

ALGORITMO PARA DETERMINAÇÃO NUMÉRICA DE FATOR DE FORMA

Lílian Dobrowolski de Carvalho Augusto

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR,80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
liliandoca@gmail.com

Bruno Giacomet

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR,80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
handriel@gmail.com

Marluz Jonsson

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR,80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
marluzjonsson@bol.com.br

Nathan Mendes

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR,80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
nathan.mendes@pucpr.br

Liliana Madalena Gramani Cumin

Departamento de Matemática, Universidade Federal do Paraná.
Caixa Postal 19044, 81531-990 Curitiba - PR – Brasil
gramani@mat.ufpr.br

Resumo. O fator de forma é determinado através da resolução de uma integral dupla. Essas soluções, para superfícies comuns, 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 se necessita 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, com geometrias simples ou complexas, utilizando-se 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 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 é também apresentada.

Palavras chave: fator de forma, superfícies com geometrias complexas, computação gráfica, malha bidimensional com elementos triangulares.

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 a outra.

Uma grande coleção de relação de fatores de forma entre duas superfícies planas é dada por Howell (1982), e Siegel & Howell (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 (1976), Shapiro (1983), Burns (1983), Emery (1986), Jensen (1987) e Chin (1992). Neste contexto deve-se citar ainda Hoays (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 cálculos 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 *et al.*, 2005) e SimSpark (Mora *et al.*, 2003)

2. 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:

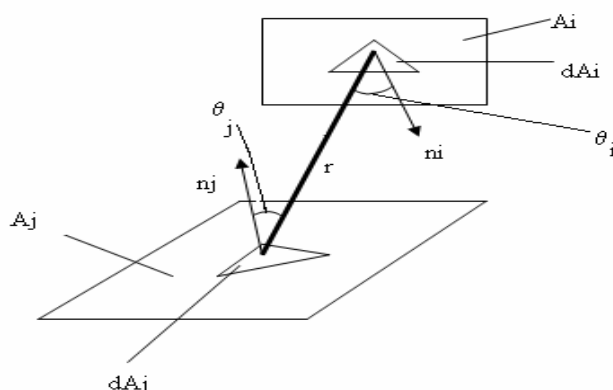


Figura 1. Esquema para a apresentação da expressão geral para determinação do fator de forma.

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 θ_i e θ_j denotam os ângulos entre r e as respectivas normais.

Analogamente, o fator de forma F_{ji} se define como a fração da radiação que sai da superfície A_j e é interceptada por A_i , dada por uma equação similar a Eq. (1) exceto que a subscrição i e j são intercaladas. Dessa forma, tem-se que:

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

Comparando-se as equações (1) e (2), surge a seguinte relação de reciprocidade:

$$A_i \cdot F_{ij} = A_j \cdot F_{ji}, \quad (3)$$

podendo ser usada para determinar o fator de forma associado a duas superfícies quaisquer que sejam emissoras e refletoras difusoras e que tenham radiosidade uniforme.

3. Determinação do fator de forma

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:

$$F_{ij} = \frac{1}{A_i} \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{\cos \theta_i \cdot \cos \theta_j}{\pi \cdot r^2} dA_i dA_j. \quad (4)$$

Para calcular o fator de forma F_{ji} dado pela Eq. (4) é necessário, portanto, determinar a distância r , os cossenos θ_i e θ_j , 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).

3.1 Determinação da distância r

Para calcular a distância r , toma-se o esquema da fig.1, onde o baricentro de cada triângulo é considerado como o ponto central e a distância r é calculada a partir do baricentro do triângulo da superfície i ao baricentro do triângulo da superfície j .

3.2 Determinação dos cossenos

Em geral, a forma de uma superfície pode ser descrita, especificamente, pela direção normal em todos os pontos na superfície. Por sua vez, a direção normal é geralmente expressa em termos dos cossenos diretores ℓ , m , e n .

O cosseno diretor ℓ é o cosseno do ângulo entre a normal da superfície e a direção x , o cosseno diretor m é o cosseno do ângulo entre a normal da superfície e a direção y e assim por diante. O vetor normal local para a superfície pode ser expresso como:

$$\mathbf{n} = \ell \mathbf{i} + m \mathbf{j} + n \mathbf{k}, \quad (5)$$

onde, \mathbf{i} , \mathbf{j} e \mathbf{k} são vetores unitários canônicos que se encontram ao longo do eixo de coordenadas do espaço \mathcal{R}^3 e ℓ , m e n são os cossenos diretores.

