NUMERICAL METHOD FOR CALCULATING VIEW FACTOR BETWEEN TWO SURFACES

Lílian Dobrowolski de Carvalho Augusto, Bruno Giacomet and Nathan Mendes

Mechanic Engineering Graduate Program, Thermal Systems Laboratory,
Pontifical University Catholic of the Paraná
R. Imaculada Conceição, 1155 Curitiba-PR, 80215-901-Brazil

ABSTRACT

Long-wave radiation heat transfer is calculated using a geometrical factor the so-called view factor, which is normally estimated in a simplified way by building simulation programs, especially when there is no information in the literature due to the complexities to solve the double integral which defines the view factor between two surfaces. Analytical solutions are only available in the literature for a restricted number of simple configurations. This simplification on the process may implicate several errors on the evaluation of the internal surface temperatures which can cause discrepancies on the prediction of thermal comfort level and energy consumption, mainly when surfaces present complex geometries, openings (windows and doors) and when there are obstructions between them. In this way, the objective of this work is to present a computational method to determine the view factor between two surfaces considering simple or complex geometries with openings and obstructions, using an algorithm to generate a finite-element mesh onto surfaces to distinguish viewed/shaded surface areas. The algorithm is fast, memory-efficient, robust and uses constrained and unconstrained Delaunay triangulations. The mesh quality is guaranteed by the use of the Rupert's Delaunay refinement algorithm that avoids undesirable small triangle angles. A comparison with literature data available only for the case of simple configurations (without openings and obstructions) shows the excellent accuracy of the proposed computational method. For those cases, a sensitivity analysis consisting in varying the mesh refinement and the distance between the two surfaces has been performed. The results obtained for some geometries considering openings and obstructions are also presented.

KEYWORDS

View factor, Numerical method, Complex geometries, Opening and obstruction, Delaunay triangulations.

INTRODUCTION

Significant efforts have been made worldwide to minimize energy consumption by promoting the use of building simulation tools. In this way, the improvement of mathematical models accuracy is also of paramount importance.

For instance, long-wave radiation heat transfer is calculated using a geometrical factor that is well-know as view factor, which is normally estimated in a simplified way by building simulation programs, especially when there is no information in the literature due to the complexities to solve the double integral which defines the view factor between two surfaces. Analytical solutions are only available in the literature for a restricted number of simple configurations.

This simplification on the calculation of the view factor may imply calculation errors on the evaluation of the internal wall temperatures which can cause discrepancies on the prediction of thermal comfort level and energy consumption, mainly when surfaces present complex geometries, openings (windows and doors) and when there are obstructions between them.

A great collection of view factors relations between two surfaces is given by Howell (1982), and Siegel & Howell (2001). There are also some commercial and non-commercial computer programs for determining view factors for some more complex geometrical forms, such as presented in Wong (1976), Shapiro (1983), Burns (1983), Emery (1986), Jensen (1987) and Chin (1992). Moreover, we should also cite Hoays (1990) that developed a model and a program for the calculation of view factors in attics.

Nevertheless, no work previously cited presented an algorithm or program to calculate in a generic way the view factor regardless the geometry and the presence of openings or obstructions.

Therefore, we propose a numerical method to determine the view factor between any pair of surfaces, by calculating numerically the double-integral definition formula, considering openings and obstructions. The only limitation presented so far is that the method does not work for non-flat surfaces.

A triangular mesh is generated by using an algorithm that is fast, memory-efficient, and uses constrained and unconstrained Delaunay triangulations. The mesh quality is guaranteed by the use of the Rupert's Delaunay refinement algorithm that avoids undesirable small triangle angles.

