

ESCOLA POLITÉCNICA
UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE ENGENHARIA MECÂNICA

PROJETO DE FORMATURA

1992

PROGRAMA COMPUTACIONAL

PARA

O CÁLCULO DE

ÁREAS DIRETAS E GLOBAIS DE TROCA DE CALOR

EM UMA FORNALHA CILÍNDRICA

ALUNO : MARCOS TAUE

Nº USP : 8212925

CURSO : ENGENHARIA MECÂNICA

OPÇÃO : ENERGIA E FLUIDOS

ORIENTADOR : PROF. DOUTOR JURANDIR I. YANAGIHARA

Índice

Parte A

I - Introdução (pág. 2)

II - Fornalha Cilíndrica (pág. 4)

II.1 - Discretização da fornalha. (pág. 6)

II.2 - Pares possíveis de troca de calor na fornalha. (pág. 8)

II.3 - Sistema de coordenadas na fornalha. (pág. 10)

III - Fundamentação Teórica (pág. 12)

III.1 - Área Direta de Troca de Calor. (pág. 13)

III.2 - Área Global de Troca de Calor. (pág. 17)

IV - Método de Monte Carlo (pág. 21)

IV.1 - Descrição. (pág. 22)

IV.2 - Roteiro. (pág. 22)

IV.3 - Comentários (pág. 24)

IV.4 - Comparações de resultados obtidos com o Método de Monte Carlo e com solução analítica. (pág. 24)

IV.5 - Análise de Sensibilidade do Método de Monte Carlo. (pág. 27)

V - Cálculo das Áreas Diretas de Troca de Calor para os elementos da fornalha. (pág 38)

V.1 - Expressões das áreas diretas de troca de calor para os 36 casos possíveis de troca de calor.

VI - Programa Computacional (pág.43)

VI.1 - Ferramentas (pág. 44)

VI.2 - Estrutura (pág.44)

VI.3 - Descrição dos Módulos (pág. 45)

VII - Apêndice (pág. 51)

VIII - Bibliografia (pág 58)

Parte B

Listagem dos Módulos do Programa Computacional (pág. 61)

I. ÁreaDireta (pág. 62)

II. Varredura (pág. 87)

III. Gauss (pág. 95)

IV. ÁreaGlobal (pág. 101)

V. Library (pág.113)

Parte C

Resultados (pág 129)

Parte A

I

Introdução

Cálculo de áreas diretas e globais de troca de calor em uma fornalha cilíndrica.

I - Introdução

O trabalho consiste em desenvolver um programa computacional para calcular as áreas diretas e globais de troca de calor entre os elementos de superfícies e volumes devidamente discretizados de uma fornalha cilíndrica. Este trabalho fornece resultados que possibilitarão os cálculos das trocas de calor entre chamas e superfícies que as envolvem, constituindo parte de um dos módulos que integram um trabalho de modelagem e simulação da fornalha descrita acima, iniciado no Agrupamento de Engenharia Térmica do Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT.

II

Fornalha

Cilíndrica

II. Fornalha Cilíndrica

O Agrupamento de Engenharia Térmica do Instituto de Pesquisas Tecnológicas possui entre outras capacitações o estudo das trocas de calor por radiação entre as chamas e as superfícies envolventes, possuindo para isto um laboratório equipado com uma fornalha cilíndrica horizontal de teste. Esta fornalha é constituída de duas partes acopláveis, cada uma com 1,0 m de diâmetro interno e 2,0 m de comprimento, sendo uma constituída por dupla carcaça metálica resfriada a água e a outra com revestimento interno de refratário sílico-aluminoso monolítico. A parte resfriada é formada por compartimentos independentes, onde para cada um deles é possível a medição das vazões e temperaturas na entrada e saída, permitindo assim a determinação dos fluxos de calor incidentes na superfície ao longo da fornalha.

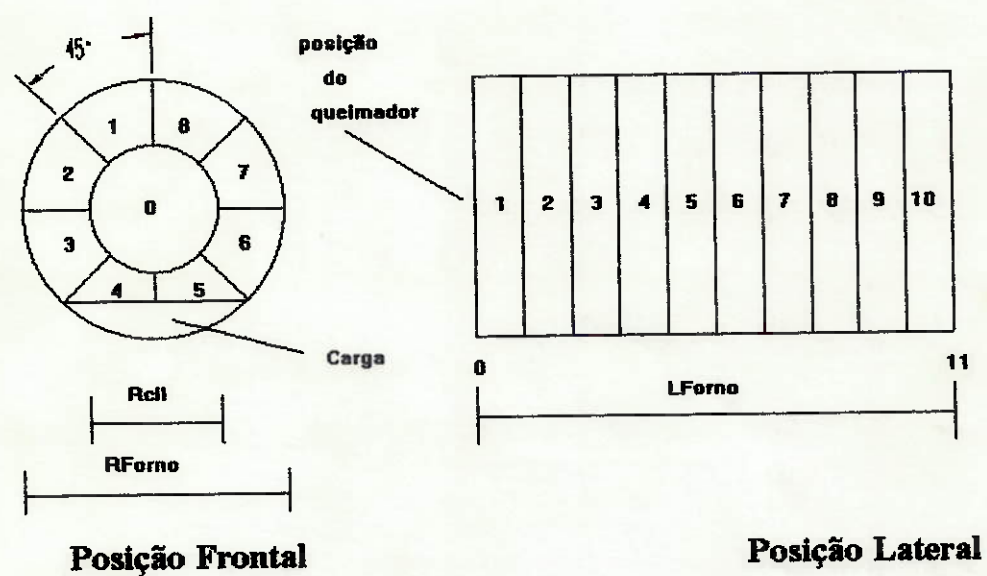
Através de orifícios e frestas na fornalha, é possível fazer o monitoramento das chamas através de aparelhos especiais tais como : pirômetro de sucção, sonda para coleta de amostra de gás, sonda de entalpia entre outras.

Entre os ensaios já realizados tem-se : experimentos com carvão mineral de 50% de cinzas, turfa e carvão de casca de coco de babaçu pulverizados, gás pobre gerado em gaseificação de lenha, alcatrão obtido do carvoejamento de casca de babaçu, emulsão de resíduo asfáltico (RASf) e óleo combustível tipo 3A, usando ar e oxigênio como comburentes.

A equipe de pesquisadores, deste Instituto, aliados a amplos conhecimentos práticos e teóricos sobre o fenômeno da radiação térmica e por uma automotivação, iniciaram em 1991 um trabalho de modelagem e simulação da fornalha de teste, objetivando um processo de interação entre o experimental e o teórico (de um lado a constatação teórica e por outro lado o aprimoramento do modelo).

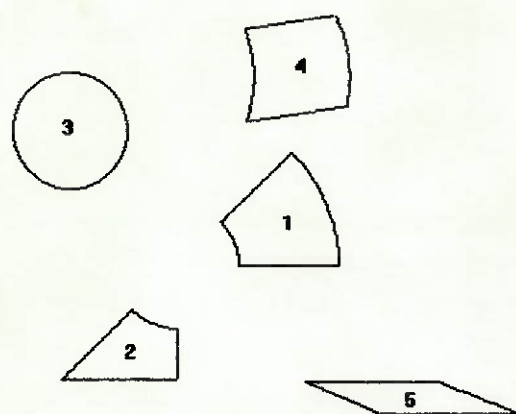
II.1 - Discretização da Fornoalha

No modelo, para efeito de simplificação, considera-se a fornalha como sendo um cilindro de diâmetro D_{forno} e comprimento L_{forno} .

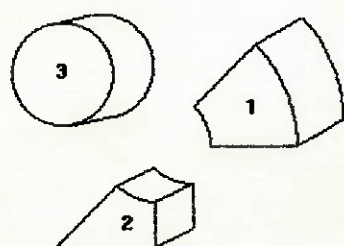


A fornalha foi discretizada de tal maneira a identificar satisfatoriamente regiões distintas e de interesse, tais como a região onde se desenvolve a chama e a região da carga (local onde se depositam materiais para a fundição), desta forma a fornalha foi segmentada longitudinalmente em 10 "porções" iguais e para cada "porção" dividiu-se em 9 "fatias" conforme mostram as figuras acima.

Podem-se identificar 5 tipos diferentes de elementos de superfícies assim classificadas:



1. Superfície Regular das Tampas:
2. Superfície Irregular das Tampas.
3. Superfície Circular das Tampas.
4. Superfície Cilíndrica Lateral.
5. Superfície do Sorvedouro.



Para os elementos de volume podem ser identificados:

1. Volume Regular.
2. Volume Irregular.
3. Volume do Cilindro Central.

Desta discretização obtém-se 98 elementos de superfícies (18 nas tampas e 80 nas laterais) e 90 elementos de volumes, totalizando 188 zonas de troca de calor.

II.2 - Pares possíveis de elementos de troca de calor na fornalha

Os possíveis pares de troca de calor entre dois elementos da fornalha são os seguintes:

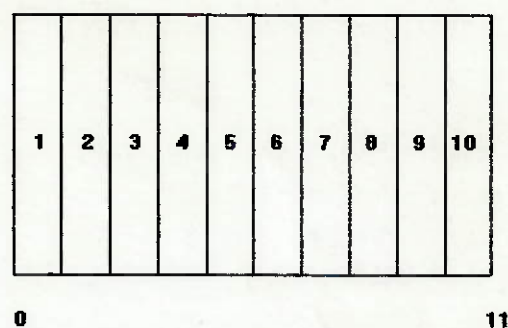
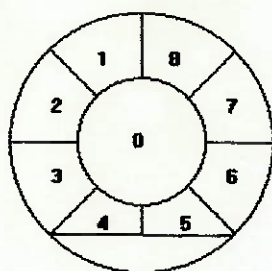
- 1. Volume Regular - Volume Regular**
- 2. Volume Regular - Volume Irregular**
- 3. Volume Regular - Volume do Cilindro Central**
- 4. Volume Regular - Superfície do Sorvedouro**
- 5. Volume Regular - Superfície Regular das Tampas**
- 6. Volume Regular - Superfície Irregular das Tampas**
- 7. Volume Regular - Superfície do Cilindro Central das Tampas**
- 8. Volume Regular - Superfície Cilíndrica Lateral**
- 9. Volume Irregular - Volume Irregular**
- 10. Volume Irregular - Volume do Cilindro Central**
- 11. Volume Irregular - Superfície do Sorvedouro**
- 12. Volume Irregular - Superfície Regular das Tampas**
- 13. Volume Irregular - Superfície Irregular das Tampas**
- 14. Volume Irregular - Superfície do Cilindro Central**
- 15. Volume Irregular - Superfície Cilíndrica Lateral**
- 16. Volume do Cilindro Central - Volume do Cilindro Central**
- 17. Volume do Cilindro Central - Superfície do Sorvedouro**
- 18. Volume do Cilindro Central - Superfície Regular das Tampas**
- 19. Volume do Cilindro Central - Superfície Regular das Tampas**
- 20. Volume do Cilindro Central - Superfície do Cilindro Central**
- 21. Volume do Cilindro Central - Superfície Cilíndrica Lateral**
- 22. Superfície do Sorvedouro - Superfície do Sorvedouro**
- 23. Superfície do Sorvedouro - Superfície Regular das Tampas**

24. Superfície do Sorvedouro - Superfície Irregular das Tampas
25. Superfície do Sorvedouro - Superfície do Cilindro Central
26. Superfície do Sorvedouro - Superfície Cilíndrica Lateral
27. Superfície Regular das Tampas - Superfície Regular das Tampas
28. Superfície Regular das Tampas - Superfície Irregular das Tampas
29. Superfície Regular das Tampas - Superfície do Cilindro Central
30. Superfície Regular das Tampas - Superfície Cilíndrica Lateral
31. Superfície Irregular das Tampas - Superfície Irregular das Tampas
32. Superfície Irregular das Tampas - Superfície do Cilindro Central
33. Superfície Irregular das Tampas - Superfície Cilíndrica Lateral
34. Superfície do Cilindro Central - Superfície do Cilindro Central
35. Superfície do Cilindro Central - Superfície Cilíndrica Lateral
36. Superfície Cilíndrica Lateral - Superfície Cilíndrica Lateral

II.3 - Sistemas de coordenadas na fornalha

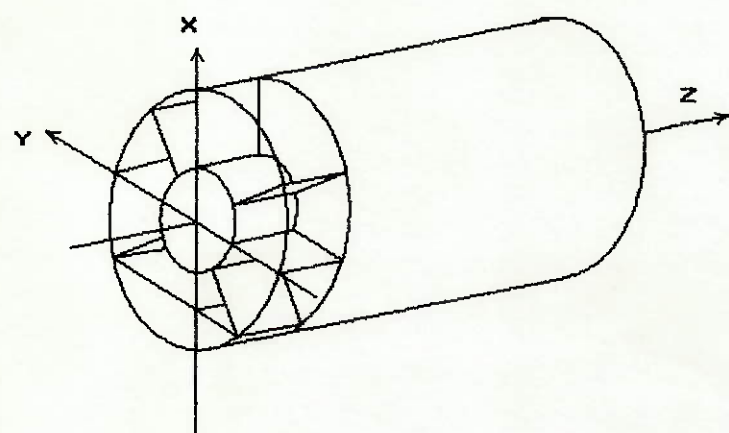
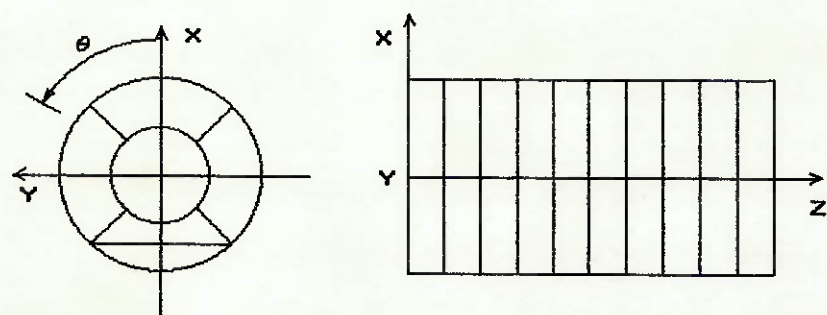
II.3.1 - Identificação dos elementos de superfícies e volumes na fornalha.

Para efeitos de simplificação e conveniência, adotou-se duas formas de identificar os elementos na fornalha, uma através da enumeração das superfícies (0 a 97) e volumes (0 a 89); e outra através da informação da seção e setor (12 seções ao longo do comprimento do forno e para cada seção 9 setores (0 a 8)), conforme as figuras abaixo.



II.3.2 - Definição e orientação de um sistema de coordenadas na fornalha.

Para efeitos de cálculos, utilizou-se conjuntamente os sistemas de coordenadas cartesianas (a origem sobre o eixo de simetria da fornalha na "tampa" onde se posiciona o queimador) e um sistema de coordenadas polares com origem do ângulo no eixo x e crescente no sentido anti-horário conforme as figuras abaixo.



III

Fundamentação

Teórica

III - Fundamentação Teórica

III.1 - Área direta de troca de calor

III.1.1 - Definição

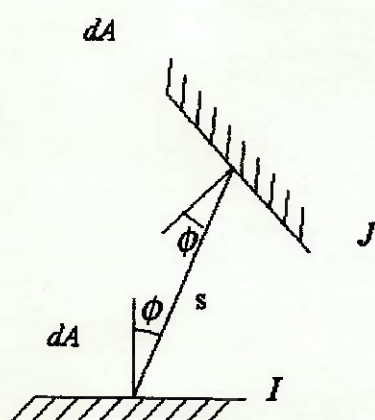
Dados dois corpos genéricos i e j (superfícies ou volumes), define-se área direta de troca de calor como sendo um valor que multiplicado pelo poder emissivo de um dos corpos obtém-se a fração do fluxo de energia emitida por este corpo e que atinge o outro.

Por exemplo entre duas superfícies :

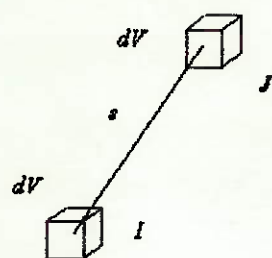
$$\dot{Q}_{i \rightarrow j} = \epsilon_i s_j \cdot E_i$$

As área diretas podem ser classificadas em três tipos :

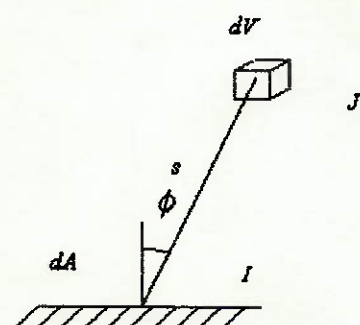
1. Entre duas superfícies (sisj).



2. Entre dois volumes (gigj).



3. Entre uma superfície e um volume.



Observa-se que a unidade das áreas diretas de troca de calor é m^2 .

Pode-se, a partir das áreas diretas, calcular o fluxo líquido trocado entre dois corpos, por exemplo, entre duas superfícies i e j .

$$Q_{\text{líquido}} = \epsilon_i s_j (E_i - E_j) = \epsilon_j s_i (E_i - E_j)$$

III.1.2 - Expressões genéricas para as áreas diretas de troca de calor.

As expressões das áreas diretas podem ser obtidas considerando-se o meio participante, ou seja, a existência dos fenômenos de absorção e espalhamento devido à presença do meio.

Para efeitos de simplificação do modelo, considerou-se neste trabalho algumas hipóteses : existência apenas do fenômeno de absorção; radiação difusa (intensidade da radiação independe do comprimento de onda) e equilíbrio dos elementos.

As expressões para as áreas diretas de troca de calor são as seguintes:

1. Entre duas superfícies

$$sisj = \int \int \frac{dA_i \cos \phi_i \cdot dA_j \cos \phi_j}{\pi \cdot s^2} \cdot \tau(s)$$

2. Entre uma superfície e um volume

$$gigj = \int \int \frac{Ak_i \cdot dV_i \cdot Ak_j \cdot dV_j}{\pi \cdot s^2} \cdot \tau(s)$$

3. Entre dois volumes

$$sigj = \int \int \frac{dA_i \cos \phi_i \cdot Ak_j \cdot dV_j}{\pi \cdot s^2} \cdot \tau(s)$$

Nomenclatura:

dA : elemento de superfície.

dV : elemento de volume.

ϕ : ângulo entre a reta normal à superfície e a reta que une os dois elementos i e j .

Ak : coeficiente de atenuação.

s : distância entre os dois elementos i e j .

τ : fração transmitida.

Das expressões genéricas para as área diretas dadas acima, verifica-se uma dificuldade relativamente grande em se obter algebricamente valores de integrais

mútiplas de funções, muitas vezes, bastante complexas; para tanto, recorreu-se a um método estocástico de cálculo de integrais chamado Método de Monte Carlo.

III.2 - Área global de troca de calor

III.2.1 - Definição

A área global de troca de calor possui uma definição semelhante à área direta, diferindo apenas no fato de que agora leva-se em consideração os efeitos da presença dos demais elementos de troca (superfícies ou volumes).

Da mesma forma as áreas globais se classificam em três tipos : superfície-superfície (SiSj), superfície-volume (SiGj) e volume-volume (GiGj). Aqui também a unidade é m^2 .

As áreas globais são identificadas com os mesmos símbolos das áreas diretas, porém em letras maiúsculas.

III.2.2 - Expressões para as áreas globais de troca de calor

Para um conjunto de n superfícies e m volumes de uma cavidade, pode-se fazer o balanço de radiação para uma superfície genérica i considerando-se o meio participante :

$$A_i \cdot W_i = A_i \cdot \varepsilon_i E_s + \rho_i \cdot \left(\sum_j s_{ji} W_j + \sum_j g_{ji} E_g \right)$$

Onde : A : área.

W : fluxo de radiação que abandona uma superfície.

ε : emissividade da superfície.

ρ : refletividade da superfície.

E_s : poder emissivo da superfície negra.

E_g : poder emissivo do volume negro.

Significado de cada parcela na expressão:

$A_i \cdot W_i$: quantidade de calor que abandona a superfície i .

$A_i \cdot \varepsilon_i \cdot E_s$: quantidade de calor emitida pela superfície i .

$\sum_j s_{j/i} \cdot W_j$: quantidades de calor que abandonando todas as outras superfícies

j , incidem na superfície i após parcialmente absorvidas pelo meio.

$\sum_j g_{j/i} \cdot E_g$: quantidades de calor que emitidas de cada um dos volumes do

meio incidem na superfície i após serem parcialmente absorvidas pelo meio.

Rearranjando a equação acima, obtém-se:

$$\sum_j (s_{j/i} - \delta_{ij} \frac{A_i}{\rho_i}) \cdot W_j = - \frac{A_i \cdot \varepsilon_i}{\rho_i} \cdot E_s - \sum_j g_{j/i} \cdot E_g$$

com :

$$\delta_{ij} = 1 \quad \text{se } i = j$$

$$\delta_{ij} = 0 \quad \text{se } i \neq j$$

Colocando-se na forma matricial tem-se que :

$$\begin{bmatrix} s_{11} - \frac{A_1}{\rho_1} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} - \frac{A_2}{\rho_2} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{nn} - \frac{A_n}{\rho_n} \end{bmatrix} \cdot \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} = \begin{bmatrix} -\frac{A_1 \cdot \varepsilon_1}{\rho_1} \cdot E_{s1} & -s_{11} \cdot E_{g1} & \dots & -s_{1n} \cdot E_{gn} \\ -\frac{A_2 \cdot \varepsilon_2}{\rho_2} \cdot E_{s2} & -s_{21} \cdot E_{g1} & \dots & -s_{2n} \cdot E_{gn} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{A_n \cdot \varepsilon_n}{\rho_n} & \dots & \dots & -s_{nn} \cdot E_{gn} \end{bmatrix}$$

Matriz de Transferência

Vetor Resposta

Matriz de Excitação

Resolvendo-se este sistema de equações lineares, obtém-se os W 's possibilitando o cálculo das trocas líquidas de calor em cada superfície (somente com prévio conhecimento das temperaturas). Como na maioria das vezes as temperaturas não são conhecidas, o sistema de equações acima pode ser utilizado para determinar as áreas globais de troca de calor entre os vários elementos.

As expressões para as áreas globais de troca de calor são dadas a seguir

Superfície-Superfície

$$S_i S_j = \frac{A_j \cdot \epsilon_j}{\rho_j} \cdot (\bar{W}_j^i - \delta_{ij} \cdot \epsilon_i)$$

Onde $\bar{W}_j^i$: corresponde ao fluxo de radiação que deixa a superfície j , considerando a superfície i a única emissora de radiação. É obtido do sistema de equações lineares, fazendo-se todos os $E_g = 0$ e todos os $E_s = 0$ com exceção de E_{si} , de modo que o vetor de excitação fica igual a:

$$\begin{bmatrix} 0 \\ \vdots \\ \frac{A_i \cdot \epsilon_i}{\rho_i} \\ \vdots \\ 0 \end{bmatrix} \cdot E_{si}$$

Superfície-Volume

$$G_i S_j = \frac{A_j \cdot \epsilon_j}{\rho_j} \cdot \bar{W}_j^i$$

Onde $\bar{W}_j^i$ corresponde ao fluxo de radiação que deixa a superfície j . É obtido do sistema de equações lineares supondo-se que apenas o volume de gás i emite

radiação, (todas os outros elementos estão presentes mas não emitem), de modo que o vetor de excitação seja :

$$\begin{bmatrix} s_1 g_i \\ s_2 g_i \\ \vdots \\ s_n g_i \end{bmatrix} \cdot E_g$$

Volume-Volume

$$G_i G_j = g_i g_j + \sum_k (s_k g_j \cdot \bar{W}_k^g) \quad k = 1 \dots n$$

Onde $\bar{W}_k^g$ corresponde ao fluxo de radiação que abandona a superfície k , supondo-se o volume i o único emissor. Obtém-se da mesma forma descrita para superfície-volume.

IV

Método

de

Monte Carlo

IV - Método de Monte Carlo

IV.1 - Descrição

O método de Monte Carlo permite de forma estocástica obter o resultado de integrais de multiplicidade genérica (simples, duplas, triplas...n-uplas), ou integrais de funções cujas primitivas são de difícil obtenção. Tal método se ajusta perfeitamente em termos computacionais, atendendo de forma satisfatória às necessidades deste trabalho.