É conveniente definir os vetores unitários \mathbf{r}_{ij} e \mathbf{r}_{ji} , que se encontram conectados pela linha r da Fig.1, dirigidos, respectivamente, do dA_i e dA_j , e vice-versa.

$$\hat{\mathbf{r}}_{ij} = \frac{1}{r} \left[(x_j - x_i) \hat{\mathbf{i}} + (y_j - y_i) \hat{\mathbf{j}} + (z_j - z_i) \hat{\mathbf{k}} \right], \quad (6)$$

$$\hat{\mathbf{r}}_{ji} = \frac{1}{r} \left[(x_i - x_j) \hat{\mathbf{i}} + (y_i - y_j) \hat{\mathbf{j}} + (z_i - z_j) \hat{\mathbf{k}} \right]. \quad (7)$$

Os cossenos θ_i e θ_j são dados pelo produto escalar $\mathbf{n}_i \cdot \mathbf{r}_{ij}$ e $\mathbf{n}_i \cdot \mathbf{r}_{ji}$ respectivamente, (Jacob, 1957). Com esses produtos internos e as equações (5) - (7), tem-se que

$$\cos \theta_i = \frac{\ell_i (x_j - x_i) + m_i (y_j - y_i) + n_i (z_j - z_i)}{r}, \quad (8)$$

$$\cos \theta_j = \frac{\ell_j(x_i - x_j) + m_j(y_i - y_j) + n_j(z_i - z_j)}{r} \quad (9)$$

3.2.1 Determinação do vetor normal

Como é de interesse construir um vetor no espaço tridimensional que é perpendicular a dois vetores, define-se uma multiplicação vetorial que produz um vetor resultante como produto, mas que é aplicável somente ao espaço tridimensional.

Sendo u e v vetores no espaço tridimensional, então se pode determinar o produto vetorial $u \wedge v$ e o vetor normal da superfície e este vetor resultante.

3.3 Determinação do dA_i e dA_j

Para determinar o elemento de área dA_i e dA_j de cada triângulo, utiliza-se novamente o produto vetorial, a partir do mesmo procedimento descrito na determinação do vetor normal (Anton, 2001). Sendo assim a área de cada triângulo é dada por:

$$A = \frac{1}{2} \|n\| = \frac{1}{2} \|u \wedge v\| \quad (10)$$

Conhecendo-se os cossenos θ_i e θ_j , a distância r , o dA_i e dA_j , pode-se calcular o fator de forma pela Eq. (4), entretanto tem-se o problema de 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 o plano xy e o algoritmo gerador de malhas será acionado. Logo após as superfícies sofrerão outra rotação, voltando à posição anterior às rotações.

4. Rotação das superfícies

4.1 Determinação do ângulo de rotação

Para determinar o ângulo α formado entre a superfície avaliada e o plano xy , encontra-se a normal do plano xy , usa-se o produto escalar, como:

$$\theta = \arccos \frac{u \cdot v}{\|u\| \cdot \|v\|} \quad (11)$$

onde u é a normal da superfície considerada e v a normal do plano xy e θ é o ângulo formado entre os vetores normais, entretanto o ângulo que se deseja determinar é formado entre a superfície e o plano xy , onde este é dado pelo suplemento do ângulo θ :

$$\alpha = \pi - \theta. \quad (12)$$

4.2 Rotação do plano

Para fazer a rotação das superfícies utilizou-se uma matriz canônica de rotação anti-horária por um ângulo α em torno de um eixo em \mathfrak{R}^3 (Newmann e Sproull, 1979), determinado por um vetor arbitrário unitário com ponto inicial na origem, a matriz M é:

$$M = \begin{bmatrix} a^2(1 - \cos \theta) + \cos \theta & a \cdot b(1 - \cos \theta) - c \cdot \text{sen} \theta & a \cdot c(1 - \cos \theta) + b \cdot \text{sen} \theta \\ a \cdot b(1 - \cos \theta) + c \cdot \text{sen} \theta & b^2(1 - \cos \theta) + \cos \theta & b \cdot c(1 - \cos \theta) - a \cdot \text{sen} \theta \\ a \cdot c(1 - \cos \theta) - b \cdot \text{sen} \theta & b \cdot c(1 - \cos \theta) + a \cdot \text{sen} \theta & c^2(1 - \cos \theta) + \cos \theta \end{bmatrix} \quad (13)$$

A maneira mais comum de descrever um eixo de rotação geral é especificando um vetor não nulo u com ponto inicial na origem e direção ao longo do eixo de rotação.

