International Journal of Heat and Mass Transfer 55 (2012) 7330–7335

Contents lists available at SciVerse ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

General procedure for calculation of diffuse view factors between arbitrary planar polygons

Sandip Mazumder *, Mahesh Ravishankar

Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH 43210, USA

ARTICLE INFO

Article history:
Received 27 May 2012
Received in revised form 19 July 2012
Accepted 24 July 2012

Keywords:
View factor
Form factor
Contour integration
Arbitrary polygon
Radiation transport
Light transport

ABSTRACT

Diffuse view factors (or form factors) between two planar polygonal surfaces placed at arbitrary orientation relative to each other serve as inputs for surface-to-surface radiation transport or other similar calculations. In this article, a general formulation and accompanying numerical procedures are presented to perform such calculations. The formulation uses a parametric vector representation of the two surfaces along with the well-known contour integral method to develop an integral formula that finally requires a numerical quadrature scheme to evaluate. The use of the parametric vector representation of the surfaces overcomes the difficulty of defining limits to the integrals in the contour integral method. Six test cases are considered, and the results are compared either to exact analytical solutions (when available) or to Monte Carlo results, which are also generated as part of this study. It is found that irrespective of the case studied, the present method matches analytical results perfectly (up to six decimal places accuracy) with a 10-point or higher Gaussian quadrature scheme. The errors in the Monte Carlo results increase when the two surfaces are either far from each other and/or at grazing angles, although the actual errors are quite small (less than 0.5%). For the cases when analytical solutions are not available, the Monte Carlo results and those generated by the present method were found to be in close agreement.

1. Introduction

Diffuse view factors, also known as form factors or shape factors, are required as inputs for the treatment of surface-to-surface exchange formulations [1,2] for radiation transport. In the radiation literature, they are sometimes also referred to as radiation configuration factors [1,2].

Most textbooks on radiation present a table of formulae to calculate diffuse view factors for a large number of scenarios for which closed-form analytical expressions can be derived. A comprehensive catalog of radiation exchange factors for which closed-form analytical solution is derivable has been assembled by Howell [3], and is also available online. While the number of scenarios covered by such catalogs is large, practical engineering calculations require much more. In fact, the shapes encountered in practical engineering applications such as combustors, rapid thermal processing chambers, or nuclear reactors are far too complex, and do not necessarily fit any of the scenarios available through catalogs. For such cases, one has to either make approximations to the actual geometry, or revert back to the basic definition of the view factor and compute the double area integrals numerically.

Numerical computation of view factors can be performed either using deterministic methods or Monte Carlo methods. The Monte Carlo method is the most general and powerful method—especially since it can also handle obstructions, and has been employed extensively for this purpose [1,2,4–10]. Unfortunately, they are computationally expensive. Furthermore, solutions generated by Monte Carlo calculations inherently contain statistical noise, especially when the solid angle subtended by one surface on the other is small, which results in small view factor values. This usually occurs when the two surfaces are placed far from each other or at grazing angles of incidence, as will also be demonstrated in this article. Some methods have been suggested to smooth fluctuations due to statistical errors [11]. Nonetheless, because of their computational cost, the Monte Carlo method is prohibitive for large scale three-dimensional calculations that are demanded by practical applications.

One of the most general deterministic methods to compute view factors numerically from first principles in three-dimensional geometries is the so-called contour integral method [1,2], first proposed by Sparrow [12]. In this method, the double surface area integrals in the definition of the view factor are converted to double line integrals through use of the Stokes' theorem. This method has found prolific use in the computation of view factors between

* Corresponding author. Address: Department of Mechanical and Aerospace Engineering, The Ohio State University, Suite E410, Scott Laboratory, 201 West 19th Avenue, Columbus, OH 43210, USA. Tel.: +1 (614) 247 8099; fax: +1 (614) 292 3163.

E-mail address: mazumder.2@osu.edu (S. Mazumder).

