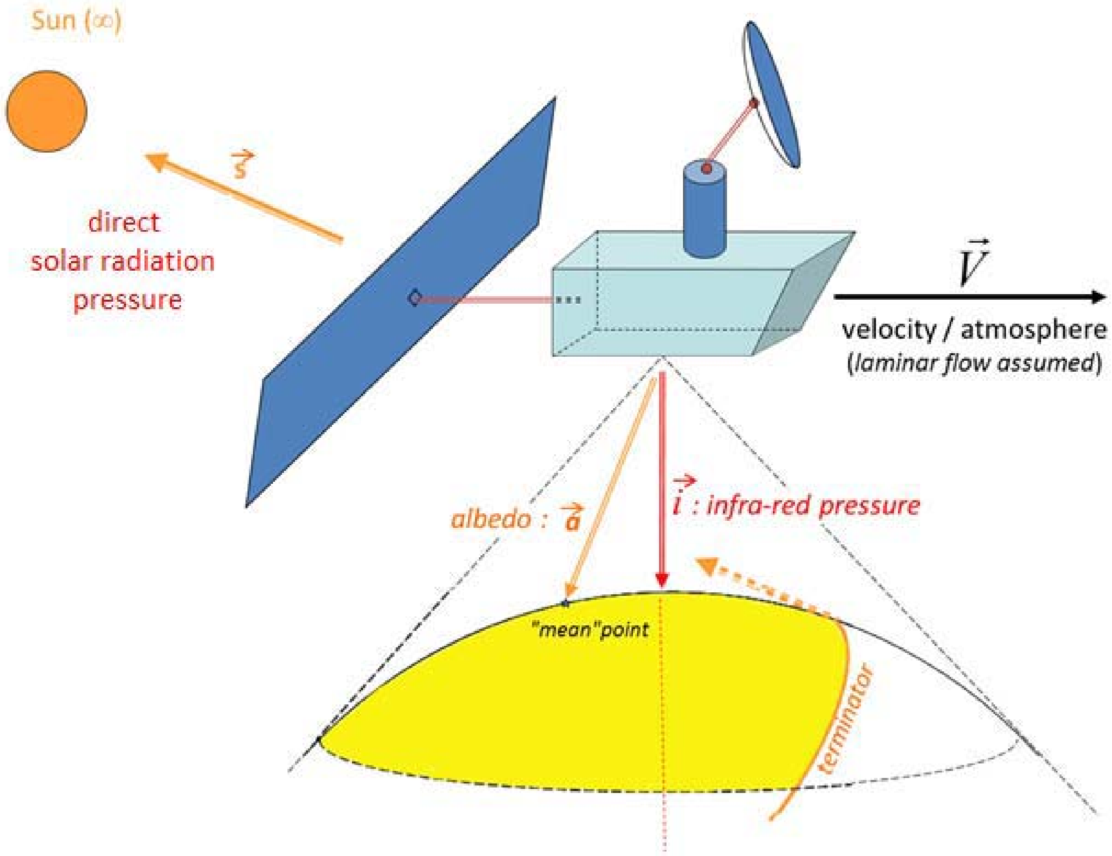


Fig. 1. Spacecraft subject to surface forces. Each element is decomposed into flat plates. For each force the shadowing of each element plate by another one is determined by the occultation conditions in the direction of the characteristic vector of the force : $\vec{V}, \vec{s}, \vec{a}$, or $\vec{i}$.

3 SPACECRAFT DECOMPOSITION INTO SURFACE ELEMENTS AND REFERENCE SYSTEMS

The geometry of a spacecraft is usually approximated by a macro-model. Such a model may consist of a bus, one or several solar panels, an external antenna, which can be approximated by simple surfaces such as polyhedrons (limited by flat polygonal plates), parts of cylinders, cones, paraboloids, etc.

Finite expressions exist for computing the surface forces acting on such elements, but taking into account self-shadowing makes these formulas useless in the case of curved surfaces; we decided to turn to a simpler approach in which each curved surface is approximated by a certain number of flat plates. Note that some of these plates must be considered twice (front and rear sides, with opposite normals) to account properly for the action of the surface forces.

With respect to the usual case (no self-shadowing) where it suffices to know the areas and orientations of the plates, we now need to define the coordinates of vertices of each plate in a given (local) reference system, for instance fixed to the bus, or linked to other spacecraft elements via intermediate reference frames.

Appendix A provides these local coordinates for flat plates arising from the decomposition of most frequently encountered structural parts, and limited by convex polygons – convexity allows to use simpler and faster algorithms in the following.

At this stage it is important to note that special cases of self-shadowing geometry (see section 4.3.1) will be quickly discriminated provided that any two surfaces cannot intersect - although they may have a common edge or their intersection may be inside one surface but outside the other (fig. 2). This may be checked at the macro-model definition level for fixed plates. For mobile parts we will apply some algorithms in assuming that the decomposition rule is not verified.

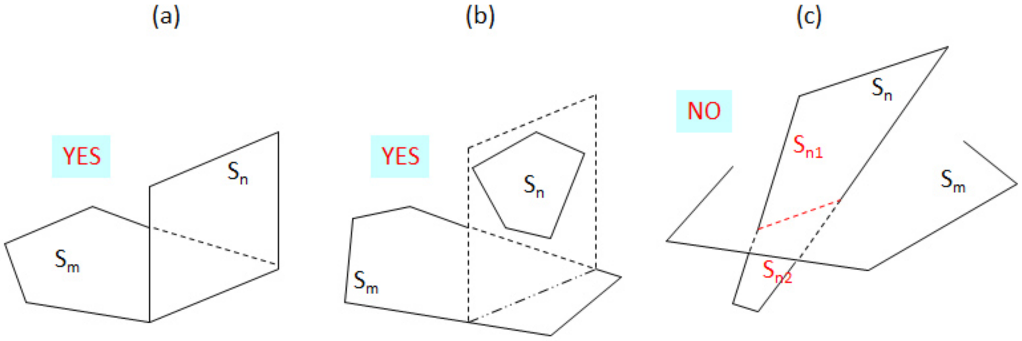


Fig. 2. Decomposition rule: (a) two plates are allowed to share one edge (or part of it); (b) the intersection of S_n and S_m is inside S_m but outside S_n ; (c) S_n is decomposed into two sub-surfaces.

All computations are to be done in a reference system (r_0) fixed with respect to the spacecraft main body $\mathcal{P}_0$ (the bus). The other parts $\mathcal{P}_s$ of the spacecraft are of two types. The fixed parts are also referenced to (r_0), whereas the mobile parts are fixed in their own reference system $(r_s) = (C_s;\ x_s, y_s, z_s)$, see figure 3, and (r_s) is mobile with respect to (r_0). A mobility index table $\mathcal{M}_s$ (0 or 1) is thus defined.

In the following, in order to avoid the use of two indices and thus to simplify notations, we will call S_j any plate resulting from the decomposition of a part $\mathcal{P}_s$ $(s \geq 0)$; S_j is in the plane Π_j , it has the same mobility index as the part to which it belongs and is referred to the same system of coordinates.

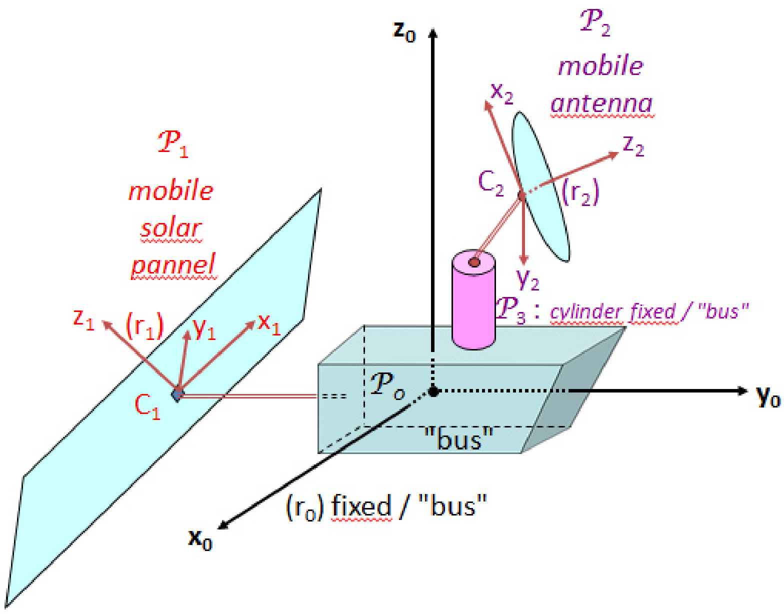


Fig. 3. Example of spacecraft structure. $\mathcal{P}_0$ is the bus, with its reference system (r_0). $\mathcal{P}_1$ and $\mathcal{P}_2$ are mobile parts, but fixed in their own reference system, (r_1) and (r_2) respectively. $\mathcal{P}_3$ is fixed in (r_0).

When S_j is fixed in (r_0), i.e. $\mathcal{M}_s(j) = 0$, the coordinates of its vertices are directly given in (r_0). For further use, we then define the new reference frame $(\sigma_j) = (O; \mathbf{u_j}, \mathbf{v_j}, \mathbf{w_j})$ where O is the origin of (r_0), $\mathbf{v_j}$ is a unit vector parallel to A_1A_2 , $\mathbf{w_j} = \mathbf{n_j}$ (the unit vector normal to Π_j, the orientation of which depends on the ordering of the vertices), and $\mathbf{u_j} = \mathbf{v_j} \times \mathbf{w_j}$ (see fig. 4a). This implies a transformation matrix $t(S_j)$ such that, for any vector **H** we have:

$$H[\sigma_j] = t(S_j)\, H[r_0] \,, \qquad (1)$$

where H[f] denotes the vector components in the reference frame (f).

For a moving surface S_j, i.e. $\mathcal{M}_s(j) = 1$, the coordinates of its polygon vertices are provided in $(r_j) = (C_j; x_j, y_j, z_j)$ where the vector OC_j is given in (r_0) - see fig. 4b ; also given is the matrix Q_j such that for any vector **H**:

$$H[r_j] = Q_j\, H[r_0] \qquad (2)$$

In practice the (r_0) and (r_j) systems are independently related to the reference system (R_0) in which the spacecraft orbit is defined, by two matrices P_0 and P_j so that: $H[r_0] = P_0\, H[R_0]$ and $H[r_j] = P_j\, H[R_0]$. Therefore:

$$Q_j = P_j\, P_0^T \qquad (3)$$

As before we define a system $(\sigma_j) = (C_j; \mathbf{u_j}, \mathbf{v_j}, \mathbf{w_j})$, this time of origin C_j, and we have a matrix $t(S_j)$ by which:

$$H[\sigma_j] = t(S_j)\, H[r_j] \qquad (4)$$

and finally:

$$H[\sigma_j] = t(S_j)\, Q_j\, H[r_0] \qquad (5)$$

The (σ_j^*) system (see fig. 4c) is sometimes needed; it is translated from the (σ_j) system and has its origin Ω_j in the Π_j plane. All points in Π_j have the same coordinate $\mathbf{w}$ in (σ_j) and $\mathbf{w}^* = 0$ in (σ_j^*); furthermore $\mathbf{u}^* = \mathbf{u}$ and $\mathbf{v}^* = \mathbf{v}$, so that $\{\mathbf{u}, \mathbf{v}\}$ and $\{\mathbf{u}^*, \mathbf{v}^*\}$ will be used synonymously.

