

## VIEW FACTOR-LST; ALGORITMO PARA DETERMINAÇÃO NUMÉRICA DE FATOR DE FORMA

Lílian Dobrowolski de Carvalho Augusto – [liliandoca@gmail.com](mailto:liliandoca@gmail.com)

Universidade Federal do Paraná, Departamento de Matemática

Bruno Giacomet – [handriel@gmail.com](mailto:handriel@gmail.com)

Pontifícia Universidade Católica do Paraná, Departamento de Ciência da Computação

Marluz Jonsson – [marluzjonsson@bol.com.br](mailto:marluzjonsson@bol.com.br)

Pontifícia Universidade Católica do Paraná, Departamento de Engenharia Mecânica

Nathan Mendes – [nathan.mendes@pucpr.br](mailto:nathan.mendes@pucpr.br)

Pontifícia Universidade Católica do Paraná, Departamento de Engenharia Mecânica

### 6.2 Análise Térmica de Edificações

**Resumo.** *ViewFactor-LST: Um programa genérico para cálculo do fator de forma entre superfícies quaisquer, considerando obstruções. O fator de forma é determinado através da resolução de uma integral dupla. Essas soluções, para algumas superfícies são apresentadas, para muitas disposições diferentes, em forma de equações, de gráficos e de tabelas na literatura. A idéia de um método numérico para se calcular o fator de forma surge quando é necessário calculá-lo para superfícies com geometrias mais complexas. Em programas de simulação de edificações, o fator de forma normalmente é calculado de maneira simplificada para estas geometrias, não considerando aberturas e obstruções. Isso provoca erros na precisão de condições de conforto térmico e em cálculo de consumo de energia. O objetivo deste trabalho é apresentar um modelo computacional capaz de determinar numericamente os fatores de forma entre duas superfícies quaisquer e dispostas em qualquer ângulo, com geometrias simples ou complexas, considerando aberturas nas superfícies e também obstruções de uma superfície em relação à outra, utilizando recursos de computação gráfica e o algoritmo gerador de malhas bidimensionais com elementos triangulares de Delaunay. Os resultados obtidos para geometrias simples são comparados com dados disponíveis na literatura e com soluções analíticas e uma excelente concordância é observada. A análise de sensibilidade em relação ao refinamento da malha e à distância entre as superfícies consideradas também é apresentada. Para geometrias com aberturas e obstruções apresentam-se resultados e uma análise sobre os mesmos.*

**Palavras-chave:** Fator de Forma, superfícies com aberturas e obstruções, computação gráfica, malha bidimensional com elementos triangulares de Delaunay

## 1. INTRODUÇÃO

É comum hoje em dia a busca para reduzir o consumo de energia elétrica e relacionado a isto, também melhorar as condições de conforto térmico. Para alcançar este objetivo, devem-se tratar adequadamente as trocas de calor por radiação, mas isso normalmente não ocorre em virtude da dificuldade de determinar o fator de forma entre superfícies quaisquer, tendo em vista que existem infinitas maneiras de uma superfície estar associada à outra.

Uma grande coleção de relação de fatores de forma entre duas superfícies planas é dada por Howell *et al.* (1982), e Siegel & Howell *et al.* (2001). Também podem ser encontrados vários programas computacionais comerciais e não comerciais para determinar fatores de forma para algumas geometrias mais complexas, tais como Wong *et al.* (1976), Shapiro *et al.* (1983), Burns *et al.* (1983), Emery *et al.* (1986), Jensen *et al.* (1987) e Chin *et al.* (1992). Neste contexto deve-se citar ainda Hoays *et al.* (1990), que descreve um modelo e um programa para a determinação de fatores

de forma em áticos. No entanto, nos trabalhos citados anteriormente não é elaborado um programa genérico que calcule numericamente o fator de forma para qualquer tipo de superfície plana, não tratando apenas de um caso específico. O desenvolvimento de um novo modelo e a elaboração de um programa genérico respectivo para determinar o fator de forma entre superfícies quaisquer teria uma grande contribuição teórica quanto à dificuldade em obter a solução da integral dupla presente na definição do fator de forma, quanto prático devido a sua utilização em cálculo de transferência de calor por radiação.