IV.2 - Roteiro

Resumidamente o método consiste nos seguintes passos descritos abaixo:

Supondo-se que deseje calcular a seguinte integral de uma função genérica de m variáveis independentes.

$$I = \int_{a_1}^{A_1} \int_{a_2}^{A_2} \dots \int_{a_m}^{A_m} f(x_1, x_2, \dots, x_m) dx_1 dx_2 \dots dx_m$$

$$a_i \leq x_i \leq A_i$$

Onde :

x : variáveis independentes.

f : função de x_i .

a_i : limite inferior de x_i .

A_i : limite superior de x_i .

Os passos são os seguintes:

1º Passo : Mudanças de variáveis.

$$a_i \leq x_i \leq A_i$$

$$\varepsilon_i = \frac{x_i - a_i}{A_i - a_i} \quad 0 \leq \varepsilon_i \leq 1$$

$$x_i = a_i + (A_i - a_i) \cdot \varepsilon_i$$

$$\frac{dx_i}{d\varepsilon_i} = A_i - a_i$$

2º Passo : Escrever a integral em função destas novas variáveis.

$$I = (A_1 - a_1) \cdot (A_2 - a_2) \dots (A_m - a_m) \int \int \dots \int f^*(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m) d\varepsilon_1 d\varepsilon_2 \dots d\varepsilon_m$$

3º Passo : Cálculo de um $f^*_{\text{médio}}$

$$f^*_{\text{médio}} = \frac{\sum^n f^*(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)}{N}$$

n : número de conjuntos de variáveis aleatórias utilizados no cálculo da $f^*_{\text{média}}$.

4º Passo : Cálculo final.

$$I = (A_1 - a_1) \cdot (A_2 - a_2) \dots (A_m - a_m) f^*_{\text{médio}} \int \int \dots \int d\varepsilon_1 d\varepsilon_2 \dots d\varepsilon_m$$

$$\text{como :} \quad \int \int \dots \int d\varepsilon_1 d\varepsilon_2 \dots d\varepsilon_m = 1$$

$$I = (A_1 - a_1) \cdot (A_2 - a_2) \dots (A_m - a_m) f^*_{\text{médio}}$$

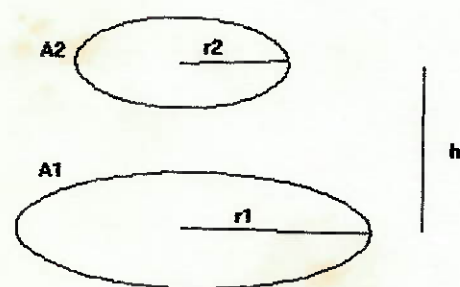
IV.3 - Comentários

O método de Monte Carlo depende fortemente do parâmetro N utilizado no 3º passo, ou seja, o valor da média da função f , é tanto mais representativa à medida em que o valor de N aumenta, tornando o resultado final da integral mais próxima do seu valor com solução analítica. Evidentemente, o aumento de N provoca um aumento também no tempo computacional, ficando a seguinte questão: até que ponto vale a pena aumentar N , sem comprometer o tempo de processamento? Esta questão pode ser respondida de acordo com as necessidades (em relação a precisão requerida) e suas disponibilidades em termos de equipamentos (microcomputador XT,286,386 etc).

IV.4 - Comparações de resultados obtidos com o Método de Monte Carlo e com solução analítica.

Para ilustrar uma aplicação do Método de Monte Carlo no cálculo de integrais e comparar o seu resultado com o da solução analítica, considere a seguinte aplicação :

Deseja-se calcular o fator de forma entre duas placas circulares concêntricas de raios r_1 e r_2 e separadas de uma distância h , conforme mostra a figura abaixo.



A expressão do fator de forma obtido algebricamente é dada pela literatura como sendo :

$$F_{12} = \frac{1}{2} \cdot \left[x - \sqrt{x^2 - 4 \cdot \left(\frac{R_2}{R_1} \right)^2} \right]$$

$$R_1 = \frac{r_1}{h}$$

$$R_2 = \frac{r_2}{h}$$

$$x = 1 + \frac{1 + R_2^2}{R_1^2}$$

Para o caso de $r_1 = 4$, $r_2 = 2$ e $h = 10$ o fator de forma obtido algebricamente é igual a : **0.0334828**.

Através do método obtém-se os seguintes valores:

N=100

$F_{12}=0.03588$

$F_{12}=0.03282$

tempo : 1 s

N=500

$F_{12}=0.03478$

$F_{12}=0.03382$

Tempo : 2 s

N=1000

$F_{12}=0.03361$

$F_{12}=0.03479$

Tempo : 3 s

N=5000

$F_{12}=0.03374$

$F_{12}=0.03347$

Tempo : 16 s

N=10000

$F_{12}=0.03347$

$F_{12}=0.03383$

Tempo : 33 s

N=100000

$F_{12}=0.03356$

$F_{12}=0.03350$

Tempo : 329 s

IV.5 - Análise de sensibilidade do Método de Monte Carlo visando a aplicação no presente trabalho.

A análise de sensibilidade do Método de Monte Carlo visa estudar o comportamento do resultado da integral à medida em que se aumenta o valor do parâmetro N (número de conjuntos de variáveis aleatórias). O objetivo final deste estudo é estabelecer um valor apropriado de N , de modo que este não seja um valor muito grande, o que tornaria "impraticável" em relação ao tempo computacional. (neste trabalho, exigirá a aplicação deste método para aproximadamente 9000 integrais, como será visto a seguir), ao mesmo tempo o valor de N não deverá ser pequeno demais, o que implicaria imprecisões maiores nos resultados. Particularmente neste trabalho, estas exigências não são muito grandes, visto que existem simplificações no desenvolvimento do modelo e que geram imprecisões de maiores grandezas.

No estudo de sensibilidade do método para o cálculo das áreas diretas de troca de calor, foram selecionadas alguns tipos de pares de troca existentes na fornalha, e para cada um deles obteve-se resultados para crescentes valores de N .

Através dos resultados obtidos para diferentes valores de N , como era previsto, à medida em que se aumenta o seu valor, menores são os desvios dos resultados em relação ao valor algébrico, isto pode ser observado através do cálculo do desvio padrão dos resultados obtidos para uma sequência de valores de uma mesma integral para um N fixo.

Análise de Sensibilidade do Método de Monte Carlo

Data : 13 / 02 / 92

Computador : At-286 / Co-Processador Matemático

Cálculo de Áreas Diretas de Troca de Calor

Tipos Estudados : 15

Descrição dos Tipos :		
Tipo		
1	Volume Regular	Volume Regular
2	Volume Irregular	Volume Irregular
3	Volume Regular Central ou Periférico	Volume Irregular
4	Superfície do Sorvedouro	Superfície Cilíndrica
5	Superfície do Sorvedouro	Superfície Regular das Tampas
6	Superfície do Sorvedouro	Superfície Irregular das Tampas
7	Superfície Cilíndrica	Superfície Cilíndrica
8	Superfície Cilíndrica	Superfície Regular das Tampas
9	Superfície Irregular das Tampas	Superfície Cilíndrica
10	Superfície Regular das Tampas	Superfície Regular das Tampas
11	Superfície Irregular das Tampas	Superfície Regular das Tampas
12	Superfície Irregular das Tampas	Superfície Irregular das Tampas
13	Volume Regular	Superfície Sorvedouro
14	Volume Regular	Superfície Cilíndrica
15	Volume Regular	Superfície Regular das Tampas

Tipo 1

N: 10000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	8,636615	8,796811	8,876719	8,473939	8,696021	0,154645
Tempo [s]	30	31	30	31	30,5	0,5

N: 100000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	8,610157	8,432312	8,558374	8,53407	8,533728	0,06468
Tempo [s]	305	307	306	307	306,25	0,829156

N: 1000000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	8,598243	8,592994	8,599428	8,645586	8,609063	0,021225
Tempo [s]	3049	3062	3062	3048	3055,25	6,759253

Tipo: 2

N: 10000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	3,223062	3,390347	3,412359	3,446366	3,368034	0,086046
N_Efetivo	4517	4617	4547	4531	4553	38,44477
N_Efetivo/N	0,4517	0,4617	0,4547	0,4531	0,4553	0,003844
Tempo [s]	25	24	25	24	24,5	0,5

N: 100000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	3,462566	3,478314	3,382072		3,440984	0,04215
N_Efetivo	45912	45754	45615		45760,33	121,3324
N_Efetivo/N	0,45912	0,45754	0,45615		0,343203	0,198151
Tempo [s]	243	244	243		243,3333	0,471405

N: 1000000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	3,460931	3,442304	3,468993	3,455379	3,456902	0,009719
N_Efetivo	455727	455994	456623	455569	455978,3	402,0456

N_Efetivo/N	0,455727	0,455994	0,456623	0,455569		0,455978	0,000402
Tempo [s]	2416	2428	2430	2416		2422,5	6,538348

Tipo: 3

N: 20000

	1	2	3	4		
Valor [10E-5]	6,42771	6,3383	6,34647	6,40966	Media	Desvio Padrao
N_Efetivo	13623	13481	13464	13619	6,380535	0,038788
N_Efetivo/N	0,68115	0,67405	0,6732	0,68095	13546,75	74,50629
Tempo [s]	53	53	53	54	0,677338	0,003725
					53,25	0,433013

N: 200000

	1	2	3	4		
Valor [10E-5]	6,37122	6,38238	6,36488		Media	Desvio Padrao
N_Efetivo	135232	135273	134972		6,372827	0,007234
N_Efetivo/N	0,67616	0,676365	0,67486		135159	133,2842
Tempo [s]	532	533	532		0,506846	0,292628
					532,3333	0,471405

N: 2000000

	1	2	3	4		
Valor [10E-5]	6,37203	6,37402	6,36764	6,36961	Media	Desvio Padrao
N_Efetivo	1350851	1351687	1350064	1351165	6,370825	0,002412
N_Efetivo/N	0,675426	0,675844	0,675032	0,675583	1350942	588,2004
Tempo [s]	5309	5332	5328	5311	0,675471	0,000294
					5320	10,12423

Tipo: 4

N: 20000

	1	2	3	4		
Valor [10E-5]	4,874102	4,843878	4,866787	4,855053	Media	Desvio Padrao
Tempo [s]	92	93	92	92	4,859955	0,011503
					92,25	0,433013

N: 200000

	1	2	3	4		
valor [10E-3]	4,871717	4,875436	4,868207	4,86639	Media	Desvio Padrao
Tempo [s]	922	922	926	926	4,870438	0,003463
					924	2

N: 2000000

	1	2	3	4
Valor [10E-3]	4,870314	4,873362	4,871681	4,873687
Tempo [s]	9226	9255	9260	9226

Media	Desvio Padrao
4,872261	0,001358
9241,75	15,8489

Tipo: 5

N: 10000

	1	2	3	4
Valor [10E-5]	9,63937	9,60352	9,60986	9,59961
Tempo [s]	43	43	44	43

Media	Desvio Padrao
9,61309	0,015607
43,25	0,433013

N: 100000

	1	2	3	4
Valor [10E-5]	9,59868	9,59758	9,60898	9,59309
Tempo [s]	431	431	432	432

Media	Desvio Padrao
9,599583	0,005816
431,5	0,5

N: 1000000

	1	2	3	4
Valor [10E-5]	9,59147	9,59797	9,59649	9,58845
Tempo [s]	4306	4306	4320	4322

Media	Desvio Padrao
9,593595	0,003825
4313,5	7,53326

Tipo: 6

N: 30000

	1	2	3	4
Valor [10E-2]	2,750149	2,481141	2,924848	
N_Efetivo	20213	20243	20264	
N_Efetivo/N	0,673767	0,674767	0,675467	
Tempo [s]	99	99	99	

Media	Desvio Padrao
2,718713	0,182501
20240	20,92845
0,506	0,29214
99	0

N: 300000

1	2	3	4
---	---	---	---

Media	Desvio Padrao
-------	---------------

Valor [10E-2]	2,753758	3,019595	2,713139		
N_Efetivo	202426	202574	203054		
N_Efetivo/N	0,674753	0,675247	0,676847		
Tempo [s]	986	990	992		
				2,828831	0,135906
				202684,7	268,0564
				0,506712	0,292551
				989,3333	2,494438

N: 3000000

	1	2	3	4		
Valor [10E-2]	2,764117	2,782634	2,707687		Media	Desvio Padrao
N_Efetivo	2026430	2027169	2026633		2,751479	0,031875
N_Efetivo/N	0,675477	0,675723	0,675544		2026744	311,7381
Tempo [s]	9900	9905	9871		0,506686	0,292535
					9892	14,98888

Tipo: 7

N: 10000

	1	2	3	4		
Valor [10E-3]	1,531839	1,534021	1,533101	1,535867	Media	Desvio Padrao
Tempo [s]	39	40	39	39	1,533707	0,001468
					39,25	0,433013

N: 100000

	1	2	3	4		
Valor [10E-3]	1,532142	1,532865	1,531004	1,532423	Media	Desvio Padrao
Tempo [s]	333	333	334	334	1,532109	0,000688
					333,5	0,5

N: 1000000

	1	2	3	4		
Valor [10E-3]	1,532627	1,532598	1,532908	1,532698	Media	Desvio Padrao
Tempo [s]	3931	3945	3946	3930	1,532708	0,000121
					3938	7,516648

Tipo: 8

N: 10000

	1	2	3	4		
Valor [10E-2]	2,195545	2,318699	1,982627	2,048023	Media	Desvio Padrao
Tempo [s]	36	35	36	36	2,136224	0,130564
					35,75	0,433013

N: 100000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-2]	2,222162	2,179607	2,201379	2,151077	2,188556	0,026356
Tempo [s]	354	354	356	355	354,75	0,829156

N: 1000000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-2]	2,30046	2,253384			2,276922	0,023538
Tempo [s]	3540	3554			3547	7

Tipo: 9

N: 30000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	4,017571	4,00607	4,042786	3,976663	4,010773	0,023752
N_Efetivo	20252	20214	20394	20075	20233,75	113,5834
N_Efetivo/N	0,675067	0,6738	0,6798	0,669167	0,674458	0,003786
Tempo [s]	91	91	91	90	90,75	0,433013

N: 300000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	4,017749	4,015538	4,021271	4,015313	4,017468	0,002393
N_Efetivo	202779	202692	202775	202590	202709	76,98376
N_Efetivo/N	0,67593	0,67564	0,675917	0,6753	0,675697	0,000257
Tempo [s]	910	914	914	913	912,75	1,63936

N: 3000000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	4,019051				4,019051	2,48E-08
N_Efetivo	2027238				2027238	0
N_Efetivo/N	0,675746				0,168937	0,292607
Tempo [s]	12698				12698	0

Tipo: 10

N:	10000						
		1	2	3	4	Media	Desvio Padrao
Valor [10E-4]		1,666199	1,672476	1,669202		1,669292	0,002563
Tempo [s]		31	31	30		30,66667	0,471405

N:	100000						
		1	2	3	4	Media	Desvio Padrao
Valor [10E-4]		1,667701	1,665636	1,663601	1,666797	1,665934	0,001533
Tempo [s]		305	306	306	306	305,75	0,433013

N:	1000000						
		1	2	3	4	Media	Desvio Padrao
Valor [10E-4]		1,665933				1,665933	ERR
Tempo [s]		3048				3048	0

Tipo: 11

N:	20000						
		1	2	3	4	Media	Desvio Padrao
Valor [10E-5]		7,30622	7,24715	7,3307		7,29469	0,03507
N_Efetivo		13478	13390	13527		13465	56,68039
N_Efetivo/N		0,6739	0,6695	0,67635		0,504938	0,291536
Tempo [s]		53	53	53		53	0

N:	200000						
		1	2	3	4	Media	Desvio Padrao
Valor[10E-5]		7,30367	7,3328	7,36238	7,322726	7,330394	0,021224
N_Efetivo		134758	135130	135601	135120	135152,3	299,3095
N_Efetivo/N		0,67379	0,67565	0,678005	0,6756	0,675761	0,001497
Tempo [s]		525	526	530	528	527,25	1,920286

N:	2000000						
		1	2	3	4	Media	Desvio Padrao
Valor [10E-5]		7,3325				7,3325	4,69E-08
N_Efetivo		1351085				1351085	0
N_Efetivo/N		0,675543				0,168886	0,292518
Tempo [s]		5260				5260	0

Tipo: 12

N: 30000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-5]	5,32681	5,35164	5,33299		5,337147	0,010554
N_Efetivo	13633	13721	13642		13665,33	39,53339
N_Efetivo/N	0,454433	0,457367	0,454733	0	0,451633	0,197245
Tempo [s]	73	74	73		73,33333	0,471405

N: 300000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-5]	5,34334	5,36326	5,36024	5,34216	5,35225	0,009569
N_Efetivo	137048	137409	137096	136863	137104	196,409
N_Efetivo/N	0,456827	0,45803	0,456987	0,45621	0,457013	0,000655
Tempo [s]	734	735	737	736	735,5	1,118034

N: 3000000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-5]	5,33852				5,33852	3,25E-08
N_Efetivo	1368862				1368862	0
N_Efetivo/N	0,456287				0,456287	0,197578
Tempo [s]	7333				7333	0

Tipo: 13

N: 10000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	8,170148	8,1942	8,167482		8,177277	0,012016
Tempo [s]	42	41	42		41,66667	0,471405

N: 100000

	1	2	3	4	Media	Desvio Padrao
Valor [10E-4]	8,167426	8,178422	8,170599	8,181684	8,174533	0,00575
Tempo [s]	416	416	417	417	416,5	0,5

N: 1000000

	1	2	3	4
Valor [10E-2]	1,075202			
Tempo [s]	3008			

Media	Desvio Padrao
1,075202	9,94E-09
3008	0

V

Expressões

das

Áreas Diretas

na

Fornalha

V - Cálculo das áreas diretas de troca de calor

V.1 - Expressões das áreas diretas de troca de calor para os 36 casos possíveis de pares de elementos na fornalha.

As expressões abaixo foram obtidas, a partir das expressões genéricas apresentadas anteriormente, considerando-se os sistemas de coordenadas predefinidos e colocando-os na forma final pelo método de Monte Carlo.

Expressões:

1. Volume Regular - Volume Regular

$$g_{ij} = \frac{Ak_i Ak_j w}{\pi} \cdot \sum \frac{r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

2. Volume Regular - Volume Irregular

$$g_{ij} = \frac{Ak_i Ak_j w}{\pi} \cdot \sum \frac{r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

3 Volume Regular - Volume do Cilindro Central

$$g_{ij} = \frac{Ak_i Ak_j w}{\pi} \cdot \sum \frac{r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

4. Volume Regular - Superfície do Sorvedouro

$$g_{isj} = \frac{Rf \cdot \cos\left(\frac{\pi}{4}\right) Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos\phi_j \cdot r_j \cdot (1 + \tan^2\phi_j)}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

5. Volume Regular - Superfície Regular das Tampas

$$g_{isj} = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

6. Volume Regular - Superfície Irregular das Tampas

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

7. Volume Regular - Superfície do Cilindro Central das Tampas

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

8. Volume Regular - Superfície Cilíndrica Lateral

$$gisj = \frac{Rf \cdot Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

9. Volume Irregular - Volume Irregular

$$gisj = \frac{Ak_i \cdot Ak_j \cdot w}{\pi} \cdot \sum \frac{r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

10. Volume Irregular - Volume do Cilindro Central

$$gisj = \frac{Ak_i \cdot Ak_j \cdot w}{\pi} \cdot \sum \frac{r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

11. Volume Irregular - Superfície do Sorvedouro

$$gisj = \frac{Rf \cdot \cos\left(\frac{\pi}{4}\right) \cdot Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot (1 + tg^2 \theta_i)}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

12. Volume Irregular - Superfície Regular das Tampas

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

13. Volume Irregular - Superfície Irregular das Tampas

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

14. Volume Irregular - Superfície do Cilindro Central

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

15. Volume Irregular - Superfície Cilíndrica Lateral

$$gisj = \frac{Rf \cdot Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

16. Volume do Cilindro Central - Volume do Cilindro Central

$$gigj = \frac{Ak_i \cdot Ak_j \cdot w}{\pi} \cdot \sum \frac{r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

17. Volume do Cilindro Central - Superfície do Sorvedouro

$$gisj = \frac{Rf \cdot \cos(\frac{\pi}{4}) \cdot Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_j \cdot (1 + tg^2 \theta_j)}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

18. Volume do Cilindro Central - Superfície Regular das Tampas

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

19. Volume do Cilindro Central - Superfície Irregular das Tampas

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

20. Volume do Cilindro Central - Superfície do Cilindro Central

$$gisj = \frac{Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

21. Volume do Cilindro Central - Superfície Cilíndrica Lateral

$$gisj = \frac{Rf \cdot Ak_i \cdot w}{\pi} \cdot \sum \frac{\cos \phi_j \cdot r_i}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

22. Superfície do Sorvedouro - Superfície do Sorvedouro

$$sisj = 0$$

23. Superfície do Sorvedouro - Superfície Regular das Tampas

$$sisj = \frac{Rf \cdot \cos(\frac{\pi}{4}) \cdot w}{\pi} \cdot \sum \frac{\cos \phi_i \cdot \cos \phi_j \cdot r_j \cdot (1 + tg^2 \theta_j)}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

24. Superfície do Sorvedouro - Superfície Irregular das Tampas

$$sisj = \frac{Rf \cdot \cos\left(\frac{\pi}{4}\right) \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j \cdot (1 + tg^2\theta_i)}{(n \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

25. Superfície do Sorvedouro - Superfície do Cilindro Central

$$sisj = \frac{Rf \cdot \cos\left(\frac{\pi}{4}\right) \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j \cdot (1 + tg^2\theta_i)}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

26. Superfície do Sorvedouro - Superfície Cilíndrica Lateral

$$sisj = \frac{(Rf)^2 \cos\left(\frac{\pi}{4}\right) \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot (1 + tg^2\theta_i)}{(n \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

27. Superfície Regular das Tampas - Superfície Regular das Tampas

$$sisj = \frac{w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

28. Superfície Regular das Tampas - Superfície Irregular das Tampas

$$sisj = \frac{w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

29. Superfície Regular das Tampas - Superfície do Cilindro Central

$$sisj = \frac{w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

30. Superfície Regular das Tampas - Superfície Cilíndrica Lateral

$$sisj = \frac{Rf \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

31. Superfície Irregular das Tampas - Superfície Irregular das Tampas

$$sisj = \frac{w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

32. Superfície Irregular das Tampas - Superfície do Cilindro Central

$$sisj = \frac{w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

33. Superfície Irregular das Tampas - Superfície Cilíndrica Lateral

$$sisj = \frac{Rf \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

34. Superfície do Cilindro Central - Superfície do Cilindro Central

$$sisj = \frac{w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_i \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

35. Superfície do Cilindro Central - Superfície Cilíndrica Lateral

$$sisj = \frac{Rf \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j \cdot r_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

36. Superfície Cilíndrica Lateral - Superfície Cilíndrica Lateral

$$sisj = \frac{(Rf)^2 \cdot w}{\pi} \cdot \sum \frac{\cos\phi_i \cdot \cos\phi_j}{(N \cdot s^2)} \cdot e^{(-Ak \cdot s)}$$

Nomenclatura:

i, j : identificação dos elementos.

$\theta_i, \theta_j, r_i, r_j$: coordenadas polares (variáveis aleatórias do método de Monte Carlo).

ϕ : ângulo formado entre a reta normal à superfície e a reta que une os dois elementos.

Ak_i, Ak_j : coeficiente de absorção da intensidade de radiação.

Ak : coeficiente de ponderação para o coeficiente de absorção.

W : produto das diferenças dos limites superior e inferior das variáveis aleatórias

Rf : raio da fornalha.

s : distância entre dois elementos (superfícies ou volumes).

N : número de conjuntos de variáveis aleatórias.

VI

Programa

Computacional

VI - Programa computacional

VI.1 - Ferramentas

O programa de cálculo das áreas diretas e globais de troca de calor foi implementado em linguagem Pascal, utilizando-se o compilador Turbo Pascal versão 6.0 e um microcomputador Pc AT-386 DX..

VI.2 - Estrutura

Basicamente a estrutura do programa está dividida nos seguintes módulos:

1. *ÁreaDireta.*
2. *Varredura.*
3. *Gauss.*
4. *ÁreaGlobal.*
5. Library

VI.3 - Descrição dos módulos

VI.3.1 - *ÁreaDireta* :

Este módulo corresponde à uma *Unit* que contém as expressões para o cálculo das áreas diretas de troca de calor para os 36 pares possíveis, sendo que estes foram agrupados em 20 subrotinas de cálculo que utilizam o Método de Monte Carlo.