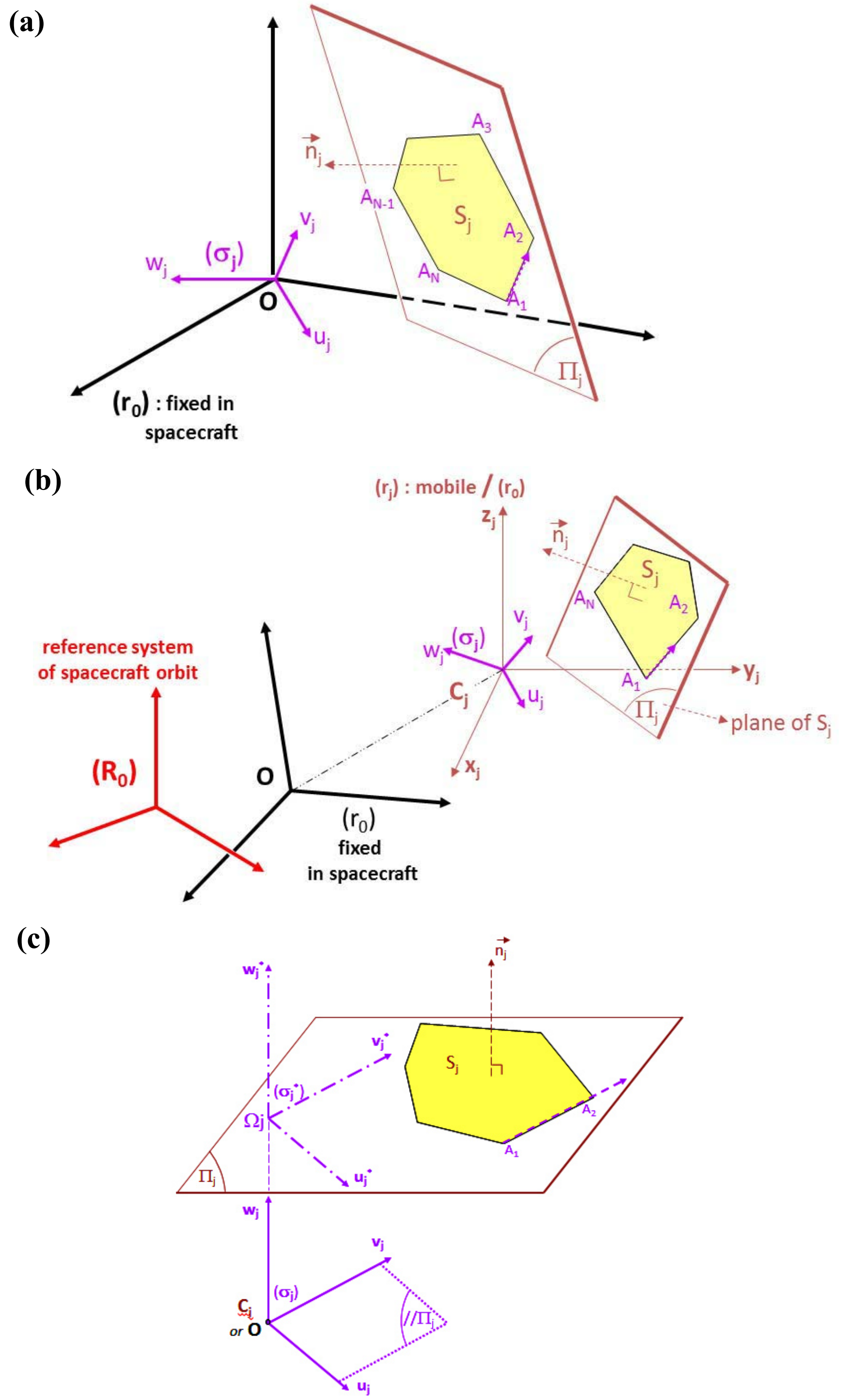


Fig. 4. The reference systems at stake: (R_0) is a reference system in which the orbit is given, (r_0) is a system fixed in the spacecraft main part (the bus). **(a)** Case of a plate fixed in the (r_0) system ; **(b)** Case of a moving plate, but fixed in its own coordinate system (r_j); **(c)** In both cases we also define the intermediate systems (σ_j) and (σ_j^*).

4 ALGORITHMS

From now on $\boldsymbol{S}_0$ will be any one of the S_j surfaces previously defined, and $\boldsymbol{S}$ will be another one. For any given vector $\vec{v}$, characteristic of one of the non-gravitational forces, we want to determine if $\boldsymbol{S}_0$ is in the shadow of $\boldsymbol{S}$, completely or partially (fig. 5).

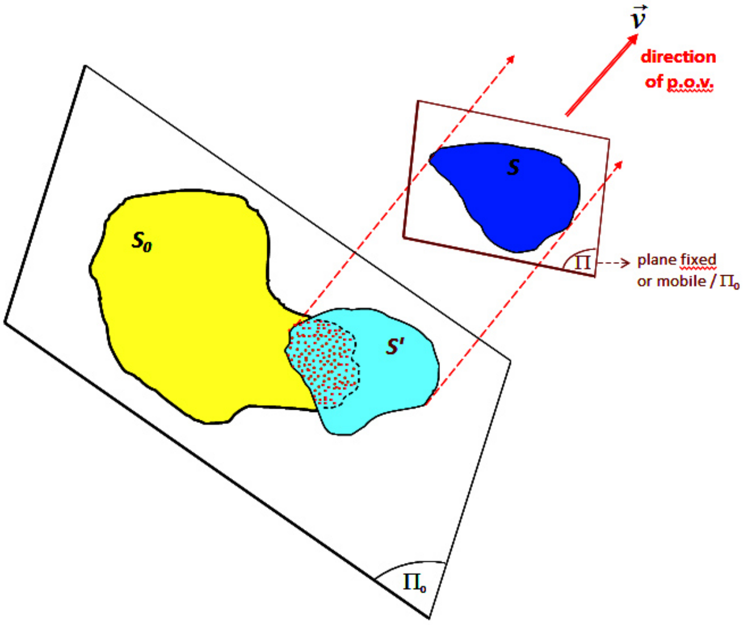


Fig. 5. Geometry of the general shadowing problem. The unit vector $\vec{v}$ is the point of view direction, $\boldsymbol{S'}$ is the oblique projection of $\boldsymbol{S}$ on Π_0 (the plane of $\boldsymbol{S}_0$). All surfaces will be convex in the rest of the paper.

The general problem is to determine $\boldsymbol{S'} = proj_{\Pi_0}^{\vec{v}}(\boldsymbol{S})$: the oblique projection (parallel to $\vec{v}$) of $\boldsymbol{S}$ onto Π_0 , the plane of $\boldsymbol{S}_0$, and then $\boldsymbol{S}_0 \cap \boldsymbol{S'}$. In the present case this is simplified since the surfaces are convex polygons and we need only to compute the ratio area($\boldsymbol{S}_0 \cap \boldsymbol{S'}$) / area($\boldsymbol{S}_0$).

4.1 Preliminary conditions

It is easier in some cases, and especially for surfaces fixed in (r_0), to firstly determine the permissible ranges of directions between two such surfaces for which shadowing may or may not occur. This accelerates the application of subsequent steps of the algorithms.

To do so we construct a discretized bundle which consists of a set of vectors $\overrightarrow{M_0M}$ where M_0 and M belong to the edges of the polygons defining $\boldsymbol{S}_0$ and $\boldsymbol{S}$, respectively (fig. 6). The number of these vectors depends on the complexity of the polygons and has been empirically determined. Then the minimum and maximum values of the elevation (e) and azimuth (a) of the M_0M vectors are used to discriminate the possible self-shadowing directions; there may be two sets of limiting values for the azimuth since it is counted in a 2π interval.

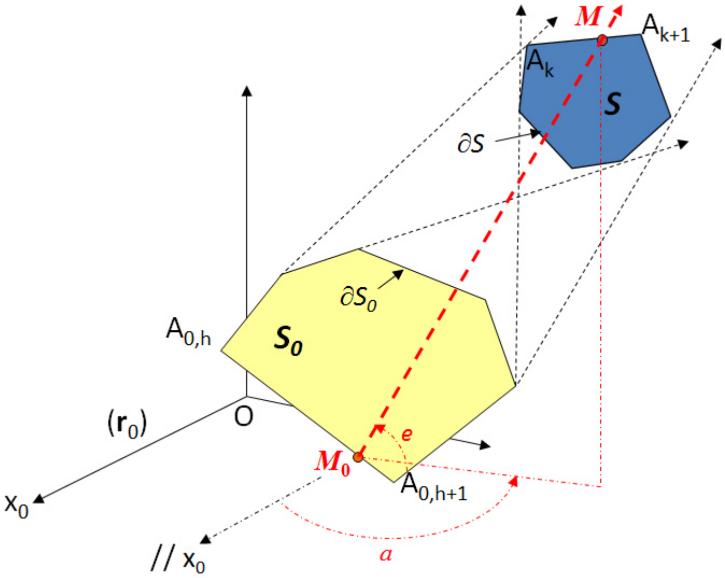


Fig. 6. The azimuth (ɑ) and elevation (*e*) of $\overrightarrow{M_0M}$ are computed for series of points M_0 and *M* on the edges ∂$\boldsymbol{S_0}$ and ∂$\boldsymbol{S}$, respectively, of the considered surfaces, here fixed in (r_0).

4.2 Basic transformation: oblique projection

The geometry of the transformation is depicted on figure 7. *P* being any point in space (subsequently it will be a vertex of $\boldsymbol{S}$) we want to determine $P' = proj_{\Pi_0}^{\vec{v}}(P)$. *P'* may be defined by:

$$\overrightarrow{G_0P'} = \overrightarrow{G_0P} + \lambda\,\vec{v}$$

$$\overrightarrow{G_0P'}\,.\,\overrightarrow{n_0} = 0\,, \qquad (6)$$

where $\overrightarrow{n_0}$ is normal to the plane Π_0 (oriented according to the $\boldsymbol{S_0}$ vertices order) and G_0 is any point in Π_0 (for instance the isobarycentre of the $\boldsymbol{S_0}$ vertices). Multiplication of the first equation by $\vec{n}_0$ then yields

$$\lambda = \left(\overrightarrow{OG_0}\,.\,\overrightarrow{n_0} - \overrightarrow{OP}\,.\,\overrightarrow{n_0}\right) / \left(\overrightarrow{n_0}\,.\,\vec{v}\right)\;, \qquad (7)$$

in which the singularity $(\overrightarrow{n_0}\,.\,\vec{v} = 0)$ is treated separately.

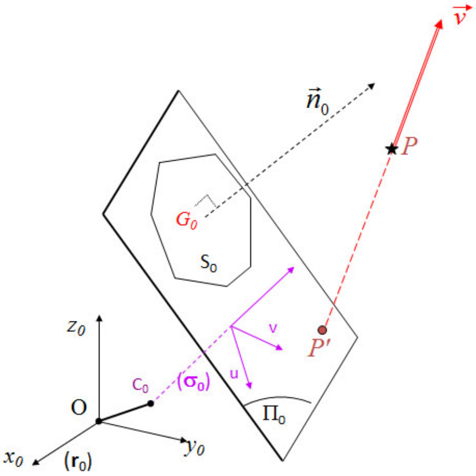


Fig. 7. Oblique projection parallel to the $\vec{v}$ direction. (σ_0) is the reference system (C_0; **u**,**v**,**w**) attached to $\boldsymbol{S_0}$; G_0 is fixed in the plane Π_0 and $\overrightarrow{n_0}$ is normal to it.

4.3 Algorithms of shadowing of S_0 by S

Here we describe in detail the shadowing occurrence conditions: shadowing (*sh*), no shadowing (*no-sh*), and possible shadowing (*p-sh*). The *p-sh* and *no-sh* cases may be determined by simple preliminary tests, and the former (*p-sh*) requires further testing (in 4.3.2 and 4.3.3) before *sh* can be confirmed and its precise effect computed.