0017-9310/$ - see front matter
http://dx.doi.org/10.1016/j.ijheatmasstransfer.2012.07.066

Nomenclature

A_Q	area of surface Q (m^2)	$\mathbf{s}_P$	vector along contour of surface P (m)
$\hat{\mathbf{n}}$	unit surface normal vector	$\mathbf{s}_Q$	vector along contour of surface Q (m)
N, M	number of vertices on surfaces P and Q, respectively	*Greek* λ_P, λ_Q	parameters representing surfaces P and Q (dimensionless)
$\mathbf{p}_{1,2}$	vector joining vertices 1 and 2 on surface P (m)	Γ_P, Γ_Q	length of contours bounding surfaces P and Q (m)
$\mathbf{q}_{1,2}$	vector joining vertices 1 and 2 on surface Q (m)		
$\mathbf{S}$	vector between differential elements [see Fig. 1] (m)		

surfaces in three-dimensional geometries, such as in [13–16] and the references cited therein. In all cases, however, the method has been employed for regular shaped surfaces, such as rectangles, cylinders, spheres, or cones. This is primarily because for regular shapes, the limits of integration in the final integral formula can be clearly defined using Cartesian, polar or spherical coordinate systems. In the case of surfaces of arbitrary shape and orientation, the most common approach has been to first tessellate the surface into smaller triangles [17,18]. This is followed by application of a coordinate transformation to numerically compute the double area integral for a pair of triangles using a method similar to the finite element method, and subsequent summation of the resulting view factors to obtain the view factor of the total surface. This procedure is not only tedious, but can result in large errors, as discussed by Ravishankar *et al.* [19] if the tessellated triangles are not small enough.

Many of the above methodologies discussed in the preceding paragraphs have been employed to compute view factors between surfaces in available codes. These codes include VISRAD [20], View3D [21], FACET [22], and MONT3D [23]. VISRAD uses a coordinate transformation to compute the double line integrals. View3D only supports triangular and quadrilateral surface elements, and employs both double area integrals and double line integrals as alternative methods [24]. Open literature on FACET is not available, and the detailed algorithms therein are not known except for the fact that it employs the double area integral method. MONT3D computes radiation exchange factors using the Monte Carlo method. A comparative study of some of the methods for computing view factors between surface elements of regular shape may be found in Emery et al. [25].

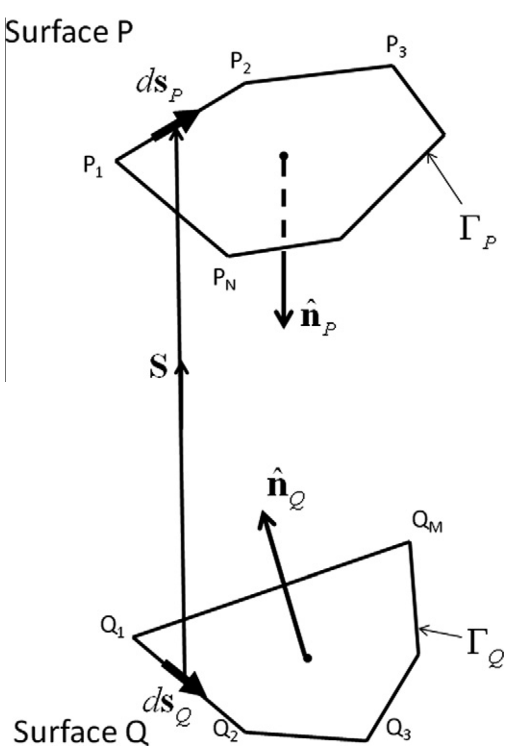


Fig. 1. Schematic representation of two polygonal surface elements showing relevant vectors used in the formulation presented in this article.

With increasing use of unstructured mesh topologies for scientific computations such as in computational fluid dynamics codes, it would be worthwhile to develop a formulation that would enable computation of view factors between arbitrary planar polygonal surfaces (could be triangular, quadrangular or other polygonal shape) placed at arbitrary orientation relative to each other. In this article, such a formulation is developed and demonstrated. The present work draws upon two previous works, at least from a motivational standpoint. The first is a conference paper by Hollands [26] that explains that the challenge in using the contour integral method is not so much in performing the integral itself as in defining the limits of the definite integral for arbitrary shapes. Hollands' paper proposes a parametric representation of surfaces to mitigate this challenge, although only regular geometries are eventually demonstrated. In a similar work, Daun and Hollands [27] used non-uniform rational B-splines to represent the surfaces in surface-to-surface radiation analysis in an enclosure. The second paper is another old conference paper [28] that outlines the mathematical procedure to use parametric representation of surfaces for arbitrary polygonal shapes, and presents a semi-analytical solution to the problem. However, the solution itself involves computations of complex functions such as the dilogarithm function, and ultimately requires a computer program to evaluate since three numerical integrations are necessary. As in other studies, the method has not been demonstrated for irregular geometries. The present work builds upon the overall idea of parametric surface representation to develop a procedure for calculation of view factors between arbitrary planar polygons. The procedure is validated and demonstrated for a number of different geometries, both regular and irregular. Comparison with the Monte Carlo method is also presented.