O módulo isoladamente não executa nenhuma subrotina, funciona apenas como um módulo-biblioteca das subrotinas de cálculo. A utilização das subrotinas é feita pelo módulo *Varredura* que será visto posteriormente.

Esta *Unit* é constituída dos seguintes subrotinas (procedures e functions):

CalcExtrTeta : calcula os extremos do ângulo θ para elementos de superfície ou volume.

CalcExtrRaio : calcula os extremos do raio r para elementos de superfície ou volumes.

CalcExtrZ : calcula os extremos da cota Z para elementos de superfície ou volume.

CalcExtrVol : calcula os extremos das variáveis que definem um elemento de volume (θ, r, Z) .

CalcExtrArea : calcula os extremos das variáveis que definem um elemento de superfície (θ, r) ou (θ, Z) .

Mc1 : calcula a área direta de troca de calor para os pares 1 , 3 e 16.

Mc2 : calcula a área direta de troca de calor para o par 9.

Mc3 : calcula a área direta de troca de calor para os pares 2 e 10.

Mc4 : calcula a área direta de troca de calor para o par 26.

Mc5 : calcula a área direta de troca de calor para os pares 23 e 25.

- Mc6** : calcula a área direta de troca de calor para o par 24.
- Mc7** : calcula a área direta de troca de calor para o par 36.
- Mc8** : calcula a área direta de troca de calor para os pares 30 e 35.
- Mc9** : calcula a área direta de troca de calor para o par 33
- Mc10** : calcula a área direta de troca de calor para os pares 27 , 29 , 34.
- Mc11** : calcula a área direta de troca de calor para os pares 28 e 32.
- Mc12** : calcula a área direta de troca de calor para o par 31.
- Mc13** : calcula a área direta de troca de calor para os pares 4 e 17.
- Mc14** : calcula a área direta de troca de calor para os pares 8 e 21.
- Mc15** : calcula a área direta de troca de calor para os pares 5 , 7 , 18 e 20.
- Mc16** : calcula a área direta de troca de calor para os pares 6 e 19.
- Mc17** : calcula a área direta de troca de calor para o par 11.
- Mc18** : calcula a área direta de troca de calor para o par 15.
- Mc19** : calcula a área direta de troca de calor para os pares 12 e 14.
- Mc20** : calcula a área direta de troca de calor para o par 13.

VI.3.2 - Varredura :

Este módulo faz a identificação de todos os pares possíveis de troca de calor entre os elementos da fornalha, levando-se em consideração o efeito da simetria entre os pares. Para cada par de troca identificado, uma subrotina de cálculo da Unit *ÁreaDireta* é executada e o valor armazenado num arquivo externo. Basicamente este método realiza 3 tipos de varreduras : identificação de pares superfície-superfície, pares superfície-volume e volume-volume. Uma vez executado este módulo, ficam disponíveis 3 arquivos (ssLoFile.txt, sgLoFile.txt e ggLoFile.txt) com os valores das áreas diretas de troca de calor para as 3 grandes "famílias" de troca de calor (Superfície-Superfície, Superfície-Volume e Volume-Volume).

Este módulo é constituído das seguintes subrotinas:

VarreSupSup : faz a varredura de todos os pares possíveis de troca entre dois elementos de superfícies da fornalha.

VarreSupVol : faz a varredura de todos os pares possíveis de troca entre um elemento de superfície e um elemento de volume da fornalha.

VarreVolVol : faz a varredura de todos os pares possíveis de troca entre dois elementos de volumes da fornalha.

VI.3.3 - *Gauss* :

Este módulo corresponde à uma *Unit* que contém a subrotina de cálculo de sistemas de equações lineares pelo método de Gauss Seidel, tal subrotina de cálculo faz parte do pacote numérico do Turbo Pascal.

VI.3.4 - *AreaGlobal* :

Este módulo calcula as áreas globais de troca de calor a partir dos valores das áreas diretas lidos através dos arquivos gerados no módulo *Varredura*. Devido à necessidade de resolver sistemas de equações lineares, este módulo depende da unit *Gauss*. Os valores das áreas globais de troca de calor são armazenados em 3 arquivos : SSbig.txt, GSbig.txt e GGbig.txt

Este módulo é constituído pelas seguintes subrotinas:

SSbig : calcula as áreas globais de troca de calor entre dois elementos de superfícies.

GSbig : calcula as áreas globais de troca de calor entre um elemento de superfície e outro de volume.

GGbig : calcula as áreas globais de troca de calor entre dois elementos de volumes.

VI.3.5 - Library :

Este módulo corresponde à uma biblioteca de subrotinas mais utilizadas:

CalcArea1 : calcula a área circular das tampas.

CalcArea2 : calcula a área periférica regular das tampas.

CalcArea3 : calcula a área periférica irregular das tampas.

CalcArea4 : calcula a área da superfície do sorvedouro.

CalcArea5 : calcula a área da superfície cilíndrica (lateral da fornalha).

CalcArea : cria um vetor com as áreas dos cinco tipos de elementos de superfícies existentes na fornalha.

ConvElemPosSup : converte o número do elemento superfície para as coordenadas *Seção* e *Setor*.

ConvElemPosVol : converte o número do elemento volume para as coordenadas *Seção* e *Setor*.

ConvPosElemSup : converte as coordenadas *Seção* e *Setor* do elemento superfície para o seu número correspondente.

ConvPosElemVol : converte as coordenadas *Seção* e *Setor* do elemento volume para o seu número correspondente.

PosSimSetor : devolve o setor simétrico.

VeriTypeArea : verifica o tipo de área dentre as cinco possíveis.

Emiss : devolve o valor da emissividade da superfície especificada.

Reflet : devolve o valor da refletividade da superfície especificada.

Es : calcula o poder emissivo de uma superfície.

Eg : calcula o poder emissivo de um volume.

CompMatTransfer : compõe a matriz de transferência do sistema de equações lineares para o cálculo de áreas globais de troca de calor.

CompVetExcitaSup : compõe o vetor de excitação (superfície) do sistema de equações lineares para o cálculo de áreas globais de troca de calor.

CompVetExcitaVol : compõe o vetor de excitação (volume) do sistema de equações lineares para o cálculo de áreas globais de troca de calor.

IniPerfilTempForno : inicializa uma distribuição de temperaturas para os elementos de superfícies e volumes da fornalha armazenando-a num arquivo.

LePerfilTempFile : faz a leitura do arquivo de distribuição de temperaturas na fornalha.

Inverte : faz a inversão das coordenadas *Seção* e *Sector* dos elementos *i* e *j*.

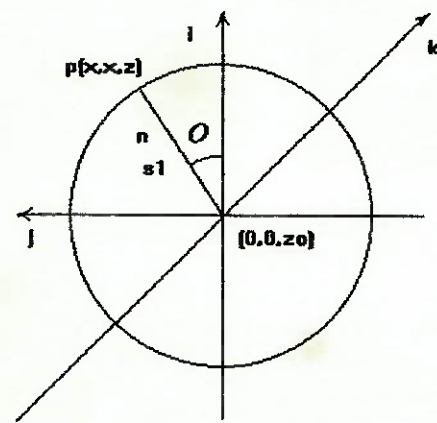
VII

Apêndice

Apêndice I

I.1 - Obtenção de versores normais às superfícies

I.1.1 - Superfície Cilíndrica



$$n = \frac{PO}{|PO|}$$

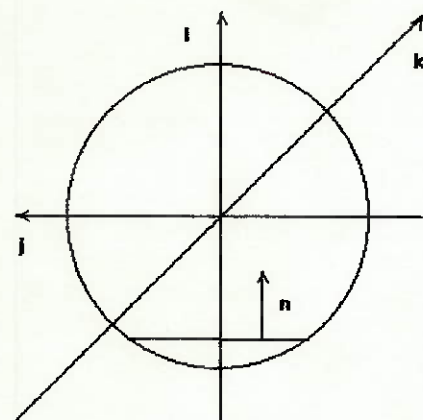
$$n = \frac{(0-x_A)}{s_1} \cdot i + \frac{(0-y_A)}{s_1} \cdot j + \frac{(z_0-z_A)}{s_1} \cdot k$$

$$z = z_0$$

$$n = -\frac{x_A}{s_1} \cdot i - \frac{y_A}{s_1} \cdot j$$

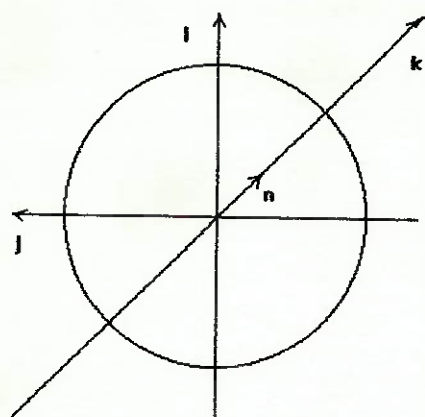
$$s_1 = \sqrt{x_A^2 + y_A^2}$$

II.1.2 - Superfície do Sorvedouro



$$n = j$$

II.1.3 - Superfície da Tampa



$$n_e = k$$

$$n_d = -k$$

II.2 - Cálculo do cosseno do ângulo formado entre a normal e areta que dois elementos

II.2.1 - Superfície Cilíndrica

$$f_A = \frac{(x_B - x_A)}{s} \cdot i + \frac{(y_B - y_A)}{s} \cdot j + \frac{(z_B - z_A)}{s} \cdot k \quad \text{vetor de módulo unitário que dois elementos}$$

$$s = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

$$n = -\frac{x_A}{s_1} \cdot i + -\frac{y_A}{s_1} \cdot j$$

$$s_1 = \sqrt{x_A^2 + y_A^2}$$

$$\cos \phi = |f \cdot n| = \left| \frac{(x_B - x_A)}{s} \cdot \frac{x_A}{s_1} + \frac{(y_B - y_A)}{s} \cdot \frac{y_A}{s_1} \right|$$

II.2.2 - Superfície do Sorvedouro

$$f_A = \frac{(x_B - x_A)}{s} \cdot i + \frac{(y_B - y_A)}{s} \cdot j + \frac{(z_B - z_A)}{s} \cdot k$$

$$s = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

$$n = i$$

$$\cos\phi = \frac{|x_B - x_A|}{s}$$

II.2.3 - Superfície da Tampas

$$f_A = \frac{(x_B - x_A)}{s} \cdot i + \frac{(y_B - y_A)}{s} \cdot j + \frac{(z_B - z_A)}{s} \cdot k$$

$$s = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

$$n_e = k$$

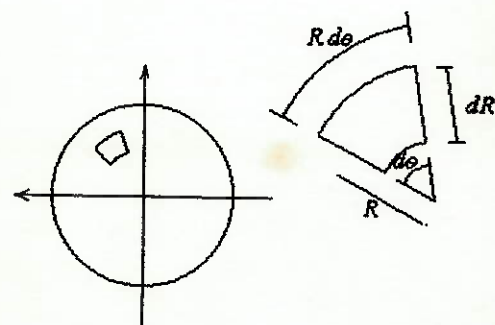
$$n_d = -k$$

$$\cos\phi = |f \cdot n| = \frac{|(z_B - z_A)|}{s}$$

Apêndice II

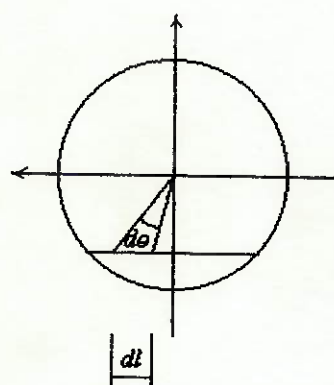
II.1 - Expressões para o cálculo dos elementos de área da fornalha

II.1.1 - Superfície da tampa



$$dA = R \cdot d\theta \cdot d\rho$$

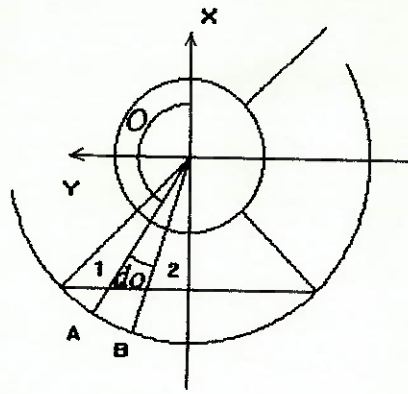
II.1.2 - Superfície do Sorvedouro



$$dl = R \cdot \frac{\sqrt{2}}{2} \cdot (1 + \tan^2 \theta) \cdot d\theta \cdot dz$$

$$dA = dz \cdot dl = R \cdot \frac{\sqrt{2}}{2} \cdot (1 + \tan^2 \theta) \cdot d\theta \cdot dz$$

Demonstração:



Determinação do ponto 1

Ponto A $(R \cdot \cos\theta, R \cdot \sin\theta)$

reta OA

$$\frac{y-0}{x-0} = \frac{R \cdot \sin\theta}{R \cdot \cos\theta} \Rightarrow y = \frac{\sin\theta}{\cos\theta} \cdot x = x \cdot \operatorname{tg}\theta$$

$$y_1 = -R \cdot \frac{\sqrt{2}}{2} \qquad y_1 = x_1 \cdot \operatorname{tg}\theta$$

Determinação do ponto 2.

Ponto B $(R \cdot \cos(\theta + d\theta), R \cdot \sin(\theta + d\theta))$

$$y_2 = x_2 \cdot \operatorname{tg}(\theta + d\theta)$$

$$x_2 = -R \cdot \frac{\sqrt{2}}{2}$$

$$y_2 = -R \cdot \frac{\sqrt{2}}{2} \cdot \operatorname{tg}(\theta + d\theta)$$

$$dl = y_1 - y_2 = -R \cdot \frac{\sqrt{2}}{2} \cdot (\operatorname{tg}\theta - \operatorname{tg}(\theta + d\theta))$$

$$dl = R \cdot \frac{\sqrt{2}}{2} \cdot (\operatorname{tg}(\theta + d\theta) - \operatorname{tg}\theta)$$

$$\operatorname{tg}(d\theta) = \frac{\operatorname{tg}(\theta + d\theta) - \operatorname{tg}\theta}{1 + \operatorname{tg}(\theta + d\theta) \operatorname{tg}\theta}$$

$$\operatorname{tg}(d\theta) \cdot [1 + \operatorname{tg}(\theta + d\theta) \cdot \operatorname{tg}\theta] = \operatorname{tg}(\theta + d\theta) - \operatorname{tg}\theta$$

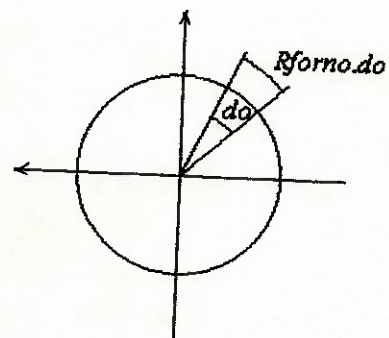
$$\operatorname{tg}(\theta + d\theta) - \operatorname{tg}\theta = (1 + \operatorname{tg}^2\theta) \cdot d\theta$$

$$d\theta \equiv \operatorname{tg}(d\theta)$$

$$\operatorname{tg}(\theta + d\theta) \equiv \operatorname{tg}\theta$$

$$dl = R \cdot \frac{\sqrt{2}}{2} \cdot (1 + \operatorname{tg}^2\theta) \cdot d\theta$$

II.1.3 - Superfície Cilíndrica



$$dA = R_{\text{forno}} \cdot d\theta \cdot dz$$

VIII

Bibliografia

VIII - Bibliografia

[1] - H.C. Hottel, A.F. Sarofim

Radiative Transfer

[2] - Souza, Francisco D.A.

Seminário "Método de Zonas aplicado às trocas de calor entre chamas e invólucros", Agrupamento de Engenharia Térmica do Instituto de Pesquisas Tecnológicas - IPT.

[3] - J.P. Holman

Tansferência de Calor

[4] - Shimizu, T.

Método de Monte Carlo - Publicação interna do laboratório de Processamentos de Dados do Instituto Tecnológico da Aeronáutica.

[5] - Schildt, H.

Turbo Pascal Avançado

[6] - User's Guide Turbo Pascal

[7] - Reference Guide Turbo Pascal

[8] - H. Miller Lawrence

Advanced Programming Design and Structure using Pascal

[9] - Catálogo do Laboratório de Combustão Industrial do Agrupamento de Engenharia Térmica do Instituto de Pesquisas Tecnológicas IPT.

Parte B

Listagem

dos

Módulos

do

Programa Computacional

I - ÁreaDireta***ÁreaDireta*****Subrotinas:*****CalcExtrTeta******CalcExtrRaio******CalcExtrZ******CalcExtrVol******CalcExtrArea******Mc1******Mc2******Mc3******Mc4******Mc5******Mc6******Mc7******Mc8******Mc9******Mc10******Mc11******Mc12******Mc13******Mc14******Mc15******Mc16******Mc17******Mc18******Mc19******Mc20***

```

{-----}
{ CalcExtrTeta : Calcula os extremos para a variavel Teta. }
{ }
{ }
{   Procedures           : - }
{   Functions           : - }
{   Variaveis Globais   : - }
{   Constantes Globais  : - }
{-----}

```

```

Procedure CalcExtrTeta(Elem      : IndicEscalar;
                      Setor      : EscVetByte;
                      Var
                        Extremos : ExtrRec);

```

```

var
  DTeta : Real;

```

```

begin
  with Extremos do
    begin
      DTeta:= 0.25 * Pi;
      if Setor[Elem] <> 0
      then
        begin
          TetaL[Elem]:= DTeta * (Setor[Elem] - 1);
          TetaG[Elem]:= TetaL[Elem] + DTeta;
        end
      else
        begin
          TetaL[Elem]:= 0;
          TetaG[Elem]:= 2 * Pi;
        end;
      end;
    end;
end;

```

```

{-----}
{ CalcExtrRaio : Calcula os extremos para a variavel R. }
{ }
{ }
{   Procedures           : - }
{   Functions           : - }
{   Variaveis Globais   : - }
{   Constantes Globais  : - RForno, RCil }
{-----}

```

```

Procedure CalcExtrRaio(Elem      : IndicEscalar;
                      Setor      : EscVetByte;
                      Var
                        Extremos : ExtrRec);

```

```

begin
  with Extremos do
    begin
      if Setor[Elem] <> 0
      then
        begin
          RL[Elem]:= RCil;
          RG[Elem]:= RForno;
        end
      else
        begin
          RL[Elem]:= 0;
          RG[Elem]:= RCil;
        end;
      end;
    end;
end;

```

```

{-----}
{ CalcExtrZ : Calcula os extremos para a variavel Z. }
{ }
{ }
{   Procedures      : - }
{   Functions      : - }
{   Variaveis Globais : - }
{   Constantes Globais : - LForno, NDivAxial }
{-----}

```

```

Procedure CalcExtrZ(Elem      : IndicEscalar;
                    Secao     : EscVetByte;
                    Var
                      Extremos : ExtrRec);
var
  DeltaZ : Real;

```

```

begin
  with Extremos do
    begin
      DeltaZ := LForno / NDivAxial;
      ZL[Elem] := DeltaZ * (Secao[Elem] - 1);
      ZG[Elem] := ZL[Elem] + DeltaZ;
    end;
end;

```

```

{-----}
{ CalcExtrVol : Calcula as tres variaveis que definem um elemento de }
{             volume. }
{ }
{ }
{   Procedures      : CalcExtrTeta, CalcExtrRaio, CalcExtrZ }
{   Functions      : - }
{   Variaveis Globais : - }
{   Constantes Globais : - }
{-----}

```

```

Procedure CalcExtrVol(Indice    : IndicEscalar;
                     Secao,
                     Setor      : EscVetByte;
                     Var
                       Extremos : ExtrRec);

```

```

begin
  CalcExtrTeta(Indice, Setor, Extremos);
  CalcExtrRaio(Indice, Setor, Extremos);
  CalcExtrZ(Indice, Secao, Extremos);
end;

```

```

{-----}
{ CalcExtrArea : Calcula as tres variaveis que definem um elemento de }
{ area. }
{ }
{ }
{ Procedures : CalcExtrTeta,CalcExtrRaio,CalcExtrZ }
{ Functions : - }
{ Variaveis Globais : - }
{ Constantes Globais : - }
{-----}

Procedure CalcExtrArea(Indice : IndicEscalar;
                      Secao,
                      Setor : EscVetByte;
                      Var
                        Extremos : ExtrRec);

begin
  if (Secao[Indice] <> 0) and (Secao[Indice] <> 11)
  then
    begin
      CalcExtrTeta(Indice,Setor,Extremos);
      CalcExtrZ(Indice,Setor,Extremos);
    end
  else
    begin
      CalcExtrTeta(Indice,Setor,Extremos);
      CalcExtrRaio(Indice,Secao,Extremos);
    end;
end;

```



```

{-----}
{ Mcl : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume (Central ou Periferica) }
{   J - Elemento de Volume (Central ou Periferica) }
{ }
{   Procedures      : CalcExtrVol }
{   Functions       : - }
{   Variaveis Globais : Variaveis, Extremos }
{   Constantes Globais : - AK, AKI, AKJ, K, N }
{-----}

Procedure Mcl;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    for Indice:= I to J do
    begin
      CalcExtrVol (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J] * DeltaZ[J];

    for a:= 1 to K do
      for b:= 1 to N do
      begin
        for Indice:= I to J do
        begin
          Teta[Indice]:= Random * DeltaTeta[Indice] + TetaL[Indice];
          R[Indice]:= Random * DeltaR[Indice] + RL[Indice];
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
          Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
        end;
        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
        Soma:= Soma + R[I] * R[J] * Exp(-AK * S1) / Sqr(S1)
      end;
    end;
    McResult:= AKI * AKJ * Wronsk * Soma / (Pi * K * N);
  end;
end;

```

```

{-----}
{ Mc2 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Irregular (Periferico Parcial) }
{   J - Elemento de Volume Irregular (Periferico Parcial) }
{ }
{   Procedures      : CalcExtrVol }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : - AK, AKI, AKJ, K, N }
{-----}

Procedure Mc2;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);

    for Indice:= I to J do
    begin
      CalcExtrVol (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J] * DeltaZ[J];
    for a:= 1 to K do
      for b:= 1 to N do
        begin
          Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
          R[I]:= Random * DeltaR[I] + RL[I];
          Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
          R[J]:= Random * DeltaR[J] + RL[J];
          If (R[I] <= abs(Rcos45 / Cos(Teta[I]))) and (R[J] <= abs(Rcos45 /
            Cos(Teta[J])))
          then
            begin
              for Indice:= I to J do
              begin
                X[Indice]:= R[Indice] * Cos(Teta[Indice]);
                Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
                Z[Indice]:= Random * DeltaZ[indice] + ZL[Indice];
              end;
              S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J])
              Soma:= Soma + R[I] * R[J] * Exp(-AK * S1) / Sqr(S1);
            end;
          end;
        end;
      McResult:= AKI * AKJ * Wronsk * Soma / (Pi * K * N);
    end;
  end;
end;

```

```

{-----}
{ Mc3 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Regular (Central ou Periferico) }
{   J - Elemento de Volume Irregular (Periferico Parcial) }
{-----}
{   Procedures      : CalcExtrVol }
{   Functions       : - }
{   Variaveis Globais : Variaveis, Extremos }
{   Constantes Globais : - AK, AKI, AKJ, K, N }
{-----}

```

```

Procedure Mc3;

```

```

begin

```

```

  Randomize;

```

```

  with Variaveis, Extremos do

```

```

    begin

```

```

      Soma:=0;

```

```

      Rcos45:= RForno * Cos(0.25 * Pi);

```

```

      for Indice:= I to J do

```

```

        begin

```

```

          CalcExtrVol (Indice, Secao, Setor, Extremos);

```

```

          DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];

```

```

          DeltaR[Indice]:= RG[Indice] - RL[Indice];

```

```

          DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];

```

```

        end;

```

```

      Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
               DeltaTeta[J] * DeltaR[J] * DeltaZ[J];

```

```

      for a:= 1 to K do

```

```

        for b:= 1 to N do

```

```

          begin

```

```

            Teta[J]:= Random * DeltaTeta[J] + TetaL[J];

```

```

            R[J]:= Random * DeltaR[J] + RL[J];

```

```

            If (R[J] <= abs(Rcos45 / Cos(Teta[J])))

```

```

              then

```

```

                begin

```

```

                  Teta[I]:= Random * DeltaTeta[I] + TetaL[I];

```

```

                  R[I]:= Random * DeltaR[I] + RL[I];