4.3.1 Preliminary discrimination

We start by testing whether S is completely opposite to S_0 with respect to $\vec{v}$, in which case there is no shadowing. If the decomposition rule is satisfied (shown before on figure 2) then it will be sufficient to consider the isobarycentre G of S and its image G' on Π_0 , the plane of S_0 (fig. 8a), defined by

$$\overrightarrow{GG'} = \lambda_G \, \vec{v} \tag{8}$$

$$\Rightarrow \ \lambda_G > 0 \ : \ \textit{no-sh}$$
$$\Rightarrow \ \lambda_G < 0 \ : \ \textit{p-sh} \, .$$

If this rule is not satisfied (frequent case of mobile parts), then a similar condition for all the vertices P of S must be tested, with $\overrightarrow{PP'} = \lambda_P \, \vec{v}$:

$$\Rightarrow \ (\forall P) \ \lambda_P \ > 0 \ : \ \textit{no-sh} \ \text{ (fig. 8a)}$$
$$\Rightarrow \ (\forall P) \ \lambda_P \ < 0 \ : \ \textit{p-sh}, \ \text{ (fig. 8b).}$$

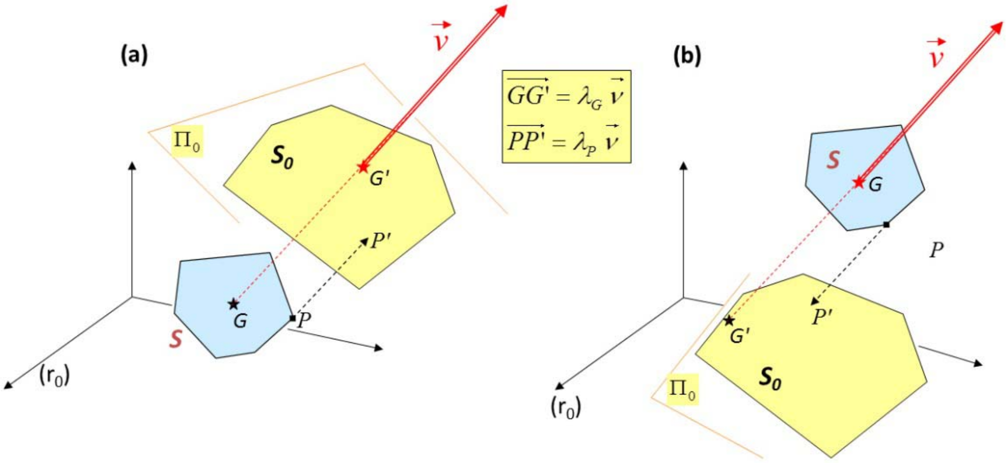

(a)
$\vec{v}$
Π_0
S_0
G'
P'
G
P
S
(r$_0$)
$\overrightarrow{GG'} = \lambda_G \, \vec{v}$
$\overrightarrow{PP'} = \lambda_P \, \vec{v}$
(b)
$\vec{v}$
S
G
P
P'
G'
S_0
Π_0
(r$_0$)

Fig. 8. Preliminary tests. **(a)** no shadowing of S_0 by S: test on G is sufficient if decomposition rule is satisfied; test on all vertices P required if not. **(b)** possible shadowing: test on isobarycentre or on all vertices depending on the case. The criteria are based on the sign of λ_G (or λ_P).

The intermediate case, when we have positive and negative λ_P values (fig. 9a), requires to also test the direction of the $\overrightarrow{QQ'}$ vector for any vertex Q of S_0 and its image Q' on Π (the plane of S), which verify:

$$\overrightarrow{QQ'} = \mu_Q \, \vec{v} \ , \tag{9}$$

and we note that all μ_Q values are here of the same sign. Then:

$$(\forall Q) \ \mu_Q \ > 0 \ : \ \textit{p-sh} \ \ \text{ (fig. 9b)}$$
$$(\forall Q) \ \mu_Q \ < 0 \ : \ \textit{no-sh} \ \text{ (fig. 9c).}$$

As before the *p-sh* case will be further analysed.

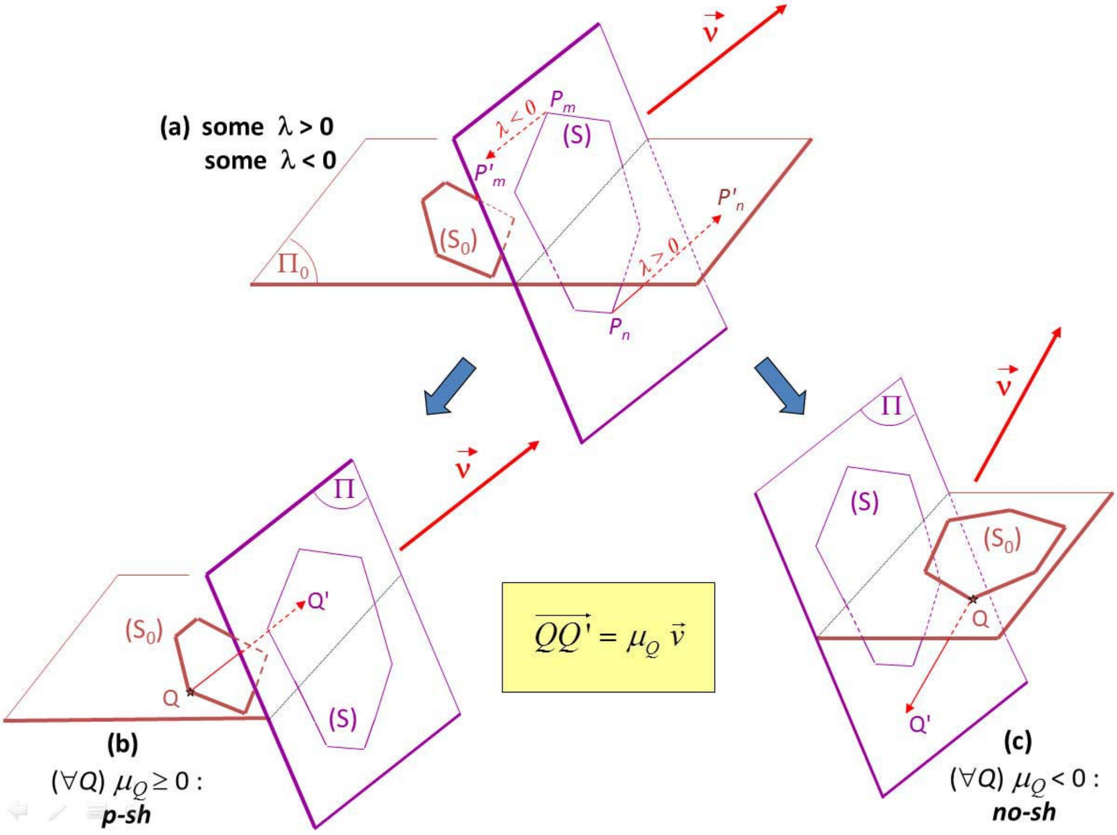


Fig. 9. Preliminary tests : intermediate case (complementary to those of fig. 8a, b) of which **(a)** shows the geometrical occurrence. **(b)** is a *p-sh* case. **(c)** is a *no-sh* case.

4.3.2 Peculiar cases

Once a *p-sh* condition is established some simple cases of *sh* and *no-sh* can be tested. For doing so we consider the (**u**,**v**) coordinates in the Π_0 plane, i.e. in (σ_0^*), of the vertices $A_{0,h}$ $(u_{0,h},v_{0,h})$ of $\boldsymbol{S_0}$, and those of the vertices A'_k (u'_k, v'_k), of $\boldsymbol{S'}$ (the image of $\boldsymbol{S}$). Then:

- $\boldsymbol{S'}$ completely outside $\boldsymbol{S_0}$ (*no-sh*), as shown on figure 10. There are two sub-cases:
  (a) $u'_{max} < u_{min,0}$ *or* $u'_{min} > u_{max,0}$ *or* $v'_{max} < v_{min,0}$ or $v'_{min} > v_{max,0}$, where the *min* and *max* subscripts correspond to the minimum and maximum values of the coordinates u_0, v_0, u', v' to which they are appended;

  (b) the conditions of case (a) are not satisfied but all vertices of $\boldsymbol{S'}$ are outside $\boldsymbol{S_0}$ *and* all vertices of $\boldsymbol{S_0}$ are also outside $\boldsymbol{S'}$: these conditions are verified by means of point-in-polygon (PIP) tests (see Appendix B) which are very simple (and fast) due to the convexity of the polygons.

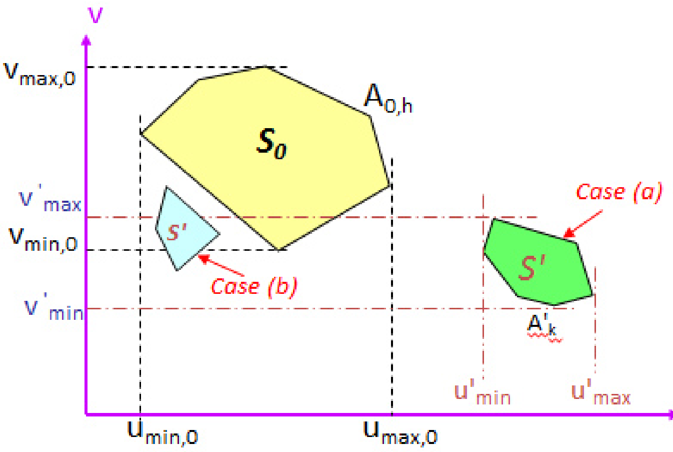


Fig. 10. Simple cases of *no-sh* : (a) is tested by comparing the extreme values of the (u,v) coordinates of $\boldsymbol{S_0}$ and $\boldsymbol{S'}$; (b) requires using a point-in-polygon algorithm.

- $\boldsymbol{S'}$ completely inside $\boldsymbol{S_0}$ (fig. 11a): this is equivalent to all image points A'_k of $\boldsymbol{S'}$ being inside the $\boldsymbol{S_0}$ ($A_{0,h}$) polygon, which is assessed by PIP testing.
- $\boldsymbol{S_0}$ totally covered by $\boldsymbol{S'}$ (fig. 11b): in this case (PIP-tested) all points $A_{0,h}$ are inside the $\boldsymbol{S'}$ polygon.

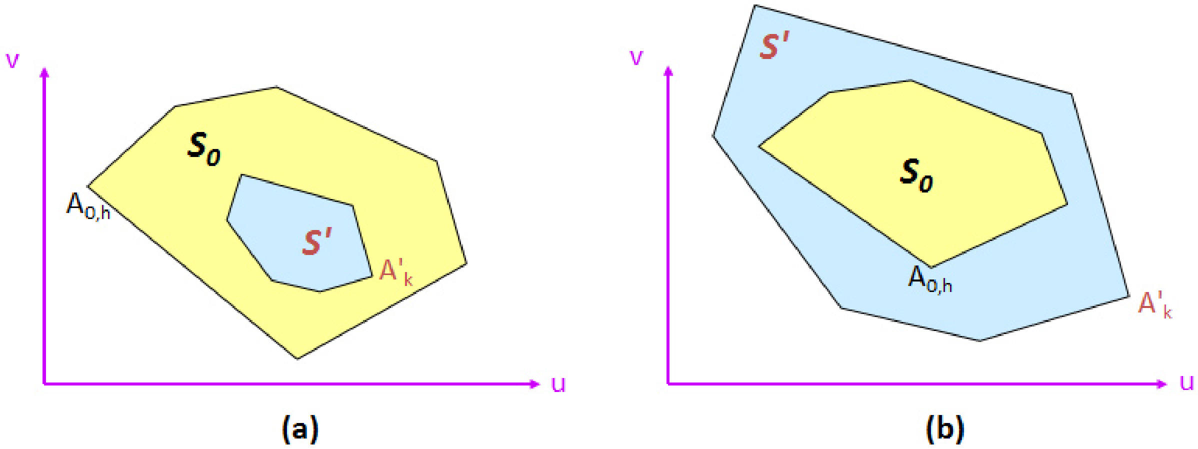


Fig. 11. Other simple cases requiring PIP testing :. **(a)** $\boldsymbol{S'}$ is completely inside $\boldsymbol{S_0}$; **(b)** $\boldsymbol{S_0}$ is totally inside $\boldsymbol{S'}$.