Fazendo-se a rotação anti-horária em torno do eixo x positivo pelo ângulo α e, portanto, utilizando-se a matriz M , Eq. (13), obtém-se:

$$M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\text{sen}\theta \\ 0 & \text{sen}\theta & \cos\theta \end{bmatrix}. \quad (14)$$

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

5. Geração da malha

5.1 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 mantêm 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. Podendo assim calcular a área de qualquer superfície, pela soma das áreas de cada elemento da malha. Para realizar a triangulação foi necessário trazer todas as faces para um plano comum, pois a biblioteca responsável pela geração das malhas de triângulos, *Triangle*, trabalha apenas em ambientes de duas dimensões. Essa rotação foi implementada na classe *rotateclass*, que possui os métodos necessários para retornar os pontos para a classe da face.

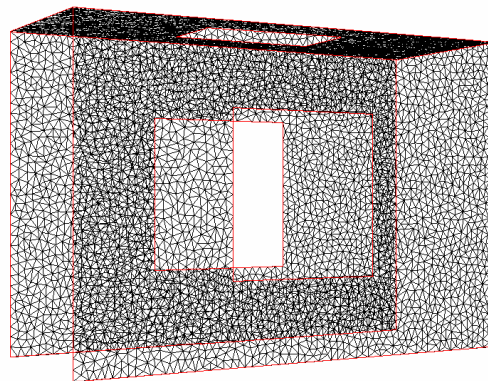
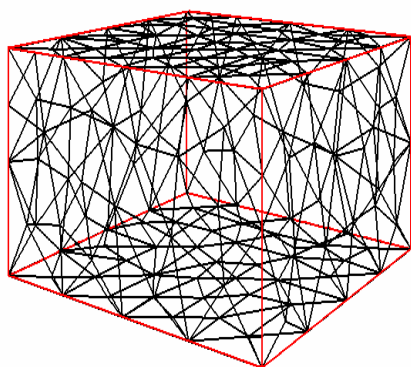
A biblioteca Triangle (Shewchuk, 2003) faz toda a parte de geração da malha utilizando os princípios de Delaunay no artigo Shewchuk (2002) e Shewchuk (1996) podem ser encontradas as informações sobre como é feita a triangulação das superfícies e o critério de escolha dos vértices para a formação dos triângulos. Para realizar uma triangulação existem vários meios de restringir o tamanho da malha, para esse trabalho foi estabelecida a restrição pela área de cada triângulo e ângulos maiores que 20 graus.

5.2 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 janelas.

Figura 2. Visualização de triangulações de superfícies.

6. Resultados e Discussões

Nesta seção apresentam-se os resultados para placas paralelas e iguais, placas quadradas paralelas concêntricas e com tamanhos diferentes e placas com um lado em comum formando um ângulo Φ .

6.1 Placas paralelas e iguais e placas quadradas paralelas concêntricas com tamanhos diferentes

Segundo Howell (1982), o fator de forma para placas paralelas iguais de área A_1 e A_2 , dimensões $b \times a$ e distâncias de c , e para placas paralelas concêntricas, quadradas e de tamanhos diferentes, com área A_1 e A_2 e distanciadas de c , conforme Fig. 3 (a) e Fig. 3 (b) são dados pelas equações (15) e (16) respectivamente, que seguem logo abaixo.