```

```

                  for Indice:= I to J do

```

```

                    begin

```

```

                      X[I]:= R[I] * Cos(Teta[I]);

```

```

                      Y[I]:= R[I] * Sin(Teta[I]);

```

```

                      Z[I]:= Random * DeltaZ[I] + ZL[I];

```

```

                    end;

```

```

                  S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]))

```

```

                  Soma:= Soma + R[I] * R[J] * Exp(-AK * S1) / Sqr(S1);

```

```

                end;

```

```

            end;

```

```

            McResult:= AKI * AKJ * Wronsk * Soma / (Pi * K * N);

```

```

          end;

```

```

        end;

```

```

      end;

```



```

{-----}
{ Mc4 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Sorvedouro }
{   J - Elemento de Superficie Cilindrica }
{ }
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK, RForno }
{-----}

Procedure Mc4;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);

    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    Wronsk:= DeltaTeta[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaZ[J];
    R[J]:= RForno;

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      R[I]:= Abs(Rcos45 / Cos(Teta[I]));
      Teta[J]:= Random * DeltaTeta[J] + TetaL[J];

      for Indice:= I to J do
      begin
        X[Indice]:= R[Indice] * Cos(Teta[Indice]);
        Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
      end;

      S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
      Albar:= (X[I] - X[J]) / S1;
      Ambar:= (Y[I] - Y[J]) / S1;
      CosTT[I]:= abs(Albar);
      Sn:= Sqrt(Sqr(X[J]) + Sqr(Y[J]));
      Aln:= X[J] / Sn;
      Amn:= Y[J] / Sn;
      CosTT[J]:= abs(Albar * Aln + Ambar * Amn);
      Tan:= Sin(Teta[I]) / Cos(Teta[I]);
      Soma:= Soma + CosTT[I] * CosTT[J] * Exp(-AK * S1) * (1 +
                                                             Sqr(Tan))/Sqr(S1);
    end;
    McResult:= RForno * Rcos45 * Wronsk * Soma / (Pi * K * N);
  end;
end;

```



```

{-----}
{ Mc5 : Calcula a area direta de troca de calor entre dois elementos: }
{      I - Elemento de Superficie Sorvedouro }
{      J - Elemento de Superficie Regiao das Tampas }
{-----}
{      Procedures      : CalcExtrArea }
{      Functions      : - }
{      Variaveis Globais : Variaveis,Extremos }
{      Constantes Globais : AK, K, N }
{-----}

Procedure Mc5;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);

    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
    end;

    DeltaZ[I]:= ZG[I] - ZL[I];
    DeltaR[J]:= RG[J] - RL[J];

    Wronsk:= DeltaTeta[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J];
    if Secao[J] = 0 then Z[J]:= 0
      else Z[J]:= LForno;
    for a:= 1 to K do
      for b:= 1 to N do
        begin
          Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
          R[I]:= Abs(Rcos45 / Cos(Teta[I]));
          Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
          R[J]:= Random * DeltaR[J] + RL[J];
          for Indice:= I to J do
            begin
              X[Indice]:= R[Indice] * Cos(Teta[Indice]);
              Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
            end;
          Z[I]:= Random * DeltaZ[I] + ZL[I];
          S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
          Albar:= (X[I] - X[J]) / S1;
          Anbar:= (Z[I] - Z[J]) / S1;
          CosTT[I]:= abs(Albar);
          CosTT[J]:= abs(Anbar);
          Tan:= Sin(Teta[I]) / Cos(Teta[I]);
          Soma:= Soma + CosTT[I] * CosTT[J] * Exp(-AK * S1) * R[J] * (1 +
                                                                Sqr(Tan))/Sqr(S1);
        end;
      end;
    McResult:= Rcos45 * Wronsk * Soma / (Pi * K * N);
  end;
end;

```

```

{-----}
{ Mc6 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Sorvedouro }
{   J - Elemento de Superficie Irregular das Tampas }
{-----}
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK, K, N }
{-----}

Procedure Mc6;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);

    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
    end;

    DeltaZ[I]:= ZG[I] - ZL[I];
    DeltaR[J]:= RG[J] - RL[J];
    Wronsk:= DeltaTeta[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J];

    if Secao[J] = 0 then Z[J]:= 0
                      else Z[J]:= LForno;
    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
      R[J]:= Random * DeltaR[J] + RL[J];
      If R[J] <= Abs(Rcos45 / Cos(Teta[J]))
      then
      begin
        Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
        R[I]:= Abs(Rcos45 / Cos(Teta[I]));
        Z[I]:= Random * DeltaZ[I] + ZL[I];

        for Indice:= I to J do
        begin
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        end;

        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]
        Albar:= (X[I] - X[J]) / S1;
        Anbar:= (Z[I] - Z[J]) / S1;
        CosTT[I]:= abs(Albar);
        CosTT[J]:= abs(Anbar);
        Tan:= Sin(Teta[I]) / Cos(Teta[I]);
        Soma:= Soma + CosTT[I] * CosTT[J] * Exp(-AK * S1) * R[J] * (1
        Sqr(Tan))/Sqr(S1);

      end;
    end;
  end;
  McResult:= Rcos45 * Wronsk * Soma / (Pi * K * N);
end;
end;

```

```

{-----}
{ Mc7 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Cilindrica }
{   J - Elemento de Superficie Cilindrica }
{ }
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,RForno }
{-----}

```

```

Procedure Mc7;

```

```

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;

    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    Wronsk:= DeltaTeta[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaZ[J];
    for a:= 1 to K do
    for b:= 1 to N do
    begin
      for Indice:= I to J do
      begin
        Teta[Indice]:= Random * DeltaTeta[Indice] + TetaL[Indice];
        X[Indice]:= RForno * Cos(Teta[Indice]);
        Y[Indice]:= RForno * Sin(Teta[Indice]);
        Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
      end;
      S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
      Albar:= (X[I] - X[J]) / S1;
      Ambar:= (Y[I] - Y[J]) / S1;
      Anbar:= (Z[I] - Z[J]) / S1;
      Sn:= Sqrt(Sqr(X[I]) + Sqr(Y[I]));
      Aln:= X[I] / Sn;
      Amn:= Y[I] / Sn;
      CostT[I]:= abs(Albar * Aln + Ambar * Amn);
      Sn:= Sqrt(Sqr(X[J]) + Sqr(Y[J]));
      Aln:= X[J] / Sn;
      Amn:= Y[J] / Sn;
      CostT[J]:= abs(Albar * Aln + Ambar * Amn);
      Soma:= Soma + CostT[I] * CostT[J] * Exp(-AK * S1) / Sqr(S1);
    end;
  end;
  McResult:= Sqr(RForno) * Wronsk * Soma / (Pi * K * N);
end;
end;

```



```

{-----}
{ Mc8 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Cilindrica }
{   J - Elemento de Superficie Regiao das Tampas }
{-----}
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,RForno }
{-----}

Procedure Mc8;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;

    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
    end;

    DeltaZ[I]:= ZG[I] - ZL[I];
    DeltaR[J]:= RG[J] - RL[J];

    Wronsk:= DeltaTeta[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J];

    R[I]:= RForno;
    if Secao[J] = 0 then Z[J]:= 0
      else Z[J]:= LForno;

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      R[J]:= Random * DeltaR[J] + RL[J];
      for Indice:= I to J do
      begin
        Teta[Indice]:= Random * DeltaTeta[Indice] + TetaL[Indice];
        X[Indice]:= R[Indice] * Cos(Teta[Indice]);
        Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
      end;
      Z[I]:= Random * DeltaZ[I] + ZL[I];
      S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
      Albar:= (X[I] - X[J]) / S1;
      Ambar:= (Y[I] - Y[J]) / S1;
      Anbar:= (Z[I] - Z[J]) / S1;
      Sn:= Sqrt(Sqr(X[I]) + Sqr(Y[I]));
      Aln:= X[I] / Sn;
      Amn:= Y[I] / Sn;
      CostT[I]:= abs(Albar * Aln+ Ambar * Amn);
      CostT[J]:= abs(Anbar);
      Soma:= Soma + CostT[I] * CostT[J] * Exp(-AK * S1) * R[J] / Sqr(S1);
    end;
    McResult:= RForno * Wronsk * Soma / (Pi * K * N);
  end;
end;
end;

```



```

{-----}
{ Mc9 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Irregular das Tampus }
{   J - Elemento de Superficie Cilindrica }
{ }
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,RForno,K,N }
{-----}

Procedure Mc9;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
    end;
    DeltaR[I]:= RG[I] - RL[I];
    DeltaZ[J]:= ZG[J] - ZL[J];
    Wronsk:= DeltaTeta[I] * DeltaR[I] *
              DeltaTeta[J] * DeltaZ[J];
    R[J]:= RForno;
    if Secao[I] = 0 then Z[I]:= 0
                      else Z[I]:= LForno;
    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      R[I]:= Random * DeltaR[I] + RL[I];
      If R[I] <= Abs(Rcos45 / Cos(Teta[I]))
      then
      begin
        Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
        Z[J]:= Random * DeltaZ[J] + ZL[J];
        for Indice:= I to J do
        begin
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        end;
        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) +
                  Sqr(Z[I] - Z[J]));
        Albar:= (X[I] - X[J]) / S1;
        Ambar:= (Y[I] - Y[J]) / S1;
        Anbar:= (Z[I] - Z[J]) / S1;
        CostTT[I]:= abs(Anbar);
        Sn:= Sqrt(Sqr(X[J]) + Sqr(Y[J]));
        Aln:= X[J] / Sn;
        Amn:= Y[J] / Sn;
        CostTT[J]:= abs(Albar * Aln + Ambar * Amn);
        Soma:= Soma + CostTT[I] * CostTT[J] * Exp(-AK * S1) * R[I] /
                  Sqr(S1);
      end
    end;
  end;
  McResult:= RForno * Wronsk * Soma / (Pi * K * N);
end;
end;

```

```

{-----}
{ Mc10 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie de Tampa }
{   J - Elemento de Superficie de Tampa }
{ }
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,RForno,K,N }
{-----}

Procedure Mc10;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;

    for Indice:= I to J do
    begin
      CalcExtrArea(Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice];
      if Secao[Indice] = 0 then Z[Indice]:= 0
      else Z[Indice]:= LForno;
    end;

    Wronsk:= DeltaTeta[I] * DeltaR[I] *
              DeltaTeta[J] * DeltaR[J];
    for a:= 1 to K do
    for b:= 1 to N do
    begin
      for Indice:= I to J do
      begin
        Teta[Indice]:= Random * DeltaTeta[Indice] + TetaL[Indice];
        R[Indice]:= Random * DeltaR[Indice] + RL[Indice];
        X[Indice]:= R[Indice] * Cos(Teta[Indice]);
        Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
      end;

      S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
      Anbar:= (Z[I] - Z[J]) / S1;
      CosTT[I]:= abs(Anbar);
      CosTT[J]:= abs(Anbar);
      Soma:= Soma + CosTT[I] * CosTT[J] * Exp(-AK * S1) * R[I] * R[J] /
              Sqr(S1);
    end;
    McResult:= Wronsk * Soma / (Pi * K * N);
  end;
end;

```

```

{-----}
{ Mc11 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Irregular das Tampas }
{   J - Elemento de Superficie Regular das Tampas }
{ }
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,K,N }
{-----}

Procedure Mc11;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[indice];
      DeltaR[Indice]:= RG[Indice] - RL[indice];
      if Secao[Indice] = 0 then Z[Indice]:= 0
      else Z[Indice]:= LForno;
    end;

    Wronsk:= DeltaTeta[I] * DeltaR[I] *
              DeltaTeta[J] * DeltaR[J];
    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      R[I]:= Random * DeltaR[I] + RL[I];
      If R[I] <= Abs(Rcos45 / Cos(Teta[I]))
      then
      begin
        Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
        R[J]:= Random * DeltaR[J] + RL[J];
        for Indice:= I to J do
        begin
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        end;
        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
        Anbar:= (Z[I] - Z[J]) / S1;
        CosTT[I]:= abs(Anbar);
        CosTT[J]:= abs(Anbar);
        Soma:= Soma + CosTT[I] * CosTT[J] * Exp(-AK * S1) * R[I] * R[J] /
                  Sqr(S1);
      end
    end;
    McResult:= Wronsk * Soma / (Pi * K * N);
  end;
end;
end;

```



```

{-----}
{ Mc12 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Superficie Irregular das Tampas }
{   J - Elemento de Superficie Irregular das Tampas }
{ }
{   Procedures      : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,K,N }
{-----}

Procedure Mc12;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    for Indice:= I to J do
    begin
      CalcExtrArea (Indice,Secao,Setor,Extremos);
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice];
      if Secao[Indice] = 0 then Z[Indice]:= 0
      else Z[Indice]:= LForno;
    end;

    Wronsk:= DeltaTeta[I] * DeltaR[I] *
              DeltaTeta[J] * DeltaR[J];

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
      R[I]:= Random * DeltaR[I] + RL[I];
      R[J]:= Random * DeltaR[J] + RL[J];
      If (R[I] <= Abs(Rcos45 / Cos(Teta[I]))) and
        (R[J] <= Abs(Rcos45 / Cos(Teta[J])))
      then
        begin
          for Indice:= I to J do
          begin
            X[Indice]:= R[Indice] * Cos(Teta[Indice]);
            Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
          end;
          S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
          Anbar:= (Z[I] - Z[J]) / S1;
          CosTT[I]:= abs(Anbar);
          CosTT[J]:= abs(Anbar);
          Soma:= Soma + CosTT[I] * CosTT[J] * Exp(-AK * S1) * R[I] * R[J] /
                  Sqr(S1);
        end
      end;
    end;
    McResult:= Wronsk * Soma / (Pi * K * N);
  end;
end;

```



```

{-----}
{ Mc13 : Calcula a area direta de troca de calor entre dois elementos: }
{      I - Elemento de Volume Regular }
{      J - Elemento de Superficie Sorvedouro }
{-----}
{
  Procedures      : CalcExtrVol
                  : CalcExtrArea
  Functions       : -
  Variaveis Globais : Variaveis, Extremos
  Constantes Globais : AK,AKI,K,N
}-----}

Procedure Mc13;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:=0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);

    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    DeltaR[I]:= RG[I] - RL[I];
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaZ[J];

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
      R[I]:= Random * DeltaR[I] + RL[I];
      R[J]:= Abs(Rcos45 / Cos(Teta[J]));
      for Indice:= I to J do
      begin
        X[Indice]:= R[Indice] * Cos(Teta[Indice]);
        Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
      end;

      S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
      Albar:= (X[I] - X[J]) / S1;
      CosTT[J]:= abs(Albar);
      Tan:= Sin(Teta[J]) / Cos(Teta[J]);
      Soma:= Soma + CosTT[J] * Exp(-AK * S1) * (1 + Sqr(Tan)) *
              R[I] / Sqr(S1);
    end;
    McResult:= Rcos45 * AKI * Wronsk * Soma / (Pi * K * N);
  end;
end;
end;

```

```

{-----}
{ Mc14 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Regular }
{   J - Elemento de Superficie Cilindrica }
{ }
{   Procedures      : CalcExtrVol }
{                   CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,AKI,RForno,K,N }
{-----}

Procedure Mc14;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:=0;
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);

    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    DeltaR[I]:= RG[I] - RL[I];
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaZ[J];
    R[J]:= RForno;

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      for Indice:= I to J do
      begin
        Teta[Indice]:= Random * DeltaTeta[Indice] + TetaL[Indice];
        R[Indice]:= Random * DeltaR[Indice] + RL[Indice];
        Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
        X[Indice]:= R[Indice] * Cos(Teta[Indice]);
        Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
      end;
      S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
      Albar:= (X[I] - X[J]) / S1;
      Ambar:= (Y[I] - Y[J]) / S1;
      Sn:= Sqrt(Sqr(X[J]) + Sqr(Y[J]));
      Aln:= X[J] / Sn;
      Amn:= Y[J] / Sn;
      CostT[J]:= abs(Albar * Aln + Ambar * Amn);
      Soma:= Soma + CostT[J] * Exp(-AK * S1) * R[I] / Sqr(S1);
    end;
    McResult:= RForno * AKI * Wronsk * Soma / (Pi * K * N);
  end;
end;

```

```

{-----}
{ Mc15 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Regular }
{   J - Elemento de Superficie Regular das Tampas }
{-----}
{   Procedures      : CalcExtrVol }
{                   : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,AKI,K,N }
{-----}

Procedure Mc15;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:=0;
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);

    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice]
    end;

    DeltaZ[I]:= ZG[I] - ZL[I];
    if Secao[J] = 0 then Z[J]:= 0
      else Z[J]:= LForno;
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
      DeltaTeta[J] * DeltaR[J];
    for a:= 1 to K do
      for b:= 1 to N do
      begin
        for Indice:= I to J do
        begin
          Teta[Indice]:= Random * DeltaTeta[Indice] + TetaL[Indice];
          R[Indice]:= Random * DeltaR[Indice] + RL[Indice];
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        end;
        Z[I]:= Random * DeltaZ[I] + ZL[I];
        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
        Anbar:= Z[I] - Z[J] / S1;
        CostT[J]:= abs(Anbar);
        Soma:= Soma + CostT[J] * Exp(-AK * S1) * R[I] * R[J] / Sqr(S1);
      end;
    end;
    McResult:= AKI * Wronsk * Soma / (Pi * K * N);
  end;
end;

```



```

{-----}
{ Mc16 : Calcula a area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Regular }
{   J - Elemento de Superficie Irregular das Tampas }
{-----}
{   Procedures      : CalcExtrVol }
{                   : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,AKI,K,N }
{-----}

Procedure Mc16;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);

    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice];
    end;

    DeltaZ[I]:= ZG[I] - ZL[I];
    if Secao[J] = 0 then Z[J]:= 0
      else Z[J]:= RForno;

    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
      DeltaTeta[J] * DeltaR[J];

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
      R[J]:= Random * DeltaR[J] + RL[J];
      If R[J] <= Abs(Rcos45 / Cos(Teta[J]))
      then
      begin
        Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
        R[I]:= Random * DeltaR[I] + RL[I];
        Z[I]:= Random * DeltaZ[I] + ZL[I];
        for Indice:= I to J do
        begin
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
        end;
        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
        Anbar:= (Z[I] - Z[J]) / S1;
        CosTT[J]:= abs(Anbar);
        Soma:= Soma + CosTT[J] * Exp(-AK * S1) * R[I] * R[J] / Sqr(S1);
      end;
    end;
    McResult:= AKI * Wronsk * Soma / (Pi * K * N);
  end;
end;
end;

```



```

{-----}
{ Mc17 : Calcula area direta de troca de calor entre dois elementos: }
{      I - Elemento de Volume Irregular                               }
{      J - Elemento de Superficie Sorvedouro                         }
{-----}
{      Procedures      : CalcExtrVol                                  }
{                      CalcExtrArea                                  }
{      Functions      : -                                           }
{      Variaveis Globais : Variaveis,Extremos                      }
{      Constantes Globais : AK,AKI,K,N                              }
{-----}

Procedure Mc17;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);
    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;
    DeltaR[I]:= RG[I] - RL[I];
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaZ[J];
    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      R[I]:= Random * DeltaR[I] + RL[I];
      If R[I] <= Abs(Rcos45 / Cos(Teta[I]))
      then
      begin
        Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
        R[J]:= Abs(Rcos45 / Cos(Teta[J]));
        for Indice:= I to J do
        begin
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
          Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
        end;
        S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
        Albar:= (X[I] - X[J]) / S1;
        CostT[J]:= abs(Albar);
        Tan:= sin(Teta[J]) / cos(Teta[J]);
        Soma:= Soma + CostT[J] * Exp(-AK * S1) * (1 +
          Sqr(Tan)) * R[I] / Sqr(S1);
      end;
    end;
  end;
  McResult:= AKI * Rcos45 * Wronsk * Soma / (Pi * K * N);
end;
end;

```

```

{-----}
{ Mc18 : Calcula area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Irregular }
{   J - Elemento de Superficie Cilindrica }
{ }
{   Procedures      : CalcExtrVol }
{                   : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,AKI,RForno,K,N }
{-----}

Procedure Mc18;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);

    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaZ[Indice]:= ZG[Indice] - ZL[Indice];
    end;

    DeltaR[I]:= RG[I] - RL[I];
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaZ[J];
    R[J]:= RForno;
    for a:= 1 to K do
      for b:= 1 to N do
        begin
          Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
          R[I]:= Random * DeltaR[I] + RL[I];
          If R[I] <= Abs(Rcos45 / Cos(Teta[I]))
          then
            begin
              Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
              for Indice:= I to J do
                begin
                  X[Indice]:= R[Indice] * Cos(Teta[Indice]);
                  Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
                  Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
                end;
              Z[I]:= Random * DeltaZ[I] + ZL[I];
              S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
              Albar:= (X[I] - X[J]) / S1;
              Ambar:= (Y[I] - Y[J]) / S1;
              Sn:=Sqrt(Sqr(X[J]) + Sqr(Y[J]));
              Aln:= X[J] / Sn;
              Amn:= Y[J] / Sn;
              CosTT[J]:= Abs(Albar * Aln + Ambar * Amn);
              Soma:= Soma + CosTT[J] * Exp(-AK * S1) * R[I] / Sqr(S1);
            end;
          end;
        end;
      McResult:= RForno * AKI * Wronsk * Soma / (Pi * K * N);
    end;
  end;
end;

```

```

{-----}
{ Mc19 : Calcula area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Irregular }
{   J - Elemento de Superficie Regular das Tampas }
{ }
{   Procedures      : CalcExtrVol }
{                   CalcExtrArea }
{   Functions      : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,AKI,K,N }
{-----}

Procedure Mc19;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);
    for Indice:= I to J do
      begin
        DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
        DeltaR[Indice]:= RG[Indice] - RL[Indice];
      end;
    DeltaZ[I]:= ZG[I] - ZL[I];
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J];
    if Secao[J] = 0 then Z[J]:= 0
      else Z[J]:= LForno;

    for a:= 1 to K do
      for b:= 1 to N do
        begin
          Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
          R[I]:= Random * DeltaR[I] + RL[I];
          If R[I] <= Abs(Rcos45 / Cos(Teta[I]))
          then
            begin
              Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
              R[J]:= Random * DeltaR[J] + RL[J];

              for Indice:= I to J do
                begin
                  X[Indice]:= R[Indice] * Cos(Teta[Indice]);
                  Y[Indice]:= R[Indice] * Sin(Teta[Indice]);

                  end;
              Z[I]:= Random * DeltaZ[I] + ZL[I];
              S1:= Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
              Anbar:= (Z[I] - Z[J]) / S1;
              CostT[J]:= Abs(Anbar);
              Soma:= Soma + CostT[J] * Exp(-AK * S1) * R[I] * R[J] / Sqr(S1);
            end;
          end;
        McResult:= AKI * Wronsk * Soma / (Pi * K * N);
      end;
    end;
  end;
end;

```



```

{-----}
{ Mc20 : Calcula area direta de troca de calor entre dois elementos: }
{   I - Elemento de Volume Irregular }
{   J - Elemento de Superficie Irregular das Tampas }
{-----}
{   Procedures      : CalcExtrVol }
{                   : CalcExtrArea }
{   Functions       : - }
{   Variaveis Globais : Variaveis,Extremos }
{   Constantes Globais : AK,AKI,K,N }
{-----}

Procedure Mc20;

begin
  Randomize;
  with Variaveis,Extremos do
  begin
    Soma:= 0;
    Rcos45:= RForno * Cos(0.25 * Pi);
    CalcExtrVol (I,Secao,Setor,Extremos);
    CalcExtrArea (J,Secao,Setor,Extremos);

    for Indice:= I to J do
    begin
      DeltaTeta[Indice]:= TetaG[Indice] - TetaL[Indice];
      DeltaR[Indice]:= RG[Indice] - RL[Indice];
    end;

    DeltaZ[I]:= ZG[I] - ZL[I];
    Wronsk:= DeltaTeta[I] * DeltaR[I] * DeltaZ[I] *
              DeltaTeta[J] * DeltaR[J];

    if Secao[J] = 0 then Z[J]:= 0
      else Z[J]:= LForno;

    for a:= 1 to K do
    for b:= 1 to N do
    begin
      Teta[I]:= Random * DeltaTeta[I] + TetaL[I];
      R[I]:= Random * DeltaR[I] + RL[I];
      Teta[J]:= Random * DeltaTeta[J] + TetaL[J];
      R[J]:= Random * DeltaR[J] + RL[J];
      If (R[I] <= Abs(Rcos45 / Cos(Teta[I]))) and
        (R[J] <= Abs(Rcos45 / Cos(Teta[J])))
      then
      begin
        for Indice:= I to J do
        begin
          X[Indice]:= R[Indice] * Cos(Teta[Indice]);
          Y[Indice]:= R[Indice] * Sin(Teta[Indice]);
          Z[Indice]:= Random * DeltaZ[Indice] + ZL[Indice];
        end;
        S1:=Sqrt(Sqr(X[I] - X[J]) + Sqr(Y[I] - Y[J]) + Sqr(Z[I] - Z[J]));
        Anbar:= (Z[I] - Z[J]) / S1;
        CosTT[J]:= Abs(Anbar);
        Soma:= Soma + CosTT[J] * Exp(-AK * S1) * R[I] * R[J] / Sqr(S1);
      end;
    end;
  end;
  McResult:= AKI * Wronsk * Soma / (Pi * K * N);
end;
end;