4.3.3 General case: pixellation

Here $\boldsymbol{S_0} \cap \boldsymbol{S'} \neq \varnothing$ but the geometry is not as simple as on figure 11. Our approach, which does not make use of any specific library, is derived from some methods used in computer graphics to superimpose or intersect images.

Firstly, and once for all, the surface $\boldsymbol{S_0}$ is discretized (pixellated) in the σ_0^* (u,v) coordinate system in assuming that we have q_{max} pixels in the largest dimension (u, or v) and a proportional number in the other one; the pixels are in general rectangular - though closer to squares as q_{max} increases. With $\{u_{min,0}; u_{max,0}\}$ and $\{v_{min,0}; v_{max,0}\}$ being the extreme values of the $\boldsymbol{S_0}$ vertices coordinates as before, and with $\Delta u_0 = u_{max,0} - u_{min,0}$, $\Delta v_0 = v_{max,0} - v_{min,0}$ we therefore define :

$$\text{- if } \Delta u_0 \geq \Delta v_0 : \quad n_{u0} = q_{max} , \quad n_{v0} = \max[2, \; n_{u0} \,.\, \Delta v_0/\Delta u_0]$$
$$\text{- if } \Delta u_0 < \Delta v_0 : \quad n_{v0} = q_{max} , \quad n_{u0} = \max[2, \; n_{v0} \,.\, \Delta u_0/\Delta v_0]. \tag{10}$$

The rectangular envelope of $\boldsymbol{S_0}$ is thus discretized in $n_{u0} \cdot n_{v0}$ pixels P_{ij} the centroids of which have coordinates, in matrix-analog form (see fig. 12):

$$\begin{aligned} v_i &= v_{max,0} - (i\text{-}1/2)\,\delta v_0\,,\ (i = 1,2,\dots\, n_{v0}) \\ u_j &= u_{min,0} + (j\text{-}1/2)\,\delta u_0\,,\ (j = 1,2,\dots\, n_{u0}) \end{aligned} \tag{11}$$

with $\delta u_0 = \Delta u_0 / n_{u0}$ and $\delta v_0 = \Delta v_0 / n_{v0}$.

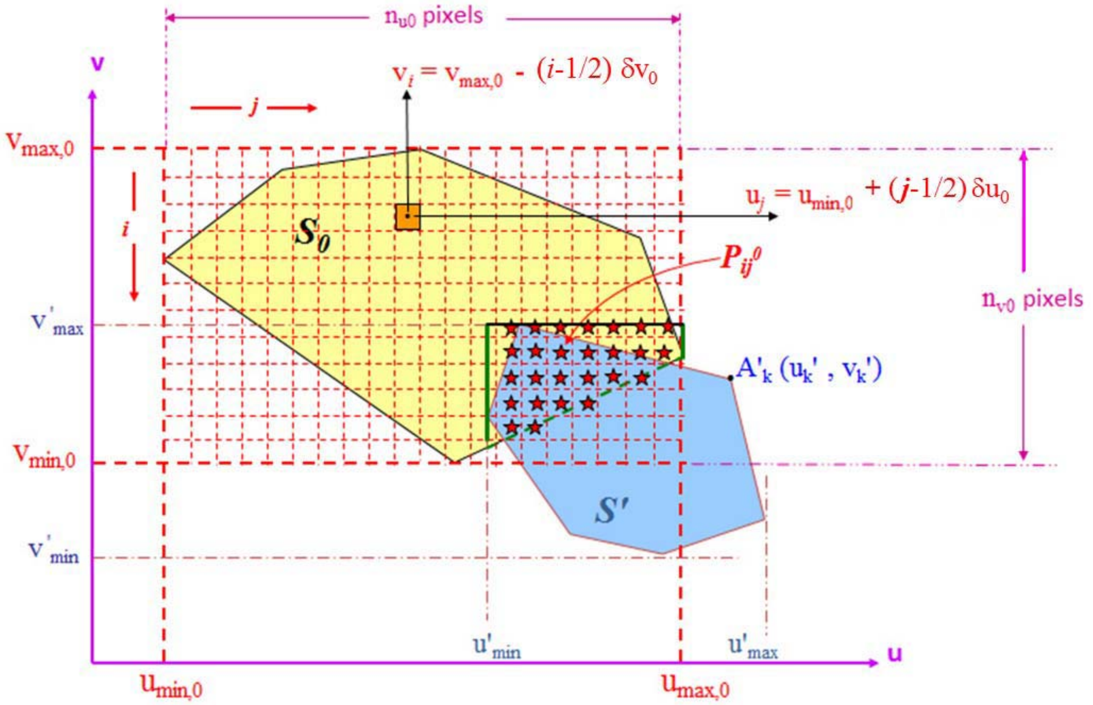


Fig. 12. General case: $\boldsymbol{S_0}$ is pixelated. The position with respect to $\boldsymbol{S'}$ (*in* or *out*) of the pixels (★) inside the part delineated by the green line, is tested by a PIP algorithm.

The pixels inside $\boldsymbol{S_0}$ form a set $\{P_{ij}^0\}$. The precision in the computation of the shadowed area goes like $1/(q_{max})^2$ at best (if $\Delta u_0 \approx \Delta v_0$) and $1/(2\, q_{max})$ at worst (if $\max[\Delta u_0/\Delta v_0\,, \Delta v_0/\Delta u_0] \geq q_{max}$). This area is equal to the area of the pixels P_{ij}^0 which are inside $\boldsymbol{S'}$, which is PIP-tested; the algorithm is further optimized by limiting the test to those pixels belonging to the intersection of the rectangular envelopes of $\boldsymbol{S_0}$ and of $\boldsymbol{S'}$, that is to:

$$\begin{aligned} &\max\,[1, (v_{max,0} - v'_{max})/\,\delta v_0] \ \leq i \leq\ \min\,[n_{v0}\,, (v_{max,0} - v'_{min})/\,\delta v_0] \\ &\max\,[1, (u'_{min} - u_{min,0})/\,\delta u_0] \ \leq j \leq\ \min\,[n_{u0}\,, (u'_{max} - u_{min,0})/\,\delta u_0]\ . \end{aligned} \tag{12}$$

with $P_{ij} \in \{P_{ij}^0\}$. The PIP Jordan algorithm (see Appendix B) is always used at this stage.

For verification and graphic display an index matrix ϖ_{ij} may be associated to the P_{ij} 's , the elements of which have the values:

$$\begin{aligned} \varpi_{ij} &= -1 \text{ if } P_{ij} \notin \{P_{ij}^0\} \\ \varpi_{ij} &= \ \ 0 \text{ if } P_{ij} \in \{P_{ij}^0\} \text{ but } P_{ij} \notin \boldsymbol{S'} \\ \varpi_{ij} &= \ \ 1 \text{ if } P_{ij} \in \{P_{ij}^0\} \text{ and } P_{ij} \in \boldsymbol{S'}. \end{aligned} \tag{13}$$

When the shadowing possibilities of $\boldsymbol{S_0}$ are analyzed for several surfaces $\boldsymbol{S_\alpha}$ simultaneously - the most frequent case, the last condition in (13) is replaced by:

$$\varpi_{ij} = \mathrm{K} \quad \text{if } P_{ij} \in \{P_{ij}^0\} \ \textit{and} \ \text{if } P_{ij} \text{ is occulted K times.} \tag{13*}$$

Figure 13 shows an example of such a case.

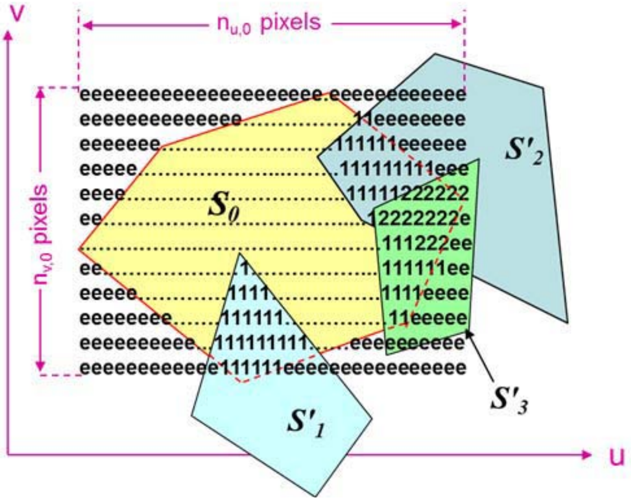


Fig. 13. Example of shadowing of $\boldsymbol{S_0}$ by three surfaces. A ϖ_{ij} matrix is associated to each pixel symbol, with $\varpi_{ij}(\mathbf{e}) = -1$, $\varpi_{ij}(.) = 0$, and $\varpi_{ij}(\mathbf{K}) = K$.

5 IMPLEMENTATION

We come back to the notations of section 3. For each surface S_j we assume that, from a preliminary analysis (in general based on the previous algorithms), we can define a shadowing index array $\Sigma(j,n,k)$ the values of which (1 or 0) tell us whether S_j may be shadowed or not by the other surfaces S_n ($n \neq j$) for any of the surface forces ($k = 1$ to 4).

5.1 Preprocessing

We firstly verify that each surface S_j, defined by its vertices in the (r_j), or (r_0) system – according to the mobility index $\mathcal{M}_s(j)$ - is planar and convex. The normal to its plane is then computed, as well as the length of the sides, the coordinates of the polygon isobarycentre, and the area. We also compute the transformation matrix from (r_j), or (r_0), to (σ_j). Then the pixellation of (S_j) is performed if $\Sigma(j,n,k) = 1$ for at least one value of n and one value of k.

5.2 Processing

At each step of the numerical integration of the spacecraft orbit, we either keep all elements (scalar, vectorial) of each surface (S_j) as determined in the preprocessing (when $\mathcal{M}_s(j)=0$), or update these elements when time varying (if $\mathcal{M}_s(j)=1$).

Then we perform, in the (R_0) reference system, the different tests detailed in section 4 for each couple of surface $\{S_j, S_n\}$ and for each force (i.e. each characteristic vector $\vec{\nu}_k$, $k = 1,\ldots 4$) when $\Sigma(j,n,k) = 1$. The detection of peculiar cases is done first, then the general case algorithm is applied if necessary.

6 EXAMPLE: POD OF TWO MARS ORBITERS

We reprocessed some deep space Doppler and range data from the NASA Mars Odyssey (ODY) and Mars Reconnaissance Orbiter (MRO) spacecraft, in orbit around planet Mars since 2002 and 2006, respectively. Such data had already been used together with those of the Mars Global Surveyor mission in order to determine a gravitational field model (MGGM08A) of Mars in spherical harmonics – its mean (static) part and the time variations of the first zonal coefficients (Marty et al., 2009). At that time we had put the emphasis on the scientific results; we here want to give more details on the POD self-shadowing effects and on the performances of our approach.

6.1 Spacecraft macro model

All geometrical and optical properties and the information about the position of movable parts were obtained from the project teams at the Jet Propulsion Laboratory and at Goddard Space Flight Center.

We restrict the description to the ODY case. The MRO spacecraft model is similar though more complex with two mobile solar panels and a larger antenna. The ODY spacecraft consists of three main parts (fig. 14): the bus, one mobile solar array and the mobile high gain antenna; the Gamma sensor head, at the end of a long boom, is here neglected. The bus consists of six rectangular faces, the solar panel is one rectangle – counted twice (front and rear sides) and the antenna is here sufficiently flat to be approximated by a disc (thus a regular polygon) with two faces also. The areas of these elements are given in table1, the coordinates of the vertices and other reference system quantities are of course introduced in the data base, together with parameters (specular and diffuse coefficients) which characterize the surface physical properties.