As análises em relação ao refinamento da malha e o erro relativo, foram feitas através dos resultados obtidos com o código ViewFactor-LST e os resultados originados das equações (15) e (16) (Howell, 1982), apresentados na Tab. 1. Essa análise pode ser visualizada na Fig. 4 para placas paralelas e iguais e na Fig. 5 para placas concêntricas quadradas e de tamanhos diferentes. As relações X e Y da Eq. (15) foram alteradas para 10; 4; 1; 0,4 e 0,1 e as relações A e B da Eq. (16) também foram alteradas para 10; 4; 1; 0,4 e 0,1, dessa forma verificam-se como elas interferem nos resultados obtidos com o algoritmo.



(a) Esquema de placas paralelas e iguais.

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

Figura 3. Esquemas de placas paralelas.

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

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

e

$$F_{1-2} = \frac{1}{\pi A^2} \left\{ \ln \left(\frac{[A^2(1+B^2)+2]^2}{(Y^2+2)(X^2+2)} \right) + (Y^2+4)^{1/2} \left[Y \tan^{-1} \frac{Y}{(Y^2+4)^{1/2}} - X \tan^{-1} \frac{X}{(Y^2+4)^{1/2}} \right] + \right. \\ \left. (X^2+4)^{1/2} \left[X \tan^{-1} \frac{X}{(X^2+4)^{1/2}} - Y \tan^{-1} \frac{Y}{(X^2+4)^{1/2}} \right] \right\}, \quad (16)$$