O objetivo deste artigo é desenvolver um modelo genérico para cálculo do fator de forma entre duas superfícies planas quaisquer utilizando um algoritmo computacional de triangulação de superfícies usando os elementos triangulares de Delaunay, contribuindo, por exemplo, em projetos de equipamentos tais como fornos, refrigeradores e ainda em dimensionamento de sistemas de climatização ou em projetos e simulações de edificações, tais como Power Domus (Mendes, 2005) e SimSpark (Mora, 2003)

## 2. DETERMINAÇÃO DE FATOR DE FORMA

### 2.1 Conceitos Básicos

O fator de forma  $F_{ij}$  é denominado como a fração da radiação, difusivamente distribuída, que sai da superfície de área  $A_i$  e incide sobre uma outra superfície de área  $A_j$ . A fim de desenvolver uma expressão geral de  $F_{ij}$ , considere a Fig. 1:

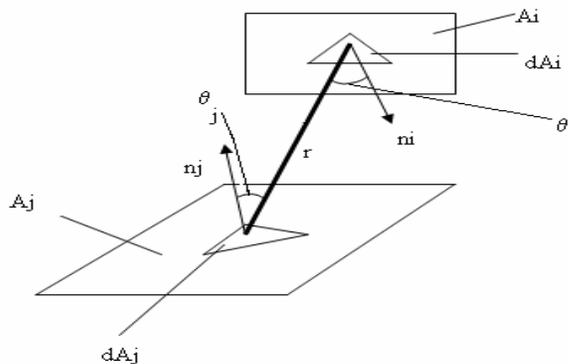


Figura 1. Esquema para a apresentação da expressão geral para determinação do fator de forma.

O fator de forma entre as superfícies de área  $A_i$  e  $A_j$  indicadas na Fig. 1, pode ser obtido através da Eq.(1),

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

onde  $r$  representa a distância entre os baricentros e os cossenos  $\theta_i$  e  $\theta_j$  denotam os ângulos entre  $r$  e as respectivas normais.

A avaliação da integral dupla, indicada na Eq.(1) é, para a maioria dos casos, geralmente complexa, sendo então comumente apresentados na forma de gráficos e de tabelas. Um procedimento alternativo consiste em utilizar recursos de computação gráfica e um algoritmo gerador de malhas bidimensionais com elementos triangulares de Delaunay, juntamente com aplicações de ferramentas matemáticas. Com base nesse procedimento, considerando a definição de integral dupla de Riemann,

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n \sum_{j=1}^m f(x_{ij}) \Delta x_i \Delta y_j = \iint_B f(x, y) dx dy,$$

a Eq.(1) transforma-se em:

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

Para calcular o fator de forma  $F_{ij}$  dado pela Eq.(2) é necessário, portanto, determinar a distância  $r$ , que será calculada a partir do baricentro do triângulo da superfície  $i$  ao baricentro da superfície  $j$ , os cossenos  $\theta_i$  e  $\theta_j$ , que são dados pelas Eq.(3) e Eq.(4),

$$\cos \theta_i = \frac{\ell_i(x_j - x_i) + m_i(y_j - y_i) + n_i(z_j - z_i)}{r} \quad \text{e} \quad \cos \theta_j = \frac{\ell_j(x_i - x_j) + m_j(y_i - y_j) + n_j(z_i - z_j)}{r}, \quad (3)$$

a área total da superfície  $A_i$  e  $A_j$  e os  $dA_i$  e  $dA_j$ , sendo que os últimos elementos são avaliados como a área de cada triângulo gerado através do algoritmo gerador de malhas bidimensionais com elementos triangulares de Delaunay (Shewchuk, 2003). Entretanto tem-se um problema, que o algoritmo gerador de malhas com elementos triangulares de Delaunay é bidimensional e as superfícies são tridimensionais, sendo assim, as superfícies consideradas sofrerão uma rotação para algum plano  $xy$  e o algoritmo gerador de malhas será acionado e logo após as superfícies sofrerão outra rotação, voltando à posição anterior às rotações.