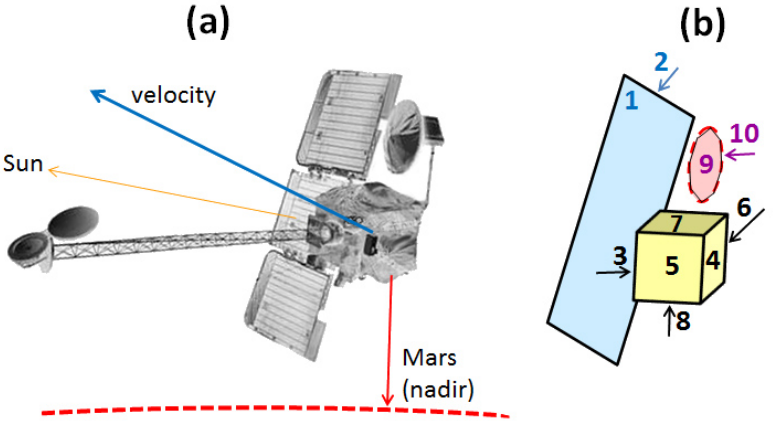


Fig. 14. The Mars Odyssey spacecraft and its modelling. In **(a)** are shown the Gamma sensor head and its boom which are omitted in the box-wing-antenna simplified model shown in **(b)**.

Table 1. Areas of the different components of the modelled Odyssey spacecraft

Solar panel	**Bus**			**Antenna**
Front (1) and rear (2) sides	Faces 3 and 4	Faces 5 and 6	Faces 7 and 8	Faces 9 and 10
11 m^2	**3 m^2**	**2.5 m^2**	**3.6 m^2**	**1.8 m^2**

The spacecraft operates on a sun-synchronous orbit (inclination 93.1°) with a periapsis frozen near the south pole at about 390 km altitude, the eccentricity is very small (~ 0.01). The orientation of the orbiter is such that the solar panel remains almost edge-on along the velocity (cf. fig. 14); as a consequence the self-shadowing effect is quite small for the drag as shown on figure 15 (*top*) where we have plotted the projection of the spacecraft elements on a plane orthogonal to the velocity vector for several positions along an orbit.

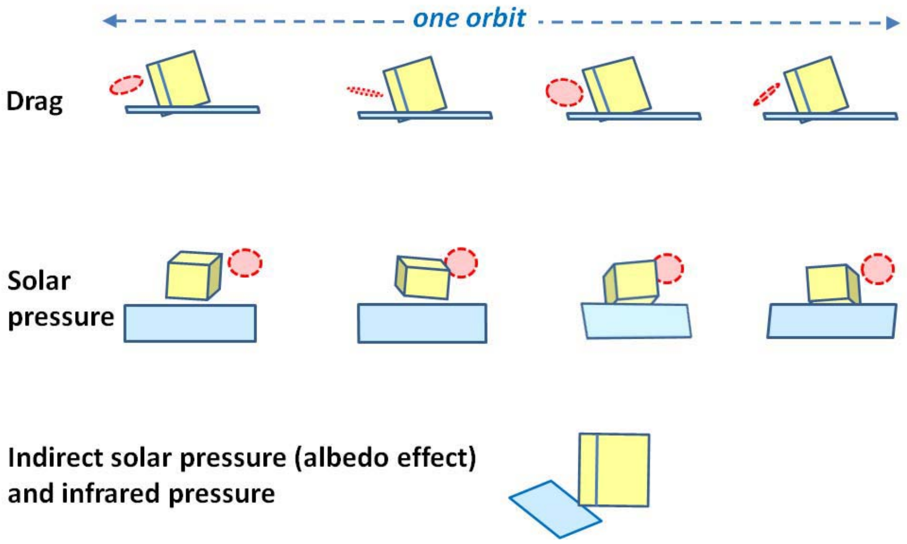

one orbit
Drag
Solar pressure
Indirect solar pressure (albedo effect) and infrared pressure

Fig. 15. Mars Odyssey spacecraft viewed from the velocity vector end (*top*), from the sun (*middle*) and from the nadir (*bottom*) for typical geometries over one orbit. The last case does not exhibit significant variations.

The situation looks different for the self-shadowing associated with the direct solar radiation pressure where the spacecraft parts occult one another significantly - depending of the sun position in a Martian year (fig. 15, *middle*). In the case of the albedo and infrared radiation pressure effects, the geometry is such that the antenna is always shadowed by the bus (fig.15, *bottom*). These have consequences on the surface force coefficients which are adjusted in the course of the POD as will be shown hereafter.

6.2 POD characteristics

The orbit data were obtained from the project teams; they are currently available from the Planetary Data System of NASA. They consist of two and three-way ramped Doppler and also two-way range measurements, all in X band and acquired by the NASA Deep Space Network (DSN) stations at the three sites of Goldstone, Madrid and Canberra. The averaging time for the Doppler data is 10 seconds over the selected period. These observations are processed in arcs of approximately four days duration each.

Corrections to the measurements include: the tropospheric delay which makes use of the meteorological data collected every half-hour at the DSN sites (or coming from 6-h grids of solar radiation pressure, temperature and humidity provided by the European Centre for Medium-range Weather Forecasts, ECMWF), antenna offset (which is peculiar to each spacecraft and may depend on the mission phase), time offsets at the DSN stations. The observables are processed according to Moyer (2000), in taking into account the precise transformation between coordinate time (the time scale of the dynamics) and atomic time (at the stations), and the relativistic effects.

The POD and the retrieval of the physical parameters (such as the gravity field spherical harmonic coefficients – not discussed here) are based on a full dynamical approach using the numerical integration of the equations of motion (and associated variational equations) combined with a least squares adjustment of the unknowns which enter in the linearized observation equations.

The orbit dynamic modelling characteristics are summarized in table 2. In addition to these, one must model the angular momentum desaturation (AMD) events to which the spacecraft is subject from time to time for its attitude control (according to information provided by the project teams).

We solve for the following parameters:

- the state vector components at epoch (beginning) for each arc;
- empirical multiplying factors, F_D , for the drag acceleration (one per 24 hours) accounting for the Drag Temperature Model (DTM) imperfections;
- empirical multiplying factors, $F_{S(d,\,a,\,IR)}$, for the radiation pressures (one per arc): direct, indirect (due to Mars albedo) and infrared;
- empirical accelerations at the AMD epochs: a vector with a priori uncertainties of 10^{-5} ms^{-2} per component (based on information about the AMD events strength);
- one bias per DSN station and per arc for all measurement types;

Due to correlations which are often high between some of these parameters - and lack of observability in some instances, a priori variances are input (as additional equations in the least squares procedure) to stabilize the inversion; this is especially critical for the AMD empirical accelerations which may impact the consistency of the solution.

Table 2. POD dynamic parameterization

Ref. Syst./Ephemeris/Forces	**Description**	**Reference**
Fixed reference system (R_0)	J2000 Mars mean equator (MME2000), X-axis: ascending node of MME2000 on Earth mean equator of J2000	Seidelmann et al., 2002
Martian rotation	Precession	Konopliv et al., 2006
	Nutations	Folkner et al., 1997
	Prime meridian: IAU 2000 conventions	Seidelmann et al., 2002
Earth	Earth rotation and polar motion	IERS, 2010
	Solid and ocean tide loading effects	
	Atmosphere loading effect (6-h radiation pressure grids from ECMWF)	Gegout, *personal comm.*, 1996
Relativistic formulation	Parameterized post-Newtonian (PPN)	Moyer, 2000
Mars gravitational field	MGGM08A	Marty et al., 2009
3rd body attraction	Sun, Moon, Planets: DE430	Folkner et al., 2014
	Phobos, Deimos: dedicated ephemeris	Lainey et al., 2007
Solid tide attraction	With Love number $k_2 = 0.12$	Marty et al., 2009
Surface forces (*with or without self-shadowing*)	Atmosphere: Drag Temperature Model (Mars DTM)	Bruinsma and Lemoine, 2002
	Direct solar radiation pressure	Marty et al., 2011
	Reflected solar + infrared radiation pressures	Lemoine, 1992

6.3 Performances

We have processed the same ODY arcs two times: once with the self-shadowing (S-SH) "on" for all surface forces, and once with no S-SH at all. The arcs are one day long, numerical integration is done with a step-size fixed to 20 seconds, which implies 4320 calls per arc and per iteration to the S-SH computing functions in the software.

The value of the q_{max} parameter (introduced in section 4.3.3) which quantifies the pixellation resolution is quite important since it drives a large part of the extra computing time

CT. With complete parameterized force models as in table 2, the CT increase for the S-SH "on" case varies between 16 % and 90 % for values of q_{max} between 13 and 50, see table 3. The extreme cases, $q_{max} = 13$ and $q_{max} = 50$ correspond to a decomposition of each surface elements into a maximum of 169 and 2500 pixels, respectively. Comparing the results from the different tests, neither the observation residuals nor the orbits show significant differences (above the noise level). A more decisive criterion would lie in the drag and solar pressure coefficients: in our example they do not show significant variations (they are below, or much below 1 %). Therefore the smallest value of q_{max} may be chosen, which makes the problem very tractable. Of course such a value must be adapted to a given spacecraft and orbit characteristics by preliminary tests.

Table 3. Effect of the q_{max} value on the computer time (CT), in percent of the total CT.

q_{max}	**relative CT increase**
13	16
25	30
50	90

Note that these tests have been conducted by assuming mean characteristic directions for the indirect solar radiation pressure and thermal emission. This is a rough simplification in particular in the case of the indirect solar pressure. But we considered that this assumption was, for the time being, good enough for our applications in view of other physical model uncertainties. Using a more accurate modeling of albedo and thermal emission would increase the total CT and drive the relative CT increases in table 3 smaller.

6.4 Effects on the accelerations

To analyze the effect of the self-shadowing on the orbit extrapolation we show in figure 16 the acceleration differences (with and without self-shadowing) for drag, direct solar radiation pressure and albedo for ODY and MRO. The infrared radiation pressure effect is very small (about one percent of the albedo) and has been omitted in these tests.

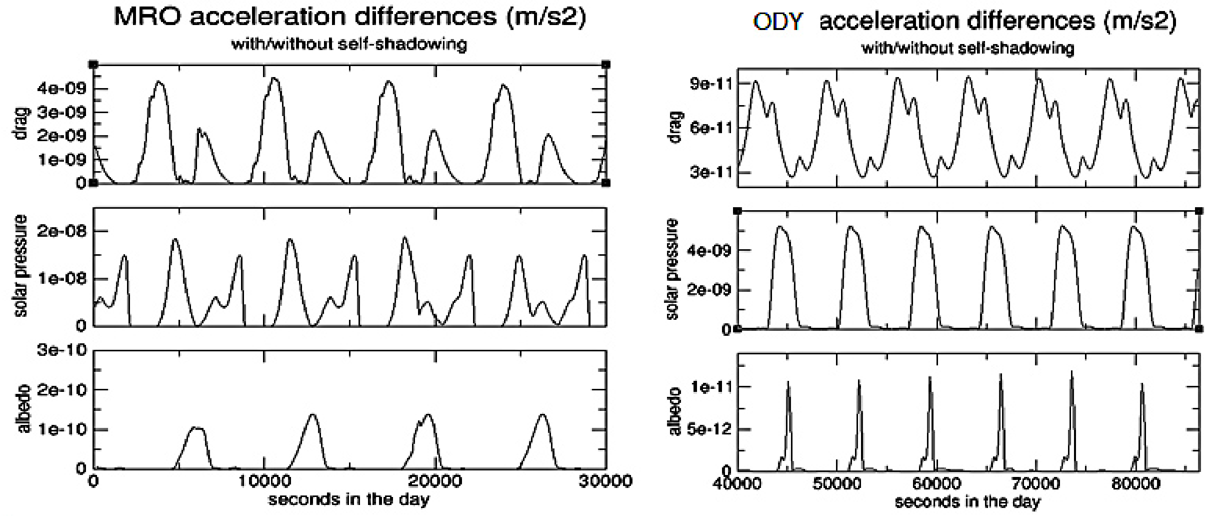


Fig. 16. Accelerations changes ($m.s^{-2}$) due to self-shadowing for the MRO (left) and ODY (right) orbiters, for drag [*top*], direct solar radiation pressure [*middle*] and albedo [*bottom*]. Starting epoch is 2003-08-22 for MRO and 2007-02-05 for ODY.