EVALUATION OF VIEW FACTOR

Basic concepts

The view factor Fij is defined as the fraction of the radiation, diffusively distributed which comes from the surface with area Ai and goes to another surface with area Aj. To find a general expression of Fij let's consider Figure 1:

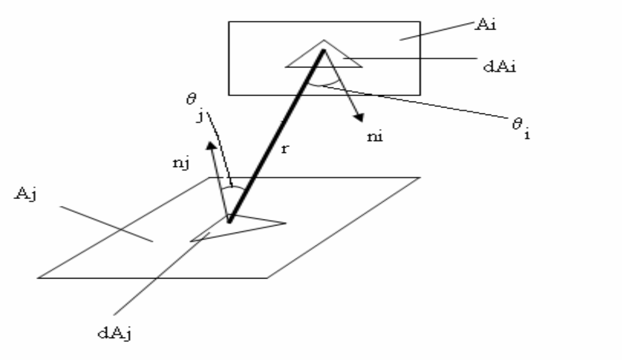


Figure 1. Needed parameters for the evaluation of the view factor.

The view factor between the two surfaces as indicated in Figure 1 can be obtained using Eq. (1),

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cdot \cos\theta_j}{\pi \cdot r^2} dA_i dA_j , \qquad (1)$$

where r represents the distance between the baricenters and the cosines θi and θj denote the angles between r and the respective normal vectors.

The evaluation of the double integral shown in Eq. 1 is, for the majority of cases, generally complex and presented for simplified configurations in tables, graphs or equations.

An alternative procedure consists of using an algorithm to generate a triangular mesh on the two surfaces to distinguish viewed/shaded surface areas and considering the approximation of double integral of Riemann,

$$\lim_{n\to\infty} \sum_{i=1}^{n} \sum_{j=1}^{m} f(x_{ij}) \Delta x_i \Delta y_j = \iint_B f(x,y)\,dxdy \; .$$

In this way, Eq.(1) can be re-written as:

$$F_{ij} = \frac{1}{A_i} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\cos\theta_i \cdot \cos\theta_j}{\pi \cdot r^2} \; dA_i dA_j . \qquad (2)$$

To calculate the view factor Fij given by Eq.(2), it is necessary, therefore, to determine the distance r, which will be calculated from the centers of the triangles i and j. The cosines θi and θj are given by Eq. (3),

$$\cos\theta_i = \frac{\ell_i (x_j - x_i) + m_i (y_j - y_i) + n_i (z_j - z_i)}{r}$$

$$\cos\theta_j = \frac{\ell_j (x_i - x_j) + m_j (y_i - y_j) + n_j (z_i - z_j)}{r} \; . \qquad (3)$$

The total surface area Ai and Aj and the differential elements dAi and dAj are evaluated by the area of each triangle generated by means of the Delaunay algorithm (Shewchuk 2003). However, the mesh generating algorithm is limited to two-dimensions and for purposes of building geometry definition, a three-dimension specification is needed so that the considered surfaces must suffer a rotation to any x-y plane and the generator algorithm of meshes will be activated and the surfaces will suffer another rotation right after, coming back to the previous position they had before the rotations.

Rotation of the Surfaces

To determine the angle α, between the surface and the x-y plane, a scalar product is used. The angle θ is formed between the normal vectors, however the desired angle is the angle formed between the surface and the x-y plane, given by the supplement of the angle θ, i.e., α = π – θ.

To rotate the surfaces, the matrix Equation (4) for an anti-clockwise rotation for the angle α bounding an axis in $\Re^3$ (Newmann and Sproull, 1979) is used,

$$M = \begin{bmatrix} a^2(1-\cos\theta)+\cos\theta & a\cdot b(1-\cos\theta)-c\cdot sen\theta & a\cdot c(1-\cos\theta)+b\cdot sen\theta \\ a\cdot b(1-\cos\theta)+c\cdot sen\theta & b^2(1-\cos\theta)+\cos\theta & b\cdot c(1-\cos\theta)-a\cdot sen\theta \\ a\cdot c(1-\cos\theta)-b\cdot sen\theta & b\cdot c(1-\cos\theta)+a\cdot sen\theta & c^2(1-\cos\theta)+\cos\theta \end{bmatrix} \qquad (4)$$

To undo the rotation, the same rotation process is used with the matrix presented in Eq. (4). However, in this case, the angle is α = 2π – θ.