## 2.2 Rotação das Superfícies

Para determinar o ângulo  $a$  formado entre a superfície avaliada e o plano  $xy$ , usa-se o produto escalar, com a normal da superfície considerada e a normal do plano  $xy$ . O ângulo encontrado  $\theta$  é o ângulo formado entre os vetores normais, entretanto o ângulo que se deseja determinar é o ângulo formado entre a superfície e o plano  $xy$ , onde este é dado pelo suplemento do ângulo  $\theta$ ,  $\alpha = \pi - \theta$ .

Para fazer a rotação das superfícies utilizou-se a matriz Eq.(4) de rotação anti-horária por um ângulo  $a$  em torno de um eixo em  $\mathfrak{R}^3$  (Newmann e Sproull, 1979), determinado por um vetor unitário com ponto inicial na origem e direção ao longo do eixo de rotação pelo ângulo  $a$ . Este vetor é determinado através do produto vetorial entre a normal da superfície que se deseja rotacionar e a normal do plano  $xy$ .

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

Para desfazer a rotação utiliza-se o mesmo processo da rotação juntamente com a matriz Eq.(4), porém o ângulo utilizado neste procedimento é  $\alpha = 2 \cdot \pi - \theta$ .

## 2.3 Determinação de fator de forma para superfícies com aberturas

A biblioteca Triangle (Shewchuk, 2003) gera uma malha com elementos triangulares de Delaunay nas superfícies, ainda não considerando que a mesma venha a ter alguma abertura. Para se detectar uma possível abertura na superfície, a própria biblioteca Triangle (Shewchuk, 2003) tem um

sistema de verificação que funciona a partir de um dado ponto central da abertura, que é calculado pelo ponto médio dos vértices da abertura, a mesma, começa a retirar os triângulos gerados, a partir deste ponto, até encontrar uma aresta formada pelos vértices da abertura e assim sucessivamente, até ter encontrado todas as arestas da abertura. Dessa forma a superfície com abertura é estabelecida e assim segue-se com a determinação de fator de forma através do procedimento descrito nas seções anteriores.

#### 2.4 Determinação de fator de forma para superfícies com obstruções

Para se determinar o fator de forma para duas superfícies planas com uma ou várias obstruções entre elas, convencionou-se tratá-las como superfícies planas também.

Calcula-se a equação do plano da superfície de obstrução, através do seu vetor normal, que é determinado pelo produto vetorial  $u \times v$ , considerando  $u$  e  $v$  como dois vetores formados pelos vértices da superfície, sendo assim, a equação do plano será dada pela Eq.(5)

$$\pi = Ax + By + Cz + D, \quad (5)$$

onde,  $D = -(A \cdot x_i + B \cdot y_i + C \cdot z_i)$ , e  $x_i$ ,  $y_i$  e  $z_i$  são os baricentros dos triângulos da superfície  $i$ .

Após isto, encontra-se a equação vetorial da reta formada pelos baricentros das superfícies avaliadas e verifica-se qual a posição relativa entre a reta e o plano. Podem-se encontrar três tipos de posição relativa entre reta e plano, concorrentes, paralelos e reta pertence ao plano. Através da Eq.(6) é possível verificar como a reta  $r$ , gerada pelos baricentros das superfícies  $i$  e  $j$ , Fig. 1, está relacionada a um eventual plano de obstrução ( $p$ ).

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

onde  $a$ ,  $b$  e  $c$  são determinados pela Eq.(7)

$$a = x_j - x_i, \quad b = y_j - y_i \quad \text{e} \quad c = z_j - z_i, \quad (7)$$

onde,  $x_i$ ,  $y_i$  e  $z_i$  são as coordenadas dos baricentros dos triângulos da superfície  $i$  e  $x_j$ ,  $y_j$  e  $z_j$  são as coordenadas dos baricentros dos triângulos da superfície  $j$ .