```


II - Varredura

Varredura

Subrotinas:

VarreSupSup

VarreSupVol

VarreVolVol

```

{-----}
{ VarreSupSup : Executa a varredura entre os elementos de superficie, }
{ obtendo-se todos os pares possiveis. Para cada par, }
{ faz-se a identificacao do tipo da subrotina de calculo }
{ que devera ser executado para a obtencao das areas }
{ diretas de troca de calor (sup - sup). Nesta varredura, }
{ leva-se em consideracao o efeito de simetria do forno, }
{ alem das superficies que nao se 'enxergam'. }
{ }
{ Procedures : ConvPosElemSup, Inverte }
{ Functions : - }
{ Var Global : - }
{ Cte Global : NDivLateral, NSecao }
{-----}

```

```

Procedure VarreSupSup;

```

```

var

```

```

    SecVet,
    SetVet : EscVetByte;
    Cont,
    ElemI,
    ElemJ : Byte;
    McResult : Real;
    OutFile : Text;

```

```

begin

```

```

    Cont:= 0;
    Assign(OutFile, 'ssLoFile.txt');
    Rewrite(OutFile);
    Close(OutFile);
    Append (OutFile);
    for SecaoI:= 0 to (NSecao - 1) do
        begin
            if SecaoI = 0
            then
                begin
                    SetorIndexIMin:= 0;
                    SecaoIndexJMin:= SecaoI + 1;
                end
            else
                begin
                    SetorIndexIMin:= 1;
                    SecaoIndexJMin:= SecaoI;
                end;
            for SecaoJ:= SecaoIndexJMin to NSecao do
                begin
                    for SetorI:= SetorIndexIMin to (NDivLateral div 2) do
                        begin
                            if SecaoJ = NSecao then SetorIndexJMin:= 0
                            else
                                if SecaoI = SecaoJ
                                then SetorIndexJMin:= SetorI
                                { Setor + 1, simplificacao possivel }
                                else SetorIndexJMin:= 1;
                            if SetorI = 0 then SetorIndexJMax:= (NDivLateral div 2)
                            else SetorIndexJMax:= NDivLateral;
                            for SetorJ:= SetorIndexJMin to SetorIndexJMax do
                                begin
                                    SecVet[I]:= SecaoI;
                                    SecVet[J]:= SecaoJ;
                                    SetVet[I]:= SetorI;
                                    SetVet[J]:= SetorJ;
                                    if SecaoI = 0

```

```

then
  case SetorI of
    0..3,6..NDivLateral :
      if SecaoJ = NSecao
        then
          case SetorJ of
            0..3,6..NDivLateral :
              Mc10(SecVet,SetVet,McResult);
            4,5 :
              begin
                Inverte(SecVet,SetVet);
                Mc11(SecVet,SetVet,McResult);
              end;
          end
        else
          case SetorJ of
            1..3,6..NDivLateral :
              begin
                Inverte(SecVet,SetVet);
                Mc8(SecVet,SetVet,McResult);
              end;
            4,5 :
              begin
                Inverte(SecVet,SetVet);
                Mc5(SecVet,SetVet,McResult);
              end;
          end;
        4,5 : if SecaoJ = NSecao
          then
            case SetorJ of
              0..3,6..NDivLateral :
                Mc11(SecVet,SetVet,McResult);
              4,5 :
                Mc12(SecVet,SetVet,McResult);
            end
          else
            case SetorJ of
              1..3,6..NDivLateral :
                Mc9(SecVet,SetVet,McResult);
              4,5 :
                begin
                  Inverte(SecVet,SetVet);
                  Mc6(SecVet,SetVet,McResult);
                end;
            end
          end
        end
      else
        case SetorI of
          1..3,6..NDivLateral:
            if SecaoJ = NSecao
              then
                case SetorJ of
                  0..3,6..NDivLateral :
                    Mc8(SecVet,SetVet,McResult);
                  4,5 :
                    begin
                      Inverte(SecVet,SetVet);
                      Mc9(SecVet,SetVet,McResult);
                    end;
                end
              else
                case SetorJ of
                  1..3,6..NDivLateral :
                    Mc7(SecVet,SetVet,McResult);
                  4,5 :
                    ;
                end
              end
            end
          end
        end
      end
    end
  end

```



```

4,5: if SecaoJ = NSecao
    then
        case SetorJ of
            0..3,6..NDivLateral :
                Mc5(SecVet,SetVet,McResult);
            4,5 :
                Mc6(SecVet,SetVet,McResult);
        end
    else
        case SetorJ of
            1..3,6..NDivLateral :
                Mc5(SecVet,SetVet,McResult);
            4,5 :
                McResult:= 0.0;
        end
    end;
ConvPosElemSup(SecaoI,SetorI,ElemI);
ConvPosElemSup(SecaoJ,SetorJ,ElemJ);
if Cont <= 2
    then
        begin
            write(OutFile,ElemI:3,ElemJ:3,' ',McResult:12);
            Cont:= Cont + 1;
        end
    else
        begin
            writeln(OutFile,ElemI:3,ElemJ:3,' ',McResult:12);
            Cont:= 0;
        end;
    end
end
end
end;
Close (OutFile);
end;

```

```

{-----}
{ VarreSupVol : Executa a varredura entre um elemento de superficie e }
{ de volume obtendo-se todos os pares possiveis. Para cada }
{ par, faz-se a identificacao do tipo da subrotina de }
{ calculo que devera ser executado para a obtencao das }
{ areas diretas de troca de calor (sup - vol). Nesta }
{ varredura, leva-se em consideracao o efeito de simetria }
{ do forno. }
{ }
{ Procedures : ConvPosElemSup, ConvPosElemVol }
{ Functions : - }
{ Var Global : - }
{ Cte Global : NDivLateral, NSecao }
{-----}

```

```

Procedure VarreSupVol;

```

```

var

```

```

  SecVet,
  SetVet : EscVetByte;
  ElemI,
  ElemJ,
  Cont : Byte;
  ElemIReal,
  ElemJReal,
  McResult : Real;
  OutFile : Text;

```

```

begin

```

```

  Cont:= 0;
  Assign(OutFile,'sgLoFile.txt');
  Rewrite(OutFile);
  Close(OutFile);
  Append(OutFile);
  for SecaoI:= 0 to NSecao do
  begin
    if (SecaoI = 0) or (SecaoI = NSecao) then SetorIndexIMin:= 0
      else SetorIndexIMin:= 1;

```

```

  for SecaoJ:= 1 to (NSecao - 1) do
  begin

```

```

    for SetorI:= SetorIndexIMin to (NDivLateral div 2) do
    begin

```

```

      if SetorI = 0 then SetorIndexJMax:= (NDivLateral div 2)
        else SetorIndexJMax:= NDivLateral;
      for SetorJ:= 0 to SetorIndexJMax do

```

```

        begin
          SecVet[I]:= SecaoI;
          SecVet[J]:= SecaoJ;
          SetVet[I]:= SetorI;
          SetVet[J]:= SetorJ;
          Inverte (SecVet,SetVet);
          if (SecaoI = 0) or (SecaoI = NSecao)
            then
              case SetorI of
                0..3,6..NDivLateral :
                  case SetorJ of
                    0..3,6..NDivLateral :
                      Mc15(SecVet,SetVet,McResult);
                    4,5 :
                      Mc19(SecVet,SetVet,McResult);
                  end;
                4,5 :
                  case SetorJ of
                    0..3,6..NDivLateral :

```

```

        Mc16(SecVet,SetVet,McResult);
        4,5
        :
        Mc20(SecVet,SetVet,McResult);
    end;
end
else
    case SetorI of
        0..3,6..NDivLateral :
            case SetorJ of
                0..3,6..NDivLateral :
                    Mc14(SecVet,SetVet,McResult);
                    4,5
                    :
                    Mc18(SecVet,SetVet,McResult);
                end;
                4,5
                :
                case SetorJ of
                    0..3,6..NDivLateral :
                        Mc13(SecVet,SetVet,McResult);
                        4,5
                        :
                        Mc17(SecVet,SetVet,McResult);
                    end;
                end;
            end;
        ConvPosElemSup(SecaoI,SetorI,ElemI);
        ConvPosElemVol(SecaoJ,SetorJ,ElemJ);
        if Cont <= 2
            then
                begin
                    write(OutFile,ElemI:3,ElemJ:3,' ',McResult:12);
                    Cont:= Cont + 1;
                end
            else
                begin
                    writeln(OutFile,ElemI:3,ElemJ:3,' ',McResult:12);
                    Cont:= 0;
                end;
            end
        end
    end
end
end;
Close(OutFile);
end;

```



```

{-----}
{  VarreVolVol : Executa a varredura entre dois elementos de volumes      }
{      obtendo-se todos os pares possiveis. Para cada par,                }
{      faz-se a identificacao do tipo da subrotina de calculo              }
{      que devera ser executado para a obtencao das areas                  }
{      diretas de troca de calor (vol - vol). Nesta varredura              }
{      leva-se em consideracao o efeito de simetria do forno                }
{-----}
{      Procedures : ConvPosElemVol                                          }
{      Functions   : -                                                      }
{      Var Global  : -                                                      }
{      Cte Global  : NDivLateral, NSecao                                    }
{-----}

```

```

Procedure VarreVolVol;

```

```

var
  OutFile   : Text;
  SecVet,
  SetVet    : EscVetByte;
  ElemI,
  ElemJ,
  Cont      : Byte;
  McResult  : Real;

begin
  Cont := 0;
  Assign(OutFile, 'ggLoFile.txt');
  Rewrite(OutFile);
  Close(OutFile);
  Append(OutFile);
  for SecaoI := 1 to (NSecao - 1) do
    for SecaoJ := SecaoI to (NSecao - 1) do
      begin
        for SetorI := 0 to (NDivLateral div 2) do
          begin
            if SetorI = 0 then SetorIndexJMax := (NDivLateral div 2)
            else SetorIndexJMax := NDivLateral;
            if SecaoI = SecaoJ then SetorIndexJMin := SetorI + 1
            else SetorIndexJMin := 0;
            for SetorJ := SetorIndexJMin to SetorIndexJMax do
              begin
                SecVet[I] := SecaoI;
                SecVet[J] := SecaoJ;
                SetVet[I] := SetorI;
                SetVet[J] := SetorJ;

                case SetorI of
                  0..3, 6..NDivLateral :
                    case SetorJ of
                      0..3, 6..NDivLateral :
                        Mc1(SecVet, SetVet, McResult);
                      4, 5 :
                        Mc3(SecVet, SetVet, McResult);
                    end;
                  4, 5 :
                    case SetorJ of
                      0..3, 6..NDivLateral :
                        begin
                          Inverte (SecVet, SetVet);
                          Mc3(SecVet, SetVet, McResult);
                        end;
                      4, 5 :
                        Mc2(SecVet, SetVet, McResult);
                    end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```
        end;  
    end;  
    ConvPosElemVol(SecaoI,SetorI,ElemI);  
    ConvPosElemVol(SecaoJ,SetorJ,ElemJ);  
    Append (OutFile);  
    if Cont <= 2  
    then  
        begin  
            write(OutFile,ElemI:3,ElemJ:3,' ',McResult:12);  
            Cont:= Cont + 1;  
        end  
    else  
        begin  
            writeln(OutFile,ElemI:3,ElemJ:3,' ',McResult:12);  
            Cont:= 0;  
        end;  
    Close (OutFile);  
end  
end  
end  
end;  
end;
```

III - Gauss*Gauss*


```

{-----}
{ Gauss : Esta Unit e apenas uma modificacao da rotina de calculo de um }
{ sistema de n equacoes lineares pelo metodo de Gauss Seidel. Tal }
{ rotina pertencente ao pacote numerico do Turbo Pascal - }
{ Numerical Methods Toolbox - Borland }
{-----}

Unit Gauss;
{$S+}
Interface

Const
    Nmax = 97;
    TNNearlyZero = 1E-07;

type
    TVetor = array[0..Nmax] of Real;
    TMatriz = array[0..Nmax] of TVetor;
    Float = Real;

procedure GaussSeidel(Dimen : integer;
    Var Coefficients : TMatriz;
    Var Constants : TVetor;
    Tol : Float;
    MaxIter : integer;
    Var Solution : TVetor;
    Var Iter : integer;
    Var Error : byte);

Implementation

procedure GaussSeidel(Dimen : integer;
    Var Coefficients : TMatriz;
    Var Constants : TVetor;
    Tol : Float;
    MaxIter : integer;
    Var Solution : TVetor;
    Var Iter : integer;
    Var Error : byte);

var
    Guess : TVetor;

procedure TestInput(Dimen : integer;
    Tol : Float;
    MaxIter : integer;
    var Coefficients : TMatriz;
    var Constants : TVetor;
    var Solution : TVetor;
    var Error : byte);

{-----}
{- Input: Dimen, Tol, MaxIter -}
{- Coefficients, -}
{- Constants -}
{- Output: Solution, Error -}
{- -}
{- test the input data for errors -}
{- The procedure also finds the -}
{- solution for the trivial case -}
{- Dimen = 0. -}
{-----}

```

```

begin
  Error := 0;
  if Dimen < 1 then
    Error := 3
  else
    if Tol <= 0 then
      Error := 4
    else
      if MaxIter < 0 then
        Error := 5;
      if (Error = 0) and (Dimen = 1) then
        begin
          if ABS(Coefficients[0, 0]) < TNNearlyZero then
            Error := 6
          else
            Solution[0] := Constants[0] / Coefficients[0, 0];
          end;
        end;
      { procedure TestInput }

```

```

procedure TestForDiagDominance(Dimen      : integer;
                               var Coefficients : TMatriz;
                               var Error      : byte);

```

```

{-----}
{- Input: Dimen, Coefficients                      -}
{- Output: Error                                   -}
{-                                                -}
{- This procedure examines the Coefficients matrix to see if it is -}
{- diagonally dominant. If it is, then the Gauss-Seidel iterative -}
{- method will converge to a solution of this system of equations; -}
{- if not, then convergence may not be possible with this method -}
{- and Error = 1 (which is a warning) is returned. If one of the -}
{- elements on the main diagonal of the Coefficients matrix is -}
{- zero, then the matrix is singular and cannot be solved and -}
{- Error = 6 is returned. In such a case, one of the direct -}
{- methods for solving systems of equations (e.g. Gaussian -}
{- elimination) should be used.                    -}
{-----}

```

```

var
  Row, Column : integer;
  Sum : Float;

```

```

begin
  Row := 0;
  while (Row < Dimen) and (Error < 2) do
    begin
      Row := Succ(Row);
      Sum := 0;
      for Column := 0 to Dimen do
        if Column <> Row then
          Sum := Sum + ABS(Coefficients[Row, Column]);
        if Sum > ABS(Coefficients[Row, Row]) then
          Error := 1; { WARNING! convergence may not be }
                     { possible because matrix isn't }
                     { diagonally dominant }
        if ABS(Coefficients[Row, Row]) < TNNearlyZero then
          Error := 6; { Singular matrix - can't be solved }
                     { by the Gauss-Seidel method. }
      end; { while }
    end; { procedure TestForDiagDominance }

```

```

procedure MakeInitialGuess(Dimen      : integer;
                           var Coefficients : TMatriz;
                           var Constants   : TVetor;

```

```

var Guess          : TVetor);

{-----}
{- Input: Dimen, Coefficients, Constants          -}
{- Output: Guess                                  -}
{-                                                -}
{- This procedure creates an initial approximation to the solution -}
{- by dividing the Constants terms by the corresponding terms      -}
{- on the main diagonal of the Coefficients matrix.                -}
{-----}

var
  Term : integer;

begin
  FillChar(Guess, SizeOf(Guess), 0);
  for Term := 0 to Dimen do
    if ABS(Coefficients[Term, Term]) > TNNearlyZero then
      Guess[Term] := Constants[Term] / Coefficients[Term, Term];
  end; { procedure MakeInitialGuess }

  procedure TestForConvergence(Dimen      : integer;
                               var OldApprox : TVetor;
                               var NewApprox : TVetor;
                               Tol          : Float;
                               var Done      : boolean;
                               var Product   : Float;
                               var Error     : byte);

  {-----}
  {- Input: Dimen, OldApprox, NewApprox, Tol, Product          -}
  {- Output: Done, Product, Error                              -}
  {-                                                -}
  {- This procedure determines if the sequence of approximations -}
  {- has converged. For convergence to occur, the relative difference -}
  {- between each Term of OldApprox and NewApprox must be less than -}
  {- the tolerance, Tol. If so, Done = TRUE is returned.          -}
  {-                                                -}
  {- This procedure also determines if the sequence of approximations -}
  {- is diverging. Product records the total fractional change from -}
  {- the initial guess to the current iteration. If Product is greater -}
  {- than 1E20, then the sequence is assumed to have diverged. If so, -}
  {- Error = 7 is returned.                                        -}
  {-----}

  var
    Term : integer;
    PartProd : Float;

  begin
    Done := true;
    PartProd := 0;
    for Term := 0 to Dimen do
      begin
        if ABS(OldApprox[Term] - NewApprox[Term]) > ABS(NewApprox[Term] * Tol) then
          Done := false;
        if (ABS(OldApprox[Term]) > TNNearlyZero) and (Error = 1) then
          { This is part of the divergence test }
          PartProd := PartProd + ABS(NewApprox[Term] / OldApprox[Term]);
        end;
        Product := Product * PartProd / Dimen;
        if Product > 1E20 then
          Error := 7 { Sequence is diverging }
        end; { procedure TestForConvergence }
      end;
    end;
  end;

```



```

procedure Iterate(Dimen      : integer;
                  var Coefficients : TMatriz;
                  var Constants  : TVetor;
                  var Guess      : TVetor;
                      Tol      : Float;
                      MaxIter   : integer;
                  var Solution   : TVetor;
                  var Iter       : integer;
                  var Error      : byte);
{-----}
{- Input: Dimen, Coefficients, Constants, Guess, Tol, MaxIter -}
{- Output: Solution, Iter, Error -}
{- -}
{- This procedure performs the Gauss-Seidel iteration and -}
{- returns either an error or the approximated solution and -}
{- the number of iterations. -}
{-----}

var
  Done : boolean;
  OldApprox, NewApprox : TVetor;
  Term, Loop : integer;
  FirstSum, SecondSum, Product : Float;

begin { procedure Iterate }
  Product := 1;
  Done := false;
  Iter := 0;
  NewApprox := Guess;
  OldApprox := Guess;
  while (Iter < MaxIter) and not(Done) and (Error <= 1) do
    begin
      Iter := Succ(Iter);
      for Term := 0 to Dimen do
        begin
          FirstSum := 0;
          SecondSum := 0;
          for Loop := 0 to Term - 1 do
            FirstSum := FirstSum + Coefficients[Term, Loop] * NewApprox[Loop];
          for Loop := Term + 1 to Dimen do
            SecondSum := SecondSum + Coefficients[Term, Loop] * OldApprox[Loop];
          NewApprox[Term] := (Constants[Term] - FirstSum - SecondSum) /
            Coefficients[Term, Term];
        end;
      TestForConvergence(Dimen, OldApprox, NewApprox, Tol, Done, Product, Error);
      OldApprox := NewApprox;
    end; { while }
    if (Iter < MaxIter) and (Error = 1) then
      Error := 0; { The sequence converged, }
                  { disregard the warning }
    if (Iter >= MaxIter) and (Error = 1) then
      Error := 1; { Matrix is not diagonally dominant; }
                  { convergence is probably impossible }
    if (Iter >= MaxIter) and (Error = 0) then
      Error := 2; { Convergence IS possible; }
                  { more iterations are needed }
    Solution := NewApprox;
  end; { procedure Iterate }

begin { procedure Gauss_Seidel }
  TestInput(Dimen, Tol, MaxIter, Coefficients, Constants, Solution, Error);
  if Dimen > 1 then
    begin
      TestForDiagDominance(Dimen, Coefficients, Error);
    end;
end;

```

```
if Error < 2 then
begin
  MakeInitialGuess(Dimen, Coefficients, Constants, Guess);
  Iterate(Dimen, Coefficients, Constants, Guess, Tol,
    MaxIter, Solution, Iter, Error);
end;
end;
end; { procedure GaussSeidel }

begin
end.
```

IV - ÁreaGlobal

ÁreaGlobal

Subrotinas:

SSbig

GSbig

GGBig


```

{-----}
{ SSbig : calcula as areas globais de troca de calor entre elementos de }
{      superficies, e sao armazenados no arquivo SSHiFile.txt.          }
{-----}
{ Procedures : - }
{ Functions   : - VerifTypeArea,Emiss,Reflet }
{ Var Global  : - }
{ Cte Global  : - }
{-----}

Procedure SSbig;
var
  NumSupI,
  NumSupJ,
  Cont      : Byte;
  OutFile   : Text;
  sWs,
  SS        : TVetor;

{-----}
{ CalcSSbig : calcula as areas globais de }
{      troca de calor entre elementos }
{      de superficies.                  }
{-----}
{ Procedures : - }
{ Functions   : - VerifTypeArea,Emiss,Reflet }
{ Var Global  : - }
{ Cte Global  : - }
{-----}

Procedure CalcSSbig(SupExcita : Byte;
                    Var SS     : TVetor;
                    sWs        : TVetor;
                    Area       : Vetor5);
var
  NumSup,
  TypeArea : Byte;
  Aux      : Real;

begin
  {CalcSSbig}
  for NumSup:= 0 to 97 do
  begin
    TypeArea:= VerifTypeArea(NumSup);
    if NumSup = SupExcita
    then Aux:= Emiss(SupExcita)
    else Aux:= 0;
    SS[NumSup]:= Area[TypeArea] * Emiss(NumSup) *
                  (sWs[NumSup] - Aux) / Reflet(NumSup);
  end;
end;

{CalcSSbig}

begin
  {SSbig}
  assign (OutFile,'SSHiFile.txt');
  rewrite (OutFile);
  Cont:= 0;
  for NumSupI:= 0 to 97 do
  begin
    CompVetExcitaSup (NumSupI,Area,Temperatura,VetExcita);
    GaussSeidel(97,MatTransfer,VetExcita,1E-3,50,sWs,Iter,Erro);
    CalcSSbig(NumSupI,SS,sWs,Area);
    for NumSupJ:= 0 to 97 do
    begin
      if Cont <= 2

```

```
    then
      begin
        write (OutFile,NumSupI:3,NumSupJ:3,' ',SS[NumSupJ]:12);
        Cont:= Cont + 1;
      end {then}
    else
      begin
        writeln (OutFile,NumSupI:3,NumSupJ:3,' ',SS[NumSupJ]:12);
        Cont:= 0;
      end; {else}
    end;
  end;
close (OutFile);
end; {SSbig}
```

```

{-----}
{ GSbig : Calcula as areas globais de troca de calor entre os elementos }
{       de volume e superficie, e sao armazenados no arquivo GSHiFile. }
{-----}
{ Procedures : - }
{ Functions  : - VerifTypeArea,Emiss,Reflet }
{ Var Global : - MatTransfer,Area,Temperatura }
{ Cte Global : - }
{-----}

```

```

Procedure GSbig;

```

```

var
  NumSup,
  NumVol,
  Cont   : Byte;
  GS,
  gWs    : TVetor;
  OutFile1,
  OutFile2 : Text;

```

```

{-----}
{ CalcGSbig : calcula as areas globais de }
{           troca de calor entre os }
{           elementos de volume e }
{           superficie. }
{-----}
{ Procedures : - }
{ Functions  : - VerifTypeArea,Emiss,Reflet }
{ Var Global : - }
{ Cte Global : - }
{-----}