Calculation of view factor between surfaces containing openings

The Triangle library (Shewchuk 2003) generates a mesh with triangular elements of Delaunay on the surfaces, but openings are not considered. To detect a possible opening on the surface, the Triangle library (Shewchuk 2003) has a system to check from a central opening point, which is calculated by the average point of the vertices of the opening, obtaining the triangles generated from this point, until finding an edge formed by the vertices of the opening and repeats this process successively, until the conditions are respected by all the edges of the opening. In this way, the surface with an opening is established followed by the view factor determination as described in the previous section.

Calculation of view factor between surfaces with obstructions in between

To determine the view factor between two plane surfaces with one or more obstructions, it was predetermined to treat these obstructions in the same way as did for the other surfaces. The first step, it is to obtain the plane equation of the considered surface as an obstruction, by means of its normal vector, that is determined by the vector product u x v, considering them as two vectors formed by the vertices of the surface, being thus, the equation of the plane given by Eq.(5),

$$\pi = Ax + By + Cz + D, \tag{5}$$

where $D = -(A \cdot x_i + B \cdot y_i + C \cdot z_i),$ and xi, yi and zi is the baricenter of the triangles at the surface i. Then, the vector equation of the line formed by the gravity centers of surfaces i and j verifies the relative position between the line and the plane. There are three kinds of relative position between line and plane, concurrent, parallel and when line belongs to the plane. Eq. (6) allows to check relative position between the lines, defined between points i and j, Fig 1, and by the plane of obstruction (π).

$$m = \frac{-(A \cdot x_1 + B \cdot y_1 + C \cdot z_1 + D)}{A \cdot a + B \cdot b + C \cdot c}, \tag{6}$$

where, a, b and c are determined by Eq.(7):

$$a = x_j - x_i,\ \ b = y_j - y_i \text{ and } c = z_j - z_i, \tag{7}$$

If the denominator of Eq.(5) is not zero, then point p of the intersection between the line and the plane can be found using the system of equations, Eq.(8):

$$x = x_i + a \cdot m\ ,\ y = y_i + b \cdot m \text{ and } z = z_i + c \cdot m\ . \tag{8}$$

Once the intersection point p is found, a certification is needed in order to know if the analyzed point belongs or not to the obstruction plane. This analysis is carried out using the barycentric coordinates, however, the method is two-dimensional as well, and, in this in case, the point p must be rotated simultaneously with triangles of the obstruction surface to guarantee that the point belongs to the same rotation plane.

The used method to determine if a point belongs or not to a triangle is broken of the fact that any point p of the triangle can be defined from the coordinates of its apexes, so that p is defined as:

$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_{3,} \tag{9}$$

where, λ1, λ2 and λ3 are real numbers so that:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{10}$$

The coefficients λ1, λ2 and λ3 are called barycentric coordinates of p in relation to p1, p2 and p3. With the coordinates of the points p, p1, p2 and p3 and Eq. (10), a system of three equations and three unknowns is set in order to find the barycentric coordinates.

$$\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 = x_p, \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 = y_p. \tag{11}$$

The system above can be determined as:

$$x_1 y_2 - x_2 y_1 + x_3 y_1 - x_1 y_3 + x_2 y_3 - x_3 y_2, \tag{12}$$

which determinant value corresponds to the double of the area of the triangle p1, p2 and p3, with a unique solution for each point p. The values of λ1, λ2 and λ3 can be obtained using the Cramer`s rule and expressed in terms of triangles areas:

$$\lambda_1 = \frac{S(pp_2p_3)}{S(p_1p_2p_3)}\ ,\ \lambda_2 = \frac{S(p_1pp_3)}{S(p_1p_2p_3)} \text{ and } \lambda_3 = \frac{S(p_1p_2p)}{S(p_1p_2p_3)} \tag{13}$$

The signal analysis of the barycentric coordinates indicates the plane region where p is found, according to the triangle coordinates p1, p2 and p3. It is observed that for the calculation of the areas the signal must to be taken into account, showing its orientation. From this result, the barycentric coordinates can be determined. If all the coordinates are greater or equal to zero, the point is within the triangle or at their boundaries. In this way, it is sufficient to check if each intersection point is on the obstruction surface (Davis, 2001).