Se o denominador da Eq.(5) for diferente de zero, então pode-se encontrar o ponto  $p$  de intersecção da reta e do plano através do sistema de equações, Eq.(8)

$$x = x_i + a \cdot m, \quad y = y_i + b \cdot m \quad \text{e} \quad z = z_i + c \cdot m. \quad (8)$$

Encontrando-se o ponto  $p$  de intersecção deve-se verificar se ele pertence ou não a obstrução. Para isso foi utilizado um método baseado em coordenadas baricêntricas, no entanto, este método também é bidimensional, neste caso o ponto  $p$  deve ser rotacionado para um plano  $xy$ , juntamente com os pontos dos triângulos da superfície de obstrução, estes devem ser rotacionados novamente junto com o ponto  $p$  encontrado, devido ao fato de que, pode acontecer que, usando os pontos dos triângulos da primeira rotação, o ponto de intersecção ser rotacionado para um plano diferente destes, ocasionando assim um resultado incorreto, isto é, o ponto de intersecção estar num plano e os pontos dos triângulos estarem em outro, dessa forma nunca o ponto de intersecção seria pertencente a um dos triângulos da superfície de obstrução e assim não teríamos obstrução. Outro fato relevante, é que os triângulos são formados após a desrotação, portanto, não tem como verificar na primeira rotação quais os pontos gerados pela Triangle (Shewchuk, 2003) formará os triângulos e por isso

não há como calcular as suas áreas, sendo que o método utilizado para fazer tal verificação necessita das áreas dos triângulos, logo é importante fazer a rotação novamente.

O método utilizado para determinar se um ponto pertence ou não a um triângulo, parte-se do fato de que qualquer ponto  $p$  do triângulo pode ser definido a partir das coordenadas de seus vértices, de modo que  $p$  seja definido como:

$$p = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3, \quad (9)$$

onde,  $\lambda_1$ ,  $\lambda_2$  e  $\lambda_3$  são números reais e

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \quad (10)$$

Os coeficientes  $\lambda_1$ ,  $\lambda_2$  e  $\lambda_3$  são denominados coordenadas baricêntricas de  $p$  em relação a  $p_1$ ,  $p_2$  e  $p_3$ .

Com as coordenadas dos pontos  $p$ ,  $p_1$ ,  $p_2$  e  $p_3$  e a Eq.(10), constrói-se um sistema de três equações e três incógnitas para encontrar as coordenadas baricêntricas

$$\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 = x_p, \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 = y_p. \quad (11)$$

O sistema de equações (11) tem por determinante exatamente,

$$x_1 y_2 - x_2 y_1 + x_3 y_1 - x_1 y_3 + x_2 y_3 - x_3 y_2, \quad (12)$$

cujos valores correspondem ao dobro da área do triângulo  $p_1 p_2 p_3$ .

A área do triângulo é não nula, pois  $p_1$ ,  $p_2$  e  $p_3$  não são alinhados por hipótese, já que são vértices de um triângulo. Assim, o sistema tem solução única para cada ponto  $p$ .

Os valores de  $\lambda_1$ ,  $\lambda_2$  e  $\lambda_3$  podem ser obtidos usando a regra de Cramer, e expresso em termos de áreas de triângulos. Tem-se, portanto,