```

```

Procedure CalcGSbig(gWs : TVetor;
                   Var GS : TVetor);

```

```

var
  NumSup,
  TypeArea: Byte;

```

```

begin
  for NumSup:= 0 to 97 do
  begin
    TypeArea:= VerifTypeArea(NumSup);
    GS[NumSup]:= Area[TypeArea] * Emiss(NumSup) * gWs[NumSup] /
                                     Reflet(NumSup);
  end
end; {CalcGSbig}

```

```

begin {GSbig}
  assign (OutFile1,'GSHiFile.Txt');
  rewrite (OutFile1);
  assign (OutFile2,'gWsFile.Txt');
  rewrite (OutFile2);
  Cont:= 0;
  for NumVol:= 0 to 89 do
  begin
    CompVetExcitaVol (NumVol,Temperatura,VetExcita);
    GaussSeidel (97,MatTransfer,VetExcita,1E-3,50,gWs,Iter,Erro);
    CalcGSbig (gWs,GS);
    for NumSup:= 0 to 97 do
    begin
      if Cont <= 2
      then

```



```
begin
  write (OutFile1,NumVol:3,NumSup:3,' ',GS[NumSup]:12);
  write (OutFile2,NumVol:3,NumSup:3,' ',gWs[NumSup]:12);
  Cont:= Cont + 1;
end {then}
else
begin
  writeln (OutFile1,NumVol:3,NumSup:3,' ',GS[NumSup]:12);
  writeln (OutFile2,NumVol:3,NumSup:3,' ',gWs[NumSup]:12);
  Cont:= 0;
end; {else}
end;
end;
close (OutFile1);
close (OutFile2);
end; {GSbig}
```

```

{-----}
{ GBig : calcula as areas globais de troca de calor entre elementos de }
{       volume, e sao armazenados no arquivo GGHFile.txt.             }
{-----}
{ Procedures : - }
{ Functions   : - VerifTypeArea,Emiss,Reflet }
{ Var Global  : - }
{ Cte Global  : - }
{-----}

```

```

Procedure GBig;

```

```

var
  Cont,
  ElemSup,
  ElemI,
  ElemJ   : Byte;
  Soma,
  GiGj    : Real;
  gg,
  sg,
  gWs     : TVetor;
  OutFile : Text;

```

```

{-----}
{ CompVetgg : Compoe um vetor com todos os }
{            possiveis pares entre dois }
{            volumes sendo um deles fixo. }
{-----}
{ Procedures : - }
{ Functions   : - }
{ Var Global  : - }
{ Cte Global  : - }
{-----}

```

```

Procedure CompVetgg(IFix   : Byte;
                   Var gg : TVetor);

```

```

var
  Cont,
  ElemI,
  ElemJ,
  Elem,
  Secao,
  Setor,
  SimElem,
  SimSetor,
  SimIFix,
  TipoCaso : Byte;
  Result    : Real;
  OutFile   : Text;

```

```

{-----}
{ Caso1 : primeiro elemento de volume }
{      e o volume do cilindro central. }
{      }
{ Procedures : ConvElemPosVol,      }
{              ConvPosElemVol,      }
{              PosSimSetor          }
{ Functions   : -                    }
{ Var Global  : Secao,Setor,SimSetor,Elem }
{              SimElem,Result       }
{ Cte Global  : -                    }
{-----}

```

```

Procedure Caso1 (Var gg: TVetor);

```

```

begin
  while not (eof(OutFile)) do
    begin
      if Cont <= 2
      then
        begin
          read (OutFile,ElemI,ElemJ,Result);
          Inc(Cont);
        end
      else
        begin
          readln (OutFile,ElemI,ElemJ,Result);
          Cont:= 0;
        end;
      if ElemI = 0 then Elem:= ElemI
        else Elem:= ElemJ;
      ConvElemPosVol (Elem,Secao,Setor);
      PosSimSetor (Setor,SimSetor);
      ConvPosElemVol (Secao,SimSetor,SimElem);
      gg[Elem]:= Result;
      gg[SimElem]:= Result;
    end;
  end;
end;

```

```

{-----}
{ Caso2 : primeiro elemento de volume }
{      esta num dos setores 1,2,3 ou 4. }
{      }
{ Procedures : ConvElemPosVol,      }
{              ConvPosElemVol,      }
{              PosSimSetor          }
{ Functions   : -                    }
{ Var Global  : Secao,Setor,SimSetor,Elem, }
{              SimElem,Result       }
{ Cte Global  : -                    }
{-----}

```



```

Procedure Caso2 (Var gg: TVetor);
begin
  while not (eof(OutFile)) do
    begin
      if Cont <= 2
      then
        begin
          read (OutFile,ElemI,ElemJ,Result);
          Inc(Cont);
        end
      else
        begin
          readln (OutFile,ElemI,ElemJ,Result);
          Cont:= 0;
        end;
      if ElemI = IFix then gg[ElemJ]:= Result;
      if ElemJ = IFix then gg[ElemI]:= Result;
      if ElemJ = SimIFix
      then
        begin
          ConvElemPosVol (Elem,Secao,Setor);
          PosSimSetor (Setor,SimSetor);
          ConvPosElemVol (Secao,SimSetor,SimElem);
          gg[SimElem]:= Result;
        end;
      end;
    end;
end;

```

```

{-----}
{ Caso3 : primeiro elemento de volume }
{      esta num setores 5,6,7,8.      }
{                                     }
{                                     }
{ Procedures : ConvElemPosVol,        }
{               ConvPosElemVol,       }
{               PosSimSetor           }
{ Functions   : -                     }
{ Var Global  : Secao,Setor,SimSetor,Elem, }
{               SimElem,Result        }
{ Cte Global  : -                     }
{-----}

```

```

Procedure Caso3 (Var gg : TVetor);
begin
  while not (eof(OutFile)) do
    begin
      if Cont <= 2
      then
        begin
          read (OutFile,ElemI,ElemJ,Result);
          Inc(Cont);
        end
      else
        begin
          readln (OutFile,ElemI,ElemJ,Result);
          Cont:= 0;
        end;
      if (ElemI = SimIFix) or (ElemJ =SimIFix)
      then
        begin
          if ElemI = SimIFix then Elem:= ElemJ
          else Elem:= ElemI;
          ConvElemPosVol (Elem,Secao,Setor);
          PosSimSetor (Setor,SimSetor);
          ConvPosElemVol (Secao,SimSetor,SimIFix);
          gg[SimElem]:= Result;
        end;
      if ElemJ = Ifix then gg[ElemI]:= Result;
    end;
  end;

begin
  FillChar(gg,SizeOf(gg),0);
  ConvElemPosVol (IFix,Secao,Setor);
  PosSimSetor (Setor,SimSetor);
  ConvPosElemVol (Secao,SimSetor,SimIFix);
  assign (OutFile,'ggLoFile.txt');
  reset (OutFile);
  Cont:= 0;
  case setor of
    0      : Caso1(gg);
    1..4   : Caso2(gg);
    5..8   : Caso3(gg);
  end;
  close (OutFile);
end;

{-----}
{ CompVetsg : Compoe um vetor com todos os }
{           possiveis pares entre uma     }
{           superficie e um volume s fixo.  }
{-----}
{ Procedures : - }
{ Functions  : - }
{ Var Global : - }
{ Cte Global  : - }
{-----}

```

```

Procedure CompVetsg(J      : Byte;
                   Var sg : Tvetor);
var
  Cont,
  ElemI,
  ElemJ,
  Secao,
  Setor,
  SimSetorJ,
  SimSetorI,
  SimElemJ : Byte;
  Result : Real;
  OutFile : Text;

begin
  ConvElemPosVol (J,Secao,Setor);
  PosSimSetor (Setor,SimSetorJ);
  ConvPosElemVol (Secao,SimSetorJ,SimElemJ);
  assign (OutFile,'sgLoFile.txt');
  reset (OutFile);
  Cont:= 0;
  while not (eof(OutFile)) do
    begin
      if Cont <= 2
      then
        begin
          read (OutFile,ElemI,ElemJ,Result);
          Inc(Cont);
        end
      else
        begin
          readln (OutFile,ElemI,ElemJ,Result);
          Cont:= 0;
        end;
      if ElemJ = J then sg[ElemI]:= Result;
      if ElemJ = SimElemJ
      then
        begin
          ConvElemPosSup(ElemI,Secao,Setor);
          PosSimSetor(Setor,SimSetorI);
          ConvPosElemSup(Secao,SimSetorI,ElemI);
          sg[ElemI]:= Result;
        end;
      end;
    close (OutFile);
  end;

{-----}
{ CompVetgWs : Compose um vetor com todos os }
{              W de cada elemento de super- }
{              ficie devido a um elemento   }
{              de volume fixo.               }
{ Procedures : -                             }
{ Functions   : -                             }
{ Var Global  : -                             }
{ Cte Global  : -                             }
{-----}

```



```

Procedure CompVetgWs(I      : Byte;
                    Var gWs : TVetor);

var
  Cont,
  ElemI,
  ElemJ : Byte;
  Result : Real;
  OutFile : Text;

begin
  assign (OutFile, 'gWsFile.txt');
  reset (OutFile);
  Cont:= 0;
  while not (eof(OutFile)) do
    begin
      if Cont <= 2
      then
        begin
          read (OutFile, ElemI, ElemJ, Result);
          Inc(Cont);
        end
      else
        begin
          readln (OutFile, ElemI, ElemJ, Result);
          Cont:= 0;
        end;
      if ElemI = 1 then gWs[ElemJ]:= Result;
      end;
    close (OutFile);
  end;

  begin
    assign (OutFile, 'GGHiFile.txt');
    rewrite (OutFile);
    Cont:= 0;
    for ElemI:= 0 to 89 do
      begin
        CompVetgg (ElemI, gg);
        CompVetgWs (ElemI, gWs);
        for ElemJ:= 0 to 89 do
          begin
            CompVetsg (ElemJ, sg);
            Soma:= 0;
            for ElemSup:= 0 to 97 do
              Soma:= Soma + sg[ElemSup] * gWs[ElemSup];
            GiGj:= gg[ElemJ] + Soma;
            if Cont <= 2
            then
              begin
                write (OutFile, ElemI:3, ElemJ:3, ' ', GiGj:12);
                Inc(Cont);
              end
            else
              begin
                writeln (OutFile, ElemI:3, ElemJ:3, ' ', GiGj:12);
                Cont:= 0;
              end;
            end;
          end;
        end;
      close (OutFile);
    end;
  end;

```

V - Library***Library*****Subrotinas:**

CalcArea1
CalcArea2
CalcArea3
CalcArea4
CalcArea5
CalcArea
ConvElemPosSup
ConvElemPosVol
ConvPosElemSup
ConvPosElemVol
PosSimSetor
VerifTypeArea
Emiss
Es
Eg
CompMatTransfer
CompVetExcitaSup
CompVetExcitaVol
IniPerfilTempForno
LePerfilTempFile
Inverte

```

{-----}
{ CalcArea1 : Calcula a area cilindrica das }
{ tampas. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : - }
{-----}

```

```
Function CalcArea1 (Raio : Real) : Real;
```

```
begin
  CalcArea1:= Pi * Sqr(Raio);
end;
```

```

{-----}
{ CalcArea2 : Calcula a area periferica }
{ regular das tampas. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : NDivLateral }
{-----}

```

```
Function CalcArea2 (RaioMaior,
                   RaioMenor : Real ) : Real;
```

```
begin
  CalcArea2:= (CalcArea1(RaioMaior) - CalcArea1(RaioMenor)) / NDivLateral;
end;
```

```

{-----}
{ CalcArea3 : Calcula a area periferica }
{ irregular das tampas. }
{ }
{ Procedures : - }
{ Functions : CalcArea1 }
{ Var Global : - }
{ Cte Global : NDivLateral }
{-----}

```

```
Function CalcArea3 (RaioMaior,
                   RaioMenor : Real): Real;
```

```
var
  AngSetor : Real;
```

```
begin
  AngSetor := 2 * Pi / NDivlateral;
  CalcArea3 := (Sqr(RaioMaior) * Sin(AngSetor) * Cos(AngSetor)) / 2 -
               CalcArea1(RaioMenor) / NDivLateral;
end;
```



```

{-----}
{ CalcArea4 : Calcula a area do sorvedouro. }
{
  Procedures : -
  Functions  : -
  Var Global : -
  Cte Global : NDivLateral, DeltaZ
}
{-----}

```

```

Function CalcArea4 (RaioMaior,
                   DeltaZ    : Real) : Real;
begin
  CalcArea4 := RaioMaior * Sin(2 * Pi / NDivLateral) * DeltaZ;
end;

```

```

{-----}
{ CalcArea5 : Calcula a area lateral. }
{
  Procedures : -
  Functions  : -
  Var Global : -
  Cte Global : NDivLateral, DeltaZ
}
{-----}

```

```

Function CalcArea5 (RaioMaior,
                   DeltaZ    : Real) : Real;
begin
  CalcArea5 := 2 * Pi * RaioMaior * DeltaZ / NDivLateral;
end;

```

```

{-----}
{ CalcArea : Devolve um vetor contendo a area de todos os tipos de }
{            superficies existentes na fornalha.                     }
{
  Procedures : -
  Functions  : CalcArea1, CalcArea2, CalcArea3, CalcArea4, CalcArea5
  Var Global : -
  Cte Global : -
}
{-----}

```

```

Procedure CalcArea (RaioMaior,
                   RaioMenor,
                   DeltaZ    : Real;
                   Var
                     Area    : Vetor5);

```

```

begin
  Area[1] := CalcArea1 (RaioMenor);
  Area[2] := CalcArea2 (RaioMaior, RaioMenor);
  Area[3] := CalcArea3 (RaioMaior, RaioMenor);
  Area[4] := CalcArea4 (RaioMaior, DeltaZ);
  Area[5] := CalcArea5 (RaioMaior, DeltaZ);
end;

```

```

{-----}
{ ConvElemPosSup : Converte o numero do elemento superficie para as }
{ coordenadas Secao e Setor. }
{ }
{ }
{   Procedures : - }
{   Functions  : - }
{   Var Global : - }
{   Cte Global : NDivLateral,NSecao }
{-----}

```

```

Procedure ConvElemPosSup(Elemento : Byte;
                          Var
                            Secao,
                            Setor  : Byte);

```

```

Var
  a : Byte;

begin
  if (Elemento > NDivLateral) and
     (Elemento < (NSecao * NDivLateral + 1))
  then
    begin
      Secao:= (Elemento - 1 ) div NDivLateral;
      a:= Elemento mod NDivLateral;
      if a = 0 then Setor:= NdivLateral
        else Setor:= a;
    end
  else
    begin
      if Elemento <= NDivLateral
      then
        begin
          Secao:= 0;
          Setor:= Elemento;
        end
      else
        begin
          Secao:= NSecao;
          if Secao = NSecao
          then Setor:= Elemento - (NSecao* NDivLateral) - 1
            else Setor:= Elemento - (NSecao* NDivLateral);
          end
        end
    end
  end;

```

```

{-----}
{ ConvElemPosVol : Converte o numero do elemento volume      para as }
{ coordenadas Secao e Setor. }
{ }
{ }
{   Procedures : - }
{   Functions  : - }
{   Var Global : - }
{   Cte Global : NDivLateral }
{-----}

```

```

Procedure ConvElemPosVol(Elemento : Byte;
                          Var
                            Secao,
                            Setor  : Byte);

```

```

begin
  Secao:= Elemento div (NDivLateral + 1) + 1;
  Setor:= Elemento mod (NDivLateral + 1);
end;

```

```

{-----}
{ ConvPosElemSup : Converte as coordenadas Secao e Setor para o numero do }
{ elemento superficie. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : NDivLateral }
{-----}

```

```

Procedure ConvPosElemSup(Secao,
                        Setor      : Byte;
                        Var
                        Elemento : Byte);

begin
  if Secao = NSecao
  then
    Elemento:= NDivLateral * Secao + Setor + 1
  else
    Elemento:= NDivLateral * Secao + Setor;
end;

```

```

{-----}
{ ConvPosElemVol : Converte as coordenadas Secao e Setor para o numero do }
{ elemento volume. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : NDivLateral }
{-----}

```

```

Procedure ConvPosElemVol(Secao,
                        Setor      : Byte;
                        Var
                        Elemento : Byte);

begin
  Elemento:= (Secao - 1) * (NDivLateral + 1) + Setor;
end;

```

```

{-----}
{ PosSimSetor : Calcula a posicao simetrica do setor. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : NDivLateral }
{-----}

```

```

Procedure PosSimSetor(Setor      : Byte;
                     Var
                     SimSetor : Byte);

begin
  if Setor <> 0 then SimSetor:= NDivLateral - Setor + 1
  else SimSetor:= Setor;
end;

```



```

{-----}
{ VerifTypeArea : Identifica o tipo de superficie.
{
{   Procedures : ConvElemPosSup
{   Functions  : -
{   Var Global : -
{   Cte Global : -
{-----}

```

```
Function VerifTypeArea(NumSup : Byte) : Byte;
```

```

var
  Secao,
  Setor : Byte;

begin
  ConvElemPosSup (NumSup, Secao, Setor);
  if (Secao = 0) or (Secao = 11)      { superficies na tampa }
  then
    case Setor of
      0       : VerifTypeArea := 1;
      1..3,6..8 : VerifTypeArea := 2;
      4,5     : VerifTypeArea := 3;
    end
  else                                { superficies laterais }
    case Setor of
      1..3,6..8 : VerifTypeArea := 5;
      4,5       : VerifTypeArea := 4;
    end;
end;
```

```

{-----}
{ Emiss : Define a emissividade para os elementos de superficie da }
{ da fornalha. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : - }
{-----}

```

```
Function Emiss(NumSup : Byte) : Real;
```

```

begin
  case NumSup of
    0..88 : Emiss:= 0.65;
    89    : Emiss:= 1.0;
    90..97 : Emiss:= 0.85;
  end
end;
```

```

{-----}
{ Reflet : Define a refletividade para os elementos de superficie da }
{ da fornalha. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : - }
{-----}

```

```
Function Reflet (NumSup : Byte) : Real;
```

```

begin
  case NumSup of
    0..88 : Reflet:= 0.35;
    89    : Reflet:= 0.0;
    90..97 : Reflet:= 0.15;
  end
end;
```

```

{-----}
{ Es : Calcula o poder emissivo da superficie negra. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : - }
{-----}

```

```
Function Es ( Temperatura : Real ) : Real;
```

```

const
  Sigma = 5.669E-8;      { Constante de Boltzmann [W / m2 T4 ] }
```

```

begin
  Es := Sigma * Exp(4 * Ln(Temperatura));
end;
```

```
{-----}
{ Eg : Calcula o poder emissivo do volume negro. }
{ }
{ Procedures : - }
{ Functions : - }
{ Var Global : - }
{ Cte Global : - }
{-----}
```

```
Function Eg ( Temperatura : Real ) : Real;
```

```
const
```

```
    Sigma = 5.669E-8;      { Constante de Boltzmann [W / m2 T4 ] }
```

```
begin
```

```
    Eg := Sigma * Exp(4 * Ln(Temperatura));
```

```
end;
```



```

{-----}
{  CompMatTransfer : Compoe a matriz de transferencia. }
{ }
{  Procedures : CalcArea }
{  Functions : - }
{  Var Global : - }
{  Cte Global : - }
{-----}

```

```

Procedure CompMatTransfer(Var Matriz : TMatriz;
                          Dimen      : Byte );

```

```

var
  TypeArea : Byte;
  Area     : Vetor5;
  Erro     : Boolean;

```

```

{-----}
{  LessLoFile : Faz a leitura do arquivo externo }
{              ssLoFile e armazena os valores   }
{              nele contidos nas posicoes cor-  }
{              retas na matriz de Transferencia. }
{ }
{  Procedures : - }
{  Functions : - }
{  Var Global : Matriz (CompMatTransfer) }
{  Cte Global : - }
{-----}

```

```

Procedure LessLoFile;

```

```

var
  OutFile : Text;
  ElemI,
  ElemJ,
  Cont    : Byte;
  Result  : Real;

```

```

begin
  Cont:= 0;
  assign (OutFile,'ssLoFile.txt');
  reset (OutFile);
  while not(eof(OutFile)) do
    begin
      if Cont <= 2
      then
        begin
          read(OutFile,ElemI,ElemJ,Result);
          Cont:= Cont + 1;
        end
      else
        begin
          readln(OutFile,ElemI,ElemJ,Result);
          Cont:= 0;
        end;
      Matriz[ElemI,ElemJ]:= Result;
    end; { while }
  close (OutFile);
end; { LessloFile }

```

```

{-----}
{ CalcSimSup : Uma vez calculado as areas      }
{      diretas de troca de calor para        }
{      para o minimo necessario de            }
{      pares de superficies, faz-se           }
{      a identificacao dos pares              }
{      simetricos, que recebem os             }
{      mesmos valores.                        }
{-----}
{ Procedures : ConvElemPosSup,PosSimSetor      }
{ Functions  : -                               }
{ Var Global : Matriz,Dimen (CompMatTransfer) }
{ Cte Global : -                               }
{-----}

```

```

Procedure CalcSimSup;

```

```

var
  SimSetorI,
  SimSetorJ,
  SimLin,
  SimCol : Byte;

```

```

begin
  for Lin:= 0 to Dimen do
    for Col:= 0 to Dimen do
      begin
        if Matriz[Lin,Col] <> 0
          then
            begin
              ConvElemPosSup(Lin,SecaoI,SetorI);
              PosSimSetor(SetorI,SimSetorI);
              ConvElemPosSup(Col,SecaoJ,SetorJ);
              PosSimSetor(SetorJ,SimSetorJ);
              ConvPosElemSup(SecaoI,SimSetorI,SimLin);
              ConvPosElemSup(SecaoJ,SimSetorJ,SimCol);
              if SimLin < SimCol then Matriz[SimLin,SimCol]:= Matriz[Lin,Col]
                else Matriz[SimCol,SimLin]:= Matriz[Lin,Col];
            end { if }
          end; { for }
        end; { CalcSimSup }
      end;
    end;
  end;

```

```

begin { Inicio CompMatTransfer }
  FillChar(Matriz,SizeOf(Matriz),0);
  LessLoFile;
  CalcSimSup;
  CalcArea (RForno,RCil,DeltaZ,Area);
  for Lin:= 0 to Dimen do
    for Col:= 0 to Dimen do
      begin
        if Lin = Col
          then
            begin
              TypeArea:= VerifTypeArea(Lin);
              Matriz[Lin,Col]:= Matriz[Lin,Col]
                - Area[TypeArea] / Reflet(Lin);
            end
          else
            Matriz[Col,Lin]:= Matriz[Lin,Col];
          end;
        end;
      end;
    end;
  end; { final CompMatTransfer }

```

```

{-----}
{  CompVetExcitaSup : Compoe o vetor de excitacao, no caso do elemento de }
{                    superficie ser o emissor de energia.                  }
{                                                                           }
{  Procedures : -                                                           }
{  Functions   : VerifTypeArea                                             }
{  Var Global  : -                                                         }
{  Cte Global  : -                                                         }
{-----}

```

```

Procedure CompVetExcitaSup (NumSup      : Byte;
                          Area         : Vetor5;
                          Var
                            Temperatura : TVetor;
                          Var
                            VetExcita   : TVetor);

var
  TypeArea : Byte;

begin
  FillChar(VetExcita,SizeOf(VetExcita),0);
  TypeArea:= VerifTypeArea(NumSup);
  VetExcita[NumSup]:= - Area[TypeArea] * Emiss(NumSup)
                      * Es(Temperatura[NumSup]) / Reflet(NumSup);
end;

```

```

{-----}
{  CompVetExcitaVol : Compoe o vetor de excitacao, no caso de elemento de }
{                    volume sendo o emissor de energia.                    }
{                                                                           }
{  Procedures : ConvElemPosSup, PosSimSetor                                }
{  Functions   : -                                                         }
{  Var Global  : -                                                         }
{  Cte Global  : -                                                         }
{-----}

```

```

Procedure CompVetExcitaVol (ColExcita   : Byte;
                          TempVet      : TVetor;
                          Var
                            VetExcita   : TVetor );

var
  OutFile  : Text;
  ElemI,
  ElemJ,
  Cont,
  ElemSup,
  SimSetorI,
  SimSetorJ,
  SimElemJ : Byte;
  sgLo      : Real;

begin
  FillChar (VetExcita,SizeOf(VetExcita),0); {zera o vetor de excitacao}
  ConvElemPosVol (ColExcita,Secao,Setor);
  PosSimSetor (Setor,SimSetorJ);
  ConvPosElemVol (Secao,SimSetorJ,SimElemJ);
  assign (OutFile,'sgLoFile.txt');
  reset (OutFile);
  Cont:= 0;
  while not(eof(OutFile)) do
  begin
    if Cont <= 2
    then