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

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

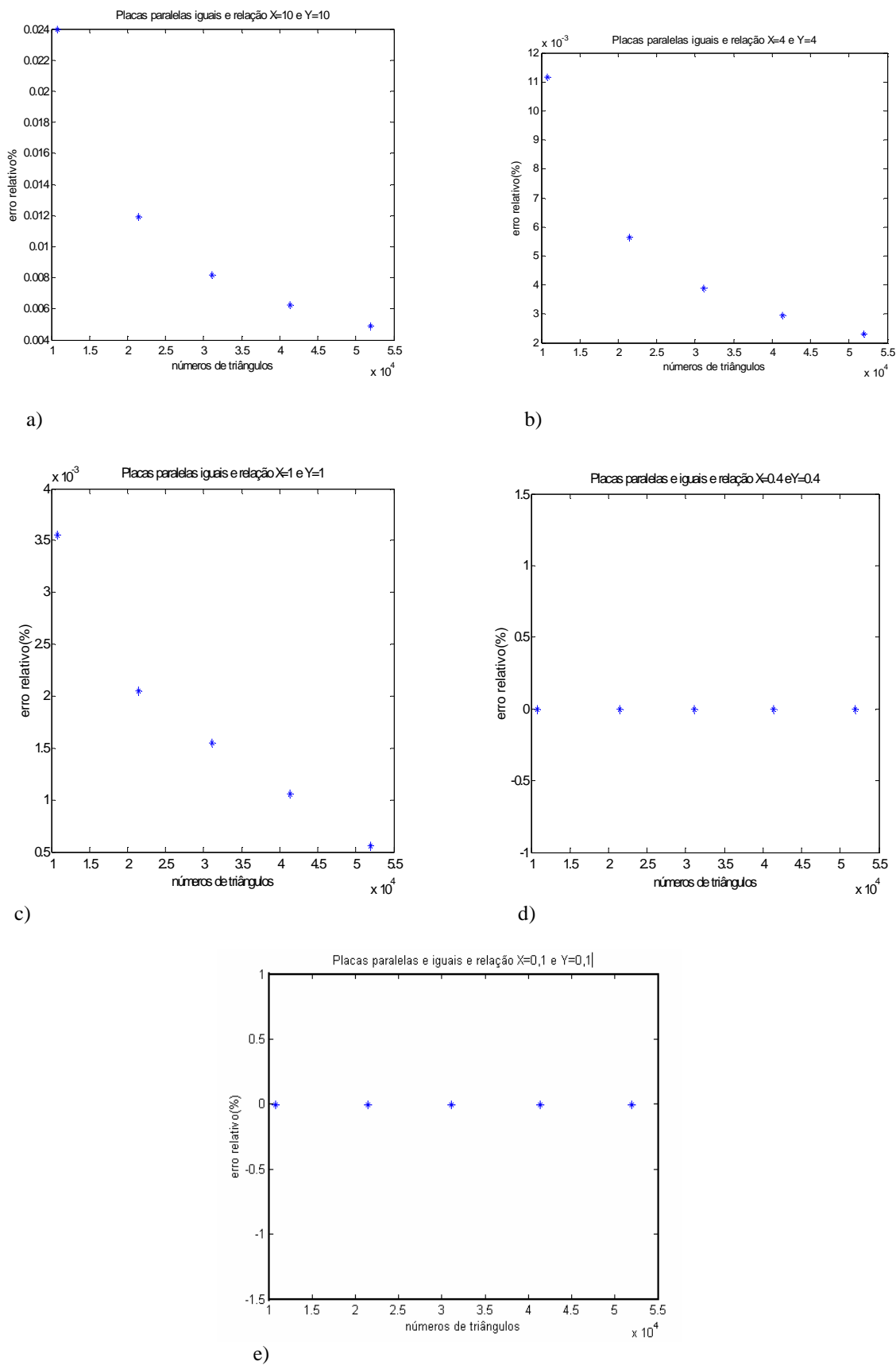


Figura 4. Conforme geometria ilustrada na Fig. (a) Relação X=10 e Y=10; (b) Relação X=4 e Y=4; (c) Relação X=1 e Y=1; (d) Relação X=0,4 e Y=0,4 e (e) Relação X=0,1 e Y=0,1.

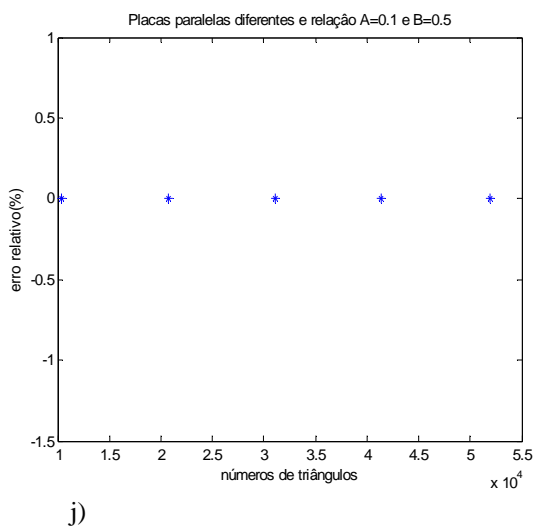
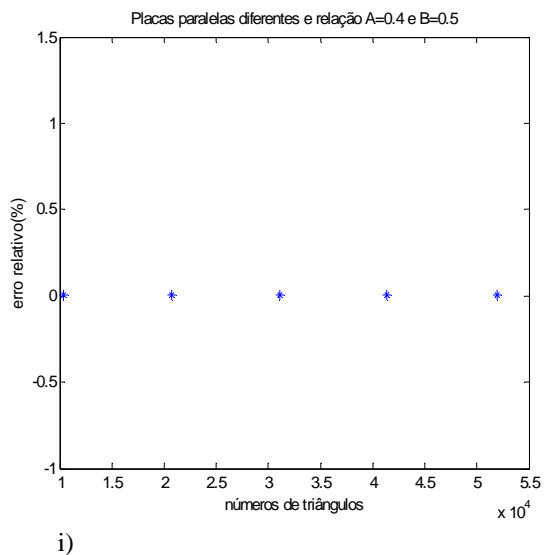
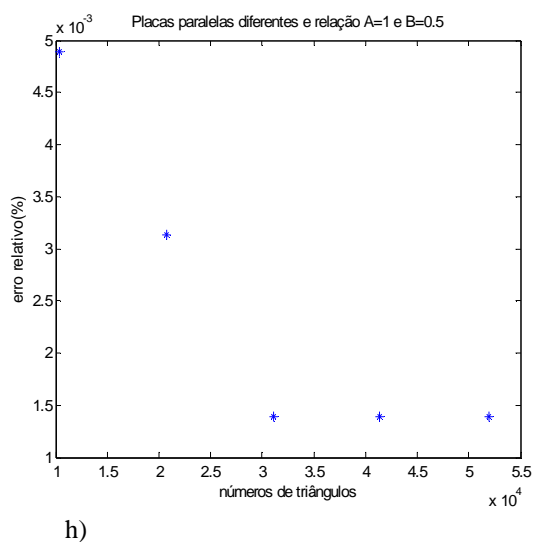
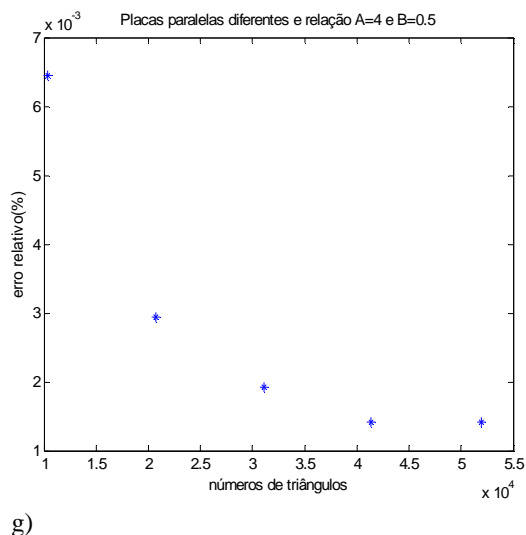
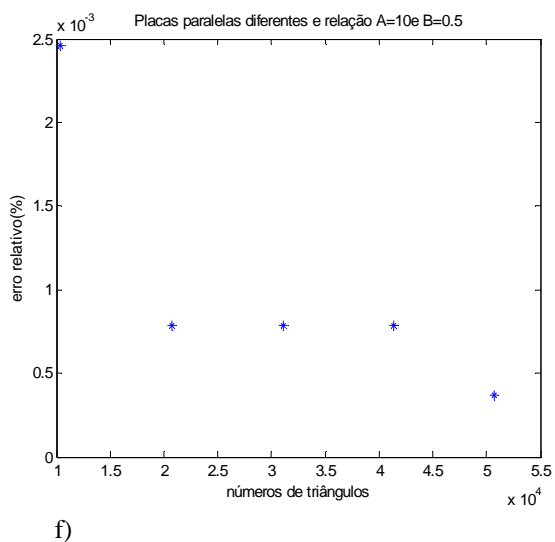