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

A análise do sinal das coordenadas baricêntricas indica a região do plano em que se encontra  $p$ , em relação ao triângulo  $p_1 p_2 p_3$ . Observe-se que, para isso, as áreas devem ser orientadas, ou seja, devem ser calculadas com sinal. Com esse resultado, basta obter as coordenadas do ponto e dos vértices do triângulo, e calcular as coordenadas baricêntricas. Se todas as três forem maiores ou iguais a zero, o ponto pertence ao interior do triângulo ou à fronteira, respectivamente, e se uma das coordenadas baricêntricas for negativa o ponto  $p$  não pertence ao triângulo. Assim basta checar cada ponto de intersecção encontrado contra cada um dos triângulos da superfície de obstrução (Davis, 2001). Agora, se o denominador da Eq.(5) for igual a zero, deve-se avaliar o numerador da equação, pois pode ocorrer neste caso duas possibilidades ou a reta é paralela ao plano ou está contida no plano. Se o numerador for igual a zero a reta está contida no plano, portanto há obstrução, se não a reta é paralela ao plano e assim não há obstrução e a determinação numérica de fator de forma segue conforme as especificações das seções 2.1 e 2.2.

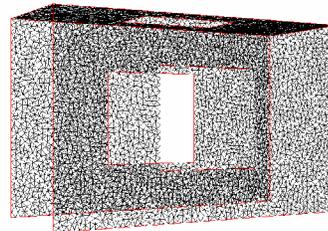
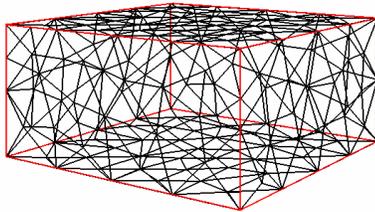
## 2.5 Estruturação do código fonte

O código computacional ViewFactor-LST desenvolvido em C++, utilizou o princípio de orientação à objetos para modelar e estruturar as várias propriedades de cada cenário como entrada para o programa.

Cenário é um arquivo de entrada que contém um determinado sólido tridimensional, ou várias faces separadas no espaço. Um exemplo de cenário seria um cubo, composto de seis faces em planos diferentes. Para isso foi criada uma classe que armazena esses dados de entrada, sendo eles o número total de vértices, o número de faces, a lista dos vértices que mantêm uma lista de faces, vindas do arquivo de cenário. Essas faces compõem uma classe específica chamada *face*. A classe das faces, por sua vez, contém apenas os pontos que formam aquela face, uma estrutura de entrada e uma de saída para a sua triangulação, e uma lista de pontos para a face triangulada. Também mantém uma lista de elementos da classe *triclass*, que guarda os triângulos que compõe a determinada face, por sua vez, calcula e armazena a área e os centros de todos os triângulos. Para fazer a rotação foi implementou-se uma classe *rotateclass*, que possui os métodos necessários para retornar os pontos para a classe da face. A biblioteca Triangle (Shewchuk, 2003) faz toda a parte de geração da malha utilizando os princípios de Delaunay. No artigo Shewchuk *et al.* (2002) pode ser encontrada as informações sobre como é feita a triangulação das superfícies e o critério de escolha dos vértices para a formação dos triângulos. Para realizar uma triangulação existem vários meios de restringir o tamanho da malha, nesse trabalho foi estabelecida a restrição pela área de cada triângulo e ângulos maiores que 20 graus.

## 2.6 Visualização

Para se obter uma visão do cenário onde será calculado o fator de forma, foi utilizado a GLUT, uma biblioteca criada dentro da OpenGL para renderização de imagens tridimensionais, conforme Fig. 2. Na visualização é necessário obter todos os pontos resultantes da triangulação de cada face do cenário e passá-los como segmentos ( $p1 \rightarrow p2$ ) até que todos eles sejam renderizados. A movimentação é simples, feita via teclado, podendo deslocar horizontalmente e verticalmente, rotacionar, e aproximar a câmera, permitindo um controle mais amigável ao usuário.

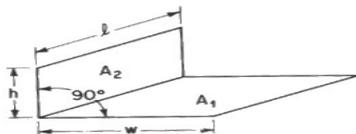


(a) Visualização de um cubo triangulado. (b) Visualização de três paredes trianguladas com suas respectivas janelas.

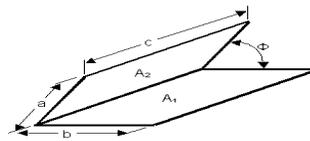
Figura 2. Visualização de triangulações de superfícies.