```



```

begin
  read (OutFile,ElemI,ElemJ,sgLo);
  Cont:= Cont + 1;
end
else
begin
  readln (OutFile,ElemI,ElemJ,sgLo);
  Cont:= 0;
end;
if ElemJ = ColExcita
then VetExcita[ElemI]:= - sgLo * Eg(TempVet[ElemJ]);
if ElemJ = SimElemJ
then
begin
  ConvElemPosSup(ElemI,Secao,Setor);
  PosSimSetor(Setor,SimSetorI);
  ConvPosElemSup(Secao,SimSetorI,ElemI);
  VetExcita[ElemI]:= - sgLo * Eg(TempVet[ElemJ]);
end;
end;
close (OutFile);
end;

```

```

{-----}
{ IniPerfilTempForno : Inicializa uma distribuicao de temperaturas para }
{ os elementos de superficie e de volume. Cria - }
{ se dois arquivos : TempSup e TempVol, temperatu- }
{ ras nos elementos de superficie e de volume do }
{ forno respectivamente. }
{ }
{ Procedures : ConvElemPosSup, ConvElemPosVol }
{ Functions : - }
{ Var Global : - }
{ Cte Global : - }
{-----}

Procedure IniPerfilTempForno;

const
  ColFile = 7; { Define o numero de colunas no arquivo }

var
  OutFile1,
  OutFile2 : Text;
  Cont,
  Secao,
  Setor,
  Elem : Byte;
  IniTemp : Real;

begin
  Cont:= 0;
  assign (OutFile1,'TempSup.txt');
  rewrite (OutFile1);
  writeln (OutFile1,ColFile);
  for Elem := 0 to 97 do
    begin
      ConvElemPosSup (Elem,Secao,Setor);
      case Setor of
        0..3,6..8 : IniTemp:= 1600.0;
        4,5 : IniTemp:= 1450.0;
      end;
      if Cont <= ColFile
      then
        begin
          write (OutFile1,IniTemp:8:1);
          Cont:= Cont + 1;
        end
      else
        begin
          writeln (OutFile1,IniTemp:8:1);
          Cont:= 0;
        end;
      end;
    end;
  close (OutFile1);
  Cont:= 0;
  assign (OutFile2,'TempVol.txt');
  rewrite (OutFile2);
  writeln (OutFile2,ColFile);
  for Elem:= 0 to 89 do
    begin
      ConvElemPosVol (Elem,Secao,Setor);
      if Setor <> 0 then IniTemp:= 1400.0
        else IniTemp:= 2000.0 - 526.0 * (1 - Exp(- 0.3 * Secao));
      if Cont <= ColFile
      then
        begin

```

```
        write (OutFile2,IniTemp:8:1);  
        Cont:= Cont + 1;  
    end  
    else  
        begin  
            writeln (OutFile2,IniTemp:8:1);  
            Cont:= 0;  
        end;  
    end;  
    close (OutFile2);  
end;
```



```

{-----}
{ LePerfilTempFile : Faz a leitura do arquivo Temp*.txt que contem o
{                   o perfil de temperatura do forno para os elementos
{                   de volume ou de superficie.
{
{   Procedures : -
{   Functions  : -
{   Var Global : -
{   Cte Global : -
{-----}

Procedure LePerfilTempFile (FileName  : String;
                           Var
                           TempVet   : TVetor);

var
  OutFile  : Text;
  Cont,
  Elem     : Byte;
  a : Real;

begin
  Elem:= 0;
  Cont:= 0;
  assign(OutFile,FileName);
  reset(OutFile);
  while not(eof(OutFile)) do
    begin
      if Cont <= 13
      then
        begin
          read(OutFile,TempVet[Elem]);
          Cont:= Cont + 1;
        end
      else
        begin
          readln(OutFile,TempVet[Elem]);
          Cont:= 0;
        end;
      Elem:= Elem + 1;
    end;
  close(OutFile);
end;

```

```

{-----}
{ Inverte: Faz a troca dos indices I e J. }
{      As rotinas de calculo das areas   }
{      diretas sao feitas para o par     }
{      tipoI - tipoJ e nao vice versa.   }
{-----}
{ Procedures : - }
{ Functions   : - }
{ Var Global  : - }
{ Cte Global  : - }
{-----}

```

```

Procedure Inverte (Var Secao,
                   Setor : EscVetByte);

```

```

var
  Aux : Byte;

begin
  Aux:= Secao[I];
  Secao[I]:= Secao[J];
  Secao[J]:= Aux;
  Aux:= Setor[I];
  Setor[I]:= Setor[J];
  Setor[J]:= Aux;
end;

```

Parte C

Resultados

Resultados

O programa computacional para o cálculo das áreas diretas e globais de troca de calor gera como saídas 6 arquivos sendo que para:

Áreas Diretas

- 1 - ssLoFile.txt
- 2 - sgLoFile.txt
- 3 - ggLoFile.txt

Áreas Globais

- 4 - SSHiFile.txt
- 5 - GSHiFile.txt
- 6 - GGHFile.txt

À título de demonstração, foi executado o programa com os seguintes parâmetros:

- Raio do Forno = 0.55 m;
- Raio do Cilindro Central = 0.2 m
- Comprimento do Forno = 2.5 m
- Coefficiente de absorção = 0.6
- Coefficiente de Ponderação = 0.8
- Número de variáveis aleatórias no método de Monte Carlo = 5

A seguir estão listados os arquivos com os valores das áreas diretas e globais de alguns pares de elementos (não foi colocado os arquivos completos, devido ao tamanho). Nos arquivos, estão representados os números dos elementos de troca seguido do respectivo valor da área direta ou global.

Arquivo sgLoFile.txt

0	0	1.39502E-03	0	1	3.02897E-04	0	2	7.44991E-04	0	3	7.10751E-04
0	4	1.45364E-03	1	0	3.33922E-04	1	1	1.15894E-04	1	2	3.02205E-05
1	3	1.79968E-05	1	4	4.87756E-05	1	5	3.04684E-05	1	6	1.39103E-05
1	7	1.86903E-05	1	8	1.51216E-05	2	0	7.70160E-05	2	1	1.51216E-05
2	2	1.56332E-05	2	3	1.36199E-05	2	4	1.26962E-04	2	5	4.83646E-05
2	6	2.18633E-05	2	7	2.37744E-05	2	8	3.07184E-05	3	0	2.58190E-04
3	1	2.16992E-05	3	2	2.89858E-05	3	3	3.40032E-05	3	4	7.88264E-05
3	5	4.86553E-05	3	6	1.76081E-05	3	7	3.33467E-05	3	8	3.53494E-05
4	0	8.29713E-04	4	1	6.55133E-05	4	2	1.03879E-04	4	3	2.22349E-04
4	4	3.05159E-04	4	5	1.28255E-04	4	6	8.72267E-05	4	7	5.52167E-05
4	8	6.73690E-05	0	9	1.12301E-03	0	10	3.72789E-04	0	11	4.54480E-04
0	12	6.41602E-04	0	13	1.64120E-03	1	9	1.13833E-04	1	10	7.15103E-05
1	11	6.51006E-05	1	12	4.97388E-05	1	13	7.86567E-05	1	14	2.76986E-05
1	15	5.97022E-05	1	16	7.53933E-05	1	17	1.00614E-04	2	9	1.21557E-04
2	10	1.00614E-04	2	11	1.01876E-04	2	12	9.55024E-05	2	13	8.42707E-05
2	14	5.61627E-05	2	15	4.86804E-05	2	16	4.44134E-05	2	17	5.62338E-05
3	9	1.58195E-04	3	10	5.62338E-05	3	11	7.50244E-05	3	12	1.35329E-04
3	13	4.51151E-05	3	14	2.75380E-05	3	15	5.92399E-05	3	16	5.49905E-05
3	17	6.17888E-05	4	9	4.42050E-04	4	10	8.67396E-05	4	11	1.30252E-04
4	12	1.31009E-04	4	13	1.71618E-04	4	14	1.30839E-04	4	15	9.78087E-05
4	16	6.41089E-05	4	17	6.62152E-05	0	18	4.99676E-04	0	19	3.66616E-04
0	20	4.17471E-04	0	21	3.86486E-04	0	22	3.01480E-04	1	18	7.94818E-05
1	19	7.02778E-05	1	20	6.58927E-05	1	21	5.54416E-05	1	22	3.35891E-05
1	23	2.98618E-05	1	24	4.05836E-05	1	25	4.64860E-05	1	26	5.63952E-05
2	18	6.95187E-05	2	19	6.33084E-05	2	20	6.82796E-05	2	21	6.23627E-05
2	22	6.64069E-05	2	23	5.30252E-05	2	24	3.74278E-05	2	25	4.07278E-05
2	26	4.74393E-05	3	18	6.47662E-05	3	19	4.74393E-05	3	20	5.44901E-05
3	21	5.20566E-05	3	22	7.90720E-05	3	23	5.18536E-06	3	24	3.72728E-05
3	25	3.30617E-05	3	26	3.28313E-05	4	18	2.16035E-04	4	19	6.41308E-05
4	20	8.81661E-05	4	21	1.17697E-04	4	22	4.80072E-05	4	23	2.00325E-05
4	24	6.56003E-05	4	25	5.08479E-05	4	26	4.58316E-05	0	27	4.79731E-04
0	28	3.88248E-04	0	29	4.03086E-04	0	30	4.09024E-04	0	31	3.40946E-04
1	27	7.15096E-05	1	28	5.82273E-05	1	29	3.28410E-05	1	30	2.98734E-05
1	31	9.11900E-06	1	32	8.61661E-06	1	33	2.28819E-05	1	34	2.46665E-05
1	35	4.57014E-05	2	27	6.14528E-05	2	28	4.57014E-05	2	29	4.88291E-05
2	30	4.77785E-05	2	31	1.61383E-05	2	32	1.01309E-05	2	33	2.18219E-05
2	34	2.22462E-05	2	35	2.36974E-05	3	27	2.95792E-05	3	28	2.36974E-05
3	29	2.03432E-05	3	30	2.08475E-05	3	31	7.70906E-06	3	32	2.69756E-05
3	33	1.77191E-05	3	34	1.73443E-05	3	35	2.88039E-05	4	27	3.61968E-05
4	28	2.06417E-05	4	29	2.29660E-05	4	30	2.48595E-05	4	31	7.73362E-06
4	32	1.47769E-05	4	33	5.43882E-05	4	34	4.66944E-05	4	35	4.46050E-05
0	36	1.33858E-04	0	37	8.60660E-05	0	38	8.72124E-05	0	39	8.87993E-05
0	40	1.00372E-04	1	36	1.65483E-05	1	37	2.45531E-05	1	38	2.33997E-05
1	39	2.18489E-05	1	40	5.77383E-06	1	41	9.72151E-06	1	42	2.15502E-05
1	43	2.39446E-05	1	44	2.55605E-05	2	36	3.54756E-05	2	37	2.55605E-05
2	38	2.64368E-05	2	39	2.59091E-05	2	40	1.01294E-05	2	41	1.43091E-05
2	42	2.06877E-05	2	43	2.12714E-05	2	44	2.27793E-05	3	36	2.64821E-05
3	37	1.66115E-05	3	38	1.76055E-05	3	39	1.79811E-05	3	40	1.05987E-05
3	41	2.97824E-06	3	42	2.11090E-05	3	43	2.04137E-05	3	44	2.11396E-05
4	36	3.75195E-05	4	37	1.95111E-05	4	38	1.96239E-05	4	39	2.14914E-05
4	40	1.66753E-05	4	41	1.60542E-05	4	42	1.92690E-05	4	43	1.11735E-05
4	44	1.08659E-05	0	45	8.15996E-05	0	46	1.00020E-04	0	47	9.61277E-05
0	48	2.06583E-04	0	49	4.71684E-05	1	45	3.82148E-05	1	46	2.80176E-05
1	47	2.75394E-05	1	48	2.58548E-05	1	49	3.60031E-06	1	50	3.33746E-06
1	51	1.60045E-05	1	52	1.67329E-05	1	53	1.75250E-05	2	45	1.77186E-05
2	46	2.00883E-05	2	47	2.07395E-05	2	48	2.04649E-05	2	49	9.81712E-06
2	50	8.02399E-06	2	51	1.78475E-05	2	52	1.81923E-05	2	53	1.90375E-05
3	45	2.19208E-05	3	46	1.90375E-05	3	47	2.02154E-05	3	48	1.87939E-05
3	49	9.53956E-06	3	50	5.54203E-06	3	51	1.67781E-05	3	52	1.65320E-05
3	53	1.68933E-05	4	45	1.26487E-05	4	46	7.86848E-06	4	47	8.47795E-06
4	48	9.13428E-06	4	49	6.79290E-06	4	50	5.56352E-06	4	51	1.45302E-05
4	52	1.34415E-05	4	53	1.29728E-05	0	54	1.73860E-04	0	55	7.69985E-05
0	56	7.59767E-05	0	57	7.55348E-05	0	58	1.42969E-05	1	54	1.15268E-05

Arquivo ssLoFile.txt

0 9	1.74843E-03	0 10	3.95456E-03	0 11	9.25119E-04	0 12	8.81194E-04
1 9	3.49566E-04	1 10	3.34780E-04	1 11	1.73106E-04	1 12	7.65683E-05
1 13	3.06605E-05	1 14	2.43624E-05	1 15	9.85377E-06	1 16	5.10558E-06
2 9	4.24161E-04	2 10	3.39829E-04	2 11	2.10656E-04	2 12	1.00632E-04
2 13	3.55432E-05	2 14	2.54241E-05	2 15	1.39799E-05	2 16	1.03113E-05
3 9	2.33416E-04	3 10	5.87437E-04	3 11	3.33353E-04	3 12	1.22626E-04
3 13	4.10608E-05	3 14	2.56172E-05	3 15	1.46637E-05	3 16	7.70955E-06
4 9	1.31110E-04	4 10	5.53917E-04	4 11	4.57009E-04	4 12	7.66273E-05
4 13	2.57393E-05	4 14	2.32194E-05	4 15	1.34563E-05	4 16	6.14764E-06
0 17	2.88103E-03	0 18	2.32164E-03	0 19	1.38428E-03	0 20	5.35931E-04
1 17	4.85062E-04	1 18	5.34685E-04	1 19	2.44404E-04	1 20	9.89575E-05
1 21	6.12049E-05	1 22	5.24386E-05	1 23	2.76725E-05	1 24	6.64402E-06
2 17	2.89075E-04	2 18	4.74597E-04	2 19	2.12568E-04	2 20	1.03355E-04
2 21	3.36257E-05	2 22	1.52948E-05	2 23	1.32629E-05	2 24	6.94069E-06
3 17	1.13509E-04	3 18	5.45006E-04	3 19	3.04474E-04	3 20	7.83856E-05
3 21	1.85964E-05	3 22	2.74538E-05	3 23	1.56882E-05	3 24	8.75074E-06
4 17	1.27394E-04	4 18	3.20430E-04	4 19	2.50474E-04	4 20	7.53759E-05
4 21	3.32433E-05	4 22	3.21239E-05	4 23	1.86374E-05	4 24	1.11566E-05
0 25	1.38796E-03	0 26	1.17034E-03	0 27	2.28242E-03	0 28	4.83645E-04
1 25	2.58093E-04	1 26	2.96213E-04	1 27	1.44512E-04	1 28	9.57257E-05
1 29	3.92083E-05	1 30	2.24058E-05	1 31	1.17838E-05	1 32	6.15273E-06
2 25	4.05561E-04	2 26	7.82956E-04	2 27	2.85975E-04	2 28	1.04915E-04
2 29	3.76673E-05	2 30	2.00353E-05	2 31	1.09692E-05	2 32	5.87918E-06
3 25	1.60496E-04	3 26	3.54624E-04	3 27	1.94068E-04	3 28	9.97356E-05
3 29	1.82701E-05	3 30	2.44476E-05	3 31	1.37790E-05	3 32	7.62046E-06
4 25	7.10694E-05	4 26	2.45456E-04	4 27	1.96959E-04	4 28	7.01266E-05
4 29	2.77783E-05	4 30	2.08887E-05	4 31	1.18479E-05	4 32	6.86383E-06
0 33	2.71238E-03	0 34	2.78797E-03	0 35	1.40414E-03	0 36	8.24866E-04
1 33	8.95983E-04	1 34	4.67383E-04	1 35	1.95074E-04	1 36	5.62982E-05
1 37	5.63697E-05	1 38	3.58626E-05	1 39	1.89080E-05	1 40	9.69310E-06
2 33	3.58413E-04	2 34	5.61775E-04	2 35	2.14304E-04	2 36	9.89204E-05
2 37	4.03968E-05	2 38	2.64085E-05	2 39	1.44256E-05	2 40	7.45443E-06
3 33	1.45945E-04	3 34	3.78684E-04	3 35	2.07058E-04	3 36	9.47996E-05
3 37	3.62637E-05	3 38	3.25581E-05	3 39	1.89260E-05	3 40	1.08998E-05
4 33	9.77636E-05	4 34	1.63739E-04	4 35	1.26129E-04	4 36	4.70793E-05
4 37	1.99064E-05	4 38	1.81577E-05	4 39	1.06926E-05	4 40	6.43686E-06
0 41	3.15106E-03	0 42	2.05125E-03	0 43	1.18073E-03	0 44	8.18581E-04
1 41	1.95536E-04	1 42	3.76265E-04	1 43	1.57142E-04	1 44	4.12453E-05
1 45	1.65597E-05	1 46	2.29785E-05	1 47	1.50510E-05	1 48	7.47989E-06
2 41	4.62719E-04	2 42	6.15313E-04	2 43	1.12617E-04	2 44	1.05620E-04
2 45	3.70376E-05	2 46	1.47242E-05	2 47	1.08245E-05	2 48	5.91749E-06
3 41	1.48338E-04	3 42	2.76426E-04	3 43	1.28244E-04	3 44	9.76661E-05
3 45	3.13734E-05	3 46	1.57331E-05	3 47	1.83749E-05	3 48	1.03183E-05
4 41	1.35521E-04	4 42	3.66575E-04	4 43	2.76172E-04	4 44	9.78596E-05
4 45	3.47567E-05	4 46	3.46613E-05	4 47	1.65171E-05	4 48	9.71204E-06
0 49	1.05458E-03	0 50	2.46980E-03	0 51	1.04723E-03	0 52	8.96460E-04
1 49	2.91364E-04	1 50	3.08232E-04	1 51	1.48727E-04	1 52	8.74603E-05
1 53	4.52105E-05	1 54	2.89111E-05	1 55	1.54334E-05	1 56	8.33098E-06
2 49	4.52939E-04	2 50	6.40589E-04	2 51	2.60071E-04	2 52	4.91074E-05
2 53	1.98472E-05	2 54	2.16930E-05	2 55	1.17908E-05	2 56	1.00093E-05
3 49	2.06910E-04	3 50	5.34827E-04	3 51	3.24057E-04	3 52	7.62774E-05
3 53	4.41286E-05	3 54	4.82698E-05	3 55	2.77860E-05	3 56	1.56510E-05
4 49	6.63290E-05	4 50	1.86642E-04	4 51	1.74139E-04	4 52	6.56219E-05
4 53	2.10066E-05	4 54	1.93386E-05	4 55	1.11332E-05	4 56	9.23503E-06
0 57	6.50907E-04	0 58	1.63300E-03	0 59	1.48677E-03	0 60	5.26924E-04
1 57	6.53708E-04	1 58	5.89112E-04	1 59	2.56615E-04	1 60	6.79150E-05
1 61	3.50162E-05	1 62	2.68622E-05	1 63	1.39141E-05	1 64	7.15785E-06
2 57	5.02167E-04	2 58	4.67452E-04	2 59	1.92495E-04	2 60	5.14480E-05
2 61	4.29389E-05	2 62	3.83868E-05	2 63	2.13043E-05	2 64	1.11960E-05
3 57	1.33112E-04	3 58	2.66307E-04	3 59	1.50421E-04	3 60	4.41849E-05
3 61	1.49728E-05	3 62	2.86365E-05	3 63	1.64148E-05	3 64	9.33003E-06

Arquivo ggLoFile.txt

0	1	2.39134E-04	0	2	4.22352E-04	0	3	7.34620E-04	0	4	1.21948E-03
1	2	7.52353E-04	1	3	2.39920E-04	1	4	1.67075E-04	1	5	5.67007E-05
1	6	8.97316E-05	1	7	1.29029E-04	1	8	4.30763E-04	2	3	1.11796E-03
2	4	1.84450E-04	2	5	1.87752E-04	2	6	6.70341E-05	2	7	8.13071E-05
2	8	1.49348E-04	3	4	7.44693E-05	3	5	4.75064E-05	3	6	8.55890E-05
3	7	7.78889E-05	3	8	8.35965E-05	4	5	1.00106E-04	4	6	5.02679E-04
4	7	1.46415E-04	4	8	1.56575E-05	0	9	1.54803E-03	0	10	1.77476E-04
0	11	7.50502E-04	0	12	2.61484E-04	0	13	1.69734E-06	1	9	3.21165E-04
1	10	3.58875E-04	1	11	3.00430E-04	1	12	1.56012E-04	1	13	1.34374E-04
1	14	5.54053E-05	1	15	6.94632E-05	1	16	1.50666E-04	1	17	2.26469E-04
2	9	1.97419E-04	2	10	4.17165E-04	2	11	3.30106E-04	2	12	3.15880E-04
2	13	1.84818E-04	2	14	4.97277E-05	2	15	6.13933E-05	2	16	8.38000E-05
2	17	1.25031E-04	3	9	5.46647E-04	3	10	1.56950E-04	3	11	8.03944E-04
3	12	6.25824E-04	3	13	4.68641E-04	3	14	6.31888E-04	3	15	8.08521E-05
3	16	6.05221E-05	3	17	8.69930E-05	4	9	6.81589E-05	4	10	1.47613E-04
4	11	1.02115E-04	4	12	8.92067E-05	4	13	6.37115E-04	4	14	5.25609E-05
4	15	3.13409E-04	4	16	2.49668E-04	4	17	6.65929E-05	0	18	4.76606E-04
0	19	1.58955E-04	0	20	1.97817E-04	0	21	1.75825E-04	0	22	8.37460E-05
1	18	9.66202E-05	1	19	2.94469E-04	1	20	1.33401E-04	1	21	9.43237E-05
1	22	5.63518E-05	1	23	4.83624E-05	1	24	5.65948E-05	1	25	8.15183E-05
1	26	1.46710E-04	2	18	2.37922E-04	2	19	2.14813E-04	2	20	2.48385E-04
2	21	1.15321E-04	2	22	1.26655E-04	2	23	1.03973E-04	2	24	4.95590E-05
2	25	5.08861E-05	2	26	6.82234E-05	3	18	1.33173E-04	3	19	8.49609E-05
3	20	1.58548E-04	3	21	2.11096E-04	3	22	1.26473E-04	3	23	1.06310E-04
3	24	4.66922E-05	3	25	3.91053E-05	3	26	5.27716E-05	4	18	1.36453E-04
4	19	5.05783E-04	4	20	1.30771E-04	4	21	1.02502E-05	4	22	3.67071E-05
4	23	6.97301E-05	4	24	1.87182E-04	4	25	1.07566E-04	4	26	6.21743E-05
0	27	1.44885E-04	0	28	5.32444E-05	0	29	9.02634E-05	0	30	4.22590E-05
0	31	7.66282E-05	1	27	5.22608E-05	1	28	6.26744E-05	1	29	5.56443E-05
1	30	4.87367E-05	1	31	4.10596E-05	1	32	6.47413E-05	1	33	2.92693E-05
1	34	3.82636E-05	1	35	7.16738E-05	2	27	6.53875E-05	2	28	5.49346E-05
2	29	9.81761E-05	2	30	7.72803E-05	2	31	1.81401E-05	2	32	2.05861E-05
2	33	2.53462E-05	2	34	3.45688E-05	2	35	3.89288E-05	3	27	3.47589E-05
3	28	4.75117E-05	3	29	9.61011E-05	3	30	7.47231E-05	3	31	2.44791E-05
3	32	2.76238E-05	3	33	3.99421E-05	3	34	3.31073E-05	3	35	2.72209E-05
4	27	1.90937E-04	4	28	4.76321E-04	4	29	6.30338E-05	4	30	7.49657E-05
4	31	3.99519E-05	4	32	1.75471E-05	4	33	4.29642E-04	4	34	2.58568E-04
4	35	1.14149E-04	0	36	3.81361E-05	0	37	2.42600E-05	0	38	2.15971E-05
0	39	3.97445E-05	0	40	3.78414E-06	1	36	2.41504E-05	1	37	3.81442E-05
1	38	2.38261E-05	1	39	2.07509E-05	1	40	1.42165E-05	1	41	1.64055E-05
1	42	1.90821E-05	1	43	2.54512E-05	1	44	2.45995E-05	2	36	4.31182E-05
2	37	3.86054E-05	2	38	3.76403E-05	2	39	3.85097E-05	2	40	1.59500E-05
2	41	8.11862E-06	2	42	1.75253E-05	2	43	1.73500E-05	2	44	2.23091E-05
3	36	4.26454E-05	3	37	3.51858E-05	3	38	2.53606E-05	3	39	2.72283E-05
3	40	2.02451E-05	3	41	2.44013E-05	3	42	1.79426E-05	3	43	1.50526E-05
3	44	1.89089E-05	4	36	1.00856E-04	4	37	3.41112E-04	4	38	1.25579E-04
4	39	1.59161E-05	4	40	2.88665E-05	4	41	1.38134E-05	4	42	3.69442E-04
4	43	1.00517E-04	4	44	3.36799E-05	0	45	3.40299E-05	0	46	2.14980E-05
0	47	1.46001E-05	0	48	2.29092E-05	0	49	8.59488E-06	1	45	2.04612E-05
1	46	2.60465E-05	1	47	2.20543E-05	1	48	1.71025E-05	1	49	4.41046E-06
1	50	1.28934E-05	1	51	9.94974E-06	1	52	1.51159E-05	1	53	1.95491E-05
2	45	2.82462E-05	2	46	1.39614E-05	2	47	1.92276E-05	2	48	1.05272E-05
2	49	1.20737E-05	2	50	1.04972E-05	2	51	9.52491E-06	2	52	1.23895E-05
2	53	1.21064E-05	3	45	1.99204E-05	3	46	1.69657E-05	3	47	1.88334E-05
3	48	1.83094E-05	3	49	1.12016E-05	3	50	6.23914E-06	3	51	1.26352E-05
3	52	1.12081E-05	3	53	8.03630E-06	4	45	4.37587E-04	4	46	7.68793E-04
4	47	1.33352E-04	4	48	9.60512E-05	4	49	4.90331E-06	4	50	9.51634E-06
4	51	2.40708E-04	4	52	1.57239E-04	4	53	1.26690E-04	0	54	2.86682E-05
0	55	7.96268E-06	0	56	9.20258E-06	0	57	1.17352E-05	0	58	6.52275E-06
1	54	7.72527E-06	1	55	1.52281E-05	1	56	8.55854E-06	1	57	8.46451E-06