However, if the Eq.-5 denominator is equal to zero, the numerator must be evaluated. In this case, two possibilities may occur. First, the straight line is parallel to the plane or is belongs to the plane. If the numerator is null, the line crosses the plane and, in this case, there is obstructions so that the calculation is not considered in the summation formula given by Eq. 2.

Computational structure

The computational code ViewFactor-LST, developed in C++, is an object-oriented program used to model all the scenery attributes.

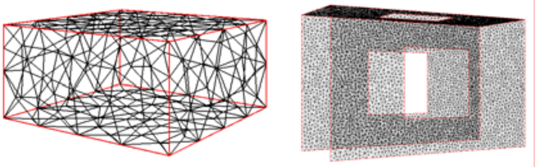
Scenery is a file that contains a determined three-dimensional solid, or several faces separated in the space. A simple example of scenery would be a cube, composed by six faces in different planes. Thus the object-oriented programming, the scenery, face and triangle classes form a hierarchy tree. All the rotation algorithms were implemented in a separated class, having one specifically for faces and another one for points.

The triangulation of the faces is made by the Triangle library as described before. Another relevant ability about the Triangle library is to manage the mesh size. There are several ways of restricting the size of the mesh (Shewchuck, 2002), for this work, it was decided to control it by the area of each triangle and angles were limited to be not greater than 20 degrees.

Visualization

In order to visualize the scenery, the Glut library has been used, based on OpenGL for three-dimensional images rendering, as presented in Fig. 2.

To visualize the scene is necessary to obtain all the resultant points of the triangulation of each face of the scenery and to pass them as segments (p1→ p2) until the whole image is rendered. The interface is user-friendly and from keyboard the user can translate, rotate and zoom the scenery.



(a)Triangulated rendered cube

(b) Three triangulated open faces

Figure 2. Triangulated surfaces visualization.

DISCUSSION AND RESULT ANALYSIS

Results for intersected surfaces of attics angle

Considering angles between 10 º and 90º, as shown in Figure (3), a comparison was made between the results from the presented algorithm and from those obtained from Hoays (1990), which describes a model and a program to determine view factor in attics.

In Fig. 4, it is shown that the results of the method compared to those presented by Hoays are in a very good agreement. There are some small discrepancies in the graphs especially because of the truncation error origin of numerical methods. Also, the triangle shape effect shows on the view factor resulting values, in this case having equilateral triangles and the attics with the following dimensions

$$\frac{B_a}{A_a}=2,\ \frac{C_a}{A_a}=1,\ \frac{D_a}{A_a}=1$$

Several evaluations have been performed with angles varying from 10º to 90º, using a 10º step. The results show that there is no need to refine the mesh to find a reliable value.