## 2.7 Resultados para placas com um lado em comum formando um ângulo F

Considerando o ângulo entre as placas de  $F=90^\circ$  e um ângulo qualquer, conforme Figuras 3(a) e 3(b), respectivamente, foi feita uma comparação entre os resultados tabelados dados por Siegel & Howell *et al.* (2001), apresentados na Tab. 1 e os obtidos com o algoritmo a comparação pode ser visualizada na Fig. 4.



(a) Esquema para placas perpendiculares.



(b) Esquema para dois retângulos com um lado em comum formando qualquer ângulo entre si.

Figura 3. Esquema de placas com um lado em comum.

Tabela 1. Resultados tabelados variando a relação A da superfície, sendo que  $A=h/l$  para  $F=90^\circ$  e  $A=a/c$  para os demais ângulos F.

Ângulo	$F_{1-2}, A=10$	$F_{1-2}, A=4$	$F_{1-2}, A=1$	$F_{1-2}, A=0,4$	$F_{1-2}, A=0,1$
$30^\circ$	0,728385	0,690387	0,619028	0,394538	0,235961
$45^\circ$	0,602836	0,560160	0,483347	0,280023	0,160116
$60^\circ$	0,485586	0,443475	0,370905	0,202035	0,112663
$90^\circ$	0,281888	0,250320	0,200044	0,101359	0,055024
$120^\circ$	0,128098	0,111512	0,086615	0,042260	0,022639
$135^\circ$	0,072612	0,062773	0,048310	0,023305	0,012436

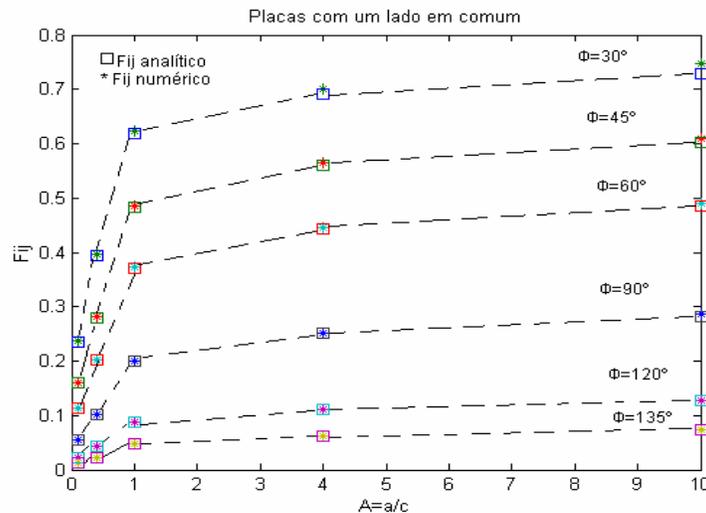


Figura 4. Comparação do fator de forma obtido pelo ViewFactor-LST e o tabelado, com aproximadamente  $4 \times 10^4$  triângulos.

Através da Fig. 4, pode-se observar que a concordância entre os resultados analítico e numérico é excelente quando se utiliza aproximadamente  $4 \cdot 10^4$  triângulos.

Foram feitas várias simulações, variando-se o número de triângulos de  $1 \cdot 10^4$ ,  $2 \cdot 10^4$ ,  $3 \cdot 10^4$ ,  $4 \cdot 10^4$  e  $5 \cdot 10^4$ . Observou-se que para  $F=135^\circ$  e relações maiores que um, a malha não precisava ser tão refinada, pois logo se encontrava uma aproximação satisfatória enquanto que se a malha fosse muito refinada acima de  $4,0 \cdot 10^4$  o resultado não era tão satisfatório devido a propagação de erro no truncamento de números irracionais utilizados no cálculo do ViewFactor-LST, isso não ocorreu apenas com o  $F=135^\circ$ , mas também com os ângulos  $F=120^\circ$  e  $F=90^\circ$ .