2. Mathematical formulation

The subject of this particular paper is calculation of the view factor between two planar (or flat) surfaces oriented arbitrarily relative to each other. This particular type of surface is of particular interest because the surface elements generated by any unstructured mesh generator are planar or flat, *i.e.*, each surface element has a unique surface normal. Furthermore, such surface elements (or faces) are convex polygons and are bounded by a discrete number of straight edges, as shown in Fig. 1. The number of edges bounding the face can be arbitrary, which essentially encompasses any unstructured mesh type. Accordingly, surface P is shown to have *N* vertices, while surface Q is shown to have *M* vertices. Under this general description, using the contour integral method, the view factor between surfaces Q and P may be written as [1]

$$F_{Q-P} = \frac{1}{2\pi A}\oint_{\Gamma_P}\oint_{\Gamma_Q} \ln S \quad d\mathbf{s}_Q \cdot d\mathbf{s}_P \tag{1}$$

where A_Q is the area of surface Q. The contours (edges) bounding surfaces P and Q are represented by Γ_p and Γ_Q respectively (Fig. 1), while $d\mathbf{s}_P$ and $d\mathbf{s}_Q$ are differential length vectors along Γ_P and Γ_Q, respectively. S is the distance between the two differential line elements, *i.e.*, $S = |\mathbf{S}|$, as shown in Fig. 1. The vector $\mathbf{S}$ may be written as

$$\mathbf{S} = -\mathbf{s}_Q + \overrightarrow{\mathrm{Q}_1\mathrm{P}_1} + \mathbf{s}_P = -\lambda_Q\,\overrightarrow{\mathrm{Q}_1\mathrm{Q}_2} + \overrightarrow{\mathrm{Q}_1\mathrm{P}_1} + \lambda_P\,\overrightarrow{\mathrm{P}_1\mathrm{P}_2} \tag{2}$$

where $\mathbf{s}_Q$ is the vector pointing from vertex Q_1 to the tail of the differential vector $d\mathbf{s}_Q$ and likewise, $\mathbf{s}_P$ is the vector pointing from vertex P_1 to the tail of the differential vector $d\mathbf{s}_P$. $\overrightarrow{\mathrm{Q}_1\mathrm{Q}_2}$ represents the vector joining Q_1 to Q_2, and may be written as $\mathbf{q}_{1,2}$. Similarly, the vector joining P_1 to P_2 may be written as $\mathbf{p}_{1,2}$. The quantity λ_Q represents the fraction of the vector $\overrightarrow{\mathrm{Q}_1\mathrm{Q}_2}$ that is equal to vector $\mathbf{s}_Q$, and likewise, λ_P represents the fraction of the vector $\overrightarrow{\mathrm{P}_1\mathrm{P}_2}$ that is equal to vector $\mathbf{s}_P$. Accordingly, $0 \le \lambda_Q \le 1$and $0 \le \lambda_P \le 1$ for the tail and tip of the vector $\mathbf{S}$ to scan from vertex 1 to vertex 2 for both surface elements. It also follows that

$$d\mathbf{s}_Q = d\lambda_Q\,\overrightarrow{\mathrm{Q}_1\mathrm{Q}_2} = d\lambda_Q\mathbf{q}_{1,2} \tag{3a}$$

$$d\mathbf{s}_P = d\lambda_P\,\overrightarrow{\mathrm{P}_1\mathrm{P}_2} = d\lambda_P\mathbf{p}_{1,2} \tag{3b}$$

From Eq. (2), it also follows that

$$\begin{aligned} S^2 = \mathbf{S}\cdot\mathbf{S} &= (-\lambda_Q\mathbf{q}_{1,2} + \overrightarrow{Q_1P_1} + \lambda_P\mathbf{P}_{1,2})\cdot(-\lambda_Q\mathbf{q}_{1,2} + \overrightarrow{Q_1P_1} + \lambda_P\mathbf{P}_{1,2}) \\ &= \lambda_Q^2|\mathbf{q}_{1,2}|^2 + \lambda_P^2|\mathbf{P}_{1,2}|^2 - 2\lambda_Q\,\overrightarrow{Q_1P_1}\cdot\mathbf{q}_{1,2} - 2\lambda_P\,\overrightarrow{Q_1P_1}\cdot\mathbf{P}_{1,2} \\ &\quad - 2\lambda_P\lambda_Q\mathbf{P}_{1,2}\cdot\mathbf{q}_{1,2} + |\overrightarrow{Q_1P_1}|^2 \end{aligned} \tag{4}$$