The patterns are different for MRO and ODY because of the more complex geometry of MRO, especially its two solar panels instead of one for ODY. The amplitude of the self-shadowing effect on drag is larger for MRO than for ODY because its altitude is lower, the drag force being three to four times larger in the present case; it is also larger for the solar radiation pressure due to the two panels and to the spacecraft attitude. The maximum effect of the self-shadowing is given in table 4, in percent of the accelerations themselves.

Table 4. Maximum relative effects (acceleration) of the self-shadowing on the ODY and MRO spacecraft over one day, in percent.

	ODY	**MRO**
Drag	6	12
Solar radiation pressure	9	25
Albedo	4	5

6.5 Effects on the adjusted drag and solar radiation pressure coefficients, and on the orbit

We finally use the self-shadowing model in orbit adjustment, solving for all parameters as listed in section 6.2, that is including drag and solar radiation pressure empirical factors F_D (one per day) and F_S (one per arc). These coefficients should be close to 1 if the drag and solar radiation pressure effects were precisely represented by the force models. Table 5 shows their mean values computed over 25 arcs of MRO and ODY. The use of self-shadowing models makes the F_D and F_S closer to 1 in all cases and especially for solar radiation pressure. For drag the thermosphere model is not precise enough to obtain coefficients much closer to one, though some improvement can be seen.

Table 5. Mean drag and solar radiation pressure coefficients computed for 25 arcs (100 days) of Mars Odyssey (ODY) and Mars Reconnaissance Orbiter (MRO), starting 2002-09-09 and 2007-01-03 respectively.

	ODY		**MRO**	
Self-shadowing	F_D	F_S	F_D	F_S
No	0.74 +/- 0.20	0.94 +/- 0.08	0.79 +/- 0.08	0.77 +/- 0.12
Yes	1.21 +/- 0.15	1.01 +/- 0.08	0.88 +/- 0.08	0.95 +/- 0.11

The Doppler residual changes are not significant (less than 10^{-6} Hz, compared to ~2 mHz residuals r.m.s.). However the orbit differences (3-D) amount to 1.5 m and 10 m typically for ODY and MRO, respectively, and they are mostly in the tangential and normal directions; they originate from self-shadowing-induced variations of the non-gravitational accelerations which cannot be captured by scaling factors. Figure 17 is an example of such differences over one arc for MRO and ODY. It shows that the differences in the radial direction are indeed one to two orders of magnitude smaller than in the other directions. These orbit differences remain at the current POD level for such planetary orbiters. They are larger, though, in the MRO case which is very likely due to its sensitivity to drag: it is much higher than for ODY because of its lower orbit (altitudes of periapsis-apoapsis are 250-320 km for MRO, whereas ODY orbit is almost circular at 400 km altitude).

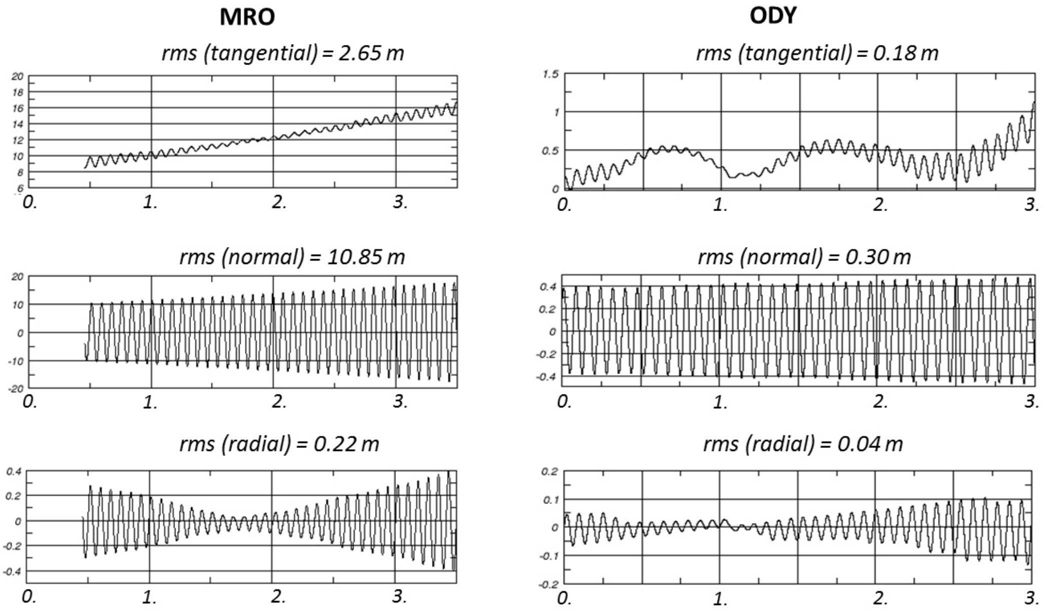


Fig. 17. Orbit differences (r.m.s. values, in meter) between POD with self-shadowing and POD without it, for the MRO (left) and ODY (right) orbiters, over a three day arc. Starting epoch is 2003-07-19 for MRO and 2007-02-05 for ODY.

7 CONCLUSION

We have developed a novel approach to model the self-shadowing effects on the surface forces acting on any spacecraft decomposed into elementary convex planar surfaces, fixed or mobile with respect to each other. The algorithms use elementary geometry and ray-casting techniques. They perform very efficiently in precise orbit determination based on numerical integration. Tests have been done with two Mars orbiters, which demonstrate our ability to use them routinely in planetary geodesy when highest precision is desired.

Acknowledgments

This study was performed at the French Space Agency (CNES) in Toulouse and supported by internal funds. We thank the reviewers for their careful analysis of the paper, thanks to which the final version was improved.

REFERENCES

Balmino, G. (2007), Auto-ombrage & occultation d'un satellite dans le calcul des forces de surface, *Tutorial, C.N.E.S.* (*revised 2009)*, 1-44.

Bruinsma, S., and F. G. Lemoine (2002), A preliminary semi-empirical thermosphere model of Mars: DTM-Mars, *J. Geophys. Res.*, 107(E10), doi:10.1029/2001JE001508.

Clavier, C. (1991), Modelisation des forces de surface sur un satellite artificiel, *Technical report, C.N.E.S.*

Folkner, W.M., R.D. Kahn, R.A. Preston, C.F. Yoder, E.M. Standish, J.G. Williams, C.D. Edwards, R.W. Hellings, T.M. Eubanks, and B.G. Bills (1997), Mars Dynamics from Earth-based trackingof the Mars Pathfinder lander, *J. Geophys. Res*. 102, 4057-4064.

Folkner W.M., Williams J.G., Boggs D.H., Park R.S., and Kuchynka P. (2014),The Planetary and Lunar Ephemerides DE430 and DE431, *IPN Progress Report* 42-196.

Gegout P. (1996), *personal comm.*

IERS Conventions (2010), Gérard Petit and Brian Luzum (eds.). *IERS Technical Note 36*, Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie. 179 pp., ISBN 3-89888-989-6.

Jordan, M.C. (1887), Cours d'analyse, Tome 3, Ecole Polytechnique, *Gauthiers-Villars Ed., Paris*.

Konopliv, A.S. C.F. Yoder, E.M. Standish, D-N. Yuan, and W.L. Sjogren (2006), A global solution for the Mars static and seasonal gravity, Mars orientation, Phobos and Deimos masses, and Mars ephemeris, *Icarus* 182, 23-50.

Konopliv, A.S., S.W. Asmar, W.M. Folkner, O. Karatekin, D.C. Nunes, S.E. Smrekar, C.F. Yoder, M. Zuber (2011), Mars high resolution gravity fields from MRO, Mars seasonal gravity, and other dynamical parameters, *Icarus*, Vol. 211, Issue 1, 401-428.

Lainey, V., V. Dehant, M. Pätzold (2007), First numerical ephemerides of the two Martian moons. *Astronom. Astrophys*. 463 (3), 1075-1084.

Lemoine, F.G. (1992), Mars: The dynamics of orbiting satellites and gravity model development, *Ph.D. thesis*, *Univ. of Colorado*, Boulder, CO.

Marty, J.C., G. Balmino, P. Rosenblatt, J. Duron, S. LeMaistre, A. Rivoldini, V. Dehant, T. Van Hoolst (2009), Martian gravity field model and its time variations from MGS and Odyssey data, *Plan. & Space Sci*., 57, 350-363.

Marty, J.C., S. Loyer, F. Perosanz, F. Mercier, G. Bracher, B. Legresy, L. Portier, H. Capdeville, F. Fund, J.M. Lemoine, R. Biancale (2011), GINS : the CNES/GRGS GNSS scientific software, 3rd International Colloquium Scientific and Fundamental Aspects of the Galileo Programme, 31 Aug.-2 Sept. 2011, Copenhagen, Denmark. *ESA Proceedings WPP326*.

Mazarico, E.M. (2008), Study of the Martian upper atmosphere using radio tracking data, *PhD thesis, MIT*, pp. 1-268.

Mazarico, E.M., M.T. Zuber, F.G. Lemoine, D.E. Smith (2009), Effects of Self-Shadowing on Nonconservative Force Modeling for Mars Orbiting Spacecraft. *J. Spacecraft Rockets,* Vol. 46, No. 3, pp. 662-669.

Moyer, T. D. (2000), Formulation for observed and computed values of Deep Space Network data types for navigation, *Monograph 2, Deep Space Communications and Navigation series.*

Salomon, K.B. (1978), An efficient point-in-polygon algorithm*,* *Computers & Geosciences*, Vol. 4, no. 2, pp. 173–175.

Seidelmann, P. K., V. K. Abalakin, M. Bursa, M. E. Davies, C. De Bergh, J. H. Lieske, J. Oberst, J. L. Simon, E. M. Standish, P. Stooke, and P. C. Thomas (2002), Report of the IAU/IAG Working Group on cartographic coordinates and rotational elements of the planets and satellites: 2000, *Celest. Mech. Dyn. Astron.*, 82, pp. 83-111.

Ziebart, M. (2001), High Precision Analytical Solar Radiation Pressure Modelling for GNSS Spacecraft. *PhD thesis*, *University of East London*.

Ziebart, M., S. Adhya, A. Sibthorpe, S. Edwards, and P. Cross (2005). Combined radiation pressure and thermal modelling of complex satellites: Algorithms and on-orbit tests, *Advances in Space Research*, 36(3):424-430.

APPENDIX A

Polyhedral decomposition of spacecraft elements

This appendix provides the formulas needed in the most frequent cases of decomposition of a spacecraft into flat elements, especially when approximating curved surfaces (cylinder, cone, paraboloid) by plates.

A1. Bus represented as an irregular polyhedron with lateral four-sided faces

The bus has two polygonal faces with N edges, and N lateral faces each with 4 edges. This is exemplified on figure A1.

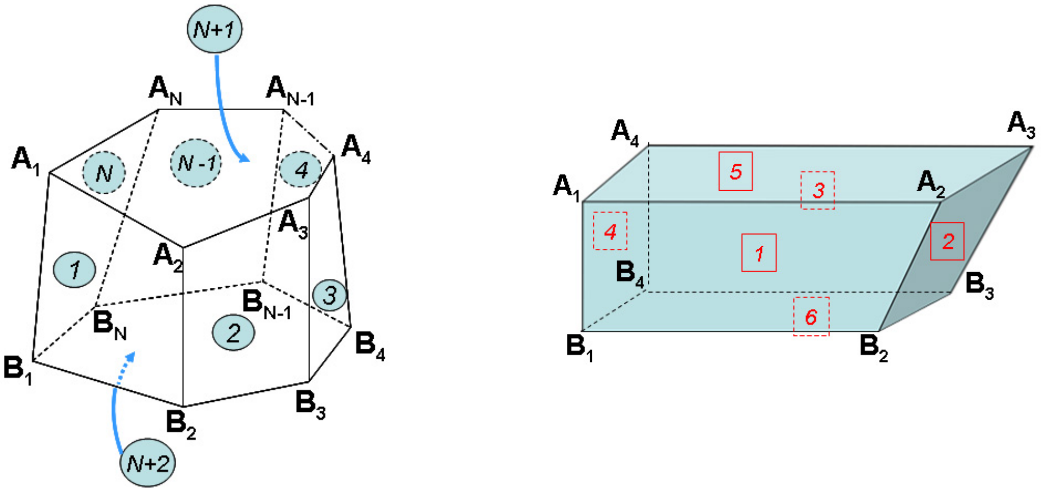


Fig. A1. Spacecraft bus with N lateral four-sided faces and two polygonal faces. *Left* : general case; *right* : simple (usual) case: N= 4 (the quadrilateral faces need not be rectangular nor trapezoidal as suggested).