Figura 5. Conforme geometria ilustrada na Fig. 3(b). (f) Relação A=10 e B=10; (g) Relação A=4 e B=4; (h) Relação A=1 e B=1; (i) Relação A=0,4 e B=0,4 e (j) Relação A=0,1 e B=0,1.

Nota-se, pela análise das figuras 4 e 5 que os resultados obtidos com o ViewFactor-LST, apresentam uma excelente concordância quando comparados com os dados originados das equações (15) e (16), apresentados na Tab. 1. O erro relativo nos casos em que as relações são iguais à 0,4 e 0,1 foi nulo, enquanto para as demais relações esse erro manteve-se inferior à 1%.

Tabela 1. Resultados obtidos com as equações (15) e (16)

| Relação | F_{1-2} para placas como Fig. 3 |
|---------------|-----------------------------------|
| X=10 e Y=10 | 0,826995 |
| X=4 e Y=4 | 0,632036 |
| X=1 e Y=1 | 0,199825 |
| X=0,4 e Y=0,4 | 0,046137 |
| X=0,1 e Y=0,1 | 0,003162 |
| Relação | F_{1-2} para placas como Fig. 4 |
| A=10 e B=10 | 0,239392 |
| A=4 e B=4 | 0,198613 |
| A=1 e B=1 | 0,057115 |
| A=0,4 e B=0,4 | 0,011946 |
| A=0,1 e B=0,1 | 0,007958 |

6.2 Placas com um lado em comum formando um ângulo Φ qualquer.

Segundo Howell (1982), tem-se que, considerando o ângulo entre as placas de $\Phi=90^\circ$ e um ângulo qualquer, conforme Figuras 6 (a) e 6 (b), respectivamente, foi feita uma comparação entre os resultados originados das equações (17) e (18), apresentados na Tab. 2 e os obtidos com o algoritmo. A comparação pode ser visualizada na Fig. 7.



(a) Esquema para placas perpendiculares.