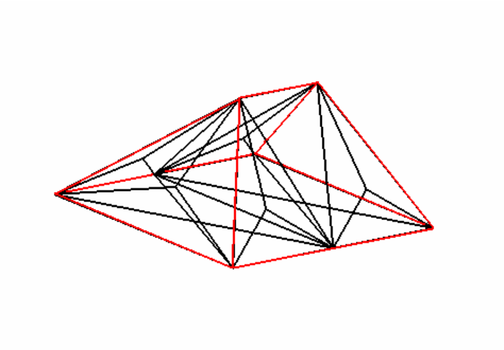


Figure 3. 45° triangulated attic

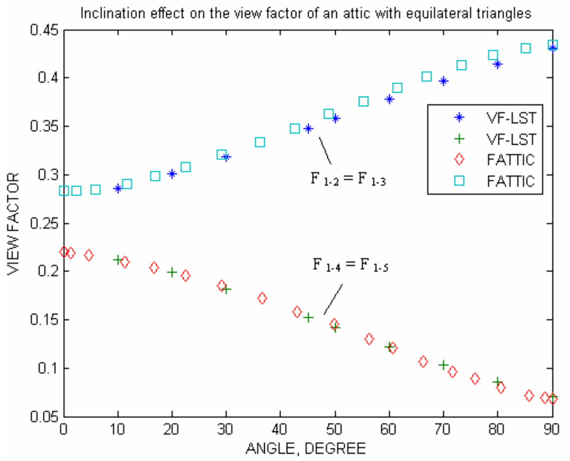


Figure 4. Inclination effect on the view factor of an attic with equilateral triangles by ViewFactor-LST (VF-LST)

Results for surfaces with openings

The view factor value has been calculated from non-opem to open surfaces as shown in Fig. 5 and the results from ViewFactor-LST are presented in Table 1. The cases are almost identical, squared shapes (2000 x 2000) and 2000 far from each other. The openings on surface 2 are supposed to vary from from small, medium, to big and the number from one to two. The results were validated in a logical way, calculating the open surface as if there was no opening surface and right after it is recalculated again, with the original opening as a surface and subtracting the first value from the second one, obtaining the results presented in Table 2. In comparison with table 1 and table 2, the error was smaller than 0.06%, proving the validity of the presented method. Only the surface with two openings has not been compared to any analytic result.

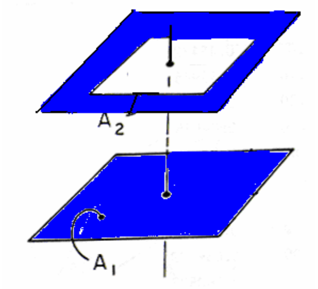
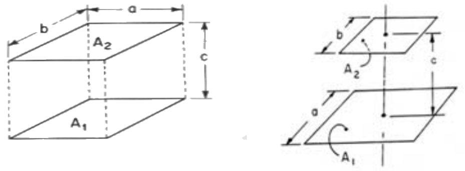


Figure 5. Open surface scene.



(a) Identical parallel surfaces scene

(b) Parallel concentric non-identical surfaces.

Figure 7. Parallel surfaces.

Table 1 Results from ViewFactor-LST with about 2.2x104.

Opening	small	medium	big	two
F_{1-2}	0.199829	0.142796	0.000331	0.15436
Opening area	4 u.a	10^6 u.a	3,99x10^6 u.a	250000 u.a

Table 2 Results obtained from Siegel et al. (2001) .

Opening	small	medium	big	two
F_{1-2}	0.199824	0.142709	0.000331	--------

Results for surfaces containing obstructions in between

In this case, a scenery as shown in Fig. 6 has been used, varying just the size of the obstruction and the distance between the surfaces, observing that the mesh on the obstruction was restricted to two triangles in all cases. The results from ViewFactor-LST are illustrated in Table 3. The values were analyzed in the same way as they were for opening cases. But in this time the results were higher than the section above because of the perspective is bigger compared to an opening. In a case that smaller values were obtained, it can be justified it because just too few triangles on one or both meshes were used. An important detail observed is that a high number of triangles is only needed when the obstruction have great dimensions.

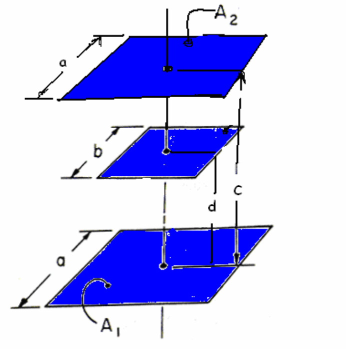


Figure 6. Surfaces with obstruction

Table3 Results obtained with ViewFactor-LST.