### 2.8 Resultados para superfícies com aberturas

Calculou-se o fator de forma de superfícies sem abertura, com abertura, conforme Fig. 5 e os resultados obtidos com o Algoritmo ViewFactor-LST encontram-se na Tab. 2. Os cenários são parecidos, placas quadradas (2000 x 2000) u.c. e distânciadas de 2000 u.c., as aberturas na superfície 2 são alteradas de pequena, média, grande e em quantidade de uma para duas aberturas. Os resultados foram comparados de maneira intuitiva, considerou-se a superfície com abertura como se não a tivesse e calculou-se o fator de forma pela Eq.(14) conforme Fig. 6(a), depois considerou a abertura como se fosse uma superfície e calculou-se o fator de forma pela Eq.(15) conforme Fig. 6(b), após isto, diminuiu o primeiro resultado do segundo e obteve-se o fator de forma para a superfície conforme Fig. 5, apresentados na Tab. 3 comparando os resultados das Tab. 2 e Tab. 3 verifica-se que o erro é inferior à 0,06%, somente a superfície com duas aberturas que não foi comparada com nenhum resultado analítico.

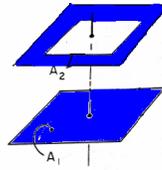


Figura 5. Esquema de superfícies com aberturas.



(a) Esquema de placas paralelas e iguais. (b) Esquema de placas quadradas tamanhos diferentes paralelas e concêntricas.

Figura 6. Esquemas de placas paralelas.

$$F_{1-2} = \frac{2}{\pi XY} \left\{ \ln \left[ \frac{(1+X^2) \cdot (1+Y^2)}{1+X^2+Y^2} \right]^{1/2} + X\sqrt{1+Y^2} \tan^{-1} \frac{X}{\sqrt{1+Y^2}} + Y\sqrt{1+X^2} \tan^{-1} \frac{Y}{\sqrt{1+X^2}} - X \tan^{-1} X - Y \tan^{-1} Y \right\}, \quad (14)$$

onde  $X = a/c$  e  $Y = b/c$ .

$$F_{1-2} = \frac{1}{\pi A^2} \left\{ \ln \left( \frac{[A^2(1+B^2)+2]^2}{(Y^2+2)(X^2+2)} \right) + (Y^2+4)^{1/2} \left[ Y \tan^{-1} \frac{Y}{(Y^2+4)^{1/2}} - X \tan^{-1} \frac{X}{(Y^2+4)^{1/2}} \right] + \right. \\ \left. (X^2+4)^{1/2} \left[ X \tan^{-1} \frac{X}{(X^2+4)^{1/2}} - Y \tan^{-1} \frac{Y}{(X^2+4)^{1/2}} \right] \right\}, \quad (15)$$

Para  $A < 0,2$ ,  $F_{1-2} = \frac{AB^2}{\pi}$

onde  $A = a/c$ ;  $B = b/a$ ;  $X = A(1+B)$ ;  $Y = A(1-B)$ .

Tabela 2. Resultados obtidos com ViewFactor-LST com aproximadamente  $2,2 \times 10^4$  triângulos

Abertura	pequena	média	grande	duas
$F_{1-2}$	0,199829	0,142796	0,000331	0,154362
Área da abertura	4 u.a.	1.000.000 u.a.	3.992.004 u.a.	250.000 u.a.

Tabela 3. Resultados obtidos efetuando a subtração especificada através das Eq.(14) e (15).