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

Figura 6. Esquema de placas com um lado em comum.

$$F_{1-2} = \frac{1}{W\pi} \left(W \tan^{-1} \frac{1}{W} + H \tan^{-1} \frac{1}{H} - \sqrt{H^2 + W^2} \tan^{-1} \sqrt{\frac{1}{H^2 + W^2}} + \frac{1}{4} \ln \left\{ \frac{(1+W^2)(1+H^2)}{1+W^2+H^2} \left[\frac{W^2(1+W^2+H^2)}{(1+W^2)(W^2+H^2)} \right]^{W^2} \left[\frac{H^2(1+H^2+W^2)}{(1+H^2)(H^2+W^2)} \right]^{H^2} \right\} \right), \quad (17)$$

onde $H=h/\ell$, $W=w/\ell$.

$$\begin{aligned}
 F_{1-2} = & -\frac{\sin 2\phi}{4\pi B} \left[AB \sin \theta + \left(\frac{\pi}{2} - \phi \right) (A^2 + B^2) + B^2 \tan^{-1} \left(\frac{A - B \cos \phi}{B \sin \phi} \right) + A^2 \tan^{-1} \left(\frac{B - A \cos \phi}{A \sin \phi} \right) \right] \\
 & + \frac{\sin^2 \phi}{4\pi B} \left\{ \left(\frac{2}{\sin^2 \phi} - 1 \right) \ln \left[\frac{(1 + A^2)(1 + B^2)}{1 + C} \right] + B^2 \ln \left[\frac{B^2(1 + C)}{(1 + B^2)C} \right] + A^2 \ln \left[\frac{A^2(1 + A^2)^{\cos 2\phi}}{C(1 + C)^{\cos 2\phi}} \right] \right\} \\
 & + \frac{1}{\pi} \tan^{-1} \left(\frac{1}{B} \right) + \frac{A}{\pi B} \tan^{-1} \left(\frac{1}{A} \right) - \frac{\sqrt{C}}{\pi B} \tan^{-1} \left(\frac{1}{\sqrt{C}} \right) \\
 & + \frac{\sin \phi \sin 2\phi}{2\pi B} AD \left[\tan^{-1} \left(\frac{A \cos \phi}{D} \right) + \tan^{-1} \left(\frac{B - A \cos \phi}{D} \right) \right] \\
 & + \frac{\cos \phi}{\pi B} \int_0^B \sqrt{1 + \xi^2 \sin^2 \phi} \left[\tan^{-1} \left(\frac{\xi \cos \phi}{\sqrt{1 + \xi^2 \sin^2 \phi}} \right) + \tan^{-1} \left(\frac{A - \xi \cos \phi}{\sqrt{1 + \xi^2 \sin^2 \phi}} \right) \right] d\xi,
 \end{aligned} \tag{18}$$

onde $A = a/c$ $B = b/c$ $C = A^2 + B^2 - 2AB \cos \Phi$ $D = (1 + A^2 \sin^2 \Phi)^{1/2}$.

Tabela 2. Resultados obtidos com as equações (17) e (18)

| Ângulo | Relação A=10 | Relação A=4 | Relação A=1 | Relação A=0,4 | Relação A=0,1 |
|--------|--------------|-------------|-------------|---------------|---------------|
| 30° | 0.728385 | 0.690387 | 0.619028 | 0.394538 | 0.235961 |
| 45° | 0.602836 | 0.560160 | 0.483347 | 0.280023 | 0.160116 |
| 60° | 0.485586 | 0.443475 | 0.370905 | 0.202035 | 0.112663 |
| 90° | 0,281888 | 0,250320 | 0,200044 | 0,101359 | 0,055024 |
| 120° | 0.128098 | 0.111512 | 0.086615 | 0.042260 | 0.022639 |
| 135° | 0.072612 | 0.062773 | 0.048310 | 0.023305 | 0.012436 |