A_i and B_i being associated as shown, the k.th face (k=1 to N+2) has the following vertices:

. $1 \leq k \leq N$: $[\ A_k\ ,\ B_k\ ,\ B_{\mathrm{mod}(k,N)+1},\ A_{\mathrm{mod}(k,N)+1}\ ]$
. $k = N+1$: $[\ A_1\ ,\ A_2\ ,\ \dots\ ,\ A_N\ ]$
. $k = N+2$: $[\ B_N\ ,\ B_{N-1}\ ,\ \dots\ ,\ B_1\ ]$.

This ordering ensures that a normal vector computed from any three points taken cyclically from one of the above sequences is indeed the outward normal to the associated face.

A2. Approximation of cylinders and cones

These surfaces may be viewed as peculiar cases of the above, where N is chosen according to the desired degree of approximation, with the two polygonal faces being regular polygons: (i) equal in the case of the cylinder, and (ii) one being reduced to a point in the conical case (fig. A2)

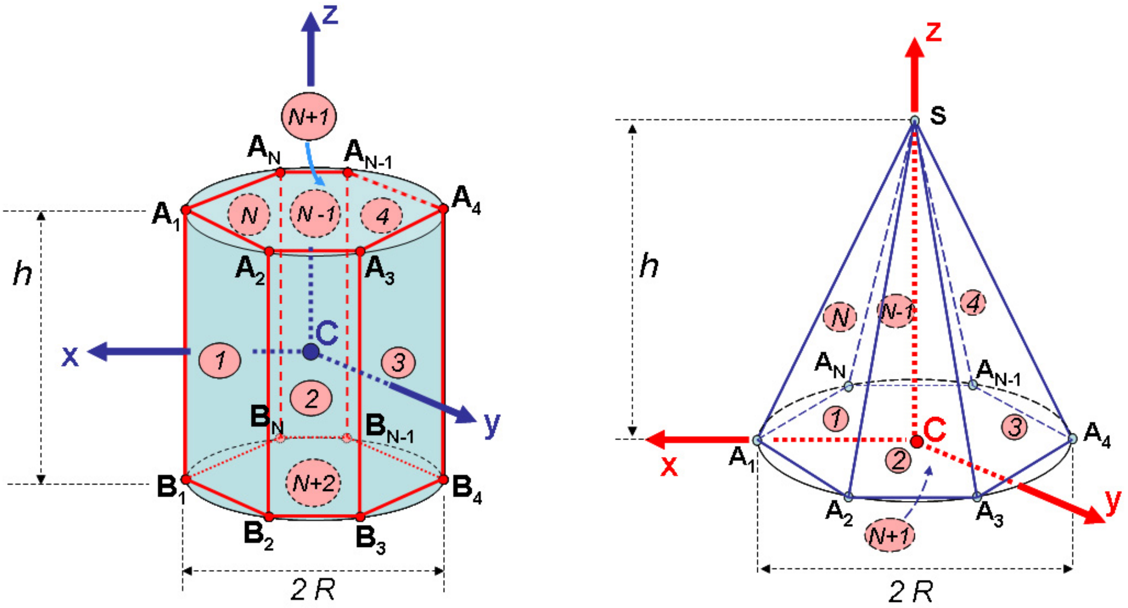


Fig. A2. Cylinder and cone in a macro-model, approximated by flat plates.

We consider the usual case of a circular cylinder or cone. In a coordinate system defined by the cylinder axis and its geometrical centre, or by the cone axis and the centre of the base) we have:

- for the cylinder:

  . A_k : $\{ x_k = R \cos [2(k-1)\pi / N] \,;\ y_k = R \sin [2(k-1)\pi / N] \,;\ z_k = h / 2 \}$
  . B_k : $\{ x_k = R \cos [2(k-1)\pi / N] \,;\ y_k = R \sin [2(k-1)\pi / N] \,;\ z_k = - h / 2 \}$
  . Area of the polygonal upper and lower faces = $N/2\, R^2 \sin 2\pi/N$
  . Area of each lateral plate = $2\, R\, h \sin \pi/N$

- for the cone:

  . S : $\{ x_S = 0 \,;\ y_S = 0 \,;\ z_S = h \}$
  . A_k : $\{ x_k = R \cos [2(k-1)\pi / N] \,;\ y_k = R \sin [2(k-1)\pi / N] \,;\ z_k = 0 \}$
  . Area of the polygonal lower face (base) = $N/2\, R^2 \sin 2\pi/N$
  . Area of each lateral plate = $R\, [\, h^2 + R^2 \cos^2 \pi/N]^{1/2} \sin \pi/N$

A3. Approximation of a parabolic antenna

High gain antennas on planetary probes and orbiters usually have a significant size and must be modelled accurately for computing the surface forces. By nature they are paraboloids of revolution with the emitting/receiving electronics located at the focus of the surface (phase centre). They may be opened to free space, or closed by a conical surface centred on the focus (fig. A3).

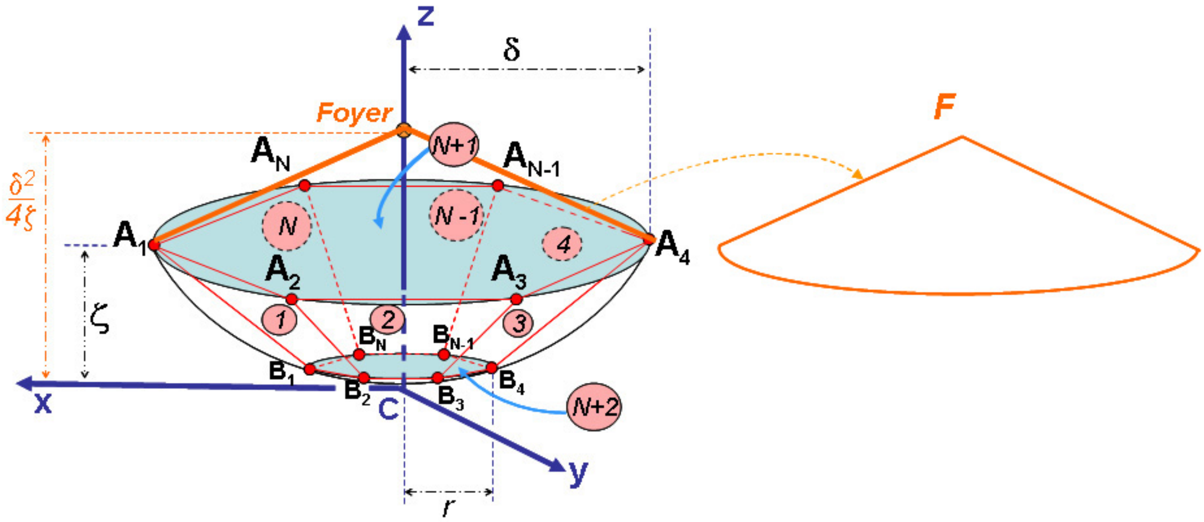


Fig. A3. Parabolic antenna surface approximated by flat plates. One has $0 < r \leq \delta/2$. A conical cover (shown on the right) may be added.

The paraboloid may be approximated by a bottom polygonal plate (at local coordinate z_0 which is function of radius r) with N edges, and N lateral trapezoidal plates, supplemented – when there is a cone covering the antenna, by:

- either an upper plate (numbered N+1) approximating the cone if it is very flat (i.e. when $\delta > 2\zeta$) : this is the polygon $A_1, A_2, \ldots A_N$,
- or N plates approximating it when $\delta \leq 2\zeta$; these are the triangles $\{ F\ A_k\ A_{\mathrm{mod}(k,N)+1} \}$ ($k = 1,2,\ldots N$).

The coordinates of the involved vertices are:

. A_k : $\{ x_k = \delta \cos [2(k\text{-}1)\pi / N]\ ;\ y_k = \delta \sin [2(k\text{-}1)\pi / N]\ ;\ z_k = \zeta \}$
. B_k : $\{ x_k = r \cos [2(k\text{-}1)\pi / N]\ ;\ y_k = r \sin [2(k\text{-}1)\pi / N]\ ;\ z_k = z_0 = r^2\zeta / \delta^2 \}$
. Area of the polygonal lower face = $N/2\ r^2 \sin 2\pi/N$
. Area of each lateral plate = $h\,(\delta + r) \sin \pi/N$, h being the height of trapeziums like $A_1A_2B_2B_1$: $h^2 = (\delta\text{-}r)^2 \cos^2\pi/N + \zeta^2\,(1\text{-}r^2 / \delta^2)^2$

In some cases the whole antenna may be viewed as a flat disc, approximated by a regular polygon.

A4. Area of a convex polygon of N vertices

The area A_N can be obtained by summing up the areas of N-2 triangles of common vertex A_1 (for instance), that is (see Fig. A4):

$$A_N = \sum_{k=3}^{N} \left[p_k (p_k - a_k)(p_k - b_k)(p_k - c_k) \right]^{1/2} \quad ,$$

with: $a_k = A_1A_{k\text{-}1}$, $b_k = A_{k\text{-}1}A_k$, $c_k = A_kA_1$

and: $p_k = (a_k + b_k + c_k)/2$.

Alternatively if (u_i, v_i) are the rectangular coordinates of A_i in any reference system in the polygon plane, then we can apply Gauss (or shoelace) formula:

$$A_N = \tfrac{1}{2}\left|\sum_{i=1}^{N}(u_i v_j - u_j v_i)\right|,$$

with $j = \mathrm{mod}(i,N) + 1$.

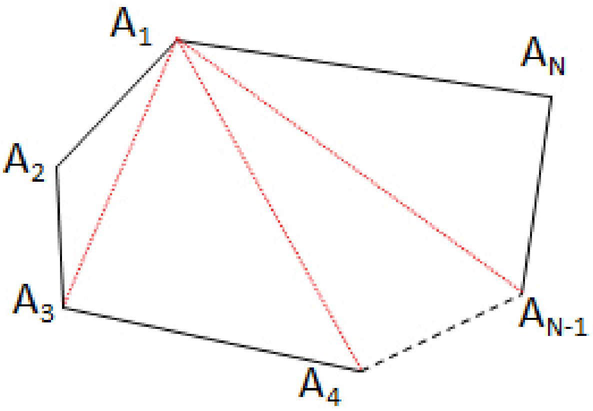


Fig. A4. Area of a convex polygon by triangular decomposition

APPENDIX B

Point-in-polygon (PIP) algorithms

The PIP problem is a particular case of determining whether a point P_0 is inside or outside a compact Ω of $\mathbf{R}^2$. There are basically two classes of methods which solve this problem:

(i) those belonging to the winding number concept, which may be derived from Cauchy integral formula, and

(ii) those belonging to ray casting techniques.