Arquivo SSHiFile.txt

0	0	6.45574E+01	0	1	3.76008E-02	0	2	3.15267E-02	0	3	2.97301E-02
0	4	2.14879E-02	0	5	2.14880E-02	0	6	2.97301E-02	0	7	3.15268E-02
0	8	3.76009E-02	0	9	1.72031E+00	0	10	9.54486E-01	0	11	6.62145E-01
0	12	1.68957E-01	0	13	1.68957E-01	0	14	6.62145E-01	0	15	9.54486E-01
0	16	1.72031E+00	0	17	7.03734E-01	0	18	8.14631E-01	0	19	5.27366E-01
0	20	1.76918E-01	0	21	1.76918E-01	0	22	5.27366E-01	0	23	8.14631E-01
0	24	7.03734E-01	0	25	5.92969E-01	0	26	8.60175E-01	0	27	6.71245E-01
0	28	2.22342E-01	0	29	2.22342E-01	0	30	6.71245E-01	0	31	8.60176E-01
0	32	5.92970E-01	0	33	7.17939E-01	0	34	9.61470E-01	0	35	5.61221E-01
0	36	2.12945E-01	0	37	2.12945E-01	0	38	5.61221E-01	0	39	9.61470E-01
0	40	7.17939E-01	0	41	3.13365E-01	0	42	9.62327E-01	0	43	5.89833E-01
0	44	2.70164E-01	0	45	2.70164E-01	0	46	5.89833E-01	0	47	9.62327E-01
0	48	3.13365E-01	0	49	9.21923E-01	0	50	1.03886E+00	0	51	7.10752E-01
0	52	2.73361E-01	0	53	2.73361E-01	0	54	7.10752E-01	0	55	1.03886E+00
0	56	9.21923E-01	0	57	5.85126E-01	0	58	1.06173E+00	0	59	4.22573E-01
0	60	1.88668E-01	0	61	1.88668E-01	0	62	4.22573E-01	0	63	1.06174E+00
0	64	5.85126E-01	0	65	4.26877E-01	0	66	1.00304E+00	0	67	7.31147E-01
0	68	2.95872E-01	0	69	2.95871E-01	0	70	7.31147E-01	0	71	1.00304E+00
0	72	4.26877E-01	0	73	1.29285E+00	0	74	9.39631E-01	0	75	7.58651E-01
0	76	3.65110E-01	0	77	3.65110E-01	0	78	7.58651E-01	0	79	9.39631E-01
0	80	1.29285E+00	0	81	9.06606E-01	0	82	1.90472E+00	0	83	5.38343E-01
0	84	4.23833E-01	0	85	4.23833E-01	0	86	5.38343E-01	0	87	1.90472E+00
0	88	9.06606E-01	0	89	2.51005E-01	0	90	3.22731E-02	0	91	2.57042E-02
0	92	2.49236E-02	0	93	2.52870E-02	0	94	2.52870E-02	0	95	2.49236E-02
0	96	2.57042E-02	0	97	3.22731E-02	1	0	3.76016E-02	1	1	5.25614E+01
1	2	4.98911E-03	1	3	4.10764E-03	1	4	2.88920E-03	1	5	4.91158E-04
1	6	5.95120E-04	1	7	6.06092E-04	1	8	6.00327E-04	1	9	1.94436E-01
1	10	1.60329E-01	1	11	8.64209E-02	1	12	1.88726E-02	1	13	1.31299E-02
1	14	1.26214E-02	1	15	7.23983E-03	1	16	5.15805E-03	1	17	1.37584E-01
1	18	1.18542E-01	1	19	6.20253E-02	1	20	2.25129E-02	1	21	1.39902E-02
1	22	1.59791E-02	1	23	7.93220E-03	1	24	4.08010E-03	1	25	2.15296E-01
1	26	9.87436E-02	1	27	9.80934E-02	1	28	4.22062E-02	1	29	1.57265E-02
1	30	1.77622E-02	1	31	7.67878E-03	1	32	4.30450E-03	1	33	1.66236E-01
1	34	8.26020E-02	1	35	9.58233E-02	1	36	2.66801E-02	1	37	1.18363E-02
1	38	1.46109E-02	1	39	9.07667E-03	1	40	4.81872E-03	1	41	2.09352E-01
1	42	1.67208E-01	1	43	7.64188E-02	1	44	4.47182E-02	1	45	1.00907E-02
1	46	1.21421E-02	1	47	9.31338E-03	1	48	3.42591E-03	1	49	3.40156E-01
1	50	1.34376E-01	1	51	5.48114E-02	1	52	3.12723E-02	1	53	2.11404E-02
1	54	1.40892E-02	1	55	7.29364E-03	1	56	5.41814E-03	1	57	2.34516E-01
1	58	1.15657E-01	1	59	5.90960E-02	1	60	3.55542E-02	1	61	1.35674E-02
1	62	1.41415E-02	1	63	8.76210E-03	1	64	4.42854E-03	1	65	1.46156E-01
1	66	2.10958E-01	1	67	6.02987E-02	1	68	3.52024E-02	1	69	1.86771E-02
1	70	1.02572E-02	1	71	8.26482E-03	1	72	5.02895E-03	1	73	2.20446E-01
1	74	1.05384E-01	1	75	7.46478E-02	1	76	3.94977E-02	1	77	1.67222E-02
1	78	1.54314E-02	1	79	7.98277E-03	1	80	3.43902E-03	1	81	1.62321E-01
1	82	1.12301E-01	1	83	7.61004E-02	1	84	3.46877E-02	1	85	1.53442E-02
1	86	1.45445E-02	1	87	8.07560E-03	1	88	4.24146E-03	1	89	4.37532E-02
1	90	3.45300E-03	1	91	1.79727E-03	1	92	1.91116E-03	1	93	2.43919E-03
1	94	2.83857E-03	1	95	4.75865E-03	1	96	4.89598E-03	1	97	3.17277E-03
2	0	3.15274E-02	2	1	4.98911E-03	2	2	5.25592E+01	2	3	3.89190E-03
2	4	2.73361E-03	2	5	4.96513E-04	2	6	5.90305E-04	2	7	6.00312E-04
2	8	6.06092E-04	2	9	8.24921E-02	2	10	1.30479E-01	2	11	9.47837E-02
2	12	2.59963E-02	2	13	8.20273E-03	2	14	1.73208E-02	2	15	1.06649E-02
2	16	5.06292E-03	2	17	1.32163E-01	2	18	1.75676E-01	2	19	7.54619E-02
2	20	3.04349E-02	2	21	1.68229E-02	2	22	1.45333E-02	2	23	8.14062E-03
2	24	6.18002E-03	2	25	1.19753E-01	2	26	1.80848E-01	2	27	4.99618E-02
2	28	2.88695E-02	2	29	8.79115E-03	2	30	1.70159E-02	2	31	6.85865E-03
2	32	4.33132E-03	2	33	1.31557E-01	2	34	1.24849E-01	2	35	8.00759E-02
2	36	5.20848E-02	2	37	8.26389E-03	2	38	1.12140E-02	2	39	9.70404E-03
2	40	5.19416E-03	2	41	1.60654E-01	2	42	1.39231E-01	2	43	6.31261E-02
2	44	2.77894E-02	2	45	1.94954E-02	2	46	1.57075E-02	2	47	6.63121E-03

Arquivo GSHiFile.txt

0	0	2.54588E+00	0	1	3.93231E-01	0	2	1.78058E-01	0	3	2.34424E-01
0	4	4.86590E-01	0	5	4.86590E-01	0	6	2.34424E-01	0	7	1.78058E-01
0	8	3.93232E-01	0	9	6.13141E+00	0	10	2.13813E+00	0	11	6.01356E-01
0	12	1.63316E-01	0	13	1.63311E-01	0	14	6.01356E-01	0	15	2.13813E+00
0	16	6.13142E+00	0	17	4.78700E+00	0	18	1.42850E+00	0	19	7.88890E-01
0	20	2.82356E-01	0	21	2.82355E-01	0	22	7.88890E-01	0	23	1.42850E+00
0	24	4.78701E+00	0	25	5.40386E+00	0	26	1.93928E+00	0	27	7.38115E-01
0	28	1.94343E-01	0	29	1.94341E-01	0	30	7.38115E-01	0	31	1.93928E+00
0	32	5.40386E+00	0	33	2.10352E+01	0	34	8.70568E-01	0	35	9.17261E-01
0	36	2.97555E-01	0	37	2.97554E-01	0	38	9.17261E-01	0	39	8.70570E-01
0	40	2.10352E+01	0	41	4.49301E+00	0	42	4.70281E+00	0	43	6.05818E-01
0	44	3.99074E-01	0	45	3.99073E-01	0	46	6.05818E-01	0	47	4.70281E+00
0	48	4.49301E+00	0	49	1.23069E+01	0	50	1.40450E+00	0	51	7.56585E-01
0	52	3.95266E-01	0	53	3.95265E-01	0	54	7.56586E-01	0	55	1.40450E+00
0	56	1.23069E+01	0	57	1.15555E+01	0	58	1.20184E+00	0	59	6.96025E-01
0	60	3.39946E-01	0	61	3.39945E-01	0	62	6.96025E-01	0	63	1.20184E+00
0	64	1.15555E+01	0	65	8.36505E+00	0	66	1.16245E+00	0	67	6.22343E-01
0	68	3.97541E-01	0	69	3.97541E-01	0	70	6.22343E-01	0	71	1.16245E+00
0	72	8.36505E+00	0	73	8.22337E+00	0	74	1.75920E+00	0	75	5.06782E-01
0	76	4.67797E-01	0	77	4.67796E-01	0	78	5.06782E-01	0	79	1.75920E+00
0	80	8.22337E+00	0	81	4.84117E+00	0	82	1.30663E+00	0	83	6.68693E-01
0	84	4.25574E-01	0	85	4.25574E-01	0	86	6.68693E-01	0	87	1.30663E+00
0	88	4.84117E+00	0	89	2.06180E-01	0	90	1.82474E-02	0	91	2.38460E-02
0	92	2.81715E-02	0	93	5.35741E-01	0	94	5.35741E-01	0	95	2.81715E-02
0	96	2.38460E-02	0	97	1.82474E-02	1	0	6.53968E-01	1	1	1.39791E-01
1	2	5.96145E-02	1	3	4.78338E-02	1	4	5.60313E-02	1	5	4.64501E-02
1	6	3.58496E-02	1	7	3.76616E-02	1	8	3.73317E-02	1	9	4.67558E+00
1	10	5.25133E-01	1	11	2.20459E-01	1	12	8.39309E-02	1	13	9.35141E-02
1	14	3.71117E-01	1	15	2.68458E-01	1	16	9.32515E-01	1	17	2.14353E+00
1	18	7.28372E-01	1	19	2.28389E-01	1	20	1.02134E-01	1	21	7.24758E-02
1	22	2.84302E-01	1	23	2.81151E-01	1	24	1.04951E+00	1	25	2.02894E+00
1	26	4.95360E-01	1	27	2.35919E-01	1	28	1.07541E-01	1	29	9.61377E-02
1	30	2.63728E-01	1	31	3.79311E-01	1	32	5.23069E-01	1	33	3.10960E+00
1	34	5.23227E-01	1	35	2.13440E-01	1	36	1.23908E-01	1	37	9.38063E-02
1	38	3.03443E-01	1	39	3.40361E-01	1	40	1.23746E+00	1	41	6.83617E+00
1	42	4.78174E-01	1	43	2.50501E-01	1	44	1.32491E-01	1	45	1.06610E-01
1	46	1.99127E-01	1	47	4.54844E-01	1	48	9.90109E-01	1	49	2.10223E+00
1	50	5.75406E-01	1	51	2.09998E-01	1	52	1.58721E-01	1	53	1.21476E-01
1	54	2.02807E-01	1	55	3.92176E-01	1	56	6.11109E-01	1	57	3.19360E+00
1	58	4.88159E-01	1	59	2.14207E-01	1	60	1.40203E-01	1	61	1.27144E-01
1	62	2.62413E-01	1	63	3.70919E-01	1	64	8.62584E-01	1	65	1.90369E+00
1	66	3.49370E-01	1	67	2.30052E-01	1	68	1.56321E-01	1	69	1.31395E-01
1	70	2.52260E-01	1	71	3.03047E-01	1	72	7.28316E-01	1	73	2.67094E+00
1	74	7.14755E-01	1	75	1.67156E-01	1	76	1.71334E-01	1	77	1.41068E-01
1	78	2.88084E-01	1	79	2.65685E-01	1	80	9.80471E-01	1	81	5.79059E+00
1	82	5.85606E-01	1	83	2.39819E-01	1	84	1.62509E-01	1	85	1.58933E-01
1	86	2.77205E-01	1	87	3.18780E-01	1	88	3.12860E+00	1	89	8.21797E-02
1	90	8.10056E-03	1	91	1.03816E-02	1	92	1.12278E-02	1	93	8.45412E-02
1	94	5.67384E-02	1	95	1.09884E-02	1	96	1.10727E-02	1	97	9.69516E-03
2	0	7.45681E-01	2	1	6.21329E-02	2	2	4.76065E-02	2	3	4.58143E-02
2	4	7.45931E-02	2	5	3.96956E-02	2	6	3.27928E-02	2	7	3.08285E-02
2	8	3.01549E-02	2	9	1.57677E+00	2	10	7.21350E-01	2	11	3.07537E-01
2	12	7.93946E-02	2	13	7.27526E-02	2	14	3.35007E-01	2	15	4.94586E-01
2	16	4.86254E-01	2	17	9.38512E-01	2	18	1.58145E+00	2	19	2.05753E-01
2	20	1.01126E-01	2	21	5.78138E-02	2	22	3.34342E-01	2	23	5.27389E-01
2	24	4.64015E-01	2	25	1.70811E+00	2	26	1.13786E+00	2	27	2.98450E-01
2	28	9.73018E-02	2	29	8.23932E-02	2	30	2.63721E-01	2	31	5.56892E-01
2	32	3.74929E-01	2	33	8.09398E-01	2	34	1.07305E+00	2	35	3.08929E-01
2	36	1.27522E-01	2	37	7.83475E-02	2	38	3.45214E-01	2	39	5.69008E-01
2	40	4.46143E-01	2	41	1.53544E+00	2	42	1.15869E+00	2	43	2.02397E-01
2	44	1.24968E-01	2	45	8.89776E-02	2	46	2.80664E-01	2	47	5.88903E-01

Arquivo GGHiFile.txt

0 0	4.87702E-06	0 1	1.60194E-05	0 2	1.46578E-04	0 3	5.02809E-04
0 4	1.00131E-04	0 5	1.00131E-04	0 6	5.02809E-04	0 7	1.46578E-04
0 8	1.60194E-05	0 9	6.88055E-05	0 10	7.18685E-05	0 11	3.20416E-05
0 12	5.64215E-04	0 13	3.01251E-04	0 14	3.01251E-04	0 15	5.64215E-04
0 16	3.20416E-05	0 17	7.18685E-05	0 18	6.04048E-04	0 19	5.48883E-05
0 20	6.54313E-05	0 21	1.87547E-04	0 22	9.59876E-04	0 23	9.59876E-04
0 24	1.87547E-04	0 25	6.54313E-05	0 26	5.48883E-05	0 27	1.65617E-04
0 28	1.12114E-04	0 29	1.03089E-04	0 30	1.10815E-04	0 31	2.72802E-04
0 32	2.72802E-04	0 33	1.10815E-04	0 34	1.03089E-04	0 35	1.12114E-04
0 36	1.28163E-04	0 37	8.08375E-05	0 38	9.98838E-05	0 39	8.66309E-04
0 40	5.47788E-05	0 41	5.47788E-05	0 42	8.66309E-04	0 43	9.98838E-05
0 44	8.08375E-05	0 45	3.03889E-04	0 46	4.65485E-05	0 47	1.79015E-04
0 48	4.21950E-04	0 49	2.10004E-04	0 50	2.10004E-04	0 51	4.21950E-04
0 52	1.79015E-04	0 53	4.65485E-05	0 54	2.08538E-04	0 55	6.21059E-05
0 56	1.64096E-04	0 57	8.90388E-04	0 58	1.37930E-04	0 59	1.37930E-04
0 60	8.90388E-04	0 61	1.64096E-04	0 62	6.21059E-05	0 63	2.87197E-04
0 64	4.42143E-05	0 65	8.94639E-05	0 66	2.63621E-04	0 67	3.11027E-04
0 68	3.11027E-04	0 69	2.63621E-04	0 70	8.94639E-05	0 71	4.42143E-05
0 72	3.05509E-04	0 73	6.09371E-05	0 74	1.17676E-04	0 75	2.27294E-04
0 76	5.27138E-04	0 77	5.27138E-04	0 78	2.27294E-04	0 79	1.17676E-04
0 80	6.09371E-05	0 81	2.98082E-04	0 82	7.75783E-05	0 83	1.54522E-04
0 84	2.97406E-04	0 85	3.61956E-04	0 86	3.61956E-04	0 87	2.97406E-04
0 88	1.54522E-04	0 89	7.75783E-05	1 0	1.50579E+03	1 1	7.66996E+02
1 2	2.47663E+02	1 3	1.30539E+02	1 4	3.19575E+01	1 5	2.53591E+01
1 6	1.55910E+02	1 7	1.15624E+02	1 8	2.97540E+02	1 9	6.76134E+02
1 10	2.14853E+02	1 11	1.95555E+02	1 12	8.62737E+01	1 13	3.13198E+01
1 14	2.62621E+01	1 15	1.14361E+02	1 16	1.01385E+02	1 17	1.49842E+02
1 18	2.35197E+02	1 19	7.17068E+01	1 20	6.64073E+01	1 21	6.22647E+01
1 22	3.28013E+01	1 23	2.30703E+01	1 24	7.63968E+01	1 25	4.60141E+01
1 26	5.46677E+01	1 27	5.48017E+01	1 28	2.55570E+01	1 29	2.95775E+01
1 30	3.16763E+01	1 31	3.63071E+01	1 32	2.60376E+01	1 33	3.27855E+01
1 34	2.13477E+01	1 35	1.92149E+01	1 36	1.36101E+01	1 37	1.04676E+01
1 38	1.07867E+01	1 39	1.39765E+01	1 40	9.96954E+00	1 41	8.36850E+00
1 42	1.37829E+01	1 43	8.89369E+00	1 44	8.46921E+00	1 45	4.58314E+00
1 46	4.51050E+00	1 47	4.47352E+00	1 48	4.62100E+00	1 49	3.95450E+00
1 50	3.98494E+00	1 51	6.23973E+00	1 52	4.05965E+00	1 53	4.02591E+00
1 54	1.97541E+00	1 55	2.32780E+00	1 56	2.14269E+00	1 57	1.75712E+00
1 58	1.96373E+00	1 59	1.93300E+00	1 60	2.66255E+00	1 61	2.03524E+00
1 62	2.04397E+00	1 63	1.05029E+00	1 64	1.25822E+00	1 65	1.19590E+00
1 66	9.90176E-01	1 67	1.15166E+00	1 68	1.10755E+00	1 69	1.47251E+00
1 70	1.12960E+00	1 71	1.14613E+00	1 72	9.28230E-01	1 73	1.08727E+00
1 74	1.04545E+00	1 75	9.57881E-01	1 76	5.87495E-01	1 77	6.42009E-01
1 78	1.19591E+00	1 79	9.55240E-01	1 80	8.45378E-01	1 81	1.25153E+00
1 82	3.04591E+00	1 83	3.36409E+00	1 84	2.58523E+00	1 85	4.82804E-01
1 86	4.46413E-01	1 87	1.85608E+00	1 88	2.29583E+00	1 89	2.52552E+00
2 0	2.60901E+02	2 1	8.66104E+01	2 2	6.20283E+01	2 3	4.11098E+01
2 4	8.85466E+00	2 5	7.82517E+00	2 6	4.00757E+01	2 7	3.96597E+01
2 8	5.38600E+01	2 9	3.42970E+02	2 10	6.12993E+01	2 11	8.74104E+01
2 12	3.84246E+01	2 13	1.14487E+01	2 14	1.02021E+01	2 15	4.25042E+01
2 16	5.60362E+01	2 17	4.40382E+01	2 18	1.53819E+02	2 19	2.56173E+01
2 20	3.57170E+01	2 21	3.44114E+01	2 22	1.21730E+01	2 23	1.10590E+01
2 24	3.73012E+01	2 25	2.86023E+01	2 26	2.13586E+01	2 27	2.99794E+01
2 28	9.45119E+00	2 29	1.42621E+01	2 30	1.58221E+01	2 31	1.47775E+01
2 32	1.31393E+01	2 33	1.67494E+01	2 34	1.28633E+01	2 35	8.17962E+00
2 36	6.25749E+00	2 37	3.87867E+00	2 38	5.38889E+00	2 39	6.39876E+00
2 40	3.91144E+00	2 41	3.62745E+00	2 42	6.54342E+00	2 43	4.91655E+00
2 44	3.40649E+00	2 45	1.99428E+00	2 46	1.69143E+00	2 47	2.10688E+00
2 48	2.19016E+00	2 49	1.51060E+00	2 50	1.45945E+00	2 51	2.47705E+00
2 52	2.03624E+00	2 53	1.51856E+00	2 54	9.01188E-01	2 55	8.90841E-01
2 56	1.04244E+00	2 57	9.05206E-01	2 58	7.58858E-01	2 59	7.49457E-01
2 60	1.09555E+00	2 61	1.00059E+00	2 62	8.16166E-01	2 63	4.79124E-01
2 64	4.89231E-01	2 65	5.52320E-01	2 66	4.77528E-01	2 67	4.02666E-01