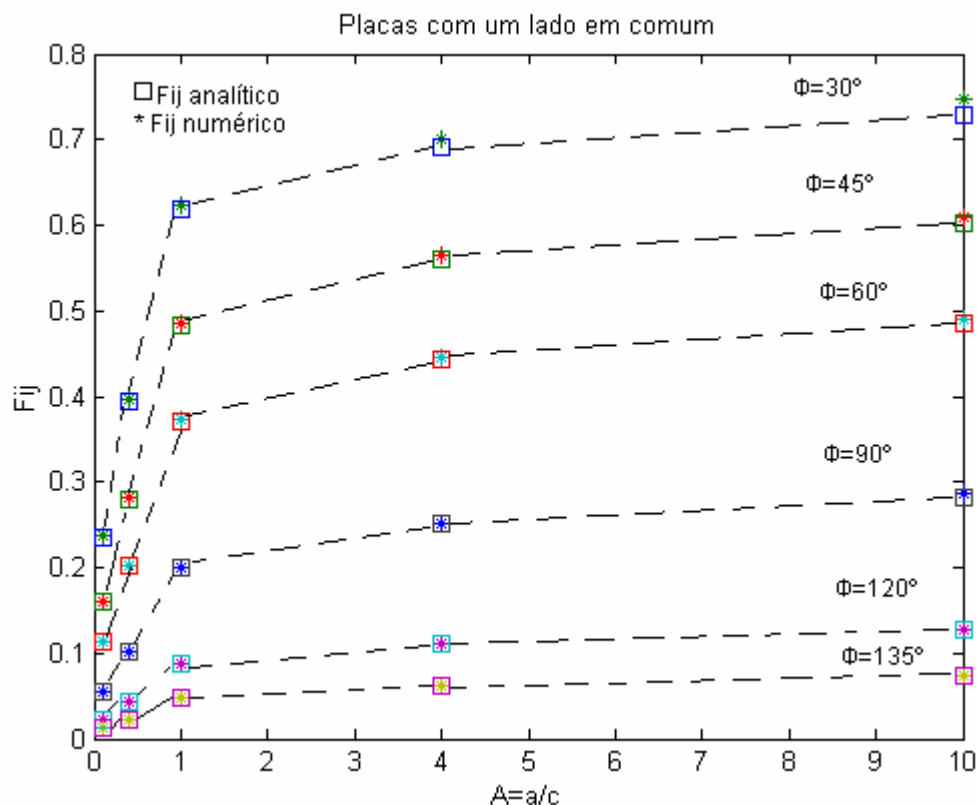


Figura 7. Comparação do fator de forma obtido pelo ViewFactor-LST e o tabelado, com aproximadamente 4×10^4 triângulos.

Através da Fig. 7, 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 $\Phi=135^\circ$ e relações A e B da Eq. (18) 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 $\Phi=135^\circ$, mas também com os ângulos 120° e 90° .

7. Conclusão

Descreveu-se um algoritmo capaz de determinar numericamente o fator de forma entre superfícies quaisquer, indicado na Fig. 1, para qualquer ângulo θ_i e θ_j . Os resultados fornecidos pelo algoritmo coincidem com os valores obtidos pelas equações (15)-(18), promovendo uma maneira alternativa de se obter esses valores e ainda ampliando possibilidades 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), considerar a presença de aberturas e obstruções nas superfícies

8. Referências

- Anton, H. and Rorres, C., 2001, "Álgebra linear com aplicações", trad. Claus Ivo Doering.-8.ed.-Porto Alegre: Bookman, Brazil.
- Burns, P. J., 1893, "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.
- 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 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 with 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., 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in " Applied Computational Geometry: Towards Geometric Engineering" (Ming C. Lin and Dinesh MANocha, editors), volume 1148 of Lecture Notes in Computer Science, pages 230-222, Springer-Verlag, Berlin, May 1996.
- 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. 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 factro program", Technical report, University of California, Lawrence Livermore National Laboratory.

ALGORITHM FOR NUMERICAL EVALUATION OF VIEW FACTOR

Lilian Dobrowolski de Carvalho Augusto

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR, 80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
liliandoca@gmail.com

Bruno Giacomet

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR, 80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
handriel@gmail.com

Marluz Jonsson

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR, 80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
marluzjonsson@bol.com.br

Nathan Mendes

Pontifícia Universidade Católica do Paraná.
Rua Imaculada Conceição 1155 Curitiba-PR, 80215-901-Brasil
Laboratório de Sistemas Térmicos-LST
nathan.mendes@pucpr.br

Liliana Madalena Gramani Cumin

Departamento de Matemática, Universidade Federal do Paraná.
Caixa Postal 19044,81531-990 Curitiba- PR – Brasil
gramani@mat.ufpr.br

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.*