Number of triangles	Distance c and d	Obstruction area (u.a.)	F_{1-2}	F_{1-2}
3000	4 and 2	160000	0.060469	0.059083
6200	2 and 1	4	0.199837	0.199824
12400	2 and 1	3992004	0.0000	0.000331
3100	4 and 2	1000000	0.009043	0.011474
3500	6 and 3	1000000	0.048731	0.042537

CONCLUSION

An algorithm has been described to determined the view factor between two any surfaces. The results provided by using the algorithm provided errors as small as 0,06 %, promoting a new alternative method to reach these values and still enlarging possibilities to obtain view factors for any type of scenery.

The algorithm was developed to make possible the computational implementation for evaluating the view factor for constructions and, consequently, for many kinds of surfaces. In addition, the method can help to generate useful values for higher accurate results.

For the future, we intend to reduce computational cost of the ViewFactor-LST, for applications in programs of simulation and to develop a model for determination of view factor applied to non-flat surfaces.

REFERENCES

Burns, P. J., 1983, MONTE-a two-dimenional radiative exchange factor code, Technical report, Colorado State University, Fort Collins.

Chin, J. H., Panczak, T. D. and Fried, L., 1992, Spacecraft thermal modeling, Int. J. Numer. Methods Eng., vol. 35, pp.641-653.

Davis, C., 2001, Geometria de triângulos e polígonos. http://www.pbh.gov.br/prodabel/cde/ publicacoes/2001/davis2001e.pdf .

Emery, A. F., 1986, VIEW-a radiation view factor program with interactive graphics for geometry definition (version .5.3), Technical report, NASA computer software management and information center, Atlanta.

Hoays, H. S., 1990, Fattic: Um programa generalizado para a determinação dos fatores de forma em áticos, Encit 1990, Itapema, Santa Catarina, Brasil.

Horward, A. and Rorres, C., 2001, Álgebra linear com aplicações, 8.ed., Porto Alegre, Rio Grande do Sul, Brazil.

Howell, J.R., 1982, A Catalog of Radiation Configuration Factors, MacGraw-Hill, New York.

Jacob, M., 1957, Heat Transfer, vol. 2, John Wiley & Sons, New York.Clark, J. A.,1986, Private Communication, University of Michigan, Ann Harbor.

Jensen, C. L., 1987, TRASYS-II user's manual-thermal radition analysis system, Technical report, Marfim Marietta Aerospace Corp., Denver.

Mendes, N., Oliveira, R. C. L. F., and Santos, G. H., 2005, Energy Efficiency and Thermal Comfort Analysis Using the Powerdomus Hygrothermal Simulation Tool, Proc.of the Nineth Building Simulation Conference (IBPSA'05), Vol. 1, pp. 9-16, Montreal, Canada.

Mora, L., Mendonça, K. C., Wurtz, E., Inard, C. SimSpark: Na object-oriented environment to predic coupled heat and mass transfer in buildings. In: International IBPSA Conference (Building Simulation), 8, 2003, Eindhoven, Holanda. Proceedings…Eindhoven:IBPSA, 2003.

Newman,W. M. and Sproull, R. F., 1979, Principles of Interactive Computer Graphics, Second Edition.

Shapiro, A. B., 1983, FACET-a computer view factor computer code for axisymmetric, 2D planar, and 3D geometries whit shadowing, Technical report, University of California, Lawrence Livermore National Laboratory, August.

Shewchuk, J.R., 2003, Triangle: http://www.cs.cmu.edu/~quake/triangle.html

Shewchuk, J.R., 2002, Delaunay Refinement Algorithms for Triangular Mesh Generation, Computational Geometry: Theory and Applications 22(1-3): 21-74, May 2002.

Siegel, R. and Howell, J. R., 2001. A Catalog of Radiation Configuration Factors, Taylor and Francis, New York, Since publication, the following modifications have been made in the catalog: http://www.me.utexas.edu/~howell/index.html

Wong, R. L., 1976, User's manual for CNVUFAC-the General Dynamics heat transfer radiation view factor program, Technical report, University of California, Lawrence Livermore National Laboratory