Since the contours Γ_P and Γ_Q are comprised of discrete straight edges in this particular scenario, the contours integrals in Eq. (1) may be replaced by discrete summations to yield

$$F_{Q-P} = \frac{1}{2\pi A}\sum_{n=1}^{N}\sum_{m=1}^{M}\oint_{\Gamma_P}\oint_{\Gamma_Q} \ln S_{m,n} \quad d\mathbf{s}_{Q,m}\cdot d\mathbf{s}_{P,n} \tag{5}$$

Generalization of Eqs. (3) and (5) to any pair of vertices on line segments n (on Γ_P) and m (on Γ_Q), followed by substitution into Eq. (5), yields

$$\begin{aligned} F_{Q,P} = \frac{1}{4\pi A_Q}\sum_{n=1}^{N}\sum_{m=1}^{M}\int_0^1\int_0^1 &\ln(\lambda_Q^2|\mathbf{q}_{m,m+1}|^2 + \lambda_P^2|\mathbf{p}_{n,n+1}|^2 \\ &- 2\lambda_Q\overrightarrow{Q_mP_n}\cdot\mathbf{q}_{m,m+1} - 2\lambda_p\overrightarrow{Q_mP_n}\cdot\mathbf{p}_{n,n+1} - 2\lambda_P\lambda_Q\mathbf{p}_{n,n+1}\cdot\mathbf{q}_{m,m+1} \\ &+ |\overrightarrow{Q_mP_n}|^2)\mathbf{p}_{n,n+1}\cdot\mathbf{q}_{m,m+1}\,d\lambda_Q\,d\lambda_P \end{aligned} \tag{6}$$

Since the coordinates of all vertices of both surfaces P and Q must be known to define the surface, all of the vectors shown in the integration kernel in Eq. (6) can be easily computed, and represent constants that change only with change in the outer summation indices. Thus, the integration kernel is actually a non-linear function of λ_Q and λ_P, and the resulting double integral can only be computed using numerical integration techniques. It is also possible to integrate one of the integrals in Eq. (6) analytically, which would result in inverse tangent functions [29]. This result may be substituted into the second integral, and the resulting integration can then be performed numerically. One final point to note is that when the last index of the summation is reached (either N or M), the subsequent index (or vertex) is replaced by the first vertex (*i.e.*, $n = 1$, and $m = 1$, rather than $n = N + 1$ or $m = M + 1$) since the vertices on each of the two surfaces form a closed loop.

A special case arises when the two surfaces share a common edge. In such a case, the distance S may be zero and, consequently, a singularity arises. In order to address such a singularity, Ambirajan and Venkateshan [30] proposed an analytical treatment, whereby the contribution to the summation in Eq. (6) by the shared edge becomes

$$\Delta F_{Q-P}\big|_{\text{shared edge}} = |\mathbf{q}_{\text{shared edge}}|^2\left(\frac{3}{2} - \frac{1}{2}\ln\left\{|\mathbf{q}_{\text{shared edge}}|^2\right\}\right) \tag{7}$$

where, as before, $|\mathbf{q}_{\text{shared edge}}|$ is the length of the shared edge.