B1. Cauchy type algorithms

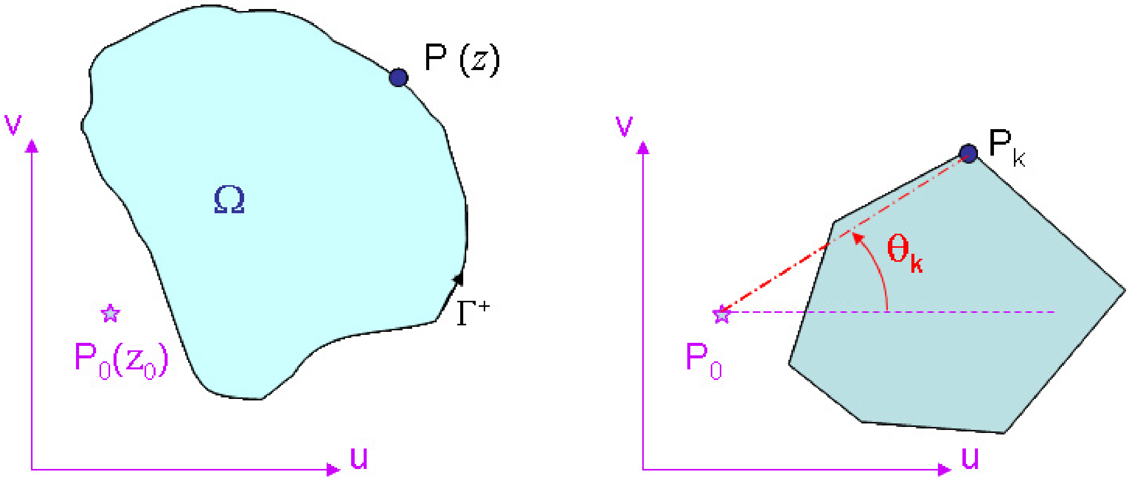


Fig. B1. Geometry for the algorithms derived from Cauchy integral. *Left* : for any curve limiting a compact Ω ; *right* : for a polygon.

Let Γ^+ be the boundary ($\partial\Omega$) of Ω, oriented clockwise (fig. B1).

In the complex plane z = u + *i* v, Cauchy's integral formula is:

$$w(\mathrm{P}_0) = \frac{1}{2i\pi}\int_{\Gamma^+}\frac{dz}{z-z_0} = \begin{cases} 0 & \text{if } \mathrm{P}_0 \text{ is outside } \Omega \\ \tfrac{1}{2} & \text{if } \mathrm{P}_0 \text{ is on } \partial\Omega \\ 1 & \text{if } \mathrm{P}_0 \text{ is inside } \Omega \end{cases}$$

$w(\mathrm{P}_0)$ is called the *winding number* of Γ^+ around P_0. For a general (closed) curve it is an integer (or an integer plus ½) representing the total number of times that curve travels counter-clockwise around the point; the winding number depends on the orientation of the curve, and is negative if the curve travels around the point clockwise. Here it is a peculiar case since Γ^+ is a simple curve (it makes only one loop).

If $\partial\Omega$ is the polygon $\mathrm{P}_1\mathrm{P}_2\ldots\mathrm{P}_N$, then:

$$w(P_0) = \frac{1}{2\pi}\sum_{k=1}^{N}\Delta\theta_k \quad = \{0 \quad \pm\tfrac{1}{2} \quad \pm 1\} \quad \text{according to the same cases,}$$

(the sign depends on the orientation of the polygonal line) where we have:

$$\begin{aligned} \Delta\theta_k = \theta_{k+1} - \theta_k &= Arc\sin q_k^{k+1} \qquad \text{if the scalar product } \vec{U}_k.\vec{U}_{k+1} \geq 0 \\ &= \pi - Arc\sin q_k^{k+1} \quad \text{if } \vec{U}_k.\vec{U}_{k+1} < 0 \end{aligned}$$

where

$\vec{U}_k = \overrightarrow{P_0P_k} / P_0P_k$ and $q_k^{k+1} = \overline{\vec{U}_k \wedge \vec{U}_{k+1}}$ (algebraic value of the cross product, i.e. the value of the determinant of the two vectors).

In the above we assume that $\mathrm{P}_{N+1} \equiv \mathrm{P}_1$.

This is a costly algorithm when directly applied like above. It can be simplified by tracking through which quadrants the polygon winds, as it turns around the test point, which makes the winding number algorithm comparable in speed to methods of class (ii) – see below. It can be further simplified if the polygon is convex (which is always the case in the present study); it is easy to see that:

- if P_0 is outside the polygon : $(\exists\ \mathrm{k})$ *sign* $\mathrm{D}[\ \mathrm{P}_0, \mathrm{P}_\mathrm{k}, \mathrm{P}_{\mathrm{mod(k,N)+1}}\ ] = -$ *sign* $\mathrm{D}[\ \mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3\ ]$,
- if P_0 is inside the polygon : $(\forall\ \mathrm{k})$ *sign* $\mathrm{D}[\ \mathrm{P}_0, \mathrm{P}_\mathrm{k}, \mathrm{P}_{\mathrm{mod(k,N)+1}}\ ] =$ *sign* $\mathrm{D}[\ \mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3\ ]$,

where for any three points A, B, C : $\mathrm{D}[A,B,C] = \overline{\overrightarrow{AB} \wedge \overrightarrow{AC}}$ is the value of a 3x3 determinant.

This the *vector form of the winding number* algorithm.

B2. Ray casting algorithms

Ray casting is a peculiar, and faster version of the ray tracing algorithm. The basis is a theorem by Jordan (1887) which applies to a compact Ω of $\mathbf{R}^2$ (see fig. B2). If Δ is any oriented straight line passing through P_0 with an arbitrary origin and if the intersecting points $\mathrm{A}_1, \mathrm{A}_2, \ldots \mathrm{A}_m$ with the boundary Γ of Ω are ordered according to increasing abscissas, then:

. m is even,
. any point outside $[\mathrm{A}_1\ \mathrm{A}_m]$ is outside Ω,
. for a point A in $[\mathrm{A}_1\ \mathrm{A}_m]$ there exists $\mathrm{p} \geq 1$ such that: either $\mathrm{A} \in [\mathrm{A}_{2\mathrm{p}-1}\ \mathrm{A}_{2\mathrm{p}}]$ and A is inside Ω, or $\mathrm{A} \in [\mathrm{A}_{2\mathrm{p}}\ \mathrm{A}_{2\mathrm{p}+1}]$ and A is outside Ω.

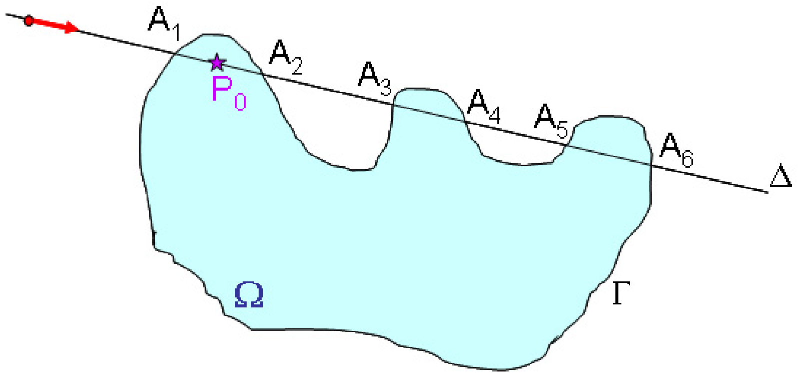
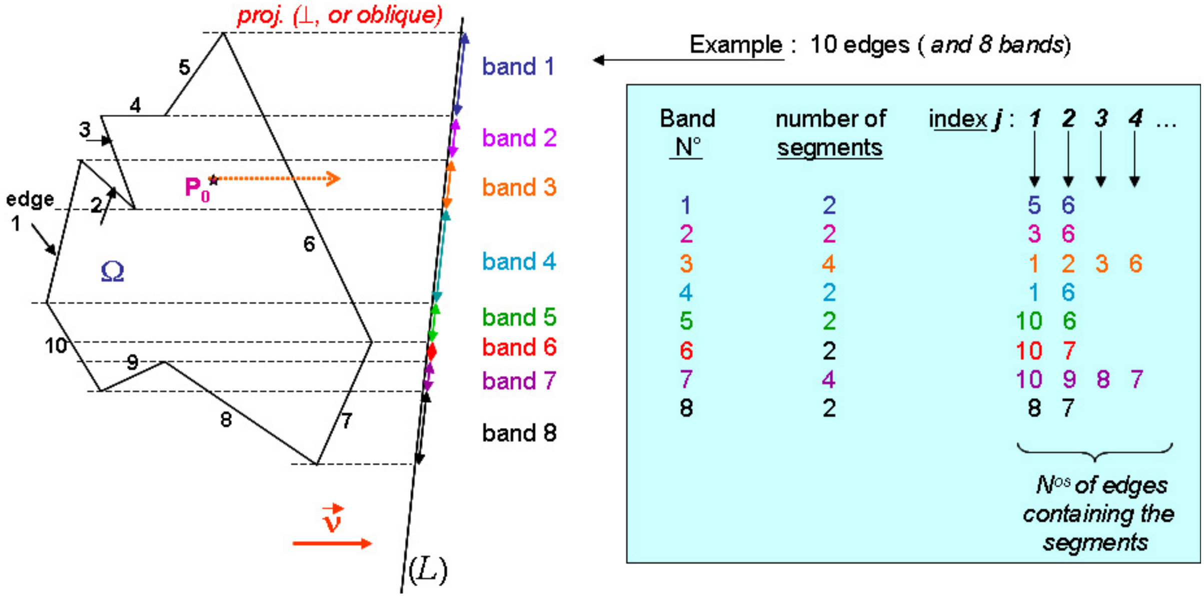


Fig. B2. Geometry of the 1887 Jordan theorem.

From this we derive the *crossing number algorithm*. For a simple polygon we count the number *n* of boundary crossings by a ray starting from the point P_0 and going in any direction. If P_0 is not on the boundary, *n* is *even* if the point is outside, and it is *odd* if P_0 is inside. This is also known as the *even-odd rule algorithm*, which is extremely simplified when Ω is convex since *m* and *n* are equal to 2 at most. In practice we take a ray parallel to an axis of coordinates and directed towards the positive values for instance.

A variant of it (for polygons), sometimes called the *dual crossing number algorithm* and introduced by Salomon (1978) makes use of the projections of the polygon edges on a straight line (*L*) which is entirely outside the polygon. The edges which are parallel to the direction of projection ($\vec{v}$) are ignored; the bands are numbered as shown in figure B3 (there are less bands than edges), and for each band the segments (pieces of edges) are ordered according to the projection direction (from left to right on fig. B3). Then we take the ray starting from the point P_0, parallel to $\vec{v}$ - going towards (*L*); if *j* is the index of the first intersected segment then: { *j* even } ⇔ { P_0 inside Ω }. This algorithm requires a pre-processing phase of the polygon but it is then very fast.



Band N°	number of segments	index *j* : *1*	*2*	*3*	*4* …
1	2	5	6		
2	2	3	6		
3	4	1	2	3	6
4	2	1	6		
5	2	10	6		
6	2	10	7		
7	4	10	9	8	7
8	2	8	7		

Fig. B3. Example showing how the Salomon (1978) algorithm works.

B3. Which one to choose ?

Our criteria has been the computing speed. Algorithms have been programmed in Fortran 90. Tests have been conducted (on PC) with various types of polygons, convex or not, and by varying the number of edges/vertices (N) and number of tested points (n_0). For convex polygons the winding number algorithm (in its vector form) and the Jordan algorithm are equivalent. Independently from the polygon type, we found that the efficiency of the Jordan algorithm vs. Salomon's method depends on the ratio $\rho = n_0 / N$; for moderate ρ (~1 to 50) our codes perform equally; for large ρ (~1000 or larger) Jordan was found to be faster; in between, relative performances depend on the polygon complexity. Considering that macro-model elements have a small number of edges and that we always work with (simple) convex polygons, we decided to use the Jordan algorithm in the POD software. However the Salomon's algorithm was chosen for off-line studies of more complex surfaces.

Received: 2017-03-09,

Reviewed: 2017-04-27, by D. Arnold, 2017-12-01, and 2018-01-03, by P.W. Kenneally,

Accepted: 2018-03-08.