Abertura	pequena	média	grande	duas
$F_{1-2}$	0,199824	0.142709	0,000331	-----

## 2.9 Resultados para superfícies com obstruções

Para superfícies com obstrução utilizou-se um cenário conforme a Fig. 7, modificando-se apenas o tamanho da obstrução e a distância entre as placas, o número de triângulos na obstrução manteve igual a dois. Os resultados obtidos com o Algoritmo ViewFactor-LST encontram-se na Tab. 4. Os resultados também foram analisados de maneira intuitiva de forma análoga aos da abertura, lo-

gicamente que os resultados seriam um pouco maior que os obtidos na seção anterior devido ao fato de que há uma perspectiva de visualização maior em relação a uma abertura. Em casos que se obtiveram resultados menores deve-se ao fato que o número de triângulos foi pequeno em relação à obstrução e para se obter um resultado mais satisfatório deveria ter um número maior de triângulos nas superfícies  $A_1$  e  $A_2$  conforme mostrado na Tab. 4. Observou-se também que os números de triângulos não precisavam ser grandes para se obter um resultado satisfatório, somente quando a obstrução tinha dimensões grandes que o número de triângulos necessitava ser bem maior.

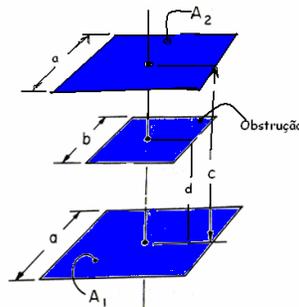


Figura 7. Esquema para obstrução.

Tabela 4. Resultados obtidos com o ViewFactor-LST

Número de triângulos em $A_1$ e $A_2$	Distância c e d, conforme Fig. 7	Área da obstrução (u.a)	$F_{1-2}$	$F_{1-2}$ utilizado para comparar
3.000	4 e 2	160000	0,060469	0,059083
6.200	2 e 1	4,0	0,199837	0,199824
12.400	2 e 1	3992004	0,000000	0,000331
3.100	4 e 2	1000000	0,009043	0,011474
3.500	6 e 3	1000000	0,048731	0,042537

## 2.10 Conclusão

Descreveu-se um algoritmo capaz de determinar numericamente o fator de forma entre superfícies quaisquer, indicado na Fig. 1, para qualquer ângulo  $\theta_i$  e  $\theta_j$  e com aberturas e obstruções. Os resultados fornecidos pelo algoritmo obtiveram o maior erro inferior a 0,06%, promovendo uma maneira alternativa de se obter esses valores e ainda ampliando possibilidades de calcular fatores de forma para qualquer tipo de cenário. O algoritmo foi desenvolvido de modo a possibilitar sua implementação em um programa de simulação dinâmica de edificações e com isso é possível avaliar o fator de forma para diversas superfícies e, conseqüentemente, calcular com maior precisão, por exemplo, a transferência de calor que ocorre em edificações.

Para o futuro, pretende-se reduzir tempo computacional do código ViewFactor-LST, para aplicações em programas de simulação tais como: Power Domus (Mendes et al., 2005) e SimSpark (Mora et al., 2003), compará-lo com outro algoritmo, por exemplo, o Fattic, desenvolvido por Hoays (1990) e desenvolver um modelo para determinação de fator de forma em superfícies curvas.

## REFERÊNCIAS

- Burns, P. J., 1983, MONTE-a two-dimensional 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, Fatic: 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 radiation 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 Ninth 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.

## ALGORITHM FOR NUMERICAL EVALUATION OF VIEW FACTOR

**Abstract.** *The view factor is determined via the resolution of a double integral. These solutions, for some surfaces, are presented for many different configurations, in form of equations, graphs and tables in the literature. The idea of creating a numerical method to calculate the view factor appears when it is needed to calculate it for surfaces with complex geometries. For instance, in building simulation programs, the view factor is normally calculated in a simplified way for these geometries, disregarding openings and obstructions. This implicates errors in the accuracy of thermal comfort calculation and energy consumption. The objective of this work is to present a computational structure to determine view factors between two surfaces, with simple or complex geometries, using resources of graphical computation to 2D meshes with triangular elements of Delaunay. The results gotten for simple geometries are compared with available data in the literature and an excellent agreement is observed. The sensitivity analysis to the mesh refinement and the distance between the surfaces is also presented.*

**Keywords:** view factor, surfaces with complex geometries, graphical computation, 2D mesh with triangular elements