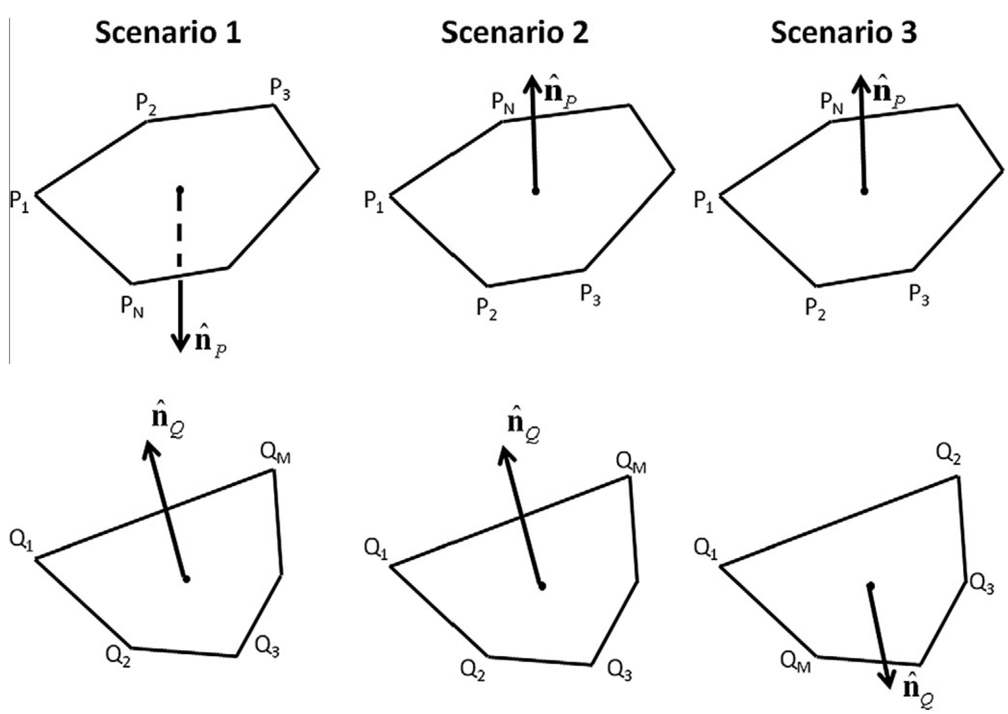
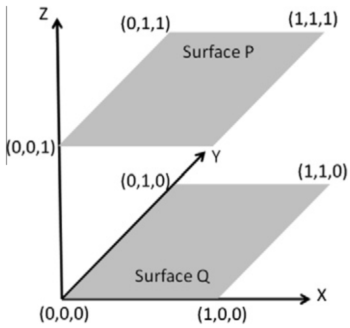
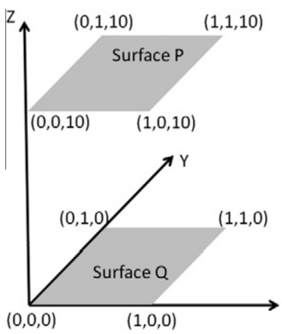
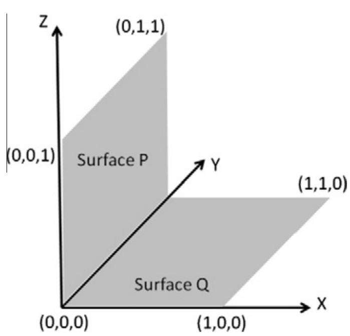
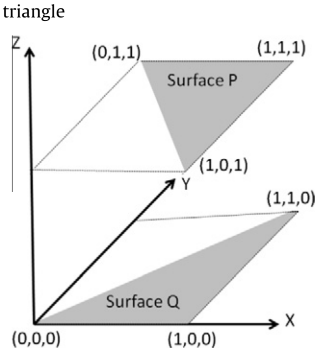


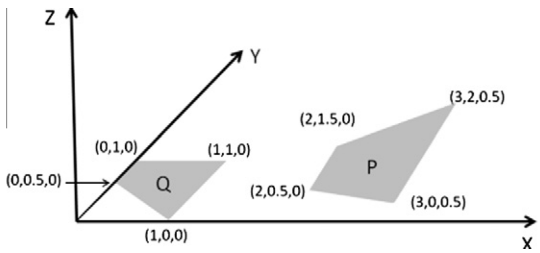
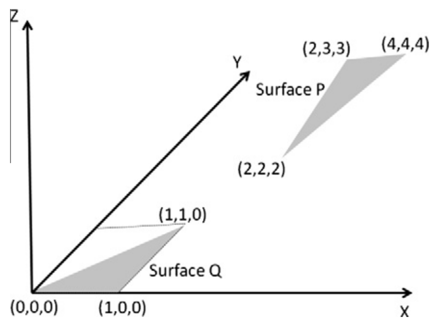
Fig. 2. Three possible surface normal orientations and vertex orders.

Table 1
View factor, F_{Q-P}, computed using the Monte Carlo method and the contour integral method (present method) for various scenarios.

<table>
<tr><th rowspan="2">Case studied</th><th rowspan="2">Exact (analytical)</th><th rowspan="2">Monte Carlo</th><th colspan="2">Contour integral (present method)</th></tr>
<tr><th>10-point Gauss</th><th>20-point Gauss</th></tr>
<tr><td>Case 1: parallel plate $1 \times 1 \times 1$

</td><td>0.199825</td><td>0.199944 (0.07%)</td><td>0.199825</td><td>0.199825</td></tr>
<tr><td>Case 2: parallel plate $1 \times 1 \times 10$

</td><td>3.162×10^{-3}</td><td>3.179×10^{-3} (0.25%)</td><td>3.162×10^{-3}</td><td>3.162×10^{-3}</td></tr>
<tr><td>Case 3: perpendicular plate $1 \times 1 \times 1$

</td><td>0.200044</td><td>0.200142 (0.06%)</td><td>0.200044</td><td>0.200044</td></tr>
<tr><td>Case 4: rotated triangle

</td><td>0.099912</td><td>0.100045 (0.11%)</td><td>0.099912</td><td>0.099912</td></tr>
</table>

(continued on next page)

Table 1 (*continued*)

Case studied	Exact (analytical)	Monte Carlo	Contour integral (present method)	
			10-point Gauss	20-point Gauss
Case 5: arbitrary quadrilateral 	Not available	3.701×10^{-3} (0.44%)	3.637×10^{-3}	3.637×10^{-3}
Case 6: arbitrary triangle 	Not available	1.083×10^{-3} (0.7%)	1.068×10^{-3}	1.068×10^{-3}

It is important to ponder the advantage of the proposed approach. While existing formulations using the contour integral method have been derived for regular polygons only, the current formulation is applicable to any arbitrary polygon as long as it is planar and is bounded by a set of discrete straight edges. The limits of the inner integral always go from zero to unity by virtue of using the parametric surface vector representation, and therein lies the advantage of the current formulation.

3. Numerical procedure

Implicit in the formulation presented above is the fact that the vertices on each surface element are placed in such an order that when the right-hand screw rule is used, the thumb points in the direction of the surface normal shown in Fig. 1, *i.e.*, inward toward each other. If the order of the vertices is not consistent with the surface normal, Eq. (6) may yield wrong results. For example, Fig. 2 shows three scenarios that may occur, when a mesh is generated using any standard mesh generator. The first scenario is the one described in Fig. 1. In the second scenario, the vertices on one of the surfaces are arranged such that the resulting surface normal is pointing outward. In this scenario, use of Eq. (6) will result in a negative view factor. In the third scenario where both surface normals are pointed outward, the resulting view factor will be positive. The negative view factor in the second scenario can be corrected simply by taking the absolute value, *i.e.*, one could compute the correct view factor (by using the absolute value of the formula) without paying any attention to which way the surface normals are pointed. However, if the surface elements share a common edge, and Eq. (7) is used, one must guarantee that both surface normals are pointed inward, and the vertices on each surface are placed in an order that obeys the right-hand screw rule. In summary, for Eqs. (6) and (7) to work in unison, one must guarantee first that the vertices on each surface element have a certain "correct" order. This is accomplished using the following algorithm:

- If vertex $P_1 = Q_1$ and $P_2 = Q_2$, then order is incorrect. Otherwise, it is correct.
- If the order is *incorrect*, the order of vertices on only one of the surface elements is to be reversed prior to any calculations. This may be either of the two surface elements.
- Irrespective of whether the order is correct or incorrect, the absolute value of the right hand side of Eq. (6) is to be used to compute the final view factor.

The computation of the integral on the right hand side of Eq. (6) requires a numerical quadrature scheme. Most previous studies indicate that Gaussian quadrature schemes [15,16] work best for this purpose. In this study, 10 and 20-point Gauss–Legendre quadrature schemes [29] were employed and compared. The overall algorithm for the view factor calculation is as follows:

Step 1: Extract coordinates of all vertices of both surfaces from the mesh generator. Initialize the view factor value to zero.
Step 2: For any edge pair (n, m), check if the edge is a shared edge.
Step 3: If the edge is a shared edge:
Step 3a: Check to see if vertices are in the "correct" order. If not, reorder vertices using the algorithm described above. This order should be retained throughout execution of the remainder of the algorithm.
Step 3b: The contribution of the shared edge to the value of the view factor is to be added using Eq. (7)
Step 4: If the edge is not a shared edge, add the contribution of that edge to the view factor value using Eq. (6).
Step 5: Repeat Step 2 and either Step 3 or 4 until all edges have been scanned.

4. Results and discussion

In order to assess the accuracy of the proposed formulation, six different test cases were selected. Since closed-form analytical solutions are available only for regular configurations, a Monte Carlo code was also developed as part of this work so that the formulation can be assessed for irregular geometries, *i.e.*, arbitrary polygons placed at arbitrary orientations. The development of the Monte Carlo code for calculating view factors required minor modifications of a previously existing Monte Carlo code [31].

Table 1 summarizes all six test cases. Of the six test cases, the first four cases have closed-form analytical solutions, and were chosen not only to assess the accuracy of the proposed formulation, but also to validate the Monte Carlo code, so that it can be used with confidence as a comparison for the other two cases for which analytical solutions are not available. The Monte Carlo results shown in Table 1 were computed after averaging 10 ensembles, *i.e.*, 10 independently (random number seed was initialized differently in each case) run simulation results were averaged. In each case, 1 million rays were traced from surface Q. The quantity within parenthesis alongside the Monte Carlo results represents the ratio of the standard deviation to the mean expressed as a percentage.

Several important conclusions can be drawn from the results shown in Table 1. First, for the first four test cases, it is clear that the results obtained using the proposed contour integration method matches analytical results exactly up to 6 decimal places. It is also evident that a 10-point Gauss Legendre quadrature scheme is sufficient for the numerical integration, and higher order integration is not necessary. The Monte Carlo results, as discussed earlier, do exhibit statistical errors, although, with 1 million rays, the errors are always small (less than 1%). One important trend to observe is that while the accuracy of the Monte Carlo method deteriorates for cases where the view factors become small (Case 2 versus Case 1), the accuracy of the proposed contour integral method is unchanged.

The final two test cases were selected to primarily demonstrate the generality of the proposed formulation. In both cases, the receiving surface (surface P) has been placed at a grazing angle of incidence. Consequently, the Monte Carlo results have relatively large errors. Although no comparison with analytical solutions is possible for these two cases, it is clear from Table 1 that the match between the Monte Carlo results and those produced by the contour integral method is very close—certainly within one standard deviation of the Monte Carlo results.

5. Summary and conclusions

A general formulation to compute diffuse view factors between two arbitrary convex planar polygons bounded by a set of discrete straight edges is developed. The formulation uses the double contour integral formula for computation of view factors as a starting point, and addresses the challenge of defining the limits to the integrals using a vector parametric representation to the surfaces. The formulation is particularly amenable to computation of view factors in three-dimensional geometries if an unstructured mesh is used for the actual radiation transport calculation.

Six test cases were considered. Closed-form analytic solution was used as the benchmark for four of the six test cases. Monte Carlo results were also generated for all six test cases. Based on the assessment of these six test cases, the following conclusions may be drawn:

- The proposed formulation using parametric representation of surfaces yields exact analytic solutions irrespective of geometry and configuration.
- It is sufficient to use a 10-point Gauss–Legendre quadrature for computation of the resultant definite integrals.
- The accuracy of the proposed method is, in general, superior to Monte Carlo results, which have inherent statistical errors associated with them.

References

[1] M.F. Modest, Radiative Heat Transfer, Second ed., Academic Press, 2003.
[2] R. Siegel, J. Howell, Thermal Radiation Heat Transfer, Fourth ed., Taylor and Francis, Hemisphere, 2001.
[3] J.R. Howell, A Catalog of Radiation Configuration Factors, McGraw-Hill, New York, 1982.
[4] M.F. Modest, Three-dimensional radiative exchange factors for nongray nondiffuse surfaces, Numer. Heat Transfer 1 (1978) 403–406.
[5] D.W. Yarbrough, C.-L. Lee, Monte Carlo calculation of radiative view factors, in: Integral Methods in Science and Engineering, Taylor and Francis, Hemisphere, Bristol, PA, 1986, pp. 563–574.
[6] J.D. Maltby, P.J. Burns, Performance, accuracy, and convergence in a three-dimensional Monte Carlo radiative transfer simulation, Numer. Heat Transfer Part B 19 (1991) 191–209.
[7] S. Miyahara, S. Kobayashi, Numerical calculation of view factors for an axially symmetrical geometry, Numer. Heat Transfer Part B 28 (1995) 437–453.
[8] G.V.G. Baranoski, J.G. Rokne, G. Xu, Applying the exponential Chevyshev inequality to the nondeterministic computation of form factors, J. Quant. Spectrosc. Radiative Transfer 69 (2001) 447–467.
[9] M.R. Vujicic, N.P. Lavery, S.G.R. Brown, View factor calculation using the Monte Carlo method and numerical sensitivity, Commun. Numer. Methods Eng. 22 (3) (2006) 197–203.
[10] M. Mirhosseini, A. Saboonchi, View factor calculation using the Monte Carlo method for a 3D strip element to circular cylinder, Int. Commun. Heat Mass Transfer 36 (6) (2011) 821–826.
[11] K.J. Daun, D.P. Morton, J.R. Howell, Smoothing Monte Carlo exchange factors through constrained maximum likelihood estimation, J. Heat Transfer 127 (10) (2005) 1124–1128.
[12] E.M. Sparrow, A new and simpler formulation for radiative angle factors, J. Heat Transfer 81 (1963) 81–87.
[13] J.B. Urquhart, View factor from conical surface by contour integration, AIAA J. 7 (11) (1969) 2157–2258.
[14] R.K. Srivastava, P.A. Ramachandran, M.P. Dudukovic, Radiation view factors in Czochralski crystal growth apparatus for short crystals, J. Cryst. Growth 74 (1986) 281–291.
[15] V. Rammohan Rao, V.M.K. Sastri, Efficient evaluation of diffuse view factors for radiation, Int. J. Heat Mass Transfer 39 (6) (1996) 1281–1286.
[16] S.B. Bopche, A. Sridharan, Determination of view factors by contour integral technique, Annals of Nuclear Engineering 36 (11–12) (2009) 1681–1688.
[17] C.K. Krishnaprakash, View-factor evaluation by quadrature over triangles, J. Thermophys. Heat Transfer 12 (1) (1998) 118–120.
[18] V.D. Hien, S. Chirarattananon, Triangular subdivision for the computation of form factors, J. Illuminating Eng. Soc. North Am. 2 (1) (2001) 41–59.
[19] M. Ravishankar, S. Mazumder, M. Sankar, Application of the modified differential approximation for radiative transfer to arbitrary geometry, J. Quant. Spectrosc. Radiative Transfer 111 (2010) 2052–2069.
[20] J.J. MacFarlane, VISRAD—A 3D view factor code and design tool for high energy density physics experiments, J. Quant. Spectrosc. Radiative Transfer 81 (2003) 287.
[21] G.N. Walton, J. Pye, (2011) View3D, <http://view3d.sourceforge.net/>.
[22] A.B. Shapiro, FACET – A radiation view factor computer code for axisymetric, 2d planar, and 3d geometries with shadowing, Lawrence Livermore National Laboratory Report UCID-19887, 1983.
[23] C.N. Zeeb, P.J. Burns, K. Branner, J.S. Dolaghan, User's manual for MONT3D—version 2.4, Colorado State University, Fort Collins, CO, 1999.
[24] G. N. Walton, Calculation of obstructed view factors by adaptive integration, Technical Report NISTIR–6925, National institute of standards and technology, Gaithersburg, MD, 2002.
[25] A.F. Emery, O. Johannson, M. Lobo, A. Abrous, A comparative study of methods for computing the diffuse radiation view factors for complex structures, J. F_{Q-P} Heat Transfer 113 (2) (1991) 413–422.
[26] K.G.T. Hollands, Application of parametric surface representation to evaluating form factors and like quantities, in: Proceedings of the International Radiation Symposium (Rad-IV), Istanbul, Turkey, 2004. DOI: http://dx.doi.org/10.1615/ICHMT.2004.RAD-4.80.
[27] K.J. Daun, K.G.T. Hollands, Infinitesimal-area radiative analysis using parametric surfaces through NURBS, J. Heat Transfer 123 (2) (2001) 249–256.
[28] P. Schroder, P. Hanrahan, On the form factor between two polygons, in: SIGGRAPH '93 Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 163–164, 1993, doi: http://dx.doi.org/10.1145/166117.166138.
[29] M. Abramowitz, I.A. Stegun (Eds.), Handbook of Mathematical Functions, Dover Publications, New York, 1965.
[30] A. Ambirajan, S.P. Venkateshan, Accurate determination of diffuse view-factors between planar surfaces, Int. J. Heat Mass Transfer 36 (8) (1993) 2203–2208.
[31] S. Mazumder, Methods to accelerate ray tracing in the Monte Carlo method for surface-to-surface radiation transport, J. Heat Transfer 128 (9) (2006) 945–952.